Building and Using a Planning Ontology from Past Data for Performance Efficiency

Bharath Muppasani[1], Vishal Pallagani[1], Biplav Srivastava[1] and Raghava Mutharaju[2]

[1]*University of South Carolina, USA*
[2]*Indraprastha Institute of Information Technology, IIIT-Delhi, India*

Abstract
Automated planning is a core task studied in artificial intelligence (AI) to help an agent rationally move from an initial to a goal state using a set of legal actions. The vast information from past data consisting of diverse planning domains, a variety of planners, and characteristics of generated plans carry essential information that can be leveraged to improve planner performance. For instance, by analyzing the performance of different planners on various problem configurations, one can identify which *type of* planners excel in particular domains and improve their efficiency. We present a planning ontology built using data from the International Planning Competition (IPC) as well as new plans and demonstrate the benefits of one of its action ordering use cases: using macros to improve planner performance.

Keywords
Ontology, Automated Planning, Planner Improvement

1. Introduction

Automating planning is a long-standing objective in the field of Artificial Intelligence (AI). The ability to generate plans and make decisions in complex domains, such as robotics, logistics, and manufacturing, has led to significant progress in the automation of planning. Currently, there are a large number of planning domains, planners, search algorithms, and associated heuristics in the field of automated planning. Each planner, in conjunction with a search algorithm and heuristic, generates plans with varying degrees of quality, cost, and optimality. The empirical results available for various planning problems, ranked by planner performance and the heuristics used as available in IPC, can provide valuable information to identify various tunable parameters such as macros, that improve planner performance. Traditionally, improving planner performance involves manually curating potential combinations to identify the optimal planner configuration. However, there has been limited effort to model the available information in a structured knowledge representation, such as an ontology [1], to facilitate efficient reasoning and further enhance planner performance.

Previous attempts to build ontologies for planning have been limited to a specific domain [2] or have not fully captured the metadata needed to assist a planner in improving its performance [3, 4]. To overcome these limitations, we propose a comprehensive ontology for AI planning. An ontology is a formal representation of concepts and their relationships [5], which, for planning, captures the features of a domain, the capabilities of planners, and how it was used to

Planning and Ontology Workshop (PLATO), ICAPS 2023, July 09-10, 2023, Prague, CZ

generate plans. This information enables systematic analysis of planning domains and planners facilitating reasoning with existing planning problems, identifying similarities, and suggesting different planner configurations. Our planning ontology is domain-independent and captures planning problems written in Planning Domain Definition Language (PDDL) [6], similar to previous work by Žáková et al. [7]. It can be a useful resource for creating new planners, as it captures essential information about planning domains and planners, which can be leveraged to design more efficient planning algorithms. In addition, the ontology can promote knowledge sharing and collaboration within the planning community.

Building a planning ontology starting with data from IPC offers several benefits such as a comprehensive coverage of planning domains, a rich source for various benchmark evaluation metrics, and documentation for the planners. However, the ontology is not necessarily limited to IPC and can be extended to any plan data source, planning domain or planner type. We also show an example scenario where the developed planning ontology could be used to extract macros to help improve planner performance. Our contributions are that we:

- propose a comprehensive ontology for AI planning built from prior data about planning domains, planners, and plans generated built using data from IPCs and any new plan
- demonstrate its usage to answer a range of queries about planner suitability for solving
- present a case study where macros are extracted, stored, and reused from past data
- provide it to the planning community to promote reuse and collaboration

In the remainder of the paper, we start with preliminaries about ontologies, followed by automated planning and IPC. We then give an overview of the existing literature on ontologies for planning. Following this, we present a detailed description of the ontology construction process and its usage. We then end the paper with a discussion of the proposed planning ontology and conclude with future research directions.

2. Ontology Construction

This section covers the construction of an ontology to capture the essential details of automated planning.

2.1. Competency Questions

Competency questions for an ontology are focused on the needs of the users who will be querying the ontology. These questions are designed to help users explore and understand the concepts and relationships within the ontology, and to find the information they need within the associated knowledge base. By answering these questions, the ontology can be better scoped and tailored to meet the needs of its users. SPARQL queries for each of these questions can be found at our GitHub Repository[1].

- C1: What are the different types of planners used in automated planning?

[1]https://github.com/BharathMuppasani/AI-Planning-Ontology/tree/main/v1.0

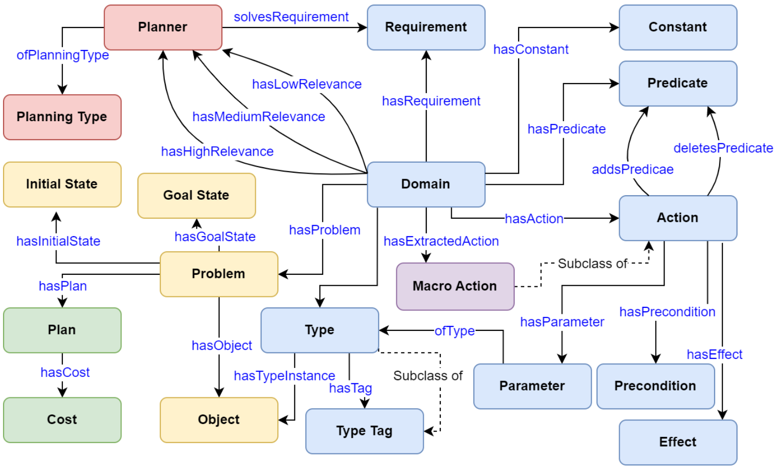

Figure 1: Ontology capturing different concepts of domain, problem, plan, and planner performance in automated planning

- C2: What is the relevance of a planner in a given problem domain?
- C3: What is the cost associated with generating a plan for a given problem?
- C4: What are the available actions for a given domain?
- C5: How many parameters does a specific action has?
- C6: What are the different `types` present in a domain?
- C7: What are all requirements a given domain has?
- C8: What planning type a specific planner belongs to?
- C9: What requirements does a given planner support?

2.2. Design

An ontology is a formal and explicit representation of concepts, entities, and their relationships in a particular domain. In this case, ontology is concerned with the domain of automated planning, which refers to the process of generating a sequence of actions to achieve a particular goal within a given set of constraints. The ontology aims to provide a structured framework for organizing and integrating knowledge about this automated planning domain.

Figure 1 shows an ontology that aims to encompass the various elements of the automated planning domain and its associated problems. The ontology comprises 19 distinct classes, `Domain`, `Requirement`, `Type`, `Type Tag`, `Constant`, `Predicate`, `Action`, `Macro Action`, `Parameter`, `Precondition`, `Effect`, `Problem`, `Object`, `Initial State`, `Goal State`, `Plan`, `Cost`, `Planner` and `Planning Type`, and 24 properties of objects. Among these, 11 classes are used to represent the problem domain (blue), 4 classes are used to denote the problem itself (yellow), 2 classes are used to represent the plan generated for the problem and its cost (green), and 2 classes are dedicated to capturing the details of planner performance from previous IPCs (red). The extracted macro actions for a domain are updated in the ontology as a subclass of the action class(purple).

To incorporate the details of planner performance into the ontology, we have used information from previous IPCs (1998, 2000, and 2002). Specifically, we have analyzed the number of problems that a given planner has successfully solved and categorized this information into three distinct levels of relevance to the planner. Planners that have solved a relatively small number of problems are classified as of low relevance (0% to 30% problems), whereas those who have solved a moderate number of problems are considered to have medium relevance (30% to 65% problems). Finally, planners that have solved a large number of problems, including many challenging ones, are classified as having high relevance (65% to 100% problems) for a given domain. By incorporating this information into the ontology, we can better assess the performance of planners in different problem domains and make more informed decisions about which planners to use for a given problem. In addition, this information can be used to guide the development of new planners and to evaluate their performance against established benchmarks.

2.3. Tools Used

In the process of creating an ontology for automated planning, several tools were used for different purposes. The ontology was created using `Protege`[2], which is a widely used open-source ontology editor and knowledge management system. `Protege` provides an intuitive user interface that allows users to easily create and edit ontologies. It also supports a wide range of ontology languages, including OWL, RDF, and RDFS.

After creating the ontology, we utilized the `rdflib`, a Python library, to access the RDF-based ontology and extract the relevant information. To begin populating the ontology, we captured the domain and problem data in JSON format. Subsequently, we incorporated the data triples from different domains into the ontology using the `rdflib` library. Additionally, we included information about the performance of various planners from previous IPCs in the ontology. Our GitHub repository[1] provides the RDF model file for the ontology, as well as Python scripts to extract domain and problem data in JSON format and add the extracted data as triples to the model ontology, creating a knowledge graph. To query the resulting knowledge graph, we utilized the SPARQL query language, which is the standard query language for RDF data.

Table 1
Extracted action relations, ordered based on their frequency, for domains blocksworld, driverlog, and grippers.

Domains	Extracted Action Relations
blocksworld	unstack * put-down * stack; put-down * unstack; stack * pick-up; unstack * stack; put-down * pick-up * unstack
driverlog	drive-truck * unload-truck; drive-truck * load-truck; board-truck * drive-truck; walk * board-truck
grippers	pick * move; move * drop

3. Extracting, Storing and Reusing Macro Operators

While automated planning has been successful in many domains, it can be computationally expensive, especially for complex problems. One approach to improve efficiency is by using

[2]https://protege.stanford.edu/

macro-operators, which are sequences of primitive actions that can be executed as a single step. However, identifying useful macro-operators manually can be time-consuming and challenging. Authors in [8] introduce a novel method for improving the efficiency of planners by generating macro-operators. The proposed approach involves analyzing the inter-dependencies between actions in plans and extracting macro-operators that can replace primitive actions without losing the completeness of the problem domain. The soundness and complexity of the method are assessed and compared to other existing techniques. The paper asserts that the generated macro-operators are valuable and can be seamlessly integrated into planning domains without losing the completeness of the problem.

Based on the ontology depicted in Figure 1, we extract macro-operators that can enhance the efficiency of planners. To demonstrate this, we have considered three different domains: `blocksworld (bw)`, `driverlog (dl)`, and `grippers (gr)`, presented in IPC-2000, 2002, and 1998 respectively. We initially developed a knowledge graph using the ontology represented in Figure 1 for the three domains of interest. Subsequently, we employed a SPARQL query to retrieve the stored plans for these domains. We then examined these plans to identify the sequences of action pairs and ranked them based on their frequency of occurrence. To improve the effectiveness of this technique, it is essential to consider both the frequency of occurrence of action pairs and the properties of the domain. Specifically, the precondition and effect of actions should be analyzed to ensure that the first action leads to the precondition of the second action in the pair. We employed another SPARQL query to extract the preconditions and effects associated with each of these actions. We analyzed the resulting action pairs to verify their validity of occurrence, thereby filtering out pairs that did not have a combined effect. The results of this extraction process are shown in Figure 1. These action relations are stored back into the knowledge graph in the `Macro Operators` class and can be utilized by planners to enhance their efficiency and produce better plans.

We have used the extracted action relations to test Plansformer [9], a generative model for AI planning. Plansformer is obtained by fine-tuning CodeT5, a Large Language Model that is pre-trained on code. We use Plansformer for our experimentation as it is easy to infuse macros by appending to the input and obtain the generated plan for validation. The results are presented in Table 2. By directly including the extracted action relations in the prompt, we can observe an increase in percentage for valid plans in `bw` domain, whereas the percentage declined for other domains.

Table 2
Table showing the results of plan validation for Plansformer with the percentage of optimal plans shown in parentheses., with an asterisk denoting domains that had extracted action relations added to the prompt.

	bw	**bw***	**dl**	dl*	**gr**	gr*
Successful	90.04% (88.44%)	**94.08% (92.36%)**	**76.56% (52.61%)**	40.08% (35.69%)	**82.97% (69.47%)**	72.42% (44.94%)
Failed	9.94%	**5.92%**	**23.44%**	42.86%	**16.61%**	21.97%
Incomplete	0.02%	**0%**	**0%**	17.06%	**0.42%**	5.61%

4. Conclusion

An ontology provides a structured representation of concepts and relations for AI planning domains, enabling the efficient extraction of domain properties. In this regard, we propose a domain-independent ontology for AI planning that captures the features of a planning domain and the capabilities of planners. Additionally, we present a use case for the ontology, which involves the extraction of macro operators using plan statistics and domain properties. In the future, we would like to extract macros for a wide variety of domains, store them in the ontology, and evaluate traditional planner performance with macros.

Ontologies will play a crucial role in advancing AI planning, enabling the creation of more intelligent and efficient planning systems. They can model knowledge from multiple domains, integrate it into a unified planning framework, and support intelligent knowledge-based systems. Future prospects for ontology in automated planning include exploring complex domains such as multi-agent systems, developing hybrid systems with machine learning, and finding relevant planners from past experiences.

Acknowledgement

Raghava Mutharaju would like to acknowledge the partial support of the Infosys Center for Artificial Intelligence (CAI), IIIT-Delhi. Biplav Srivastava would like to acknowledge the partial support of Cisco Research and Vajra (India) program.

References

[1] R. Gayathri, V. Uma, Ontology based knowledge representation technique, domain modeling languages and planners for robotic path planning: A survey, ICT Express 4 (2018) 69–74.

[2] A. Valente, T. Russ, R. MacGregor, W. Swartout, Building and (re) using an ontology of air campaign planning, IEEE Intelligent Systems and their Applications 14 (1999) 27–36.

[3] Y. Gil, J. Blythe, Planet: A shareable and reusable ontology for representing plans, in: AAAI Workshop on Representational Issues for Real-world Planning Systems, volume 114, 2000.

[4] M. Babli, E. Onaindia, E. Marzal, Extending planning knowledge using ontologies for goal opportunities, arXiv preprint arXiv:1904.03606 (2019).

[5] N. Guarino, D. Oberle, S. Staab, What is an ontology?, Handbook on ontologies (2009) 1–17.

[6] D. McDermott, M. Ghallab, C. Knoblock, D. Wilkins, A. Barrett, D. Christianson, M. Friedman, C. Kwok, K. Golden, S. Penberthy, D. Smith, Y. Sun, D. Weld, PDDL - The Planning Domain Definition Language, Technical Report, Technical Report, 1998.

[7] M. Žáková, P. Křemen, F. Železnỳ, N. Lavrač, Automating knowledge discovery workflow composition through ontology-based planning, IEEE Transactions on Automation Science and Engineering 8 (2010) 253–264.

[8] L. Chrpa, Generation of macro-operators via investigation of action dependencies in plans, The Knowledge Engineering Review 25 (2010) 281–297.

[9] V. Pallagani, B. Muppasani, K. Murugesan, F. Rossi, L. Horesh, B. Srivastava, F. Fabiano, A. Loreggia, Plansformer: Generating symbolic plans using transformers, arXiv preprint arXiv:2212.08681 (2022).