Challenges on Deriving Planning Problems from Ontologies

Milene Santos Teixeira[1], Michael Welt[1], Raphael Chis[1] and Birte Glimm[1]

[1]Institute of Artificial Intelligence (Ulm University), Ulm, Germany

Abstract

The integration of automated planning and ontologies can benefit systems in aspects like additional expressivity and the handling of huge domains. However, the mapping between these techniques might not be straightforward. This work identifies some of the core challenges on the mapping of domain-specific ontologies into planning problems. Here, we identify challenges from both, the lenses of automated planning and domain-specific requirements. This work is intended to raise the discussion on these challenges and motivate future work on this direction.

Keywords

automated planning, ontology, plan engineering, plan modeling

1. Introduction

Automated planning [1] consists of a search problem that describes a sequence of *actions* to be applied in the world aiming to achieve a *goal*. Planning considers that the world has different *states*, which are changed through the application of *actions*. An action is applicable in the current state as long as its *preconditions* are met. Actions also present *effects*, i.e. a list of add and delete values that describe what changes in the world as a consequence of their execution.

Plan-based systems incorporate knowledge about an application domain, which is used to automatically compute a course of actions that can achieve a given goal. To build such systems, knowledge engineers must comprehend the domain knowledge and find the best way to model the expected behavior of the system as a planning problem, which is a complex task [2]. To be able to express their mental model into planning terms, knowledge engineers must also be aware of planning capabilities and constraints. The integration of further mechanisms that formalize the knowledge of a domain, such as ontologies [3], can aid this process. In addition to their reasoning capabilities, ontologies may contribute to plan-based systems in aspects like:

- providing additional expressiveness to represent the domain as a planning problem;
- resorting to established means of ontology-based explanations and respective modeling support to explain planning domains and results;
- mapping the planned state into the real-world state;
- providing or enriching the domain (ground instances) for planning instances;
- covering large domains that otherwise would be limited by the plan scalability.

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

milene.santos-teixeira@uni-ulm.de (M. S. Teixeira); michael.welt@uni-ulm.de (M. Welt); raphael.chis@uni-ulm.de (R. Chis); birte.glimm@uni-ulm.de (B. Glimm)

While planned states can still be mapped to real-world states through a planning specification that is independent from the ontology components [4, 5], maintenance is a challenge. As changes are not automatically reproduced when either the ontology or planning domain needs updates, there is an additional human cost to keep the specifications aligned and to update systems deploying it. A more flexible strategy would be to automate the plan-ontology integration, i.e., deriving the planning specification from the knowledge within the ontology (and/or vise-versa). However, given the differences between the two techniques, mapping the knowledge from an ontology into a planning problem is not a straightforward task. This paper first discusses existing strategies and, next, identifies some of the core challenges of mapping ontological knowledge into a planning problem.

2. Related Work

Recent literature on the integration of planning and ontologies is not extensive. We base this review on the work of Teixeira [6], but we keep our focus on the mapping strategies adopted by these works for the integration of planning and ontologies.

One of the identified strategies consists of formalizing the planning technique itself as an ontology. For example, Freitas et al. [7] propose an ontology that formalizes hierarchical task network (HTN) planning [8]. The ontology makes it possible to automatically derive a planning specification. The HTN formalization was provided in the schema part of the ontology (called the terminological part or TBox). This way, it is enough to populate the ontology's facts (in the assertional part – the ABox) with domain-specific knowledge to generate HTN plans for different domain applications. The work of Bermejo-Alonso et al. [9] focuses on emergency scenarios and includes planning concepts in a domain-specific ontology. This formalization enables the mapping of ontology concepts into planning terms to generate a corresponding plan. A disadvantage of these approaches is that they require knowledge engineers to not only be familiar with ontological modeling, but they must also comprehend planning in order to populate the ontology. That is, before instantiating the domain-specific knowledge, the knowledge engineer must think about how this domain can best be represented as a planning problem.

Some works rely on an external mapping of predefined components of the ontology into planning terms. This mapping can be exploited both ways, that is, either to generate a planning problem based on the ontology or to map planned actions into ontology instances. PORSCE II [10], for example, is a system to perform automatic semantic web service composition that relies on an ontology that models web services. In this work, the knowledge is easily mapped into planning actions since the ontology already describes which are the expected inputs (preconditions) and outputs (effects) of each web service. Planned actions can also easily be mapped back to their corresponding web service during execution. A similar strategy was used by Žáková et al. [11], who focuses on the automated construction of knowledge discovery workflows. Specific classes within an ontology that formalizes knowledge discovery are translated into planning domain and problem instances. For example, actions are translated from instances of `NamedAlgorithm`, while their preconditions and effects are generated from ontology properties called `input` and `output`, respectively. The resulting classical plans correspond to the expected workflows and they also can be employed to extract knowledge from the ontology. Similar mappings were also a common strategy adopted by works within the dialogue community [12, 13].

Another interesting strategy can be observed in the work of Behnke et al. [14], which integrates planning and an ontology in the context of a companion system. By relying on an ontology as a common knowledge source, separated models are automatically generated (and also extended) for the plan and a dialogue component. In this process, ontology and planning aspects are translated through a mapping defined by the authors. The definition of predicates in the planning domain, for example, is mapped to pre-defined OWL (Web Ontology Language) properties. Meanwhile, arguments of a predicate can be obtained from annotations in the ontology. By keeping a shared vocabulary and model between ontology, plan, and dialogue components, this approach can reproduce possible domain updates and reduce costs on maintenance. A similar strategy is adopted by Schiller et al. [15], which also addresses companion systems. This work exploits a conceptual model (ontology) to provide domain-specific knowledge and a planning model to address procedural knowledge. The planning model is automatically generated by exploiting the most generic concepts within the ontology. The integration of conceptual and procedural knowledge expands the possibility of delivering explanations of the resulting plans.

By reviewing the literature and to the best of our knowledge, there is no consensus on the best strategy to derive planning problems from ontologies. Similar to the decision on the type of planning to be employed in the application (e.g. classical, probabilistic or HTN), the suitability of one strategy or another depends on aspects like the domain requirements and system's purpose. The next section discusses the main challenges we encountered while exploiting a reusable mapping of a domain-specific ontology into an HTN planning specification.

3. From an e-Learning Ontology to an HTN Planning Problem

Ontologies are well-estabilished within the learning community (see, e.g., [16] for a survey), where they enhance learning systems through their reusability and reasoning ability.

The diagram in Figure 1 represents a simplified ontology for the e-Learning domain, which is used in our early-stage work within the 2LIKE project[1] for deriving planning problems from the ontology. This simplified diagram is enough to introduce some of the challenges that arise in this context. In summary, the diagram represents a *course*, which is divided into *concepts* (topics). A concept might be a pre-requisite of another concept. Each concept is linked to a set of *resources*, that correspond to learning materials (e.g. lectures, quizzes) to be presented to the *learner*.

A *learning path* consists of a sequence of concepts or resources to be studied by a learner. The heterogeneity of learners' profiles raises the need for adaptive learning paths [17]. For example, at the beginning of a course, a learner might already have knowledge of a given concept belonging to this course. In this case, this concept should not be included in this learner's learning path. As learning paths can be intuitively interpreted as planning problems, i.e., a problem with an initial state for which a sequence of actions should be applied in order to

[1]https://www.uni-ulm.de/en/in/institute-of-artificial-intelligence/research/projects/2like/

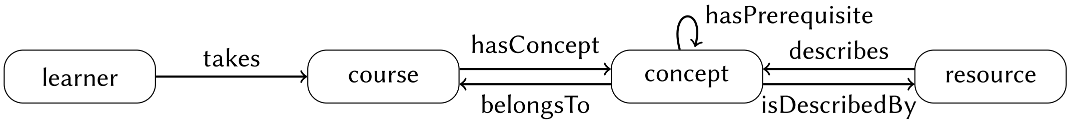

Figure 1: Simplified representation of an e-Learning ontology

achieve a goal, several works have exploited plan-based solutions to adapt these paths [17, 4]. Within the scope of the 2LIKE project, we exploit the integration of an e-Learning ontology and automated planning to dynamically generate adaptive learning paths.

One of the first tasks when specifying any domain as a planning problem, is to identify the most suitable planning type. Given the hierarchical nature of learning problems, i.e., a course can be decomposed into topics, which can be decomposed into smaller learning objects, we identify HTN planning as a suitable approach for this domain. In HTN planning, one distinguishes *abstract tasks* and *primitive actions*. Abstract tasks can be decomposed by *methods* into sub-tasks until the decomposition eventually leads to *primitive actions*, i.e., actions that cannot be further decomposed and which are directly executable (in our case, performed by the learner). HTN plans can also be represented on different levels of detail to users.

While, ideally, an approach to map ontologies into planning terms should be reusable and general, a couple of challenges arise. We next discuss theses challenges, which are not limited to the considered e-Learning domain.

Plan-specific requirements: To achieve the expected plan, a few planning aspects that are not part of domain-specific knowledge must be addressed in the planning model. These aspects are often related to the type of planning and to the capabilities of the automated planner employed. Examples include disjunctive statements (not covered by many planners), variable updates (addressed as add and delete effects in planning), or states that are logically possible but do not happen in the real world. Since these aspects are not part of the domain knowledge, they might not be foreseen in the ontology and, therefore, cannot be easily mapped from it. Instead, they must be predicted by the knowledge engineer, who has to be aware of specific planning requirements.

In the mapping of the e-Learning ontology into an HTN model, to avoid dead-ends that prevent plan generation, we identified the need of specifying additional decomposition methods that address conditions to be *ignored* by the plan:

```
(:method dummyStudyConcept
  :parameters (?l - learner ?c - concept)
  :task (AchieveStudyConcept ?l ?c)
  :precondition  (and (study ?l ?c)(seen ?l ?c) )
  :ordered-subtasks (doNothing) )
```

This sample[2] shows a *dummy* method that was specified to decompose the abstract task `AchieveStudyConcept`. This method will ignore (`doNothing` operator) a concept that, during the search, was flagged to be studied but, as it was already seen by the learner, should not

[2]Samples displayed in this work follow the HDDL syntax [18].

be included in the learning path (i.e., the plan). The original ontology does not handle this unwanted combination, therefore, no direct mapping is possible. Note that, for generality purposes, we avoid including planning aspects into the e-Learning ontology.

Domain-specific requirements: Domain-specific requirements might also prevent a one-to-one mapping from a domain-specific ontology into a planning model. An example is related to the possibility of using lifted or grounded parameters. The actual method in charge of decomposing the task `AchieveStudyConcept` could be specified with lifted parameters as follows:

```
(:method studyConcept
  :parameters (?l - learner ?c - concept)
  :task (AchieveStudyConcept ?l ?c)
  :precondition  (and (study ?l ?c) (not(seen ?l ?c)) )
  :ordered-subtasks (studyConceptOp ?l ?c) )
```

However, since a concept might have prerequisites, those cannot be specified in a lifted way. Instead, a modeling strategy would be to specify a method for each concept, listing the specific prerequisites of this concept as preconditions of the method. The example below describes a method to address the concept `tree` (i.e., the concept of trees as a data structure), which has as its prerequisite the concept `list`. Note that the `learner` parameter must stay lifted.

```
(:method studyConceptTree
  :parameters (?l - learner)
  :task (AchieveStudyConcept ?l tree)
  :precondition  (and (study ?l tree) (not(seen ?l tree)) (seen ?l list))
  :ordered-subtasks (studyConceptOp ?l tree) )
```

Although this is not an uncommon strategy, the challenge here is knowing which parameters should be lifted and which should be grounded. Unless the ontology flags it, we rely, once again, on the knowledge engineer to specify this.

4. Conclusion

This paper discussed some of the occurring challenges in the alignment of domain-specific knowledge and planning-specific vocabulary. By exploiting the mapping of an ontology for the e-Learning domain into an HTN planning problem, we identified challenges observed from both, the lenses of planning and domain-specific requirements. We concluded that the identified challenges limit the creation of a general and reusable framework for such a mapping, which remains an open problem. This work is aimed to motivate the discussion and future investigation to find solutions for these challenges and similar ones related to this integration.

References

[1] M. Ghallab, D. Nau, P. Traverso, Automated Planning: Theory and Practice, Elsevier, 2004.

[2] S. Sreedharan, T. Chakraborti, C. Muise, Y. Khazaeni, S. Kambhampati, – d3wa+ – a case study of XAIP in a model acquisition task for dialogue planning, Proc. of the Int. Conf. on Automated Planning and Scheduling (ICAPS'20) 30 (2020) 488–497.

[3] T. R. Gruber, A translation approach to portable ontology specifications, Knowledge Acquisition 5 (1993) 199–220.

[4] E. Kontopoulos, D. Vrakas, F. Kokkoras, N. Bassiliades, I. Vlahavas, An ontology-based planning system for e-course generation, Expert Systems with Applications 35 (2008) 398–406.

[5] T. W. Bickmore, D. Schulman, C. L. Sidner, A reusable framework for health counseling dialogue systems based on a behavioral medicine ontology, Journal of Biomedical Informatics 44 (2011) 183–197.

[6] M. S. Teixeira, Automating the Generation of Goal-Oriented Dialogue Managers for Healthcare, Ph.D. thesis, University of Trento, 2022.

[7] A. Freitas, D. Schmidt, A. Panisson, F. Meneguzzi, R. Vieira, R. H. Bordini, Semantic representations of agent plans and planning problem domains, in: Int. Workshop on Engineering Multi-Agent Systems, Springer, 2014, pp. 351–366.

[8] K. Erol, J. Hendler, D. Nau, Complexity results for HTN planning, Annals of Mathematics and AI 18 (1996) 69–93.

[9] J. Bermejo-Alonso, J. Salvador, R. Sanz, Towards an ontology for task and planning in autonomous systems: An emergency scenario, in: Iberian Robotics Conf., Springer, 2017, pp. 429–440.

[10] O. Hatzi, D. Vrakas, N. Bassiliades, D. Anagnostopoulos, I. Vlahavas, The PORSCE II framework: Using AI planning for automated semantic web service composition, The Knowledge Engineering Review 28 (2013) 137–156.

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

[12] M. S. Teixeira, V. Maran, M. Dragoni, The interplay of a conversational ontology and AI planning for health dialogue management, in: The 36th ACM/SIGAPP Symposium On Applied Computing (SAC'21), ACM, 2021, p. 611–619.

[13] V. Franzoni, A. Milani, J. Vallverdú, Emotional affordances in human-machine interactive planning and negotiation, in: Proceedings of the Int. Conf. on Web Intelligence, ACM, 2017, pp. 924–930.

[14] G. Behnke, D. Ponomaryov, M. Schiller, P. Bercher, F. Nothdurft, B. Glimm, S. Biundo, Coherence across components in cognitive systems – one ontology to rule them all, in: Proc. of the 24th Int. Joint Conf. on Artificial Intelligence, 2015, pp. 1442–1449.

[15] M. Schiller, G. Behnke, M. Schmautz, P. Bercher, M. Kraus, M. Dorna, W. Minker, B. Glimm, S. Biundo, A paradigm for coupling procedural and conceptual knowledge in companion systems, in: 2017 Int. Conf. on Companion Technology (ICCT'17), IEEE, 2017.

[16] G. George, A. M. Lal, Review of ontology-based recommender systems in e-learning, Computers & Education 142 (2019).

[17] V. Caputi, A. Garrido, Student-oriented planning of e-learning contents for moodle, Journal of Network and Computer Applications 53 (2015) 115–127.

[18] D. Höller, G. Behnke, P. Bercher, S. Biundo, H. Fiorino, D. Pellier, R. Alford, HDDL: An extension to PDDL for expressing hierarchical planning problems, Proc. of the AAAI Conf. on Artificial Intelligence 34 (2020).