A Unified Knowledge Base for Companion-Systems – A Case Study in Mixed-Initiative Planning

Gregor Behnke*, Marvin Schiller*, Denis Ponomaryov+, Florian Nothdurft†,
Pascal Bercher*, Wolfgang Minker†, Birte Glimm*, and Susanne Biundo*

*Institute of Artificial Intelligence,
† Institute of Communications Engineering,
Ulm University, D-89069 Ulm, Germany,
e-mail: firstName.lastName@uni-ulm.de
+A.P. Ershov Institute of Informatics Systems,
Novosibirsk State University, 630090 Novosibirsk, Russia,
e-mail: ponom@iis.nsk.su

Abstract. Companion systems aim to extend the abilities of ordinary technical systems, for instance by modeling the user's situation, by recognizing the user's intentions, and by being able to interact with the user and to adapt to her/him. Such a system depends on planning capabilities to determine which actions are necessary to achieve a particular goal. In many situations it may not be appropriate for a companion system to develop plans on its own, but instead it has to integrate the user while creating the plan, i.e., it needs to be mixed-initiative. Based on earlier work, we demonstrate how a central knowledge base for a mixed-initiative planning system can be designed. We outline various benefits our approach brings to bear within a companion system. Lastly, we present several requests a user might issue towards the mixed-initiative planning system and how they can be answered by harnessing the knowledge base.

1 Introduction

Most state-of-the-art planners and planning applications work in a black-box fashion. They receive a planning problem from the user and compute a solution without further interaction, which in turn is conveyed to the user. While this scheme is sometimes sufficient, it poses significant problems in situations where the final decision on which action to execute should rest with a human user. Such situations usually occur if grave risks are involved (see e.g. [11, 1]) or the plan to be developed is of a more personal nature, e.g., a personalized training plan. Here it is necessary to integrate the user directly into the process of generating a plan via interaction. A system that possesses this capability is called a Mixed-Initiative Planning System (MIPS). In addition to the pure planning capability, a MIPS that can be successfully applied in real world scenarios must incorporate advanced user interaction and explanation facilities. Each of the several components of a MIPS usually has its own domain model, specifically tailored to information relevant for that particular component. This can cause significant problems if these representations are not coherent which can, e.g., occur if the domain has to be changed. In previous work [2] we have shown how an ontology can be utilized as the central knowledge base for all relevant domain knowledge and how specialized models can be generated based upon it.

In this paper we assume that the MIPS uses a hierarchical planning approach, as we believe that it is more intuitive for the user when the planner solves the problem in a similar way experienced human problem-solvers tend to approach problems, i.e., top-down [5]. At the same time, hierarchical planning provides an expressive formalism [9], which can also be exploited when planning for humans [3]. More specifically we employ Hybrid Planning [4], an extension of Hierarchical Task Network (HTN) planning [7, 8], where tasks are arranged in a hierarchical fashion using so-called decomposition methods. A method $\mathcal{A} \mapsto_{\prec} \mathcal{B}_1, \ldots, \mathcal{B}_n$ describes that the abstract task $\mathcal{A}$ can be achieved by executing $\mathcal{B}_1, \ldots, \mathcal{B}_n$ – which may be primitive or abstract – in any order compatible with $\prec$.

This paper starts with a brief overview of the previously developed technique [2]. Next, we discuss its specific advantages for mixed-initiative planning, while using the same application domain: fitness training. Finally, we outline avenues for future applications of our approach that promise benefits for MIPS.

2 Integrating Planning Knowledge into Ontologies and Retrieving it

In this section we present an approach [2] to using an ontology as the central knowledge base for a MIPS. First we show how the planner's domain model, which encompasses most of the system's knowledge, can be represented as part of an ontology. Next we describe how that model can be retrieved and passed on to the MIPS' planning component in its required formalism and integrity, and how an initial specification of a planning model can be automatically extended using reasoning. Finally we demonstrate how verbal explanations for plans generated by the planner can be enhanced using ontology verbalizations.

2.1 Creating the Ontology

The ontology is constructed such that it contains a suitable encoding of the planning domain. The concept hierarchy of ontologies resembles the task hierarchy of HTN planning in that both represent a hierarchical order from the most abstract towards more concrete objects. In keeping with this analogy, planning tasks are represented by concepts in the ontology, while decomposition methods are represented by concept inclusions. Simple methods, i.e., $\mathcal{A} \mapsto \mathcal{B}$, are translated into axioms of the form $\mathcal{B} \sqsubseteq \mathcal{A}$. To represent more complex methods in a semantically correct way, the onlysome construct [10, 2] – written as $\mathrm{O}\mathit{includes}.[\mathcal{C}_1, \ldots, \mathcal{C}_n]$ – is applied, which represents a set of concepts $\mathbf{C} = \{\mathcal{C}_1, \ldots, \mathcal{C}_n\}$ connected by the role *includes*. A method $\mathcal{A} \mapsto_{\prec} \mathcal{B}_1, \ldots, \mathcal{B}_n$ for $n \geq 2$ is thus represented by $\mathrm{O}\mathit{includes}.[\mathcal{B}_1, \ldots, \mathcal{B}_n] \sqsubseteq \mathcal{A}$. The order $\prec$ specified for individual methods cannot be directly modeled in the ontology due to the tree-model property, but is instead stored as annotations. The ontology may also contain further axioms, representing preconditions and effects of actions, as well as additional domain knowledge, e.g., from specialized models of other components.

Having obtained a unified knowledge model in form of an ontology, retrieving models for the several components of the planner is straightforward by analyzing the subsumptions contained in the model. Most notably, this refers to the dialog model, which determines all possible interactions between the user and the planner. To support every possible interaction, the dialog model incorporates a dialog for every action and every decomposition method of the planning model. Using additional data – like textual descriptions, images and videos – the dialog management system can convey actions and plans suitably to the user. For further details concerning to the dialog model, we refer to Nothdurft et al. [12].

2.2 Extending the Planning Domain

We now consider an important benefit of using ontologies: reasoning. The reasoning task of classification computes all subsumptions between concepts in the ontology logically implied by its axioms. In keeping with the aforementioned analogy, we can interpret newly inferred subsumptions as new decomposition methods, which are automatically added to the planning domain. However, classification only generates subsumptions between named concepts and thus generates only decomposition methods with only a single task (i.e. $\mathcal{A} \mapsto \mathcal{B}$). In previous work, we have described a scheme that allows for inferring more complex decomposition methods by adding new concepts into the ontology [2]. Herein lies a significant advantage of our approach, as the planning model needs only to be specified partially while the reasoner creates most of the decomposition methods automatically. This eases the process of modeling the planning domain, as general descriptions of abstract tasks are used to determine which tasks can serve to achieve them. In Section 3 we will describe an example use-case.

2.3 Explanations

The ability to explain its behavior is an important capability of a companion system [13]. For a MIPS this is the ability to explain plans. Using the fact that the decomposition methods in the planning domain are based on concept subsumptions, we introduced a scheme (in [2]) to integrate plan- [15] and ontology-explanations [14] to improve on traditional plan explanation. To start with, plans contain causal and decomposition relationships that can be conveyed to the user. For example, a plan might include an exercise that serves to warm up a muscle needed by another exercise. Here, the plan explanation would state:

> "The runner's calf stretch is necessary as it ensures that the gastrocnemius muscle is warmed up, which is needed by the skip rope jumping."

Such relationships are verbalized by traversing the generated plan, extracting the formal relationships between its elements and by applying text templates to them. Decomposition relationships between tasks are verbalized similarly:

> “The cardio workout no. 1 is necessary, since it is part of the lower body training.”

Since these decomposition relationships are derived using background domain knowledge in the ontology, such assertions can be justified in more detail. This is the task of an ontology-explanation module, which provides a verbalization of the reasoning steps that serve to justify the decomposition relationship (for example, to explain why the example workout is considered a lower body training). For instance, consider the following (verbalized) fact in the ontology:

> “According to its definition, cardio workout no. 1 includes skip rope jumping and stationary bike exercise.”

Inference rules are used to infer intermediate facts from the relevant domain axioms, for example:

> “Skip rope jumping engages the gastrocnemius muscle, which is something that is part of the lower body. Therefore skip rope jumping engages something that is part of the lower body.”

The formal representation (proof trees) of such arguments is transformed into texts by applying patterns specified for each inference rule. A more thorough discussion of the underlying inference mechanism is provided in [14]. Whereas at current, the ontology-explanation module is used to extend plan explanations and implemented only as a prototype, such a service could potentially also be used to provide more general kinds of explanations related to the domain.

3 Possible Applications

In this section we elaborate on several scenarios where the proposed approach to use an ontology as the central knowledge base bears advantages. During the mixed-initiative planning process, users might request changes to the current plan. A possible request is to replace an action in a plan with another action. Users are often not able to precisely designate the new action, but instead to provide some description of it. For example, he or she might request to replace a weight-lifting exercise in the current plan with “a stamina exercise that does not use free weights and can be done while sitting”. Furthermore, these descriptions may not be in line with the descriptions of actions in the planning model or the referenced information may not be contained in the planning model at all. To handle the request, i.e., to identify the action the user had in mind, additional background knowledge stored in the ontology as well as its reasoning capabilities are utilized. Here, the idea is to offer the user the possibility to input descriptions in natural language, from which a concept expression is generated using techniques developed in the field of ontology learning [6]. For this concept expression, all subconcepts can be determined using an automated reasoner and presented to the user for selection. This set contains all concepts that fulfill the user’s description. Only direct subconcepts need to be considered, effectively grouping several more concrete tasks (i.e. actions) into a more general one. This set can subsequently be presented to the user as a selection, or an action is chosen by the planner according to some preference measure. A similar situation may occur if the user is asked to select a method to decompose an abstract task, e.g., “Which workout do you want to use for this strength training?” The planner has to deal with the possibility that the user is not content with any of the presented options. A standard MIPS would fail in this situation. Using the ontology and web-retrieval techniques, the system can browse the web to obtain new concepts describing matching workouts and integrate them into the ontology. As every component’s domain model in the system is derived from the central ontology, all these models can be updated synchronously and in a consistent way. Using the newly extended model additional options can be presented to the user.

The user might also propose decompositions himself, e.g., a new workout. This description can be translated into a tentative decomposition method $\mathcal{A} \mapsto_{\prec} \mathcal{B}_1, \ldots, \mathcal{B}_n$. Both the ontology and the planner can be utilized to determine whether the newly proposed method is valid in the context of the domain. First, using a reasoner it can be checked whether $\mathrm{O}\mathit{includes}.[\mathcal{B}_1, \ldots, \mathcal{B}_n] \sqsubseteq \mathcal{A}$ holds, which would be required if the methods were part of the ontology. This essentially performs a check of high-level domain constraints encoded in the ontology, e.g., lower-body workouts must only contain exercises for the lower body. If not, the method is rejected and the user should be provided with a suitable explanation for the rejection, potentially using ontology explanations. Thereafter the planner determines whether the new method fulfills certain legality constraints [4] ensuring that the new method is valid for the abstract one it decomposes. If the method has passed these tests it is added to the ontology as a legal method.

4 Conclusion

In this paper we have discussed how a previously developed technique to use an ontology as the single knowledge base for a planning-based companion system can be suitably applied to a MIPS such that the models for the several components can be extracted from it. We have outlined the basic principles of this approach and showed how two of its advantages – automatically extended planning domains and improved plan explanations – provide benefits for a MIPS. We argued how this integration of knowledge helps when offering choices to the user, illustrated different kinds of explanations such a system can offer, and outlined how such a system can be enabled to incorporate new elements into the planning domain on request (based on the user’s input or browsing the web).

Acknowledgments

This work was done within the Transregional Collaborative Research Centre SFB/TRR 62 “A Companion-Technology for Cognitive Technical System” funded by the German Research Foundation (DFG).

References

1. Ai-Chang, M., Bresina, J., Charest, L., Chase, A., Hsu, J.J., Jonsson, A., Kanefsky, B., Morris, P., Rajan, K., Yglesias, J., et al.: Mapgen: Mixed-initiative planning and scheduling for the mars exploration rover mission. Intelligent Systems, IEEE 19(1), 8–12 (2004)
2. Behnke, G., Ponomaryov, D., Schiller, M., Bercher, P., Nothdurft, F., Glimm, B., Biundo, S.: Coherence across components in cognitive systems - one ontology to rule them all. In: Proc. of the 25th Int. Joint Conf. on Artificial Intelligence (IJCAI). AAAI Press (2015)
3. Bercher, P., Höller, D., Behnke, G., Biundo, S.: User-centered planning – a discussion on planning in the presence of human users. In: Proc. of the Int. Symposium on Companion Technology (ISCT 2015) (2015)
4. Biundo, S., Schattenberg, B.: From abstract crisis to concrete relief (a preliminary report on combining state abstraction and HTN planning). In: Proc. of the 6th European Conference on Planning (ECP). pp. 157–168. AAAI Press (2001)
5. Byrne, R.: Planning meals: Problem solving on a real data-base. Cognition 5, 287–332 (1977)
6. Cimiano, P., Mädche, A., Staab, S., Völker, J.: Ontology learning. In: Handbook on Ontologies, pp. 245–267. Springer (2009)
7. Erol, K., Hendler, J.A., Nau, D.S.: UMCP: A sound and complete procedure for hierarchical task-network planning. In: Proc. of the Int. Conference on AI Planning & Scheduling (AIPS). pp. 249–254 (1994)
8. Geier, T., Bercher, P.: On the decidability of HTN planning with task insertion. In: Proc. of the 22nd Int. Joint Conf. on Artificial Intelligence (IJCAI). pp. 1955–1961. AAAI Press (2011)
9. Höller, D., Behnke, G., Bercher, P., Biundo, S.: Language classification of hierarchical planning problems. In: Proc. of the 21st European Conf. on Artificial Intelligence (ECAI). vol. 263, pp. 447–452. IOS Press, Amsterdam (2014)
10. Horridge, M., Drummond, N., Goodwin, J., Rector, A., Stevens, R., Wang, H.: The Manchester OWL syntax. In: Proc. of the Workshop on OWL Experiences and Directions (2006)
11. Myers, K.L., Jarvis, P., Tyson, M., Wolverton, M.: A mixed-initiative framework for robust plan sketching. In: Proc. of the 13th Int. Conf. on Automated Planning and Scheduling (ICAPS). pp. 256–266 (2003)
12. Nothdurft, F., Behnke, G., Bercher, P., Biundo, S., Minker, W.: The interplay of user-centered dialog systems and AI planning. In: Proc. of the 16th Annual Meeting of the SIG on Discourse and Dialogue (SIGDIAL). pp. 344–353. Association for Computational Linguistics (2015)
13. Nothdurft, F., Richter, F., Minker, W.: Probabilistic human-computer trust handling. In: Proc. of the 15th Annual Meeting of the SIG on Discourse and Dialogue (SIGDIAL). pp. 51–59. Association for Computational Linguistics (2014)
14. Schiller, M.R.G., Glimm, B.: Towards explicative inference for OWL. In: Proc. of the 26th Int. Workshop on Description Logics (DL). CEUR Workshop Proceedings (2013)
15. Seegebarth, B., Müller, F., Schattenberg, B., Biundo, S.: Making hybrid plans more clear to human users – a formal approach for generating sound explanations. In: Proc. of the 22nd Int. Conf. on Automated Planning and Scheduling (ICAPS). pp. 225–233. AAAI Press (2012)