Human-in-the-Loop Domain-Model Acquisition

Yuncong Li and **Hankz Hankui Zhuo**[*]
School of Data and Computer Science, Sun Yat-sen University, Guangzhou, China
liyc5@mail2.sysu.edu.cn, zhuohank@mail.sysu.edu.cn

Abstract

Using planning language, such as PDDL (Planning Domain Description Language), to build domain models from scratch is challenging for engineers, which impedes the applications of planning techniques in various domains. In this demonstration, we design a user-friendly visualized system to help alleviate the burden of building domain models, via (1) graph-based user interaction, and (2) consistency detection and model learning.

1 Introduction

Exploiting planning languages, such as PDDL [Fox and Long, 2003], to build domain models by hand is often difficult, even for domain experts, which impedes planning applications in various real-world domains (c.f. [Yang *et al.*, 2007; Zhuo *et al.*, 2010]). Automatically learning domain models from historical data can indeed help reduce domain modelling efforts (c.f. [Zhuo and Yang, 2014]). It requires, however, users are able to provide large enough structured training data (e.g., plan traces (c.f. [Zhuo and Yang, 2014])). On the other hand, the domain models learnt are generally not one-hundred percent accurate, i.e., they need to be further revised by users before being used to generate solutions to planning problems.

To alleviate users' burden of building domain models, we build a graph-based visualized user-interaction system to consider human-in-the-loop by integrating techniques of consistency detection (c.f. [Bacchus *et al.*, 2017]) and model learning. Different from our system, previous tools, such as itSIMPLE (c.f. [Vaquero *et al.*, 2013]) and VIZ (c.f. [Vodrázka and Chrpa, 2010]), do not consider effective consistency-detection techniques and model-learning approaches. We call our system KAVI, which stands for **K**nowledge **A**cquisition with **V**isualized **I**nteraction. The framework of KAVI is shown in Figure 1, which is composed of seven main components, i.e., domain visualized modelling, data convertor, domain knowledge base, consistency detection, plan generation, plan validation, model learning (or fine-tuning). We will introduce each component in detail in the following sections.

[*]Corresponding Author

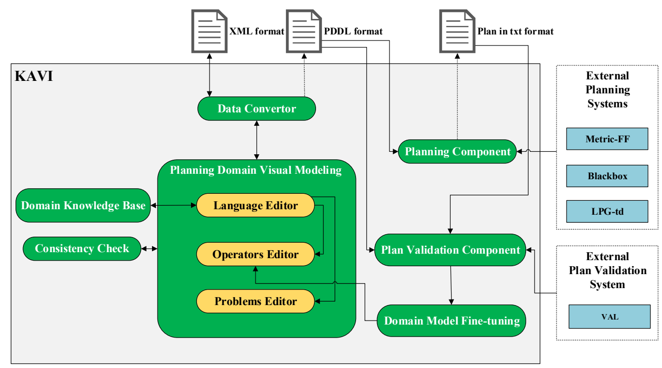

Figure 1: The architecture of KAVI

2 Graph-Based User Interactions

This component provides a graphical user interface for description of planning domains and problems, which uses simple diagrams to generate domain models represented with the PDDL language. We divide complex tasks into three levels (which is similar to VIZ (c.f. [Vodrázka and Chrpa, 2010])):

- defining *classes* and *predicates*;
- defining planning operators with *variables* and *predicates*;
- defining planning problems with *objects* and *predicates*.

To reduce the effort of specifying the above-mentioned three levels, we build a knowledge base, which is incrementally added, to automatically fill "parts" of domain models based on users' current input. The knowledge base can be categorized into two types (or templates):

The template of TYPE This type of template denotes a unique type in real-world applications.

The template of PREDICATE This type of template denotes a predicate with zero or more parameters in the form of "[*identifier*]([*spaces*][*parameter's type*])*". For example, (*at physobj place*) is a predicate template with *at* as the predicate's identifier, *physobj* and *place* as the types of two parameters.

Based on the knowledge base, when users input *classes* or *predicates* to our system, KAVI is capable of automatically completing other parts and drawing associated diagrams, as shown in Figure 2. When inputting “a”, our system will automatically recommend a predicate “(at physobj place)”. The

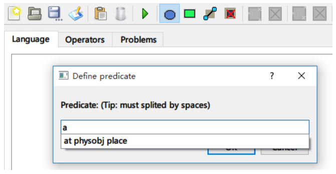

Figure 2: Auto-completion for predicate definition

result after automatically completing the definition of predicate “(at physobj place)” is shown in Figure 3.

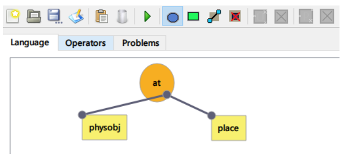

Figure 3: The result after automatically completing the definition of predicate “(at physobj place)”

3 Consistency Detection

When users defining predicates or action models, inconsistencies, such as missing arguments of predicates, conflicts on the PDDL model definition (e.g., predicates cannot be deleted and added simultaneously), etc., can be introduced into the system. We would like to automatically detect those inconsistencies immediately when they are introduced. To do this, we build a set of weighted constraints based on the knowledge base, current domain models, and the input information, and solve the constraints using off-the-shelf MAX-SAT solvers [Borchers and Furman, 1998; Bacchus *et al.*, 2017]. According to the solution of the MAX-SAT solver, we propose inconsistencies with highest weights for users to rectify.

4 Model Learning

Once domain models (which may be incomplete or noisy), initial states and goals are defined, we hope users can utilize off-the-shelf planners to generate plan solutions. We thus integrate three planners (i.e., metric-FF [Hoffmann, 2003], Blackbox [Kautz and Selman, 1998], LPG-td [Gerevini *et al.*, 2004]) into KAVI, which users can select to generate plan solutions. As domain models are noisy, plan solutions are often incorrect. We thus show the plan solutions for users to do adaptations. Users can swap actions, correct actions, add new actions and delete actions.

With domain models, initial states, goals, rectified plan solutions, we call a plan validation component VAL (c.f. [Howey *et al.*, 2004]) to visualize conflicts among domain models and plan solutions, and provide suggestions for solving the conflicts. Users can resolve the conflicts by rectifying either domain models or plan solutions based on the suggestions or their own domain knowledge. The process of rectification is shown in Figure 4, where the right column is the plan to be rectified, the top right part is the rectification suggestion, the middle right part is the action model to be rectified, and the bottom right part is the state before the action is executed.

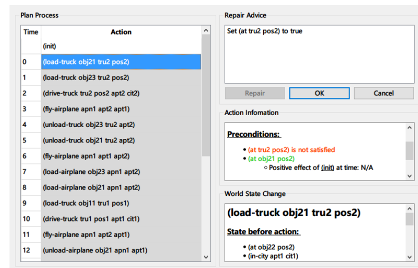

Figure 4: The visualized rectification of plans and domain models

After users’ rectification, if there are still conflicts, we view the plan solutions as new training data and build graphical models based on those new training data and current domain models. We learn the graphical models using an EM-style framework, as done by [Zhuo and Kambhampati, 2013]. We then convert the learnt graphical models to new models.

5 Final Remarks

In this demonstration, we design a novel system KAVI for building domain models based on visualized interaction, consistency detection, and model learning techniques. We exhibit that our KAVI system can indeed effectively build domain models by considering human-in-the-loop. In the future we would like to consider the following aspects:

- When using off-the-shelf planners to generate plan solutions, it is highly possible that there are no solutions generated given noisy domain models. In the future we will consider to exploit model-lite planners [Kambhampati, 2007; Zhuo and Kambhampati, 2017] to help generate plan solutions.
- Since plan solutions are generally “incorrect” based on noisy domain models, to reduce the burden of users rectifying plan solutions, we can exploit plan recognition approaches [Kautz and Allen, 1986; Tian *et al.*, 2016; Zhuo, 2017] to “preprocess” the plans (i.e., recognize the underlying correct plans), and show the recognized plans to users.

Acknowledgements

We thank the support of the National Key Research and Development Program of China (2016YFB0201900), National Natural Science Foundation of China (U1611262), Guangdong Natural Science Funds for Distinguished Young Scholar (2017A030306028), Pearl River Science and Technology New Star of Guangzhou, and Guangdong Province Key Laboratory of Big Data Analysis and Processing for the support of this research.

References

[Bacchus *et al.*, 2017] Fahiem Bacchus, Antti Hyttinen, Matti Järvisalo, and Paul Saikko. Reduced cost fixing in maxsat. In *Principles and Practice of Constraint Programming - 23rd International Conference, CP*, pages 641–651, 2017.

[Borchers and Furman, 1998] B. Borchers and J. Furman. A two-phase exact algorithm for MAX-SAT and weighted MAX-SAT problems. *J. Comb. Optim.*, 2(4):299–306, 1998.

[Fox and Long, 2003] Maria Fox and Derek Long. Pddl2.1: An extension to pddl for expressing temporal planning domains. *Journal of Artificial Intelligence Research (JAIR)*, 20:61–124, February 2003.

[Gerevini *et al.*, 2004] Alfonso Gerevini, Alessandro Saetti, Ivan Serina, and Paolo Toninelli. Lpg-td: a fully automated planner for pddl2. 2 domains. In *In Proc. of the 14th Int. Conference on Automated Planning and Scheduling (ICAPS-04) International Planning Competition abstracts*. Citeseer, 2004.

[Hoffmann, 2003] Jörg Hoffmann. The metric-ff planning system: Translating"ignoring delete lists"to numeric state variables. *Journal of artificial intelligence research*, 20:291–341, 2003.

[Howey *et al.*, 2004] Richard Howey, Derek Long, and Maria Fox. Val: Automatic plan validation, continuous effects and mixed initiative planning using pddl. In *Tools with Artificial Intelligence, 2004. ICTAI 2004. 16th IEEE International Conference on*, pages 294–301. IEEE, 2004.

[Kambhampati, 2007] Subbarao Kambhampati. Model-lite planning for the web age masses: The challenges of planning with incomplete and evolving domain theories. In *Proceedings of AAAI*, 2007.

[Kautz and Allen, 1986] Henry A. Kautz and James F. Allen. Generalized plan recognition. In *Proceedings of AAAI*, 1986.

[Kautz and Selman, 1998] Henry Kautz and Bart Selman. Blackbox: A new approach to the application of theorem proving to problem solving. In *AIPS98 Workshop on Planning as Combinatorial Search*, volume 58260, pages 58–60, 1998.

[Tian *et al.*, 2016] Xin Tian, Hankz Hankui Zhuo, and Subbarao Kambhampati. Discovering underlying plans based on distributed representations of actions. In *Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems, Singapore, May 9-13, 2016*, pages 1135–1143, 2016.

[Vaquero *et al.*, 2013] Tiago S Vaquero, José R Silva, Flavio Tonidandel, and J Christopher Beck. itsimple: towards an integrated design system for real planning applications. *The Knowledge Engineering Review*, 28(2):215–230, 2013.

[Vodrázka and Chrpa, 2010] Jindrich Vodrázka and Lukáš Chrpa. Visual design of planning domains. In *Proceedings of ICAPS 2010 workshop on Scheduling and Knowledge Engineering for Planning and Scheduling (KEPS)*, pages 68–69, 2010.

[Yang *et al.*, 2007] Qiang Yang, Kangheng Wu, and Yunfei Jiang. Learning action models from plan examples using weighted MAX-SAT. *Artificial Intelligence Journal*, 171:107–143, February 2007.

[Zhuo and Kambhampati, 2013] Hankz Hankui Zhuo and Subbarao Kambhampati. Action-model acquisition from noisy plan traces. In *IJCAI 2013, Proceedings of the 23rd International Joint Conference on Artificial Intelligence, Beijing, China, August 3-9, 2013*, pages 2444–2450, 2013.

[Zhuo and Kambhampati, 2017] Hankz Hankui Zhuo and Subbarao Kambhampati. Model-lite planning: Case-based vs. model-based approaches. *Artificial Intelligence*, 246:1–21, 2017.

[Zhuo and Yang, 2014] Hankz Hankui Zhuo and Qiang Yang. Action-model acquisition for planning via transfer learning. *Artificial intelligence*, 212:80–103, 2014.

[Zhuo *et al.*, 2010] Hankz Hankui Zhuo, Qiang Yang, Derek Hao Hu, and Lei Li. Learning complex action models with quantifiers and implications. *Artificial Intelligence*, 174(18):1540 – 1569, 2010.

[Zhuo, 2017] Hankz Hankui Zhuo. Human-aware plan recognition. In *AAAI*, pages 3686–3693, 2017.