École doctorale : MSTII

Doctorat

Comité de Suivi Individuel

Maxence GRAND

le 9 avril 2021

Apprentissage par démonstration de modèle d'actions pour la programmation de cobots

Directeur de thèse : **Damien PELLIER**
Co-encadrant de thèse : **Humbert FIORINO**

Examinateur Externe,	Olivier	BUFFET
Respondable de spécialité,	Nicolas	GAST

Table des matières

*

Chapitre 1

Introduction

Sommaire

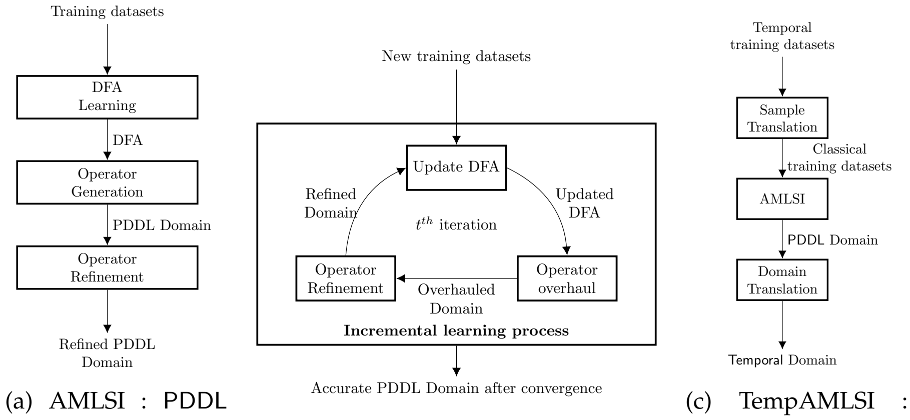
1.1 Context

1.2 Problem Statement

Hand-coding Planning domains is generally considered difficult, tedious and error-prone. The reason is that experts in charge modelling domains are not always PDDL experts and vice versa. To overcome this issue, two main approaches have been proposed. One is to develop knowledge engineering tools facilitating PDDL writing. These tools provide support for consistency and syntactic error checking, domain visualisation etc. An inconvenient aspect of these tools is that they require PDDL expertise and background in software engineering [1].

The other approach is to develop machine learning algorithms to automatically generate Pla domains as, for instance, ARMS [2], SLAF [3], LSONIO [**?**], LOCM [4]. These algorithms use training datasets made of possibly noisy and partial state/action sequences generated by a planner, or randomly generated. Classically, IPC benchmarks are used to generate training datasets. The performance of these algorithms is then measured as the syntactical distance between the learned domains and the IPC benchmarks. Machine learning approaches have three main drawbacks.

First, most of these approaches require a lot of data to perform the learning of Planning domains and in many real world applications, acquiring training datasets is difficult and costly, e.g., Mars Exploration Rover operations [**?**]. Also, dataset acquisition is a long term evolutive process : in real world applications, training data become available gradually over time. Moreover, in practice, it is important to be able to update learned PDDL domains to new incoming data without restarting the learning process from scratch.



(a) AMLSI : PDDL domain learning overview (b) IncrAMLSI : Incremental PDDL domain learning overview (c) TempAMLSI : Temporal domain learning overview

FIGURE 1.1 – Contributions overview

Second, the learned domains are not *accurate* enough to be used "as is" in a planner : a step of expert proofreading is still necessary to correct them. Even small syntactical errors can make sometime the learned domains useless for planning. Therefore, we consider that domain *accuracy*, that we define as the capacity of a learned domain to solve planning problems that were not used in the training dataset, is a better performance indicator than syntactical distance in practice.

Third, even if some approaches, e.g., [**?**, **?**, 5] are able to learn from noisy and/or partially observable data, few approaches are able to handle very high levels of noise and partial observations as can be encountered in real world applications, especially in robotics where observations are extracted using miscalibrated or noisy sensors.

1.3 The AMLSI Approach

To address these challenges, we propose a novel Planning domain learning algorithm called AMLSI (*Action Model Learning with State machine Interaction*). A key idea in AMLSI is that real world can be modelled as state machines and that retro-engineering real world state machines is analogous to learning a Deterministic Finite Automaton (DFA), which is equivalent to a regular grammar : we argue that (1) it is possible to learn a DFA by querying a real world state machine (see Figure **??**), and (2) to induce a PDDL domain from this regular grammar. AMLSI does not require any prior knowledge regarding the feasibility of actions in a given state, and state observations can be partial and noisy. AMLSI is highly accurate even with highly partial and noisy state observations. Thus it minimizes PDDL proofreading and correction for domain experts. Our experiments show that in many cases AMLSI does not even require any correction of the learned domains. AMLSI is lean and efficient on data consumption. It uses a supervised learning approach based on grammar induction. Training data are action/state (possibly partial and noisy) sequences labeled as feasible or infeasible. Both, feasible and infeasible action/state sequences are used by AMLSI to learn PDDL domains, thus maximizing data usability.

1.3.1 Observation Generation

AMLSI produces a set of observations by using a random walk. AMLSI is able to efficiently exploit these observations by taking into account not only the fact that some actions are applicable in certain states but also that others are not. More precisely, to generate the observations, AMLSI uses random walks by applying a randomly selected action to the initial state of the problem. If this action is feasible, it is appended to the current action sequence. This procedure is repeated until the selected action is not feasible in the current state. The feasible prefix of the action sequence is then added to the set of positive samples, and the complete sequence, whose last action is not feasible, is added to the set of negative samples. This generation method allow to maximize data usability in situations where obtaining training datasets is costly and/time-consuming.

1.3.2 Domain Learning

PDDL Learning An overview of the AMLSI [?, ?] algorithm is shown in Figure 1.1a. AMLSI is composed of three steps :

1. *DFA Learning :* AMLSI uses an alternative version of the AMLSI algorithm with the DFA learning algorithm proposed by [?].
2. *Operator Generation :* AMLSI generates the set of PDDL operators from the learned grammar DFA with the observed states.
3. *Operator Refinement :* AMLSI refines the operator preconditions and effects. This refinement steps is necessary to address partial and noisy observations.

Incremental Learning An overview of the IncrAMLSI algorithm [?] is shown in Figure 1.1b. IncrAMLSI learns incrementally the PDDL domain from incoming data and does not restart from scratch when new data become available at each iteration t. More precisely, IncrAMLSI consists in incrementally updating the PDDL domain with the new incoming training datasets available at iteration t to produce the new domain. This algorithm is made up of three steps :

1. *Update of the DFA (regular grammar)* with the DFA learning algorithm proposed by [?].
2. *Overhaul of the PDDL operators* in order to add new operators and remove preconditions and effects that are no longer compatible with positive and negative samples, and the updated DFA at iteration t.
3. *Refinement of the PDDL operators* as in AMLSI algorithm to deal with noisy and partial states in observations.

The incremental process is operated each time new training datasets are input and until convergence of the PDDL domain.

Temporal Learning Some planners, such as Crikey [?], solve Temporal Problems by using non-temporal planners. To that end, they convert Temporal Problems into PDDL planning problems, solve them with a non-temporal planner. Then they convert the classical plan into a temporal plan with rescheduling techniques. TempAMLSI [?] reuse this idea in order to learn Temporal domains : TempAMLSI learns a PDDL domain and then infer a Temporal domain. TempAMLSI (summarized in Figure - 1.1c) has three steps.

1. *Sample Translation :* After having generated the samples of *temporal* sequences (including both feasible and infeasible sequences), TempAMLSI translates them into non-temporal sequences.

2. *PDDL Domain Learning :* TempAMLSI uses AMLSI to learn an intermediate and classical PDDL domain.
3. *Domain Translation :* TempAMLSI translates the PDDL domain into a Temporal domain.

This algorithm is able to learn both Sequential and SHE accurate Temporal domains.

1.4 Document Organization

Première partie

Literature Review

Chapitre 2

Automated Planning

Chapitre 3

Regular Grammar Induction

Chapitre 4

Action Model Acquisition

Deuxième partie

Contributions

Chapitre 5

AMLSI - STRIPS Domain Learning

Chapitre 6

Incremental Learning

Chapitre 7

ADL Domain Learning

Chapitre 8

Temporal Domain Learning

Chapitre 9

HTN Domain Learning

Chapitre 10

Conclusion & Futur Work

Annexes

Bibliographie

[1] M. Shah, L. Chrpa, F. Jimoh, D. Kitchin, T. McCluskey, S. Parkinson, and M. Vallati, "Knowledge engineering tools in planning : State-of-the-art and future challenges," *Knowledge engineering for planning and scheduling*, vol. 53, p. 53, 2013.

[2] Q. Yang, K. Wu, and Y. Jiang, "Learning action models from plan examples using weighted max-sat," *Artificial Intelligence*, vol. 171, no. 2-3, pp. 107–143, 2007.

[3] D. Shahaf and E. Amir, "Learning partially observable action schemas," in *Proceedings of the National Conference on Artificial Intelligence*, vol. 21, p. 913, Menlo Park, CA ; Cambridge, MA ; London ; AAAI Press ; MIT Press ; 1999, 2006.

[4] S. N. Cresswell, T. L. McCluskey, and M. M. West, "Acquiring planning domain models using locm," *The Knowledge Engineering Review*, vol. 28, no. 2, pp. 195–213, 2013.

[5] C. Rodrigues, P. Gérard, and C. Rouveirol, "Incremental learning of relational action models in noisy environments," in *ILP 2010*, pp. 206–213, 2010.

[6] B. Claverie, B. L. Blanc, and P. Fouillat, "La cobotique," *Presses univ. de Bordeaux "Communication & Organisation"*, 2013.

[7] M. Ghallab, C. Knoblock, D. Wilkins, A. Barrett, D. Christianson, M. Friedman, C. Kwok, K. Golden, S. Penberthy, D. Smith, Y. Sun, and D. Weld, "Pddl - the planning domain definition language," 08 1998.

[8] M. Fox and D. Long, "PDDL2.1 : an extension to PDDL for expressing temporal planning domains," *J. Artif. Intell. Res.*, vol. 20, pp. 61–124, 2003.

[9] D. MCDERMOTT, *OPT Manual version 1.7.3*. 2005.

[10] E. P. Pednault, "Adl and the state-transition model of action," *Journal of logic and computation*, vol. 4, no. 5, pp. 467–512, 1994.

[11] B. D. Argall, S. Chernova, M. Veloso, and B. Browning, "A survey of robot learning from demonstration," *Robotics and autonomous systems*, vol. 57, no. 5, pp. 469–483, 2009.

[12] C. J. V. Rijsbergen, *Information Retrieval*. Butterworth-Heinemann, 2nd ed., 1979.

[13] D. Pellier and H. Fiorino, "Pddl4j : a planning domain description library for java," *Journal of Experimental & Theoretical Artificial Intelligence*, vol. 30, no. 1, pp. 143–176, 2018.

[14] M. Ghallab, D. Nau, and P. Traverso, *Automated Planning : theory and practice*. Elsevier, 2004.

[15] D. Pellier, *Contributions à la planification automatique pour la conduite de systèmes autonomes*. 2018.

[16] P. Dupont and L. Miclet, "Inférence grammaticale régulière : fondements théoriques et principaux algorithmes," tech. rep., 1998.

[17] E. M. Gold, "Language identification in the limit," *Information and control*, vol. 10, no. 5, pp. 447–474, 1967.

[18] S. M. Ali and S. D. Silvey, "A general class of coefficients of divergence of one distribution from another," *Journal of the Royal Statistical Society : Series B (Methodological)*, vol. 28, no. 1, pp. 131–142, 1966.

[19] J. R. Quinlan, "Induction of decision trees," *Machine learning*, vol. 1, no. 1, pp. 81–106, 1986.

[20] J. E. Hopcroft, *Introduction to automata theory, languages, and computation*. Pearson Education India, 2008.

[21] A. Cornuéjols, L. Miclet, and V. Barra, . *Apprentissage artificiel-3e édition : Deep Learning, Concepts et algorithmes*. 2018.

[22] P. Dupont, "Regular grammatical inference from positive and negative samples by genetic search : the gig method," in *International Colloquium on Grammatical Inference*, pp. 236–245, Springer, 1994.

[23] J. Oncina and P. Garcia, "Inferring regular languages in polynomial updated time," pp. 49–61, 1992.

[24] P. Dupont, "Incremental regular inference," in *International Colloquium on Grammatical Inference*, pp. 222–237, Springer, 1996.

[25] K. Lang, "Evidence driven state merging with search," *Rapport technique TR98–139, NECI*, vol. 31, 1998.

[26] D. Angluin, "A note on the number of queries needed to identify regular languages," *Information and control*, vol. 51, no. 1, pp. 76–87, 1981.

[27] R. Parekh, C. Nichitiu, and V. Honavar, "A polynomial time incremental algorithm for learning dfa," pp. 37–49, 1998.

[28] P. Garcia and E. Vidal, "Inference of k-testable languages in the strict sense and application to syntactic pattern recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 9, pp. 920–925, 1990.

[29] M. Kudo and M. Shimbo, "Efficient regular grammatical inference techniques by the use of partial similarities and their logical relationships," *Pattern recognition*, vol. 21, no. 4, pp. 401–409, 1988.

[30] P. Garcia, E. Segarra, E. Vidal, and I. Galiano, "On the use of the morphic generator grammatical inference (mggi) methodology in automatic speech recognition," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 4, no. 04, pp. 667–685, 1990.

[31] M. Mitchell, *An introduction to genetic algorithms*. MIT press, 1998.

[32] J. Pearl, *Probabilistic reasoning in intelligent systems : networks of plausible inference*. Elsevier, 2014.

[33] R. S. Sutton, "Learning to predict by the methods of temporal differences," *Machine learning*, vol. 3, no. 1, pp. 9–44, 1988.

[34] B. Bonet, H. Palacios, and H. Geffner, "Automatic derivation of memoryless policies and finite-state controllers using classical planners," in *Nineteenth International Conference on Automated Planning and Scheduling*, 2009.

[35] M. I. Jordan, "Serial order : A parallel distributed processing approach," in *Advances in psychology*, vol. 121, pp. 471–495, Elsevier, 1997.

[36] B. Borchers and J. Furman, "A two-phase exact algorithm for max-sat and weighted max-sat problems," *Journal of Combinatorial Optimization*, vol. 2, no. 4, pp. 299–306, 1998.

[37] M. Richardson and P. Domingos, "Markov logic networks," *Machine learning*, vol. 62, no. 1-2, pp. 107–136, 2006.

[38] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A training algorithm for optimal margin classifiers," in *Proceedings of the fifth annual workshop on Computational learning theory*, pp. 144–152, ACM, 1992.

[39] N. Ranasinghe and W.-M. Shen, "Surprise-based learning for developmental robotics," in *2008 ECSIS Symposium on Learning and Adaptive Behaviors for Robotic Systems (LAB-RS)*, pp. 65–70, IEEE, 2008.

[40] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on knowledge and data engineering*, vol. 22, no. 10, pp. 1345–1359, 2009.

[41] T. M. Mitchell, "Version spaces : A candidate elimination approach to rule learning," in *Proceedings of the 5th international joint conference on Artificial intelligence-Volume 1*, pp. 305–310, Morgan Kaufmann Publishers Inc., 1977.

[42] K. Driessens, J. Ramon, and H. Blockeel, "Speeding up relational reinforcement learning through the use of an incremental first order decision tree learner," in *Machine Learning : EMCL 2001, 12th European Conference on Machine Learning*, pp. 97–108, 2001.

[43] T. Ribeiro and K. Inoue, "Learning prime implicant conditions from interpretation transition," in *Inductive Logic Programming - 24th International Conference, ILP*, pp. 108–125, 2014.

[44] Y. Gil, "Learning by experimentation : Incremental refinement of incomplete planning domains," in *Machine Learning Proceedings 1994*, pp. 87–95, Elsevier, 1994.

[45] R. Simpson, "Gipo graphical interface for planning with objects," in *Proceedings of the International Conference for Knowledge Engineering in Planning and Scheduling*, Citeseer, 2005.

[46] R. M. Simpson, D. E. Kitchin, and T. L. McCluskey, "Planning domain definition using GIPO," *Knowl. Eng. Rev.*, vol. 22, no. 2, pp. 117–134, 2007.

[47] A. Arora, H. Fiorino, D. Pellier, and S. Pesty, "A review on learning planning action models for socio-communicative hri," *arXiv preprint arXiv :1810.09245*, 2018.

[48] M. Shah, L. Chrpa, F. Jimoh, D. Kitchin, T. McCluskey, S. Parkinson, and M. Vallati, "Knowledge engineering tools in planning : State-of-the-art and future challenges," *Knowledge engineering for planning and scheduling*, vol. 53, p. 53, 2013.

[49] X. Wang, "Learning by observation and practice : An incremental approach for planning operator acquisition," in *Machine Learning Proceedings 1995*, pp. 549–557, Elsevier, 1995.

[50] P. Gregory and A. Lindsay, "Domain model acquisition in domains with action costs," in *ICAPS*, pp. 149–157, 2016.

[51] P. Gregory and S. Cresswell, "Domain model acquisition in the presence of static relations in the LOP system," in *ICAPS*, pp. 97–105, 2015.

[52] S. Cresswell and P. Gregory, "Generalised domain model acquisition from action traces," in *ICAPS*, 2011.

[53] H. H. Zhuo, T. Nguyen, and S. Kambhampati, "Refining incomplete planning domain models through plan traces," in *Twenty-Third International Joint Conference on Artificial Intelligence*, 2013.

[54] T. L. McCluskey, N. E. Richardson, and R. M. Simpson, "An interactive method for inducing operator descriptions.," in *AIPS*, pp. 121–130, 2002.

[55] H. H. Zhuo, Q. Yang, D. H. Hu, and L. Li, "Learning complex action models with quantifiers and logical implications," *Artificial Intelligence*, vol. 174, no. 18, pp. 1540–1569, 2010.

[56] H. H. Zhuo and S. Kambhampati, "Action-model acquisition from noisy plan traces," in *Twenty-Third International Joint Conference on Artificial Intelligence*, 2013.

[57] K. Mourão, R. P. A. Petrick, and M. Steedman, "Learning action effects in partially observable domains," in *ECAI 2010 - 19th European Conference on Artificial Intelligence, Lisbon, Portugal, August 16-20, 2010, Proceedings*, pp. 973–974, 2010.

[58] K. Mourão, L. S. Zettlemoyer, R. P. A. Petrick, and M. Steedman, "Learning STRIPS operators from noisy and incomplete observations," in *Proceedings of the Twenty-Eighth Conference on Uncertainty in Artificial Intelligence, Catalina Island, CA, USA, August 14-18, 2012*, pp. 614–623, 2012.

[59] M. Molineaux and D. W. Aha, "Learning unknown event models," in *Twenty-Eighth AAAI Conference on Artificial Intelligence*, 2014.

[60] H. H. Zhuo, Q. Yang, R. Pan, and L. Li, "Cross-domain action-model acquisition for planning via web search," in *Twenty-First International Conference on Automated Planning and Scheduling*, 2011.

[61] T. J. Walsh and M. L. Littman, "Efficient learning of action schemas and web-service descriptions.," in *AAAI*, vol. 8, pp. 714–719, 2008.

[62] R. E. Fikes and N. J. Nilsson, "Strips : A new approach to the application of theorem proving to problem solving," *Artificial intelligence*, vol. 2, no. 3-4, pp. 189–208, 1971.

[63] A. Arora, H. Fiorino, D. Pellier, and S. Pesty, "Action model acquisition using sequential pattern mining," in *Joint German/Austrian Conference on Artificial Intelligence (Künstliche Intelligenz)*, pp. 286–292, Springer, 2017.

[64] H. H. Zhuo, "Crowdsourced action-model acquisition for planning," in *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

[65] H. H. Zhuo, J. Peng, and S. Kambhampati, "Learning action models from disordered and noisy plan traces," *CoRR*, vol. abs/1908.09800, 2019.

[66] D. Aineto, S. J. Celorrio, and E. Onaindia, "Learning action models with minimal observability," *Artif. Intell.*, vol. 275, pp. 104–137, 2019.

[67] D. Aineto, S. Jiménez, and E. Onaindia, "Learning STRIPS action models with classical planning," *CoRR*, vol. abs/1903.01153, 2019.

[68] S. J. Celorrio, T. de la Rosa, S. Fernández, F. Fernández, and D. Borrajo, "A review of machine learning for automated planning," *Knowledge Eng. Review*, vol. 27, no. 4, pp. 433–467, 2012.

[69] A. Arora, H. Fiorino, D. Pellier, and S. Pesty, "Action model acquisition using LSTM," *CoRR*, vol. abs/1810.01992, 2018.

[70] J. Á. Segura-Muros, R. Pérez, and J. Fernández-Olivares, "Learning numerical action models from noisy and partially observable states by means of inductive rule learning techniques," in *KEPS 2018*, pp. 46–53, 2018.

[71] N. Balac, D. M. Gaines, and D. Fisher, "Learning action models for navigation in noisy environments," in *ICML Workshop on Machine Learning of Spatial Knowledge*, 2000.

[72] N. Balac, D. M. Gaines, and D. Fisher, "Using regression trees to learn action models," in *Proceedings of the IEEE International Conference on Systems, Man & Cybernetics : "Cybernetics Evolving to Systems, Humans, Organizations, and their Complex Interactions"*, pp. 3378–3383, 2000.

[73] J. Kucera and R. Barták, "LOUGA : learning planning operators using genetic algorithms," in *PKAW 2018*, pp. 124–138, 2018.

[74] C. Rodrigues, P. Gérard, C. Rouveirol, and H. Soldano, "Incremental learning of relational action rules," in *The Ninth International Conference on Machine Learning and Applications, ICMLA 2010, Washington, DC, USA, 12-14 December 2010*, pp. 451–458, 2010.

[75] C. Rodrigues, P. Gérard, C. Rouveirol, and H. Soldano, "Active learning of relational action models," in *Inductive Logic Programming - 21st International Conference, ILP 2011, Windsor Great Park, UK, July 31 - August 3, 2011, Revised Selected Papers*, pp. 302–316, 2011.

[76] S. J. Celorrio, F. Fernández, and D. Borrajo, "The PELA architecture : Integrating planning and learning to improve execution," in *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence, AAAI 2008, Chicago, Illinois, USA, July 13-17, 2008*, pp. 1294–1299, 2008.

[77] T. Croonenborghs, J. Ramon, H. Blockeel, and M. Bruynooghe, "Online learning and exploiting relational models in reinforcement learning," in *IJCAI 2007, Proceedings of the 20th International Joint Conference on Artificial Intelligence, Hyderabad, India, January 6-12, 2007*, pp. 726–731, 2007.

[78] H. Pasula, L. S. Zettlemoyer, and L. P. Kaelbling, "Learning probabilistic relational planning rules," in *Proceedings of the Fourteenth International Conference on Automated Planning and Scheduling (ICAPS 2004), June 3-7 2004, Whistler, British Columbia, Canada*, pp. 73–82, 2004.

[79] H. M. Pasula, L. S. Zettlemoyer, and L. P. Kaelbling, "Learning symbolic models of stochastic domains," *J. Artif. Intell. Res.*, vol. 29, pp. 309–352, 2007.

[80] D. Martínez, T. Ribeiro, K. Inoue, G. Alenyà, and C. Torras, "Learning probabilistic action models from interpretation transitions," in *Proceedings of the Technical Communications of the 31st International Conference on Logic Programming (ICLP 2015)*, 2015.

[81] W. Shen, "Discovery as autonomous learning from the environment," *Mach. Learn.*, vol. 12, pp. 143–165, 1993.

[82] H. H. Zhuo, H. Muñoz-Avila, and Q. Yang, "Learning action models for multi-agent planning," in *10th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, pp. 217–224, 2011.

[83] A. Garland and N. Lesh, "Learning hierarchical task models by demonstration," *Mitsubishi Electric Research Laboratory (MERL), USA–(January 2002)*, 2003.

[84] H. H. Zhuo, D. H. Hu, C. Hogg, Q. Yang, and H. Munoz-Avila, "Learning htn method preconditions and action models from partial observations," in *Twenty-First International Joint Conference on Artificial Intelligence*, 2009.

[85] R. Garcia-Martinez and D. Borrajo, "An integrated approach of learning, planning, and execution," *Journal of Intelligent and Robotic Systems*, vol. 29, no. 1, pp. 47–78, 2000.

[86] B. Bonet and H. Geffner, "Learning first-order symbolic representations for planning from the structure of the state space," in *ECAI 2020 - 24th European Conference on Artificial Intelligence, 29 August-8 September 2020, Santiago de Compostela, Spain, August 29 - September 8, 2020 - Including 10th Conference on Prestigious Applications of Artificial Intelligence (PAIS 2020)*, pp. 2322–2329, 2020.

[87] S. Cresswell, M. Fox, and D. Long, "Extending tim domain analysis to handle adl constructs," *Knowledge Engineering Tools and Techniques for AI Planning : AIPS*, vol. 2, 2002.

[88] D. Rao and Z. Jiang, "Learning planning domain descriptions in RDDL," *Int. J. Artif. Intell. Tools*, vol. 24, no. 3, pp. 1550002 :1–1550002 :29, 2015.

[89] P. Verma, S. R. Marpally, and S. Srivastava, "Asking the right questions : Learning interpretable action models through query answering," in *Proc. AAAI*, 2021.

[90] A. Lindsay, S. Franco, R. Reba, and T. L. McCluskey, "Refining process descriptions from execution data in hybrid planning domain models," in *Proceedings of the Thirtieth International Conference on Automated Planning and Scheduling, Nancy, France, October 26-30, 2020*, pp. 469–477.

[91] S. Miglani and N. Yorke-Smith, "Nltopddl : One-shot learning of pddl models from natural language process manuals," in *ICAPS'20 Workshop on Knowledge Engineering for Planning and Scheduling (KEPS'20)*, ICAPS, 2020.

[92] W. Feng, H. H. Zhuo, and S. Kambhampati, "Extracting action sequences from texts based on deep reinforcement learning," in *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden*, pp. 4064–4070, 2018.

[93] M. Asai, "Unsupervised grounding of plannable first-order logic representation from images," in *Proceedings of the International Conference on Automated Planning and Scheduling*, vol. 29, pp. 583–591, 2019.

[94] T. S. Vaquero, J. R. Silva, F. Tonidandel, and J. C. Beck, "itsimple : towards an integrated design system for real planning applications," *Knowl. Eng. Rev.*, vol. 28, no. 2, pp. 215–230, 2013.

[95] T. S. Vaquero, V. Romero, F. Tonidandel, and J. R. Silva, "itsimple 2.0 : An integrated tool for designing planning domains," in *Proceedings of the Seventeenth International Conference on Automated Planning and Scheduling, ICAPS 2007, Providence, Rhode Island, USA, September 22-26, 2007*, pp. 336–343, 2007.

[96] A. González-Ferrer, J. Fernández-Olivares, L. Castillo, *et al.*, "Jabbah : a java application framework for the translation between business process models and htn," *Proceedings of the 3rd International Competition on Knowledge Engineering for Planning and Scheduling (ICKEPS'09)*, 2009.

[97] T. Plch, M. Chomut, C. Brom, and R. Barták, "Inspect, edit and debug pddl documents : Simply and efficiently with pddl studio," *System Demonstrations and Exhibits at ICAPS*, pp. 15–18, 2012.

[98] J. Vodrázka and L. Chrpa, "Visual design of planning domains," in *Proceedings of ICAPS 2010 workshop on Scheduling and Knowledge Engineering for Planning and Scheduling (KEPS)*, pp. 68–69, 2010.

[99] D. E. Smith, J. Frank, and W. Cushing, "The anml language," in *The ICAPS-08 Workshop on Knowledge Engineering for Planning and Scheduling (KEPS)*, 2008.

[100] J. Barreiro, M. Boyce, M. Do, J. Frank, M. Iatauro, T. Kichkaylo, P. Morris, J. Ong, E. Remolina, T. Smith, *et al.*, "Europa : A platform for ai planning, scheduling, constraint programming, and optimization," *4th International Competition on Knowledge Engineering for Planning and Scheduling (ICKEPS)*, 2012.

[101] O. Ilghami, D. S. Nau, H. Muñoz-Avila, and D. W. Aha, "Camel : Learning method preconditions for HTN planning," in *Proceedings of the Sixth International Conference on Artificial Intelligence Planning Systems, April 23-27, 2002, Toulouse, France*, pp. 131–142, 2002.

[102] O. Ilghami, H. Muñoz-Avila, D. S. Nau, and D. W. Aha, "Learning approximate preconditions for methods in hierarchical plans," in *Machine Learning, Proceedings of the Twenty-Second International Conference (ICML 2005), Bonn, Germany, August 7-11, 2005*, pp. 337–344, 2005.

[103] S. Yoon and S. Kambhampati, "Towards model-lite planning : A proposal for learning & planning with incomplete domain models," in *ICAPS2007 Workshop on Artificial Intelligence Planning and Learning*, 2007.

[104] A. Garrido and S. Jiménez, "Learning temporal action models via constraint programming," in *ECAI 2020 - 24th European Conference on Artificial Intelligence, 29 August-8 September 2020, Santiago de Compostela, Spain, August 29 - September 8, 2020 - Including 10th Conference on Prestigious Applications of Artificial Intelligence (PAIS 2020)*, pp. 2362–2369.

[105] S. J. Celorrio, T. de la Rosa, S. Fernández, F. Fernández, and D. Borrajo, "A review of machine learning for automated planning," *Knowl. Eng. Rev.*, vol. 27, no. 4, pp. 433–467, 2012.

[106] R. Jilani, A. Crampton, D. E. Kitchin, and M. Vallati, "Automated knowledge engineering tools in planning : state-of-the-art and future challenges," 2014.

[107] A. Arora, H. Fiorino, D. Pellier, M. Métivier, and S. Pesty, "A review of learning planning action models," *Knowl. Eng. Rev.*, vol. 33, p. e20, 2018.