Towards Transparent Planning and Legible Plan Representations — a Rule Based Planning Approach

Alan Lindsay

Automated Planning Lab,
Heriot-Watt University, Edinburgh, Scotland, UK
alan.lindsay@hw.ac.uk

1 Introduction

Within Explainable AI (XAI), XAI Planning [3] focuses on explanation for the various aspects of the AI Planning problem and process. Problems in XAI Planning include explaining the decision making of the planner or making it transparent and effectively communicating plans, including justifying plan actions. Although there has been some progress in providing access to some of the planner's decisions [12, 2],there is still a large gap between the current explanations and achieving transparency of the planning process.

In this work we take inspiration from a strategy that was proposed during early work in XAI: to start from approaches that are inherently explainable [5]. To this end we consider rule based policies (RBPs), which have proven effective for solving planning problems [1, 6, 16, 13, 8]. RBPs are based on individual rules that each identify a specific situation and the appropriate action to be taken in that situation. At each step of the plan, the selected action can be justified by the applied rule. These individual rules are combined into a single strategy, using a resolution strategy and previous work has demonstrated that a simple priority ordering is sufficient to capture effective planning strategies [8]. In this work we outline our case that RBPs can support transparent planning.

Planning model representations are not always appropriate for planning, instead planning models are designed so that plans provide a concise sequence of actions required to reach the goal. This can be insufficient for supporting effective RBPs, which build a complete strategy through incremental steps, incorporating both decisions and action. In past work this has been tackled through the enhancement of the planning model [6, 7, 11, 13]. Although these requirements are needed so the RBPs can build the plan incrementally, we observe that the problem of plan observing (or understanding) mirrors this problem and requires the observer to incrementally understand each of the plan actions.

In this work we consider transparency of the planning approach and legible plan representations. We firstly examine the enhancement of the planning model, which has two important functions in this work: it supports the concise representation of effective rules; and leads to different plan languages, which can result in plans that communicate both the underlying decisions, as well as the required plan steps. We then present RBPs as an appropriate action selection method for supporting transparent planning.

2 Background

A planning problem can be defined by a tuple, $\Sigma = \langle F, A, I, G \rangle$, with fluents, F, actions, A, initial state, I, and goals, G. Actions are represented by three sets of propositions: the precondition (a_{PRE}) and the add (a_{ADD}) and delete (a_{DEL}) effects. We use $s' = a(s)$ as a function that returns the state, s' after the application of action, a to state, s ($s' = (s \backslash a_{DEL}) \cup a_{ADD}$), defined for applicable actions, such that $a_{PRE} \subset s$. A macro action, $a_0, \ldots, a_n$, is a composition of actions, with the interpretation that they are applied in sequence. States are sets of propositions and the set of *reachable states* is defined as any state that can be reached from I through repeated application of applicable actions.

A policy, π, is a partial mapping from states to actions and a solution to a planning problem is a policy, such that it will lead a simple executive to the goal: $a_0 = \pi(I), s_1 = a_0(I) \, . \, \ldots \, . \, a_{n-1} = \pi(s_{n-1}), s_n = a_{n-1}(s_{n-1}) \, . \, g \subset s_n$. It is common to simply represent a solution as the actions that lead to the goal: $\langle \pi \rangle = a_0, \ldots, a_n$.

In this work we use an RBP [6] to compute the mapping of policies. An RBP is an ordered set of parameterised rules of the form: $\psi \rightarrow \mathbf{A}$, where ψ is a condition and $\mathbf{A}$ is the associated parameterised operator. The set of rules are attempted in order until there is a rule that unifies with the current state and goal. For example, the following rules demonstrate a strategy for a simple transportation problem (where trucks are used to redistribute packages to specific goal locations):

1. Drop-off a package at its goal
2. Pickup a misplaced package
3. Move to pickup a misplaced package
4. Move to drop-off a package at its goal
5. Move a truck home

A deterministic choice is made between multiple bindings. We adopt the rule representation that has been used in several previous works [6, 7, 11, 4] and an example is presented in Figure 2.

3 Enhancing the Planning Model

Following [8], we consider a hierarchy of model enhancements, $\Sigma_0, \ldots, \Sigma_n$, that starts with the modelled domain (Σ_0) and incrementally adds task level concepts into the model. These concepts aim to progressively modify the resulting plan structures, so that the plan mirrors the series of conceptual steps required to build the plan. For example, consider the simple transportation problem with two trucks and a package in Figure 1. In the plan for Σ_0, the first action in the

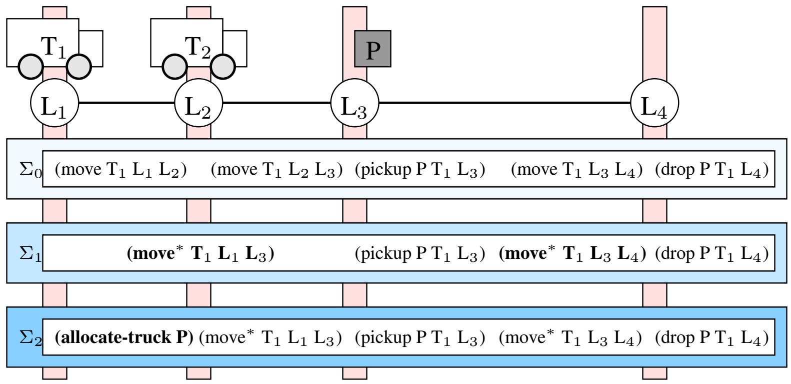

Fig. 1. An enhancement hierarchy, $\Sigma_0, \Sigma_1, \Sigma_2$, for a transportation domain. The first enhancement (Σ_1) abstracts the move actions between all locations pairs that are connected. The second enhancement (Σ_2) introduces resource allocation actions.

plan is to move T_1. However, this action has been preceded by the decision of which truck to allocate to the delivery task. A plan can provide a much richer record of the choices made during the planning process, which could provide more explicit justification for each plan step. For example, in the transportation task, the model can explicitly represent the allocation of a truck to the package. This is realised in the enhancement step from Σ_1 to Σ_2, which introduces bookkeeping propositions that record which truck has been allocated to deliver package, P. The key to selecting the appropriate hierarchy in the context of RBPs is that an effective RBP can be supported and that the model can be computed efficiently.

3.1 Realising the Enhancements

There have been various approaches to extending planning domain models in this context, including extending by hand [6], learning [13, 8] and selecting through domain analysis [11, 8]. In the case of [11] (domain analysis), key classes of planning domain subproblems (e.g., manipulation, transportation), were identified. Appropriate enhancements were designed for the class of subproblem (e.g., resource allocation and graph abstractions) and invoked in problems that contain the subproblem [11].

Plan Execution The model extensions add structure into the model. Although the same sequences of original actions will typically be possible during planning, the added structure may lead to complications at execution time. For example, if a truck breaks down and cannot service one of its deliveries, the commitments captured in the model will prevent an alternative truck being used. This provides a key opportunity to add explanations into the planning problem that can account for altered circumstances and changed decisions. In particular, one or more explanation actions (with high cost) can be added into the domain that allow resource commitments to be removed.

```
(:rule move_to_drop_off_package
  :parameters (?truck - truck ?obj - obj
               ?loc_from ?loc_to - location)
  :condition (and (in ?obj ?truck) (at ?truck ?loc_from))
  :goalCondition (and (at ?obj ?loc_to))
  :action (move* ?truck ?loc_from ?loc_to))
```

Fig. 2. The parameterised move to drop-off package rule. The rule selects to move a truck to ?*loc_to*, if there is a package in the truck and its goal is ?*loc_to*.

3.2 Plan Legibility

We are interested in investigating how the plan representation can impact on plan legibility. In particular, whether being more explicit about planning decisions makes the implicit structure of plans more understandable. In terms of plan understanding, the exact steps on the hierarchy might be different. However, similar concepts have appeared useful in the context of plan explanation [9, 10]. One possible avenue is to consider how the alternative plan languages can be used in the context of plan verbalisation [14, 15]. For example, how alternative plan descriptions generated using different hierarchy levels can be used within the framework in [14].

4 Transparent Rule Based Planning

The enhanced model, developed in the previous section, is not only supported by rule based planning approaches, it can benefit the performance and applicability of rule based planning approaches [8]. Moreover, the enhanced model supports individual rules to be used to incrementally make decisions about the used strategy. In this section we consider the transparency provided by rule based planning and how rule based action selection can support plan legibility.

Transparency of the Rules Each rule is associated with an applicability condition, which is a pair of conjunctions of predicates (the state and goal conditions). An example, is presented in Figure 2. For a rule to fire, each of its conditions must be satisfied. It is therefore straightforward to demonstrate whether a rule is applicable or not by considering each condition in turn. For example, using colour coding to annotate the condition (e.g., green for satisfied conditions and red for unsatisfied).

Transparency of the Planner The rule-based planner that we have selected uses a simple action selection process. A simple executive performs repeated actions, operating from a generalised policy, it performs the following actions:

1. look up the current state in the policy;

2. apply the mapped action;
3. repeat (until goal).

The rule selection uses a priority ordering, which means that the first applicable rule is used to select the next action. As such, each action that is applied is directly attributable to the proposing rule and the proposing rule provides the justification for selecting the action.

4.1 Plan Representation and Rule Annotations

We now consider the application of a rule system to the simple transportation problem presented in Figure 1 (the goal requires package P at L_4). We use the RBP described in the Background section, with an additional rule that allocates trucks to unallocated misplaced packages (this becomes the first rule). In this case there is a single package to allocate and once a truck is allocated to the package, the truck is moved to reposition the package at its goal location. The plan has 5 steps and the following table lists the fired rule and the selected action for each step (parameters are omitted for presentation).

Step	Fired Rule	Selected Action
1	**allocate_truck_to_misplaced_package**	`allocate-truck`
2	**move_to_pickup_misplaced_package**	`move*`
3	**pickup_misplaced_package_with_allocated_truck**	`pickup`
4	**move_to_drop_off_package**	`move*`
5	**drop_package_at_goal**	`drop`

Notice that each plan actions can be used for different reasons. In isolating the appropriate circumstance for the rule to fire, each rule provides more context, that narrows down the reason for the action's use. We also note that each of these rules could be associated with simple textual descriptions that makes use of the parameters and provides a more accessible description of the rule intention.

5 Conclusion

In this work we have examined the potential of using a rule based planning approach to provide transparency in planning decisions. We first examined the use of model enhancements and how these led to plans that capture a much richer record of the choices made during the planning process. In this work the model enhancements are used in combination with the rule based planning approach, allowing concise expression of rules. In particular, each rule is used to capture the appropriate course of action in a specific scenario. Each action selection is associated with a rule, which provides a traceable and transparent record of each decision made to construct the plan. Our future work will investigate whether rule based policies are transparent to users in practice (e.g., in the context of learned rule systems and with appropriate visualisations). We will also consider how the plan representation level can be varied to select more legible plans.

6 Acknowledgements

Funded by the ORCA Hub (orcahub.org), under EPSRC grant EP/R026173/1.

References

1. Bacchus, F., Kabanza, F.: Using temporal logics to express search control knowledge for planning. Artificial Intelligence **116**(1), 123–191 (2000)
2. Eifler, R., Steinmetz, M., Torralba, A., Hoffmann, J.: Plan-space explanation via plan-property dependencies: Faster algorithms & more powerful properties. In: Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20 (2020)
3. Fox, M., Long, D., Magazzeni, D.: Explainable planning. arXiv preprint arXiv:1709.10256 (2017)
4. Galea, M., Humphreys, D., Levine, J., Westerberg, H.: Evolutionary-based learning of generalised policies for AI planning domains. In: Proceedings of the Genetic and Evolutionary Computation Conference (2009)
5. Gunning, D., Aha, D.: Darpa's explainable artificial intelligence (xai) program. AI Magazine **40**(2), 44–58 (Jun 2019)
6. Khardon, R.: Learning action strategies for planning domains. Artificial Intelligence **113**(1-2), 125–148 (1999)
7. Levine, J., Humphreys, D.: Learning action strategies for planning domains using genetic programming. In: Proceedings of the 4th European Workshop on Scheduling and Timetabling (EvoSTIM 2003) (2003)
8. Lindsay, A.: Problem Models for Rule Based Planning. Ph.D. thesis, Department of Computer and Information Sciences, Strathclyde University, UK (2015)
9. Lindsay, A.: Towards exploiting generic problem structures in explanations for automated planning. In: Proceedings of the 10th International Conference on Knowledge Capture (2019)
10. Lindsay, A.: Using generic subproblems for understanding and answering queries in XAIP. In: ICAPS 2020 Workshop on Knowledge Engineering for Planning and Scheduling (KEPS) (2020)
11. Lindsay, A., Fox, M., Long, D.: Lifting the limitations in a rule-based policy language. In: Proceedings of the 22nd International Florida Artificial Research Society Conference (2009)
12. Magnaguagno, M.C., Pereira, R.F., Móre, M.D., Meneguzzi, F.: WEB PLANNER: A tool to develop classical planning domains and visualize heuristic state-space search. In: Proceedings of the Workshop on User Interfaces and Scheduling and Planning, UISP. pp. 32–38 (2017)
13. de la Rosa, T., McIlraith, S.A.: Learning Domain Control Knowledge for TLPlan and Beyond. In: Proceedings of the International Conference on Automated Planning and Scheduling, Workshop on Planning and Learning (PAL) (2011)
14. Rosenthal, S., Selvaraj, S.P., Veloso, M.: Verbalization: Narration of autonomous robot experience. In: Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence. IJCAI'16, AAAI Press (2016)
15. Sridharan, M., Meadows, B.: Towards a theory of explanations for human–robot collaboration. KI-Künstliche Intelligenz **33**(4), 331–342 (2019)
16. Yoon, S., Fern, A., Givan, R.: Learning control knowledge for forward search planning. Journal of Machine Learning Research **9**, 683–718 (Jun 2008)