Using Generic Subproblems for Understanding and Answering Queries in XAIP

Alan Lindsay
Department of Computer Science, Heriot-Watt University, Edinburgh, Scotland, UK
alan.lindsay@hw.ac.uk

Abstract

The relative transparency and flexibility of the planning process have been identified as key aspects suggesting that Automated Planning (AP) is well positioned to make an important contribution in Explainable AI. However, there is still a wide gap between explanations that can be directly extracted from AP models and effective explanations. There are a growing number of frameworks that are considering the problem from both the user side, where it is interesting to understand the form that an explanation might take; as well as the planner side, which must be able to explain various decisions and related properties. So far approaches have focused on single domain settings, where substantial domain specific content is produced, or at a general level, where only abstract planning concepts can be used. We aim to develop an abstraction layer that sits between these and exploits the often overlapping concepts and structures that exist between many planning domains. We propose exploiting domain analysis techniques in order to identify common roles and generic problem structures (GPSs). By attaching the concepts used for explanation to these structures we can exploit the contextual information supported by the structure, and also reduce the burden of constructing explanations in domains where these structures exist. In this work we explore the opportunities for exploiting GPSs in Explainable Planning.

Introduction

The increased adoption of AI Planning in real world applications and the drive to see wider adoption has led to more focus on issues around co-operation, transparency and trust. These are particularly relevant when humans are involved in the process and must be persuaded of the appropriateness of a particular plan. Explainable Planning (XAIP) (Fox, Long, and Magazzeni 2017) is a growing focus within planning, which aims to address these problems.

There is currently a gap between domain tailored explanations, such as the augmented reality system presented in (Chakraborti et al. 2017a) and general approaches based on planning concepts, such as (Magnaguagno, Pereira, and Meneguzzi 2016; Magnaguagno et al. 2017). Although tailoring explanations for each domain individually will lead to effective explanations, this is effort intensive, increasing the modelling burden. At the other extreme, basing explanations on general planning concepts is a general approach, but the resulting explanations are not generally understandable.

It was observed in the early days of the international planning competition (ICP)[1] that the majority of planning benchmarks could be grouped into common structures and themes (Helmert 2001), e.g., transportation (Driverlog etc.) and manipulation problems (Blocksworld etc.). The benchmark domains have continued to develop in complexity and diversity; however, key concepts such as traversal and resource management are still prevalent in planning domains, including real world applications, e.g, urban traffic control (Vallati et al. 2016), autonomous underwater vehicle missions (Cashmore et al. 2017) and mining operations (Lipovetzky et al. 2014).

We aim to exploit this commonality of structure within planning problems in order to support the functions of an explanation system. In this work we investigate how *generic problem structures* (GPSs) such as resource management and transportation, can support interpreting and disambiguating a user query, as well as providing context for generating specialised explanations. We demonstrate that explanations can both exploit the context of the GPS and be instantiated for a specific situation, greatly reducing redundancy and allowing the effort of developing explanations to be used effectively. An early version of this work was presented in (Lindsay 2019).

In the following sections we first provide the background in XAIP and domain analysis (for identifying GPSs), we then explore the role of GPSs in query interpretation and explanation construction. We present the framework for our system and some preliminary results, then we present the related work and finish with our conclusions and proposed future work.

Background

We first introduce the area of XAIP and then present supporting work in domain analysis.

XAIP

The aim of XAIP is to allow a user or observer to understand how a planning system made a certain decision or why it behaved in a certain way. In (Madumal et al. 2018) they have examined examples of explanation dialogue and used them to construct an explanation dialog model, which provides a

[1] www.icaps-conference.org/index.php/Main/Competitions

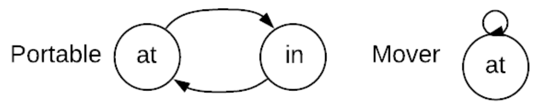


Figure 1: Movers (e.g., trucks) transition between different located properties and Portables (e.g., packages) transition between being located and being in a Mover.

general model for an explanation. Their model is formed of two main sections: for explanation dialog and argumentation dialog and includes looping, where further clarifications are sought or arguments are presented. In (Madumal et al. 2018) they predict that basing explanation systems for AI systems on human explanation will lead to more intuitive and understandable explanations for humans.

As part of the ground work in the area, Fox et al. posed 6 question types to frame the likely user queries for XAIP (Fox, Long, and Magazzeni 2017):

- Q1: Why did you do that?
- Q2: And why didn't you do something else (that I would have done)?
- Q3: Why is what you propose to do more efficient/safe/cheap than something else (that I would have done)?
- Q4: Why can't you do that?
- Q5: Why do I need to replan at this point?
- Q6: Why do I not need to replan at this point?

There has been some progress towards addressing these questions. As observed in (Fox, Long, and Magazzeni 2017), there is existing work that provides justification for why problems cannot be solved and therefore provides a good starting point for answering questions of type Q4, e.g., (Göbelbecker et al. 2010). In (Borgo, Cashmore, and Magazzeni 2018) they amalgamate questions Q1 and Q2 and explore a specific form of this question.

Domain Analysis: Identifying GPSs

TIM (Fox and Long 1998) is a domain analysis tool that uncovers certain (traditionally) implicit properties of a domain model, such as invariants and the type hierarchy. TIM constructs a collection of finite state machines (FSMs) that describe how the objects in a problem can transition between different sets of relationships, e.g., Portables (packages) can transition between `in` and `at` relationships (Figure 1).

Identifying GPSs In (Fox and Long 2001) it was demonstrated that this analysis can be extended and used to identify generic roles and generic problem structures (GPSs). They demonstrate that common aspects of planning models such as transportation and resource management can be identified within the planning problem model's structure. This has supported the effective description of specialised solutions to various problems including improving weak heuristics (Coles and Smith 2006), inducing specialised heuristics (Fox and Long 2001) and model extensions for control knowledge (Lindsay, Fox, and Long 2009). In particular, the solution can be specified at the level of the GPS, but can be instantiated and exploited at the specific instance level.

- Q7: Why did you go through L3 on route from L1 to L2 and not L4?
- Q8: Why is package P1 allocated to Truck1 and not Truck2?
- Q9: Why were packages P1 and P2 delivered by Truck2 and not transferred to Truck2 at L1?

Figure 2: An example of three queries for a Transportation domain that require concepts that will typically be missing from the domain (i.e., a route, an allocation of trucks to packages and a redistribution hub).

The key to the success of these approaches is that there is a clear description of whether the solution is appropriate for the current problem. This description is captured by a set of criteria, called a *fingerprint*, which establishes the requirements of the GPS (Long and Fox 2002). For example, a Transportation GPS is characterised by sets of Movers and Portables. A Mover exhibits a graph traversing behaviour, which is characterised by an FSM with a singleton state and a single self-transition on that state (illustrated in Figure 1). Certain objects, called Portables (transportable objects), can be attached to these Movers and moved to other locations. Portables are characterised by an FSM with two states that are linked to each other in both directions (illustrated in Figure 1). If a problem domain satisfies the fingerprint then the domain exhibits the associated GPS.

Exploiting GPSs in query interpretation

Planning domain models capture considerable knowledge about the problem domain. They are represented in a structured format, typically using human readable labels for their components (e.g., actions, predicates and types) that are meaningful in the problem domain. The planning model therefore presents a starting point for interpreting user queries, which are also likely to reference concepts from the problem model. This connection between planning as a knowledge representation and as an operational definition of a problem is not typically exploited. In (Siddle et al. 2017), clinical guidelines were mapped onto the planning operators, as part of a visualisation pipieline and in (Porteous et al. 2020; 2015) they exploited linguistic resources (such as Wordnet and Conceptnet) in partnership with domain analysis in order to propose model extensions for Interactive Storytelling domain models. Existing work attempting to map from natural language to planning models has used various supporting information: training examples (described in the target formalism) (Goldwasser and Roth 2014) and common sense knowledge (Siddle et al. 2017), or otherwise periods of observations, e.g., (Oates 2001). This requires substantial domain specific structured content or time in order to support the interpretation of the user's intentions. In this section we discuss the use of GPSs, which support a behavioural layer that can support mapping and disambiguation.

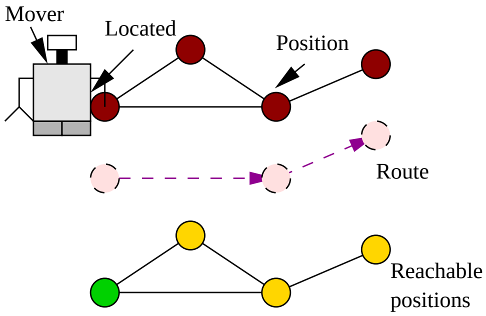


Figure 3: Key concepts for a Traversal subproblem.

Increasing the known concepts

Connecting the user query with the planning model relies on appropriate target entities existing in the planning model or these must be generated as part of the mapping process. Of course the user is likely to use concepts that are not modelled in the planning model, e.g., see Figure 2. We observe that through increasing the number of interpretations that we have of our domain model GPSs can reduce this problem of language acquisition. In this section we consider contrastive queries of the form:

- QS1: Why X and not Y?

This template structure might be used to ask a variety of questions (see Figure 2) and it is clear that creating a mapping from e.g., query, Q7, onto a meaningful query in terms of concepts at the general planning level, e.g., actions, propositions and objects, is not trivial.

Figure 3 presents important concepts defined for the Traversal subproblem (Lindsay 2015), including route, movement and reachability. More generally, the key concepts used in the queries Q7-9: route, resource allocation and distribution hubs, have all been defined for the Transportation subproblem (Lindsay 2015). Of course interpreting these queries does not come for free when a subproblem is identified. We are currently extending the capabilities of the GPS layer so that we can organise the relevant knowledge content, including concepts and reference structures.

Our approach is to organise the parsing of user queries by allowing the GPSs to define relationships between different concepts. For example, the following template defines a *Path Option*, which relates a specific movement (i.e., part of a route) and a specific location within that movement:

```
@relation: Path Option
 movement: Movement
 through:  Position
```

Position is a concept of the GPS and is automatically mapped to the relevant problem constants. In the case of Movement we provide alternative descriptions to mirror the different ways that movements can be described (e.g., endpoints, as used in Q7 or as a sequence). These relationships are used to match parts of the query to concepts during parsing. Each GPS reports the templates that it can interpret by specifying the type of concepts for each of the template's slots. For example, the Traversal subproblem will register the template:

- QS2: Why ⟨Path Option⟩ and not ⟨Location⟩?

This template could be used to parse Query Q7. Notice that although this template is defined for a relatively precise query, each time a Traversal subproblem exists in a domain then the concepts are automatically defined and queries like this can be interpreted.

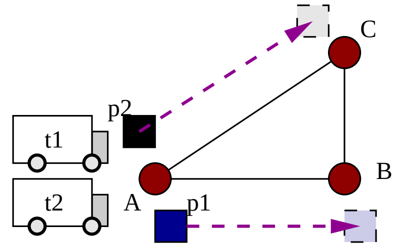


Figure 4: Part of the Transportation problem from (Borgo, Cashmore, and Magazzeni 2018).

Query disambiguation

In (Miller 2018a) it is observed that explanations are often contrastive, in the sense that the explanation should be relative to a provided alternative, reducing some of the ambiguity of the query (Miller 2018b). In (Borgo, Cashmore, and Magazzeni 2018) they developed an approach for resolving a specific form of the contrastive query, QS1:

- Q10: Why does the plan contain action A rather than action B (that I would expect)?)

Although the question is posed in a convenient form for answering using general concepts of planning problems, (i.e., it is expressed in terms of planning actions) they identify 4 possible interpretations of the user's intentions of a query of this form and there may be more. So, even when the elements that the query is referring to can be clearly identified in the underlying model, there is a lot of ambiguity in how to interpret the query.

A transportation problem is used in (Borgo, Cashmore, and Magazzeni 2018) to demonstrate their system. The key parts of this problem are illustrated in Figure 4. In the example two package deliveries must be made and there are two trucks that can be used for the deliveries. The initial plan proposed by the planner uses $t2$ to deliver $p1$ and $t1$ to deliver $p2$. An example user query is used: why the action A=(`load-truck` $p1$ $t2$ a) is in the plan and not an alternative: B=(`load-truck` $p1$ $t1$ a). They propose 4 alternative interpretations of the query. Each has an associated approach for generating a plan that includes action B; e.g., planning from the initial state and forcing the planner to use action B somewhere in the plan. In each case there is discussion around the planner's response to the remodelled problem that stems from the ambiguity of what the user intended.

We observe that one interpretation is that the user is querying why the planner did not use $t1$ to deliver $p1$. The Transportation subproblem defines concepts such as resource allocation and its relationship with the pickup and

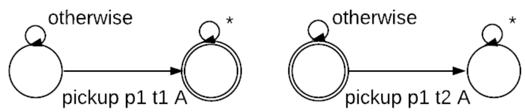


Figure 5: FSMs for forcing and preventing specific actions.

drop-off operators. As part of the fingerprint matching process, these *generic operator roles* are themselves related to the specific operators if the domain contains a Transportation subproblem. This provides the context to identify that the pair of actions are alternative loading actions and that the query can be interpreted as relating to the resource allocation procedure: In this case the allocation of $t1$ to $p2$ and $t2$ to $p1$. The system can then propose a return question:

- SQ1: Are you suggesting allocating truck, $t1$, to deliver $p1$ and $p2$?.

Depending on the complexity of the situation there might be several clarifications generated by one or more subproblems. When a clarification aligns with the user's intended query the subproblem that generated the clarification can play an active role in constructing the appropriate explanation.

GPSs for Explanation as a Service

In (Cashmore et al. 2019) they outline an approach to XAIP as a service: a framework for explanation generation as a wrapper around the planning system. Their approach is to convert the user's query into a set of constraints that are added to the existing planning model. For example, for a Q10 query they might propose making a constraint that insists on B being in the plan and prevents A. A new model is then constructed with the constraints and used to create a new plan. The comparison between the plans is then used to answer the original query. In this section we consider how GPSs can support this approach.

Specifying the Appropriate Constraint

The approach in (Cashmore et al. 2019) relies heavily on appropriate constraints being added into the model, however, the generated constraints act only on the specific actions mentioned in the user query. This limits the control that they can have over the plans that are generated. The GPS layer provides a context to construct more specific constraints.

Forcing a specific action to appear in any plan, or preventing an action from appearing in any plan can be achieved by representing the constraints as FSMs and compiling them into the domain model. For example, Figure 5 presents an FSM to force that t_1 picks up p_1 from location A and an FSM to prevent t_2 from picking up p_1 from location A. The FSMs are encoded in PDDL by adding predicates to represent the FSM states and extending the action effects so that they implement the corresponding FSM transitions.

Consider again the package truck allocation problem discussed in (Borgo, Cashmore, and Magazzeni 2018) and illustrated in Figure 4. The approaches in (Cashmore et al. 2019; Borgo, Cashmore, and Magazzeni 2018) will focus on the 2 pickup actions mentioned in the user query. However, the context of the Transportation subproblem allows a richer interpretation, where we can also consider the wider implications of resource allocation. From the Transportation subproblem context we can consider adding constraints to both pickups and drop-off actions, providing a deeper control over the next plan generated.

In the context of this query it seems likely that the original resource allocations should remain. This can be achieved by forcing all of the existing pickup and drop-off actions and preventing other trucks from initial position pickups and goal drop-offs. However, there might be more subtle situations where there are more than one appropriate set of constraints. For example, a more complex transportation problems with redistribution of packages at hub nodes etc. In cases like this the GPS layer provides an ideal place for incorporating additional supporting knowledge and reasoning (e.g., common sense reasoning) in order to select appropriate default behaviours and alternative options for the user.

Exploiting GPSs in explanation construction

The existing approaches for generating explanations include general approaches to proposing excuses for failure to find a plan, e.g., (Göbelbecker et al. 2010) and approaches to visualising state spaces, e.g., (Magnaguagno et al. 2017). However, the explanations are generated using the context supported by general planning concepts, which limits the depth of their interpretation. Similarly, an explanation that is based on AP algorithm concepts, such as heuristic value, is not going to be typically understood by a user (Fox, Long, and Magazzeni 2017). In this section we show how GPSs can be used to support the generation of explanations that exploit both the concepts defined by the GPS and its connection to the implementing domain.

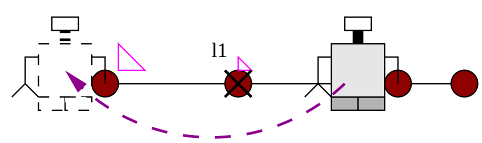


Figure 6: An example Grid problem where the robot cannot reach its goal.

Textual Explanations

In this section we will consider Q4 type queries, related to unsolvability. For example, consider the situation of the robot traversing in the Grid domain presented in Figure 6. The approach presented in (Göbelbecker et al. 2010) will generate an excuse such as: change `(not(open` $l1$`))` to `(open` $l1$`)` in the initial state. The investigation is handed back to the user at this point. They must examine the problem model and discover why location $l1$ is important in solving this problem and further why it is not possible for it to become open. Traversal problems involving opening locations with *keys* are identified in (Long and Fox 2002) within the network of 16 roles related to the Traversal and Transportation subproblems. A more useful explanation might therefore be attached to this specific form of Traversal subproblem. Consider an alternative explanation: *'robby cannot reach its goal location. There is no way to open the location, l1, which is on the path to the goal. A relevant key (triangle-key) is not reachable.'*. This explanation could be constructed using problem specific language and provides a deeper explanation for why the problem is not solvable. As the structure and vocabulary of the explanation are linked to a generic subproblem the explanation can be written exploiting the concepts of that subproblem.

Visual Explanations

Of course the most appropriate way of communicating with the user will depend on the type of system being used and the concept to be communicated. For example, it might be determined that communicating concepts related to graph traversing, such as reachability can be more naturally communicated in a visual way. Of course once we have developed a visualisation to establish effective communication for the GPS, it is available to be used in any instantiating domains.

Figure 7: A visualisation of the reachability analysis for the robot in the Grid domain.

We have used a basic visualisation approach to support explanations of reachability in Traversal subproblems. In this case, the current location is green and the goal is indicated with red. The reachable locations are yellow. Figure 7 presents the reachability for the robot in Figure 6.

Communicating Complexity

We have used simple examples to illustrate the idea. However, a major difficulty of building effective explanations is in the complexity of the information to be conveyed. The use of subproblems does not reduce this complexity, instead it provides a more appropriate layer for supporting communication and designing an effective form of explanation.

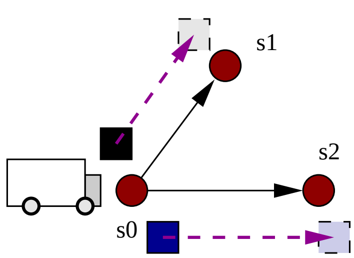


Figure 8: An example where not all of the deliveries can be serviced.

In Figure 8 a truck can only make one of the deliveries. If it delivers the blue package then it cannot reach the goal of the black package and if it delivers the black package then it cannot reach the goal of the blue package. The problem is subtle because each of the deliveries can be achieved in isolation. It therefore requires the user to understand the possible partial solutions. At the abstract planning level we need to establish basic properties in terms of actions, e.g., (at $bluebox$ $s2$) is achieved by (unload $bluebox$ $truck1$ $s2$), which relies on (at $truck1$ $s2$) and so on. If we exploit the Transportation subproblem, we can point directly at the issue and make an explanation as above (e.g., 'In Figure 8..'). The subproblem provides the context to design an appropriate and clear explanation for the situation.

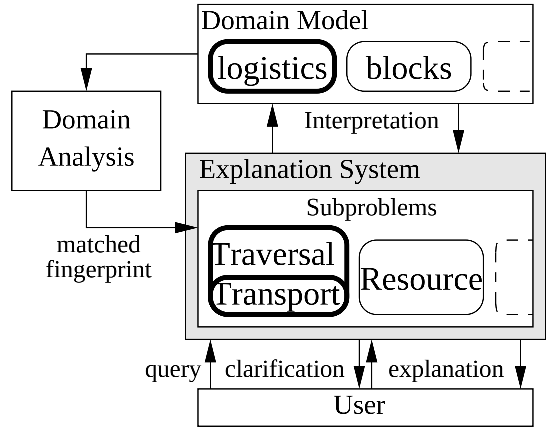


Figure 9: A framework for exploiting subproblem interpretation for explanations.

A framework for Using GPSs in Explanations

In this section we develop a framework for using subproblems in explanations. We focus on two cases: 'Why action A and not action B?' and 'Why is no solution found?'. Figure 9 presents an overview of the framework. At the heart are the subproblems, which provide alternative interpretations on a user's query. Given a specific domain, the system uses fingerprint matching to determine the appropriate subproblems. For example, the Logistics domain matches the fingerprint for the Transportation subproblem and is therefore appropriate to use in this domain, whereas the fingerprint for the Resource Management subproblem does not match. The fingerprint analysis identifies important structure within the domain, such as the constraints on traversing, and this allows the fundamental concepts of the subproblem to be evaluated (e.g., reachability).

The framework extends a typical query and response model, by including the opportunity for the system to seek clarification, termed a return question in (Madumal et al. 2018). The user can make queries of the system (these are basic method calls at this stage). The system might then ask for clarification about the query and then create an explanation. Each subproblem is given the opportunity to respond to a query. Whether a subproblem can make a response will depend if the query is relevant (e.g., for the transportation this might be a query about the allocation of trucks to packages) and if the query can be resolved in the subproblem (e.g., if the transportation aspect of the problem is unsolvable).

```
function qWhy_Unsolvable(P, π) :
    exps = []                                              ▷ explantions
    for <t in Traversers> do
        if <hasGoal(t)> then
            if <! reachable(t, t_loc, t_gloc)> then
                exps += expCantReachGoal(t)
            end if
        end if
    end for
    return exps
end function
```

Figure 10: Pseudocode for generating unreachability explanations for Traversal subproblems

Figure 10 presents pseudocode for generating explanations for unsolvability in the Traversal subproblem. In this example, the set of traversers is looped through and it is determined whether there are any that cannot achieve their goal location. The reachability between positions is computed using the analysis of the traversing action and its associated constraints (for a more detailed discussion see (Lindsay 2015)). The `expCantReachGoal` method builds an explanation using a template, e.g.,

"The locations that ⟨mover⟩ can traverse (⟨moveAction⟩) from its..."

filling in the situational information and generating any relevant visualisation (e.g., producing a DOT graph). This demonstrates how the properties of a subproblem can be used in order to build an explanation. Of course the situation might be more complex. For example, in a Transportation subproblem it must be established whether a resource allocation exists (e.g., checking that trucks can service the packages allocated to them). However, once a concept is defined for a subproblem and a suitable explanation made then this can be used for any of the instantiating domains.

Discussion: generality of subproblems

The benefit of identifying a specific subproblem within a planning domain will provide most benefit when the subproblem forms part of several planning domains. For real world applications certain of the general behaviours might be specialised further (e.g., using grid or map based visualisations) but these solutions can be supported by the general concepts, which are already defined for the domain.

In order that the explanations can be considered in the context of the subproblem it is also important that the subproblem is not tightly coupled with other parts of the domain. However, we do not believe this is an inherent constraint on the use of subproblem interpretations. Although it is outside of the scope of the current work, we are particularly interested in considering how subproblems can be considered together. For example, the work in (Seo et al. 2015), combines text and image based components in order to solve combined diagram and textual geometry questions. We believe that identifying the structural aspects of the subproblems within a shared model provides a powerful base for exploiting concepts of the subproblems within a global interpretation.

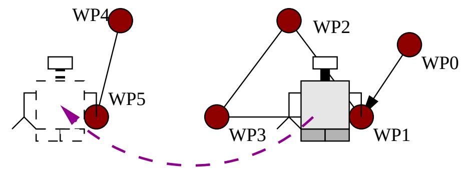


Figure 11: Example traversal problem in the Rovers domain.

Preliminary Investigation

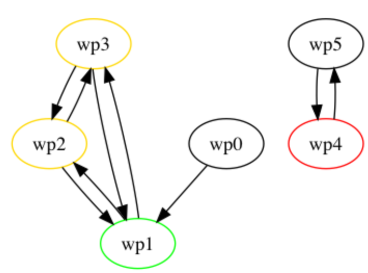
We have created preliminary implementations of the components of the framework, presented above, and present some results in order to ground the ideas. We first test the generation of an appropriate explanation for queries in unsolvable scenarios from Traversal and Transportation domains and then test the generation of an appropriate disambiguation return question.

"Why can't you do that?"

We now consider the case of queries of the type Q4, where the planner cannot find a solution and specifically, situations where the problem's unsolvability can be found within a single subproblem. We have implemented this explanation for the Traversal and Transportation subproblems. Here we ground these implementations in specific situations.

To test this scenario we first identified the set of appropriate subproblems for the domain using fingerprint analysis. Each of the identified subproblems was then asked to determine if their subproblem was solvable and if not to generate an explanation of why not.

Traversal In the first scenario, presented in Figure 11, a traversing robot must move to a location that is disconnected from its network (set as a problem in the Visitall domain). The system identified the Traversal GPS and determined the problem was not solvable. It generated the explanation (instantiating the template from the framework section):

> "The locations that robby can traverse (move) from its current position are illustrated in the diagram below:
>
> 
>
>
> The initial position of robby is shown in green and the reachable positions are shown in yellow. The goal position of robby is shown in red. This shows that robby cannot traverse to its goal."

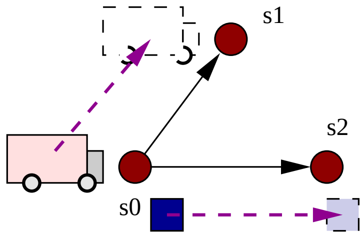


Figure 12: Example transportation problem in the Driverlog.

Transportation The second scenario is the situation presented in Figure 12, where the transportation problem, set in the Driverlog domain, has conflicting goals. The system identified the Transportation subproblem and determined it was not solvable. It generated the following explanation:

> "Each obj needs to have a truck allocated to it in order to move it to its goal. If a truck has a goal location then it can only be allocated to a obj if its goal is reachable from the goal location of the obj. In the current situation package1 cannot be delivered:
>
> - truck1 can pickup package1. However, if it delivers the obj (at s2) it cannot then reach its goal (at s1).
>
> No other trucks can pickup package1. "

In this explanation the concept of truck allocation is used to support the explanation.

These examples demonstrate how the explanations can exploit the context of the subproblem, while extracting specific information from the specific domain (e.g., traversing action name) and instantiated situation. This means that time and resources committed to designing and implementing a solution at the subproblem level are shared between the implementing domains. The use of context becomes even more important if the user wants to delve deeper, e.g., perhaps asking about a specific resource allocation.

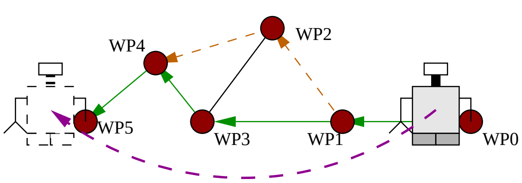


Figure 13: Example traversal problem with routing options: planned route in green and user's alternative in Orange.

"Why does the plan contain action A rather than action B (that I would expect)?)"

In this part we use our system to assist in query disambiguation. In particular, we take a query that questions the use of action A rather than action B (as was examined in (Borgo, Cashmore, and Magazzeni 2018)) and we propose alternative interpretations based on the appropriate subproblems. To test this scenario we identified the set of subproblems for the domain using fingerprint analysis. Each of the identified subproblems was asked for clarification options.

We examine query disambiguation in a situation from the traversal scenario presented in Figure 13. The traversing robot has performed a path between s0 and s5 through the central location s3. The user query is: Why (move robby WP1 WP3) and not (move robby WP1 WP2)?

Our system identified the Traversal subproblem and that the query might be relevant to the subproblem. It isolated routes where A was relevant and generated the following disambiguation question:

> "Did you mean, why did robby follow route: WP0, WP1, WP3, WP4 then WP5, starting from plan step 0, rather than WP0, WP1, WP2, WP4 then WP5?"

This has demonstrated that a query can be interpreted within the context of a relevant subproblem and that this can be used to develop richer understanding of user intention. In particular, where there is ambiguity, the relevant subproblems can present the questions that they can answer and the user can decide the answer that is most relevant. The generated query demonstrates that the subproblem can construct paths through points, allowing it to quantify the difference in cost between the two routes, which might form the basis of an explanation in this case.

Related Work

Our work builds on research in the area of XAIP (Fox, Long, and Magazzeni 2017). Recent work in this area has considered both the form that the explanation should take, e.g., (Miller 2018b) provides insights into explanations from social sciences and (Madumal et al. 2018) investigates typical explanation structures; and work that has considered how the content of an explanation can be generated from a planning model, e.g., (Borgo, Cashmore, and Magazzeni 2018). Our work aims to bridge the gap between domain tailored explanations, such as (Chakraborti et al. 2017a) and general approaches based on planning concepts, such as (Magnaguagno, Pereira, and Meneguzzi 2016; Magnaguagno et al. 2017). The typical focus of work in this area has assumed that explanation is a process that happens after planning has finished. An alternative is proposed in (Chakraborti et al. 2017b), which approaches the problem as a model reconciliation problem, where the planner should reason with the cost of explaining any actions that the user will find unintuitive.

Conclusion and Future Work

Generic subproblems are common structures in Automated Planning (AP) domain models; they establish an abstraction layer, allowing the concepts of the subproblem to be defined across various instantiating domain models. In this work we introduce the idea of exploiting this abstraction layer during query interpretation, both by extending the concepts defined for the domain and through query disambiguation, and in constructing explanations, through supporting contextual information. We have presented a framework for exploiting subproblems in explanation generation and we presented the results of a preliminary investigation, which indicated the prevalence of subproblems in domain models and demonstrated the system in both explanation and clarification generation. The current framework exploits the subproblems in isolation. We are particularly interested in investigating combining subproblems in order to allow disambiguating and constructing explanations where a query relates to several subproblems, e.g., (Seo et al. 2015). We intend to investigate how the contextual layer supported by subproblems can be exploited within a framework for supporting additional knowledge for explanations, e.g., (Vallati, McCluskey, and Chrpa 2018). We are also interested in generalising our work to exploit other AP structures, such as temporal constraints, numeric resources and continuous change.

References

Borgo, R.; Cashmore, M.; and Magazzeni, D. 2018. Towards providing explanations for AI planner decisions. *arXiv preprint arXiv:1810.06338*.

Cashmore, M.; Fox, M.; Long, D.; Magazzeni, D.; and Ridder, B. 2017. Opportunistic planning in autonomous underwater missions. *IEEE Transactions on Automation Science and Engineering* 15(2):519–530.

Cashmore, M.; Collins, A.; Krarup, B.; Krivic, S.; Magazzeni, D.; and Smith, D. 2019. Towards explainable ai planning as a service. In *Proceedings of the 2nd ICAPS Workshop on Explainable Planning (XAIP)*.

Chakraborti, T.; Sreedharan, S.; Kulkarni, A.; and Kambhampati, S. 2017a. Alternative modes of interaction in proximal human-in-the-loop operation of robots. *arXiv preprint arXiv:1703.08930*.

Chakraborti, T.; Sreedharan, S.; Zhang, Y.; and Kambhampati, S. 2017b. Plan explanations as model reconciliation. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI)*.

Coles, A. I., and Smith, A. J. 2006. Generic types and their use in improving the quality of search heuristics. In *Proceedings of the 25th Workshop of the UK Planning and Scheduling Special Interest Group*, PlanSIG 2006.

Fox, M., and Long, D. 1998. The automatic inference of state variables in TIM. *Journal of Artificial Intelligence Research* 9:367–421.

Fox, M., and Long, D. 2001. Hybrid STAN: Identifying and managing combinatorial sub-problems in planning. In *Proceedings of 17th International Joint Conference on Artifical Intelligence*. Morgan Kaufmann Publishers.

Fox, M.; Long, D.; and Magazzeni, D. 2017. Explainable planning. *arXiv preprint arXiv:1709.10256*.

Göbelbecker, M.; Keller, T.; Eyerich, P.; Brenner, M.; and Nebel, B. 2010. Coming up with good excuses: What to do when no plan can be found. In *Proceedings of the International Conference on Automated Planning and Scheduling*.

Goldwasser, D., and Roth, D. 2014. Learning from natural instructions. *Machine learning* 94(2):205–232.

Helmert, M. 2001. On the complexity of planning in transportation and manipulation domains. Diploma thesis, Albert-Ludwigs-Universität Freiburg.

Lindsay, A.; Fox, M.; and Long, D. 2009. Lifting the limitations in a rule-based policy language. In *FLAIRS Conference*.

Lindsay, A. 2015. *Problem Models for Rule Based Planning*. Ph.D. Dissertation, Department of Computer and Information Sciences, Strathclyde University, UK.

Lindsay, A. 2019. Towards exploiting generic problem structures in explanations for automated planning. In *Proceedings of the 10th International Conference on Knowledge Capture*.

Lipovetzky, N.; Burt, C. N.; Pearce, A. R.; and Stuckey, P. J. 2014. Planning for mining operations with time and resource constraints. In *Twenty-Fourth International Conference on Automated Planning and Scheduling*.

Long, D., and Fox, M. 2002. *Planning with Generic Types*. Exploring Artificial Intelligence in the New Millenium. Morgan Kaufmann. chapter 4, 103–138.

Madumal, P.; Miller, T.; Vetere, F.; and Sonenberg, L. 2018. Towards a grounded dialog model for explainable artificial intelligence. *arXiv preprint arXiv:1806.08055*.

Magnaguagno, M. C.; Pereira, R. F.; Móre, M. D.; and Meneguzzi, F. 2017. 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*, 32–38.

Magnaguagno, M. C.; Pereira, R. F.; and Meneguzzi, F. 2016. Dovetail–an abstraction for classical planning using a visual metaphor. In *The Twenty-Ninth International Flairs Conference*.

Miller, T. 2018a. Contrastive explanation: A structural-model approach. *arXiv preprint arXiv:1811.03163*.

Miller, T. 2018b. Explanation in artificial intelligence: Insights from the social sciences. *Artificial Intelligence*.

Oates, T. 2001. *Grounding knowledge in sensors : unsupervised learning for language and planning*. Ph.D. Dissertation, University of Massachusetts at Amherst.

Porteous, J.; Lindsay, A.; Read, J.; Truran, M.; and Cavazza, M. 2015. Automated extension of narrative planning domains with antonymic operators. In *Proceedings of the International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*.

Porteous, J.; Ferreira, J. F.; Lindsay, A.; and Cavazza, M. 2020. Extending narrative planning domains with linguistic resources. In *Proceedings of the International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*.

Seo, M.; Hajishirzi, H.; Farhadi, A.; Etzioni, O.; and Malcolm, C. 2015. Solving geometry problems: Combining text and diagram interpretation. In *Proceedings of Conf. on Empirical Methods in Natural Language Processing*.

Siddle, J.; Lindsay, A.; Ferreira, J. F.; Porteous, J.; Read, J.; Charles, F.; Cavazza, M.; and Georg, G. 2017. Visualization of patient behavior from natural language recommendations. In *Proceedings of the Knowledge Capture Conference*, 25. ACM.

Vallati, M.; Magazzeni, D.; De Schutter, B.; Chrpa, L.; and McCluskey, T. L. 2016. Efficient macroscopic urban traffic models for reducing congestion: A pddl+ planning approach. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Vallati, M.; McCluskey, T. L.; and Chrpa, L. 2018. Towards explanation-supportive knowledge engineering for planning. In *The EXplainable AI Planning Workshop (XAIP-18)*.