What *was* I planning to do?

Mark Roberts[1] and **Isaac Monteath**[2] and **Raymond Sheh**[2] and **David W. Aha**[1]
Piyabutra Jampathom[1] and **Keith Akins**[1] and **Eric Sydow**[1]
Vikas Shivashankar[3] and **Claude Sammut**[4]
[1]The U.S. Naval Research Laboratory; Washington, DC, USA | {first.last}@nrl.navy.mil
[2]Curtin University; Bentley, WA, Australia | isaac.monteath@postgrad.curtin.edu.au, raymond.sheh@curtin.edu.au
[3]Amazon Robotics; North Reading, MA, USA | vikshiv@amazon.com
[4]The University of New South Wales; Sydney, NSW, Australia | claude@cse.unsw.edu.au

Abstract

Adjusting commitments to ongoing plans can occur frequently when executing these plans in a dynamic environment. Often, an agent will repair its plan or replan vis-a-vis such change, which is a type of planning-specific adjustment. However, adjusting commitments to the goals by regoaling, transforming goals, or deferring goals may also be needed. As discussed in the literature on plan explanation, an agent may be asked to account for plan adjustments. In this position paper, we advocate for considering the full suite of possible adjustments in explanation. This includes plan repair, replanning, deferring, regoaling, and abandoning goals. Using an example from the RoboCup Rescue Agent Simulator (`Roborescue`), we leverage a goal lifecycle, extended with time-based Chronicles and transitions goals, for explanation. We propose an explanation taxonomy that spans three dimensions and illustrate the use of this taxonomy for many possible explanations in `Roborescue`. Finally, we highlight several possible user interfaces we intend to build.

1 Introduction

Autonomous agents will be increasingly called upon to react to ever more challenging, dynamic environments. Execution rarely proceeds according to plan, resulting in the need of an agent to adjust its commitments. When called upon to explain these changes in commitments, the agent needs to account for the reasons behind the change.

Following the approach of Langley et al. (2017), we maintain that explaining plans requires more than only explaining the choices of the planner. In many applications, deliberation – commonly referred to as planning, problem solving, or reasoning – occurs at many levels of a situated autonomous system (Ghallab, Nau, and Traverso 2016). Pollack and Horty (1999) argue that there is more to planning than simply making plans, noting that plans are meant to be executed and a variety of interesting problems occur when plans fail. In order to explain their choices, agents must be able to recreate the context and history of the moment for each commitment.

Accordingly, we argue that a wider perspective, one that considers the role of the planning system integrated *within* a situated system, as necessary for effective explanation of plans and the planning systems. Much of the literature around plan revision focuses on plan repair or replanning, and much discussion about explaining plans and planner choices emphasizes repairing or replanning. A similar argument is echoed in the literature on explanation (Miller 2017; Abdul et al. 2018). However, we offer a distinct perspective and solution for solving the problem. In the broader sense, an agent may deliberate about its goals, performing what we call goal reasoning (e.g., (Hawes 2011; Vattam et al. 2013) (Ghallab, Nau, and Traverso 2016, Section 7.2.3), (Aha 2018)).

We describe a research plan that will integrate explanation in a goal reasoning context, expanding plan explanation to encompass the agent formulating new goals, considering alternative plans (before and during execution), repairing plans, replanning from scratch, deferring or abandoning goals, switching between multiple goals, reformulating revised goals, and learning when to apply each of these possible transitions. Our proposed approach complements this literature by embedding explanations within a goal lifecycle that makes specific commitments to hierarchical, temporal, and numeric planning and characterizing explanations within to a 3-axis taxonomy. The contributions of this paper include:

- An extension of the goal lifecycle by Roberts et al. (2015; 2016) to include Chronicles from recent developments in planning and acting (Ghallab, Nau, and Traverso 2016);
- an informal description of how the extended goal lifecycle facilitates explanation using *transition goals*;
- a 3-axis taxonomy for explanations that considers their source, scope, and depth;
- an exploration of using the taxonomy to categorize various explanations; and
- a proposed extension to a user interface, the Virtual Mission Operation Console, to support explanations.

We begin by describing our proposed application, `Roborescue`, for exploring explanation (Section 2), which we follow with the assumptions for our solution (Section 3). Based on these assumptions, we introduce the goal lifecycle (Section 4), extend it to incorporate the above assumptions (Section 4.1), and demonstrate how it provides a natural mechanism for supporting anticipatory, ad-hoc, and post-hoc explanations (Section 4.2). We introduce a 3-axis taxonomy of explanations and demonstrate how explanations can be generated (Sections 5 and 6). These assumptions and commitments are not unique and much research has advanced one or more of these areas, as is elaborated in our discussion of related work (Section 7).

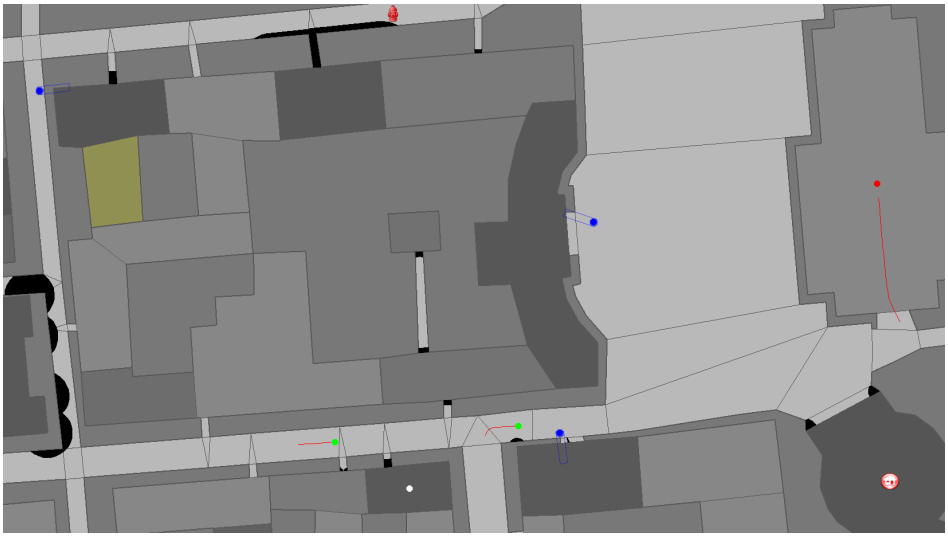

Figure 1: A single city block of the Agent Simulator running on the Berlin scenario.

2 `Roborescue`: A testbed for explanation

The RoboCup Rescue Agent Simulator, or `Roborescue`, models a situation immediately following a natural disaster (Sheh, Schwertfeger, and Visser 2016). For example, Figure 1 displays a `Roborescue` scenario using the 2017 simulator. It is the basis of the RoboCup Rescue Simulation League (Akin et al. 2012), an international competition for collaborative AI agents that has been running since 2000[1]. Inspired by the events of the 1995 Great Hanshin earthquake in Kobe, it aims to provide a venue where the performance of different algorithms for coordinating and controlling teams of simulated agents can be compared. The competition is held annually with an international championship fed by numerous regional opens. The Simulation League, which focuses on environments the size of a few city blocks, is just one of three competitions under the RoboCup Rescue umbrella. The Virtual Robot League is a competition based on a more detailed, high fidelity physics simulator within the confines of one city block. In the Robot League, physical robots compete within an arena of standard test methods for response robots.

In the Simulation League (cf. Figure 1), specific agents work together to help civilians (green dots) move to a safe area (the lower-right building). Civilians will move along roads (light gray) if they are able but may become trapped under or behind rubble (solid black). Fires (yellow building on the left) can cause further harm and produce rubble. If a civilian is trapped for too long or suffers too much damage then they turn black, indicating that they have died. Agents must communicate and cooperate to transport as many victims as possible to safety while also gathering information about the disaster area. Each agent may have its own objectives that can be interrupted by events such as requests to help other agents, discovering a victim, or coming across rubble in its path. The challenge is responding in a manner that best assists the entire team and communicating those responses.

[1] `http://wiki.robocup.org/Rescue_Simulation_League`

Controllable agents consist of police (blue dots), ambulance (white dots), and fire patrols (red dots). All three agents can `scout` an area, but only police can `clear` rubble, ambulances can `transport` a single civilian, and fire patrols can `douse` fires using water from hydrants (top). Not shown are centralized agents that serve as dispatchers each type. In this figure, two healthy civilians (bottom middle) are moving to the safe area. To their right, two police clearing rubble. Below them, an ambulance is stuck behind rubble.

2.1 Running Example

We illustrate the core explanation challenges resulting from execution failures and opportunities for an ambulance agent, although similar situations could occur for a police or fire agents. Failures can occur when a planned task is blocked from completing, a task takes longer than anticipated, a maintenance goal is violated, or a resource is blocked. Opportunities may appear when an agent discovers a previously infeasible goal is possible, through the arrival of new tasks, or while merging plans to complete multiple goals.

Figure 2 demonstrates the goal network of an ambulance agent for the following simplified scenario; we will formalize goal networks in Section 4. Suppose an ambulance agent, who can only scout or carry civilians, is commanded to scout a 'far away district' for potential civilians. After generating three alternative paths, A, B, and C, to the district, it starts down path A (solid boxes) but comes across rubble that must be cleared. It could here switch to an alternative route or ask the police to help clear the rubble. Suppose it selects path B (gray boxes) but cannot communicate this change to its dispatcher because communication is blocked by the buildings around it.

While traversing path B it identifies a civilian needing assistance. It could ignore the civilian and continue its current goal or transport the civilian to a shelter. Suppose that it chooses to defer its original goal (i.e., scout 'far away') to transport the civilian, generates a plan to the nearest shelter,

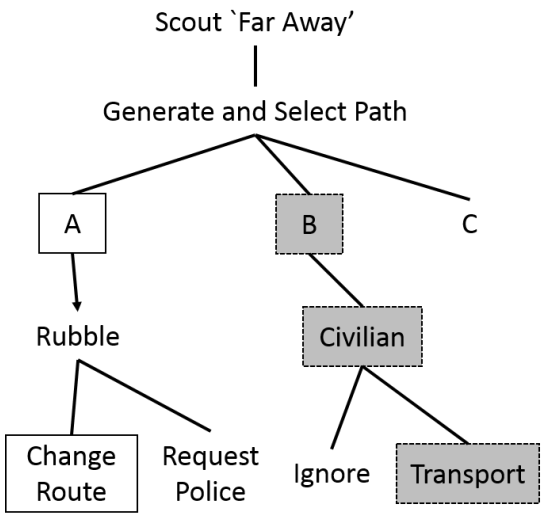

Figure 2: A goal network for the ambulance example.

Shelter 1, and picks up the civilian; again it cannot communicate this change because its radio is blocked. Upon reaching the Shelter 1, the agent discovers the shelter is full and must transport the civilian to Shelter 2, which it proceeds to do. Along the way, it identifies several more civilians needing assistance. Once the transport to Shelter 2 has completed, it could determine whether to return to the civilians it found, abandon path B, select the other alternative path C, or replan from the shelter.

2.2 Challenges in Explanation

Even for the above small example, the agent may be asked to explain its choices. Miller (2017), in surveying the relevant literature on explanation, suggests that explanations are contrastive, selected, and social. In line with this view, we imagine some possible kinds of explanations that may surface during or after the above example for the ambulance agent, which made many decisions, modified or abandoned some commitments, deferred others, and generated new goals in response to unexpected events. How does the agent explain its choice to change to a different path, switch goals by locating an agent who can clear the rubble, or attend to the civilian it just discovered? What if the agent is asked to explain which alternatives it considered when it found the first civilian? Suppose the agent did not know that a better path to 'far away' has become available and the dispatcher conveys that information (because the radios work at shelters); what if the agent sticks to its plan for path B and is later asked why it did not take the more direct route?

When appearing at the Shelter 1 with the civilian, suppose the agent is asked to report the damage exists in the 'far away', a type of *selected* explanation, which it will not know but could probably have anticipated. In this example, we imagined the agent tried to communicate changes to its plan but failed. Suppose the agent could communicate. From a *social* perspective, when should the agent proactively explain itself? Too many plan updates would likely be overwhelming but too few will result in reduced situational awareness for the entire team. Even if the agent waits for a request to explain itself, it must still anticipate when an explanation may be needed and facilitate such explanations by storing not only the plan trace but also the context that lead to changes in its commitments. Additional deliberation may be required to recreate the context of the decision if, for example, the agent is asked to consider other ways it could have solved the problem (e.g., during a post-hoc debriefing), which strengthens the important role of automated planning for explanation.

Learning can also play a role even for planning systems and plan explanation. Suppose the agent had applied machine learning to make some of its prior decisions, then it may also need to explain the learned model it used to prioritize its decisions (e.g., to select one goal over another) without being explicitly commanded to do so. For example, it may have learned that seriously wounded civilians in this situation do not live as long as usual compared to an expected model, leading it to choose to transport a critically injured civilian because it reasoned that no other agent could help in time.

Finally, consider an even more abstract view of a commander managing the entire disaster, who must communicate with dispatchers as well as agents. The commander may desire answers to *contrastive* questions such: *Why doesn't this task fit in the current solution?* or *What change is needed (or which constraint should I relax) to fit this task into the solution?* The commander may need to consider unit placement if new resources become available. For example, suppose the commander receives an offer of 5 ambulance teams from a neighboring city, a natural question question would be *Where should I place these units?* And lastly, the commander may want to anticipate possible future situations based on past data (where are fires most likely to start, for example), which is a variant of anticipatory planning.

3 Solution Assumptions

We make a number of assumptions in our proposed solution to help make solving the problem tractable. We assume agents interact with other agents and the environment in a distributed manner. Each agent generates their own plan(s) according the local information available – that is, we don't assume a globally optimal criteria. An agent may modify its goals in response to notable events that impact goal processing and can occur as a result of the environment or other agents.

The situation is oversubscribed as there will typically be more goals than resources to complete them, so an agent seeks to achieve or maintain high-quality, satisficing solutions. Goals can be hard goals (i.e., requirements), maintenance goals, and soft goals (i.e., preferences). Agents may consider a set of criteria when evaluating whether to pursue a goal. This means an agent may need to generate alternatives, consider tradeoffs, and commit to plans; such reasoning requires what is often called numeric planning or planning with metric quantities, which has been examined in the automated planning literature (e.g., (Fox and Long 2003)). However, the prevalent modeling approach of using action-costs prohibits more than one metric; this has led to benchmark problems with relatively "flat" objective spaces that often correlate with

plan length. (Radzi 2011). Although some planners generate alternatives according to multiple metrics, more work is needed to incorporate trade-off analysis between competing metrics. The agent may desire to maintain commitments it has already made to itself, to future projected events, or to other agents.

We commit to hierarchical structures to facilitate memory, planning, and retrieval. This commitment to a hierarchical approach provides for the use of domain-dependent knowledge to guide the search process in selecting and planning for goals; we leave it to future work to generalize the approach to domain-independent planning.

Agents maintain a history (or future, as appropriate) of: relevant world events, attempts to achieve goals, past experiences, other agents' actions, and commitments to other agents. This assumption requires temporal reasoning for the past, current, and future. The stored history may be compressed through learning, which we plan to address in future work. Explanations may be required before, during, and after completion of one or more goals. This requires an agent to perform introspection about its own history.

4 Goal Networks and the Goal Lifecycle

Our agents reason over goal networks, which are partially ordered sets of goals. For historical reasons, goal-networks are called Hierarchical Goal Networks (HGNs) (Shivashankar et al. 2013a; 2013b), although the decomposition of a goal network need not be strictly hierarchical. Alford et al. (2016) showed that HGN planning has the same expressiveness and complexity class, semi-decidable, as task-network planning, commonly referred to as hierarchical-task networks (HTNs). Further, their work proposed a hybrid formalism, goal-task networks (GTNs), which unified goal-network and task-network planning under sharing or insertion semantics.

Roberts et al. (2016) extended goal-task networks to model the goal lifecycle for a goal-reasoning agent. We restrict our attention to goal networks. Figure 3 (left) shows a subset of that goal lifecycle, which models the main decision points of goal reasoning (Hawes 2011; Vattam et al. 2013; Aha 2018). Goals transition between modes (rounded boxes) via strategies (arcs); colored strategies highlight the focus of this study. Figure 3 (right) shows a timeline of plan revision with the same colors. Goals are FORMULATED, SELECTED, and then EXPANDED (i.e., planned) for possible ways of achievement. Alternative plans are EXPANDED (magenta), and the system commits to a single plan (black line), which is then DISPATCHED for online execution. An objective might fail (hollow circle) due to events such as cloud cover, which may force the system to EVALUATE (wide gold line) revision of prior strategies. To move the failed task (hollow circle) to a new slot (solid circles), the system might: CONTINUE by ignoring the problem or falling back to a secondary task (red, dotted); REPAIR the plan by reusing a previous alternative (purple, dashed); REPLAN by expanding multiple alternatives (green, dot-dashed); or DEFER the objective (gray, long-dashed).

A goal transitions through the lifecycle as a *goal node* $\check{g} = (g_i, N, C, o, X, x, q)$ where: g_i is the goal to be achieved; $N = (s, gtn)$ is a gtn-node for g_i; C is the set of constraints on g_i and gtn; o is the current mode of g_i, defined below; X is the set of expansions that could achieve g_i, defined below; $x \in X$ is the committed expansion along with any applicable execution status; and q is a vector of quality metrics. Metrics could be domain-dependent (e.g., priority, cost, value, risk, reward) and are associated with achieving g_i. We used $\mathcal{N}$ in prior work but find $\check{g}$ more natural.

Figure 3 (left) displays the possible refinement strategies, where an actor's decisions consist of applying one or more refinements from R (the arcs) to transition $\check{g}$ between modes (rounded boxes). Strategies are denoted using small caps (e.g., FORMULATE) with the modes in monospace (e.g., FORMULATED). Execution may require adjustments, via the resolve strategy, to prior commitments. The degree of revision determines how far back in the goal lifecycle the goal must transition, and it is not always clear which resolution is best.

A *goal memory* $M = \{\check{g}_1, \check{g}_2, .., \check{g}_m\}$ for $m \geq 0$ holds the active goal nodes for an agent. Most refinements modify the goal memory by modifying a node within memory, in which case we write $M \rightarrow_R M'$ for a single strategy application of R resulting in M' and $M \rightarrow_R^* M''$ for a sequence of applied strategies resulting in M''.

4.1 Extending the Goal Lifecycle

Following the assumptions from Section 3, we make a specific commitment to temporal reasoning with timelines and discrete resources. We accomplish this with a hybrid planning approach called Constraint-Based Interval (CBI) Planning, which is commonly used in NASA (e.g., systems such as Europa or Aspen) and MBARI missions (e.g, (Smith, Frank, and Jónsson 2000; Rajan, Py, and Barreiro 2013)). Central representations in CBI planning include a variable, often called a token, and temporal constraints, called intervals, over the variable. A token is an object or resource in the system that needs to be tracked (e.g., the ambulance status). Tokens can be assigned attributes to indicate state (e.g., transporting, clearing or scouting). Constraints restrict the set of valid solutions by limiting resource usage (e.g,. keep health above 74 units), temporal extent (e.g., duration of 90 minutes), or ordering (e.g., scout before transport). To incorporate temporal constraints, we will extend the goal node with Chronicles as described by Ghallab et al. (2016). Decision points of the agent follow many critical points the goal lifecycle (cf. Figure 3). We next describe how these decision points will support explanation.

4.2 Facilitating Explanation with the Goal Lifecycle

The goal lifecycle with temporal constraints provides a natural mechanism for tracking changes during execution, such as those highlighted in the example from Section 2.1. We partition the goal memory into the past, present, and future, $M = M_{\text{past}} \cup M_{\text{present}} \cup M_{\text{future}}$, segmented by time: $t_0 < M_{\text{past}} < t_{\text{now}} \leq M_{\text{present}} < t_{\text{horizon}} \leq M_{\text{future}} < t_{\text{end}}$, so that t_0 and t_{end} indicate the start and end of time, t_{now} indicates now, t_{horizon} indicates the end of the furthest plan into the future. The system stores the transitions of all goals in the appropriate memories. Goals that interact will have appropriate references to their related (sub)goals.

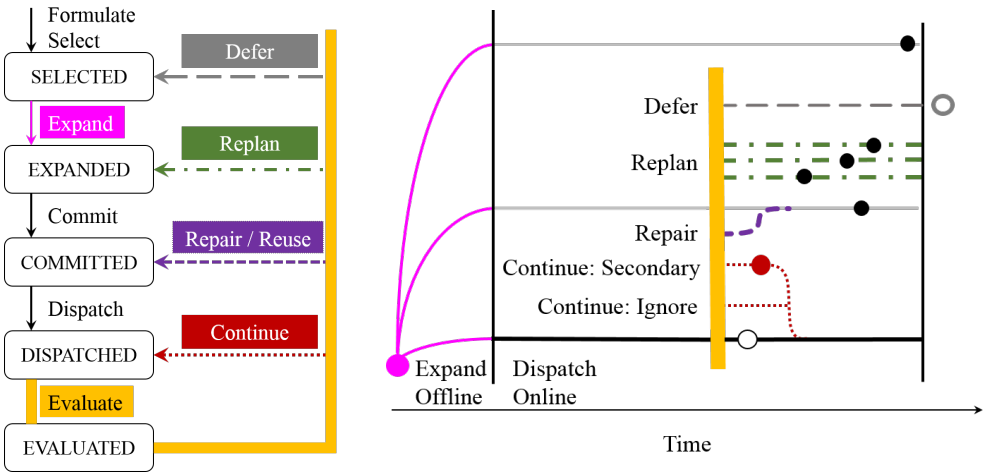

Figure 3: Left: The goal lifecycle of Roberts et al. (Roberts et al. 2016) Right: A plan revision timeline.

Explanations will often hinge on transitions in the goal lifecycle, since these capture changes in commitments. For example, the agent may be later asked *Why did you defer this goal?* or *Which alternatives did you consider at time point t?* The goal lifecycle supports explanation for such situations by creating *transition goals* to achieve each *important* transition, where importance is defined by the scope, depth, and source of expected explanations as defined later in Section 5. The agent will only store transitions that may require future explanations. It may help to see transition goals as "meta" goals, though we do not necessarily make a commitment to meta-reasoning specifically.

Transition Goal Example: Returning to the example of the ambulance agent in Figure 2. When the agent sees rubble along path A, it needs to deliberate about possible alternatives. This places `transit_path(A)` into `EVALUATED` to resolve. Since this decision is one that may require explanation, the agent creates a transition goal `resolved(rubble_blocking_path_A)`, which follows the same lifecycle as any other goal. During the expansion of this new goal, it generates the possibilities (i.e., expansions) shown under the rubble node in Figure 2; (it generates many more but only two are shown). Supposing it resolves the `transit` goal by changing the route, then `resolved` goal is now finished, the `transit_path(A)` goal is dropped, and a new `transit_path(B)` is formulated. In essence, the entire set of "goal" transitions has been captured from the moment the rubble was detected to the point where a new path was chosen. It is easy to see that transition goals can be created for similar situations.

We argue that the transition goal – the `resolved` goal in the above example – is vital for any reasonable explanation. When the system is asked to explain itself, it can simply refer to this goal to elucidate its entire deliberation. When used in this way, the goal lifecycle exposes key constraints that guide solutions and justify changes in commitments to goals, plans, or actions. M stores these transitions for future reference.

A limitation of transition goals is the increased memory needed to store additional information. This is especially true if all transitions are stored. We mentioned earlier that the agent should only store *important* transitions. We now turn our attention to an explanation taxonomy that will aid in determining those important transitions.

5 A Proposed Explanation Taxonomy

A key objective of this work is to develop agents that can perform a given task and explain their decision-making process to humans and other agents in the scenario. Providing effective explanations depends on a variety of factors, such as the requirements of the consumer and the agent's explanatory capabilities. We categorize explanations along three dimensions that identify its source, its depth and its scope. This categorization is inspired by Sheh (2017) and each dimension is currently segmented into a binary split:

- **Source**: *Post-Hoc Rationalization* vs. *Introspective*
- **Depth**: *Attribute (Entity/Use)* vs. *Model*
- **Scope**: *Justification* vs. *Teaching*

Source identifies the type of data used to generate the explanation. The source is *Introspective* if the provided information is based on a full trace of the underlying decision process. This means that the explanation must accurately reflect the actual decision process that was taken to arrive at a particular decision. The source is *Post-Hoc Rationalization* if it is based on observing the external behavior of the agent as a black box. These types of explanation do not necessarily reflect the actual decision process taken by the agent but rather seek to rationalize the result. An explanation's source can be somewhere in between if it has access to intermediate representations or processes but not the full "trace" of the decision-making process. Note that this is defined relative to a useful level of abstraction. After all, any program running on a deterministic computer can be traced, even if it's at the level of individual instructions, but this "execution" explanation is less useful in most applications.

Depth identifies the granularity of the explanation. The depth is called *Attribute Identity* if the explanation is based

on identifying the attributes, or features, that were used to make a decision. For example, saliency maps in object recognition tasks are considered Attribute Identity explanations. It is *Attribute Use* if it also includes information on how these attributes were used, such as decision boundaries and relationships. A Support Vector Machine might report the hyperplanes bordering a region containing the decision instance, for example. It is *Model* if the explanation extends to how the model, on which these decisions are based, was itself generated from the training data and other background information. A Decision Tree might report the training examples that flowed through to a given decision node to explain that node's decision boundary, for example.

Scope identifies the purpose of the explanation. The scope is *Justification* if the purpose is to justify a particular decision or set of decisions. It is *Teaching* if the explanation is about something beyond a decision. In general, teaching explanations aim to teach a user (or another agent) something that the explainer has learned, such as synthesizing new examples or answering hypothetical questions.

6 Generating Explanations

Using the taxonomy just described, we highlight some possible explanations an agent could provide. Explicitly stating the kinds of explanations we expect the system to provide elucidates which information will be required and the type of model that would need to be used to enable the explanation. Returning to the `Roborescue` scenario, consider a single police agent en-route to clear rubble in pile_A that receives a report of another nearby rubble pile_B, which is preventing a fire truck from accessing a burning building. The police agent revises its commitments by deferring its goal of `cleared`(pile_A) in order to achieve `cleared`(pile_B) first. This results in creating a plan with a detour to clear pile_B. We next provide several examples of complex queries and explanations using the taxonomy.

6.1 Justification scope

To assist in the explanation, we will begin with explanations that have Justification scope because the explainer is being asked to explain why it selected one plan over another or provide a reasonable rationale of its decisions.

Why did you decide to switch to a new goal? [Introspective, Attribute] This explanation considers why the agent decided to change its plan. e.g. Another agent is waiting for me to clear the rubble AND the time required for the detour is less than 15 minutes.

What do you think this other agent is likely to do? [Post-hoc, Attribute] The explanation has attribute depth because the agent does not have access to the internal model of the target agent. It has a post-hoc source because the agent lacks internal access to the decision-making of the target agent.

Asking the agent to explain its internal model/rationalizations for how other agents act in the world. For example, a user could ask whether the police agent thinks that the dispatcher is likely to send another police agent to assist. The agent might respond by stating that, in its experience, assistance is in areas that are far from the center of the city. Therefore, it might conclude that the dispatcher is unlikely to send assistance.

This type of explanation could be useful for debugging purposes, especially where the user has introspective access into these other agents. This can assist the user in finding discrepancies between how the agent believes other agents reason and how they actually reason. In the example above, a user may query the dispatcher for an introspective explanation on whether it is likely to dispatch another police agent to the area. If the explanations does not match that of the police agent, it may provide an important insight into the behavior of the police agent.

How confident are you that your plan will work? This explanation involves the agent computing a measure of how confident it is in the selected plan. For example, a human operator might be doubtful about the effectiveness of the police agent's selected plan and request that the agent to report its confidence. This will help the human operator decide whether to trust the agent or override it. The scope of this explanation is Justification because the confidence measure is used to justify the agent's decision. The source is Introspective in most cases as this is the only way to measure the agent's confidence. However, a measure of *reasonableness* of a given plan may be computed externally in a Post-Hoc fashion. The depth can be either Attribute or Model depending on how the confidence metric is calculated.

6.2 Teaching Scope

We now switch to explanations that follow from a teaching scope because the explainer is informing another user (or agent) about other available decisions rather than justifying a particular one. Some questions from a justification scope could be asked in a teaching scope because of a different motivation.

Why did you decide to switch goals? Agents may be able to use their own local knowledge to check for incorrect assumptions in other agents. For example, suppose a police agent informs the dispatcher of switching from clearing rubble for an ambulance to clearing rubble for a fire patrol. The dispatcher may not know the state of the fire patrol and the explanation needs to shift from justifying to teaching scope, where the agent may need to clarify its closeness to the fire.

Which other similar goals are possible? This involves the agent proposing new goals to the user that it believes are similar to its current goal. This may be particularly useful in cases where the agent cannot complete its current goal and believes it should switch to a new goal.

Extending the example, suppose that the police agent reports to the human operator that it has decided to switch goals to assist the fire truck. The human operator may know that there are many alternative routes for the fire truck to take. The operator knows that there are probably more productive goals for the police agent, but doesn't necessarily know what they are. In this case, the police agent can provide a list of goals that it believes to be similar for the human operator to decide from. These goals might include scouting a nearby area or clearing rubble for another agent.

Interventions: Explanations may relate to the use of inter-

ventions. Questions along these lines ask the agent to consider possible alternatives, some hypothetical.

How would the scenario need to change for you to retain your original goal? Given that an agent is proposing to change its goal, this query requires the agent to propose hypothetical scenarios where it would not change goals. For example, Figure 4 shows a police agent that decides to change plans from Plan 1 to Plan 2. It may explain that if the middle house was not on fire or if the distance was 1 kilometer further, it would have continued with Plan 1 instead.

How would the scenario need to change for you to change to this specific goal? This explanation type is the reverse of the previous explanation type. Given that an agent is proposing to retain its existing goal, a human operator may wish to know which scenarios would cause the agent to change to a specific goal. For example, suppose that the police agent decided to retain its Plan 1 goal in Figure 4. The human operator may ask what would need to change about the current situation for the police agent to consider switching to the goal of Plan 2. An example response might be that in order to switch to the goal of Plan 2, there would need to be more houses on fire in the area.

Counterfactuals: Other explanations relate to the use of counterfactuals. Questions along these lines ask the agent to consider alternatives in the face of negative possible worlds.

What do you predict the scenario will look like after a particular action is taken? [Introspective, Model] This explanation makes a comparison of what the model thinks would happen as a result of executing the proposed plan instead of the current plan (Model explanation) (e.g. show me the predicted outcome of executing both plans – the user really just wants to know which plan is likely to prevent the most damage in the least amount of time).

What would you do in this situation if I changed your priorities? This explanation involves the agent responding to a hypothetical scenario in which its own priorities have been altered. For example, a human operator might ask the police agent what action it would take if its highest priority was to minimize civilian casualties. This type of explanation has an Introspective source. The depth depends on the implementation and could be either Attribute or Model.

6.3 Visualizing the schedule changes

We plan to explore two kinds of visualization for explanations of competing objectives and the resulting plans. The first is exemplified by Figure 4. Agents consist of police (blue) or ambulance (red), and roads are blocked by rubble (black). Buildings can be normal (white), burning (yellow), or destroyed (black). The situation is relatively simple, but the information required to generate this explanation is complex.

The second visualization relies on an existing software product called the Virtual Mission Operations Center (VMOC), which has its origins in satellite scheduling. Figure 5 shows a mockup of an existing interface we will extend. VMOC is a space-qualified, U.S. government-owned, satellite mission management and scheduling framework. It is designed to be reusable and extensible, architected to easily incorporate new missions, optimization algorithms, and visualization applications. VMOC employs an HTML5-compliant user interface which provides users with mission management capabilities that include collection requirement (goals) management, payload ahead-of-time and real-time scheduling (planning), and stakeholder situational awareness. VMOC has been providing mission management services since the mid 2000s and currently supports several operational satellites.

VMOCs requirements management applications are graphical user interfaces for viewing, creating, and refining satellite mission tasking and requirements (goals). Users can choose to be guided through requirement generation or take an advanced view where they are able to create, read, update, and delete requirements.

VMOCs scheduling applications are centered around satellite schedule generation (planning), visualization, comparison, and real-time task (goal) execution status. During candidate schedule generation, the scheduling application returns both planned and unplanned activities, the latter including an explanation for why the activity was not scheduled. Real-time changes to schedules (plans) are pushed to the users browsers for situational awareness. Because it supports a long lifecycle for tasks, we plan to extend VMOC to support the kind of plan (and goal) revision described in Figure 3.

7 Related Work

Our proposed focus builds upon replanning and plan repair, which has a long history going back to the 1990s (e.g., (Hammond 1990; Hanks and Weld 1995; Koenig, Furcy, and Baue 2002; Horty and Pollack 2001)). Comparisons of these approaches exist (e.g., (Fox et al. 2006)), and work by Nebel & Koehler (1995) formally analyzed problems showing where reuse can be as difficult as replanning from scratch. In addition to plan repair and replanning, the goal lifecycle represents changes in commitments that defer, reformulate, or abandon goals. This fits well with the theme of planning as an iterative process advocated by Smith (2012), where the objectives shift along with exploration of the possible plans. We plan to more fully explore this distinction in future work.

Several areas of research have considered how to select between alternative objectives or adjusting commitments during execution. The most closely related to our approach is that of partial satisfaction planning by Benton et al. (2009). This approach selects goals in the face of oversubscription or preferences (soft goals) using the net utility, where the agent tries to maximize the utility of achieving goals less the action costs and penalties for unachieved goals. In contrast, our approach separates goal utilities from action costs through the use of distinct quality metrics for each.

Several studies have examined generating multiple alternative plans in the context of multiple criteria (e.g., (Do and Kambhampati 2003; Nguyen et al. 2012; Srivastava et al. 2007; Coman and Munoz-Avila 2011; Roberts, Howe, and Ray 2014). Plans are often generated according to a diversity measure that seeks to generate distinct alternatives. Our approach naturally accounts for generating multiple plans, what we call goal expansion, in the goal lifecycle. While there is work on creating flexible plans for dispatching (e.g., (Conrad

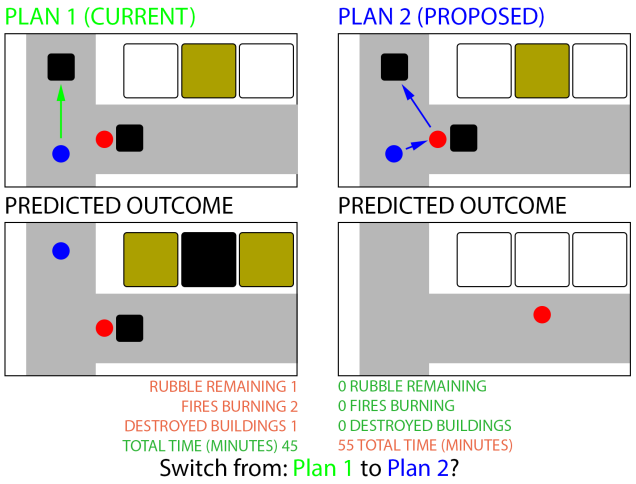

Figure 4: Possible user interface for explaining alternative plans according to multiple objectives.

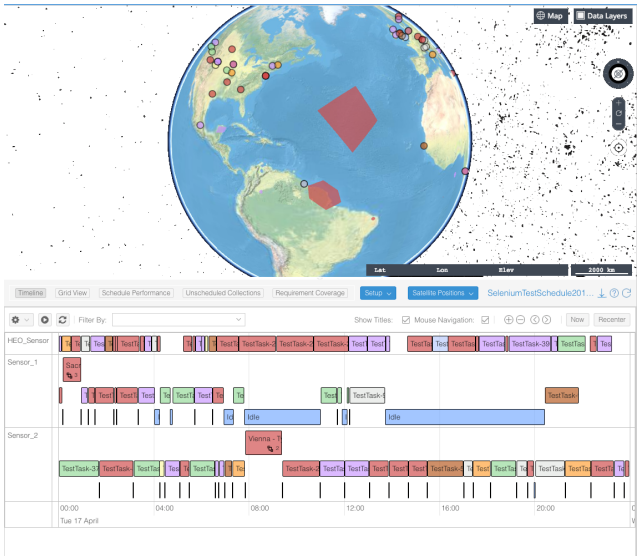

Figure 5: Existing VMOC interface we plan to extend for exploring alternative objectives.

and Williams 2011)), few studies have examined generating diverse alternatives relative to execution flexibility.

Explanations for plans have been studied in the context of generating preferred explanations (Sohrabi, Baier, and McIlraith 2011), explainable agency (Langley et al. 2017), generating trust with humans (Floyd and Aha 2017), and model reconciliation (Chakraborti et al. 2017). Much of this work is concerned with how information is selected and presented to the user. Seegebarth et al.(2012) present a formal model demonstrating how to use the plan structure from a hybrid HTN-POCL planning system to generate explanations; this approach bears the closest resemblance to our effort. Rosenthal, Selvaraj and Veloso (2016) categorize explanations provided by a mobile robot according to three parameters: abstraction (governing vocabulary), locality (governing the relevant portion of the plan) and specificity (governing detail). More recently, Fox et al. (2017) have characterized the kinds of explanations that may be needed in such systems along with initial results and a roadmap advocating for a sustained effort in Explainable Planning.

Work from a cognitive science perspective, such as that of Keil (2003), relate to focusing explanation on what is appropriate for a given audience and application. In contrast, Tolchinsky et al. (2012) focus on the use of explanation to make dialog more convincing. While these categorizations are useful to determine what a user might need, they are less useful at figuring out what techniques, or ”hooks”, need to be present in the underlying AI in order to furnish this information.

Some work has been done along the lines of a ”requirements and capabilities analysis.” Bibal (2016) share this view that existing semantics do not suit this purpose for many applications although they do not propose a unified set of categories that do so. One important distinction is made by Montavon et. al. (2018) where they make the distinction between an explanation (what we call an Attribute Identity explanation) and an interpretation (which aims to make the final output of the AI system more understandable). While this is useful, we feel it may be too narrow. Sometimes an explanation needs to contain detail about the identity of attributes or features, in the interpretable domain, and sometimes an explanation changes based on how it is used (what we call Attribute Use explanations) and why it is used in that manner (what we call Model explanations). Perhaps closest to our work, at least in terms of purpose, is that of Doran et. al. (2017) with four categories: Opaque (no insight), Interpretable (mathematically analyzable), Comprehensible (producing user-interpretable symbols) and Explainable (automated reasoning to produce explanations).

8 Summary

We have advocated a broader perspective when considering explainable planning that includes the spectrum of commitment adjustments that can take place while executing a plan. These adjustments include not just plan repair and replanning but also goal deferment, regoaling, and goal abandonment. We demonstrated that this larger perspective can be captured as part of the goal lifecycle from prior work and showed how the lifecycle can facilitate explanation. We then categorized the kinds of explanation that could take place according to a taxonomy of three dimensions: source, depth, and scope. Using this taxonomy, we examined many kinds of explanations. Finally, we showed two possible visualizations we plan to explore as our project continues.

References

Abdul, A.; Vermeulen, J.; Wang, D.; Lim, B.; and Kankanhalli, M. 2018. Trends and trajectories for explainable, accountable, and intelligible systems: An HCI research agenda. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*.

Aha, D. W. 2018. Goal reasoning: Foundations, emerging applications, and prospects. *AI Magazine* to appear.

Akin, H. L.; Ito, N.; Jacoff, A.; Kleiner, A.; Pellenz, J.; and Visser, A. 2012. Robocup rescue robot and simulation leagues. *AI magazine* 34(1):78.

Alford, R.; Shivashankar, V.; Roberts, M.; Frank, J.; and Aha, D. 2016. Hierarchical planning: Relating task and goal decomposition with task sharing. In *Proc. IJCAI*, 3022–3028.

Benton, J.; Do, M.; and Kambhampati, S. 2009. Anytime heuristic search for partial satisfaction planning. *AIJ* 173(5-6):562–592.

Bibal, A. 2016. Interpretability of Machine Learning Models and Representations: an Introduction. In *Proc. ESANN*, 77–82.

Chakraborti, T.; Sreedharan, S.; Zhang, Y.; and Kambhampati, S. 2017. Plan explanations as model reconciliation. In *Proc. IJCAI*, 156–163.

Coman, A., and Munoz-Avila, H. 2011. Generating diverse plans using quantitative and qualitative plan distance metrics. In *Proc. AAAI*, 946–951.

Conrad, P. R., and Williams, B. C. 2011. Drake: An Ecient Executive for Temporal Plans with Choice. *JAIR* 42:607–659.

Do, M. B., and Kambhampati, S. 2003. SAPA: A multi-objective metric temporal planner. *JAIR* 20:155–194.

Doran, D.; Schulz, S.; and Besold, T. R. 2017. What Does Explainable AI Really Mean? A New Conceptualization of Perspectives. *arXiv preprint arXiv:1710.00794*.

Floyd, M. W., and Aha, D. W. 2017. Using explanations to provide transparency during trust-guided behavior adaptation. *AI Communications* 30(3-4):281–294.

Fox, M., and Long, D. 2003. PDDL2.1 : An extension to PDDL for expressing temporal planning domains. *JAIR* 20:61–124.

Fox, M.; Gerevini, A.; Long, D.; and Serina, I. 2006. Plan stability: Replanning versus plan repair. In *Proc. ICAPS*, 212–221. AAAI Press.

Fox, M.; Long, D.; and Magazzeni, D. 2017. Explainable planning. In *Working notes of the IJCAI workshop on Explainable AI*.

Ghallab, M.; Nau, D.; and Traverso, P. 2016. *Automated Planning and Acting*. Cambridge University Press.

Hammond, K. J. 1990. Explaining and repairing plans that fail. *Artificial Intelligence* 45(1-2):173–228.

Hanks, S., and Weld, D. S. 1995. A domain independent algorithm for plan adaptation. *JAIR* 2:319–360.

Hawes, N. 2011. A survey of motivation frameworks for intelligent systems. *AIJ* 175(5-6):1020–1036.

Horty, J. F., and Pollack, M. E. 2001. Evaluating new options in the context of existing plans. *AIJ* 127(2):199–220.

Keil, F. C. 2003. Folkscience: coarse interpretations of a complex reality. *Trends in Cognitive Sciences* 7(8):368–373.

Koenig, S.; Furcy, D.; and Baue, C. 2002. Heuristic search-based replanning. In *ICAPS*.

Langley, P.; Meadows, B.; Sridharan, M.; and Choi, D. 2017. Explainable Agency for Intelligent Autonomous Systems. In *Proceedings of the Twenty-Ninth Conference on IAAI*, 4762–4764.

Miller, T. 2017. Explanation in artificial intelligence: Insights from the social sciences. *arXiv preprint arXiv:1706.07269*.

Montavon, G.; Samek, W.; and Müller, K. R. 2018. Methods for interpreting and understanding deep neural networks. *Digital Signal Processing: A Review Journal* 73:1–15.

Nebel, B., and Koehler, J. 1995. Plan reuse versus plan generation: a theoretical and empirical analysis. *AIJ* 76(1-2):427–454.

Nguyen, T.; Do, M.; Gerevini, A.; Serina, I.; Srivastava, B.; and Kambhampati, S. 2012. Generating diverse plans to handle unknown and partially known user preferences. *AIJ* 190:1–31.

Pollack, M., and Horty, J. 1999. There's more to life than making plans: Plan management in dynamic, multiagent environments. *AI Magazine* 20(4):71–83.

Radzi, M. 2011. *Multi-objective planning using linear programming*. Ph.D. Dissertation, Univ. of Strathclyde.

Rajan, K.; Py, F.; and Barreiro, J. 2013. *Towards Deliberative Control in Marine Robotics*. New York, NY: Springer New York. 91–175.

Roberts, M.; Apker, T.; Johnson, B.; Auslander, B.; Wellman, B.; and Aha, D. 2015. Coordinating robot teams for disaster relief. In *Proc. FLAIRS*. Hollywood, FL: AAAI Press.

Roberts, M.; Shivashankar, V.; Alford, R.; Leece, M.; Gupta, S.; and Aha, D. 2016. Goal reasoning, planning, and acting with actorsim, the actor simulator. In *Poster Proceedings of the Fourth Annual Conf. on Advances in Cognitive Systems*.

Roberts, M.; Howe, A.; and Ray, I. 2014. Evaluating diversity in classical planning. In *Proc. ICAPS*. Portsmouth, NH, USA: AAAI Press.

Rosenthal, S.; Selvaraj, S. P.; and Veloso, M. 2016. Verbalization: Narration of Autonomous Robot Experience. In *Proc. IJCAI*, 7.

Seegebarth, B.; Müller, F.; Schattenberg, B.; and Biundo, S. 2012. Making hybrid plans more clear to human users – a formal approach for generating sound explanation. In *Proc. ICAPS*, 225–233.

Sheh, R.; Schwertfeger, S.; and Visser, A. 2016. 16 years of robocup rescue. *KI - Künstliche Intelligenz* 30(3):267–277.

Sheh, R. K. 2017. Different XAI For Different HRI. *AAAI Fall Symposium Series*.

Shivashankar, V.; Alford, R.; Kuter, U.; and Nau, D. S. 2013a. The GoDeL Planning System: A More Perfect Union of Domain-Independent and Hierarchical Planning. In *IJCAI*, 2380–2386.

Shivashankar, V.; UMD EDU, R. A.; Kuter, U.; and Nau, D. 2013b. Hierarchical goal networks and goal-driven autonomy: Going where ai planning meets goal reasoning. In *Goal Reasoning: Papers from the ACS Workshop*.

Smith, D. E.; Frank, J.; and Jónsson, A. K. 2000. Bridging the gap between planning and scheduling. *The Knowledge Engineering Review* 15(1):47–83.

Smith, D. E. 2012. Planning as an iterative process. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence*, AAAI'12, 2180–2185. AAAI Press.

Sohrabi, S.; Baier, J. A.; and McIlraith, S. A. 2011. Preferred Explanations: Theory and Generation via Planning. In *Proc. AAAI*, 261–267.

Srivastava, B.; Kambhampati, S.; Nguyen, T.; Do, M.; Gerevini, A.; and Serina, I. 2007. Domain independent approaches for finding diverse plans. In *IJCAI*, 2016–2022.

Tolchinsky, P.; Modgil, S.; Atkinson, K.; McBurney, P.; and Cortés, U. 2012. Deliberation dialogues for reasoning about safety critical actions. *AAMAS* 25(2):209–259.

Vattam, S.; Klenk, M.; Molineaux, M.; and Aha, D. 2013. Breadth of approaches to goal reasoning: A research survey. In Aha, D.; Cox, M.; and Muoz-Avila, H., eds., *Goal Reasoning: Papers from the ACS Workshop (Technical Report CS-TR-5029). College Park, MD: University of Maryland, Department of Computer Science*, 222–231.