Provenance-Based Assessment of Plans in Context

Scott E. Friedman, Robert P. Goldman, Richard G. Freedman, Ugur Kuter, Christopher Geib, Jeffrey Rye
{ sfriedman, rpgoldman, rfreedman, ukuter, cgeib, jrye } @ sift.net
SIFT, LLC
Minneapolis, MN, USA

Abstract

Many real-world planning domains involve diverse information sources, external entities, and variable-reliability agents, all of which may impact the confidence, risk, and sensitivity of plans. Humans reviewing a plan may lack context about these factors; however, this information is available during the domain generation, which means it can also be interwoven into the planner and its resulting plans. This paper presents a provenance-based approach to explaining automated plans. Our approach (1) extends the SHOP3 HTN planner to generate dependency information, (2) transforms the dependency information into an established PROV-O representation, and (3) uses graph propagation and TMS-inspired algorithms to support dynamic and counter-factual assessment of information flow, confidence, and support. We qualified our approach's explanatory scope with respect to explanation targets from the automated planning literature and the information analysis literature, and we demonstrate its ability to assess a plan's pertinence, sensitivity, risk, assumption support, diversity, and relative confidence.

Introduction

In complex, dynamic, and uncertain environments, it is critical that human operators understand machine-generated plans, including their sensitivity to world changes, their reliance on individual actors, their diversity of information sources, their core assumptions, and how risky they are. This paper contributes an approach to dynamically explain and explore machine-generated single- or multi-agent, single- or multi-goal plans using *provenance-based* analysis and visualization strategies.

Most prior work on explainable planning focuses on inspecting algorithms (*i.e.*, explicating the decision-making process), synchronizing mental models (*e.g.*, because the user views the problem differently than the planner), and improving usability (*e.g.*, making complex plans more interpretable) (Chakraborti, Sreedharan, and Kambhampati 2020) and assumed fixed background domain knowledge. In contrast, our provenance-based approach treats the plan as a tripartite dependency graph that helps explain the foundations, reliability, and sensitivity of the information that comprises the plan's states and actions.

We use the definition of "provenance" from the Provenance Data Model (PROV-DM): "information about entities, activities, and people involved in producing a piece of data or thing, which can be used to form assessments about its quality, reliability or trustworthiness" (Moreau and Missier 2013). We describe the formal PROV-DM relationships among these elements later, as shown in Figure 1. We have augmented the Hierarchical Task Network (HTN) planner SHOP3 (Goldman and Kuter 2019) with the ability to annotate its plans with provenance by recording, on the fly, (1) causal dependencies, (2) dependencies from plan components onto aspects of the model (domain) from which they derive, and (3) sources of information used by the planner in checking preconditions and deriving beliefs.

The provenance of the SHOP3 plan feeds into our downstream provenance analysis, which uses PROV-DM to represent beliefs, planned activities, and actors, and the recent DIVE ontology (Friedman et al. 2020) to represent assumptions, confidence, and likelihood of those PROV-DM elements. Our approach combines truth maintenance (Forbus and de Kleer 1993) and provenance propagation (Singh, Cobbe, and Norval 2018; Gehani and Kim 2010; Pasquier et al. 2016) to estimate the confidence in the correctness of planned actions, and counterfactually assess the *sensitivity* of the plan to [the absence of] various data sources, actors, events, and beliefs.

Our central claim is that tracking and analyzing a plan's provenance can improve the interpretation of plans— along dimensions of confidence, information dependency, risk, and sensitivity— without reducing the efficiency of the planner or the complexity of the search space. To support this claim, we demonstrate our approach within a provenance visualization environment (Friedman et al. 2020). This provenance-based approach is especially useful for explaining plans with multiple goals and for plans with multiple actions to achieve a given goal. While our demonstration uses provenance analysis after planning completes, we identify future avenues for using provenance *within* a planner to advise search heuristics, mixed-initiative planning, contingency planning, and replanning.

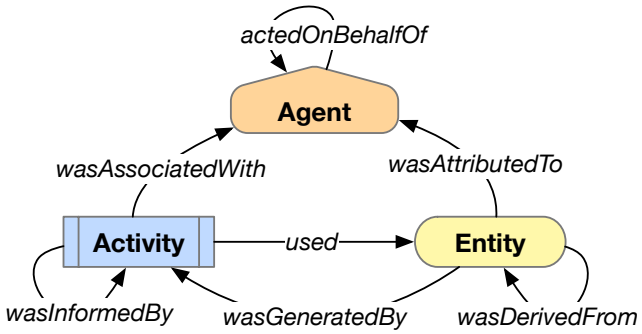

Figure 1: PROV ontology subset used in our approach.

We continue with a review of relevant background in provenance-tracking and HTN planning. We then describe our approach using provenance as a platform for plan explanation and assessment, qualifying the types of planning questions that our approach addresses. We demonstrate our system facilitating plan assessment, and we review the results and outline future work in our conclusion.

Background

Provenance-Tracking

We utilize the PROV-O ontology (Lebo et al. 2013), which expresses PROV Data Model's entities and relationships using the OWL2 Web Ontology Language. The PROV Data Model includes the following three primary classes of elements to express provenance:

1. **Entities** are real or hypothetical things with some fixed aspects in physical or conceptual space. These may be beliefs, documents, databases, inferences, *etc.*.
2. **Activities** occur over a period of time, processing and/or generating entities. These may be inference actions, judgment actions, planned (not yet performed) actions, *etc.*.
3. **Agents** are responsible for performing activities or generating entities. These may be humans, machines, rovers, web services, *etc.*.

The primary relationships over these three classes in PROV are shown in Figure 1, as detailed in the W3C PROV-O recommendation.[1]

Systems that utilize PROV-O, as specified in Figure 1, can represent long inferential chains, formally linking conclusions (*e.g.*, a downstream belief) through generative activities (*e.g.*, inference operations) and antecedents, to source entities and assumptions. This comprises a directed network of provenance that we can traverse in either direction to answer questions of foundations, derivations, and impact.

The DIVE Ontology

The DIVE ontology (Friedman et al. 2020) extends the PROV ontology with additional classes and relationships to appraise information and validate information workflows.

[1]https://www.w3.org/TR/2013/REC-prov-o-20130430/

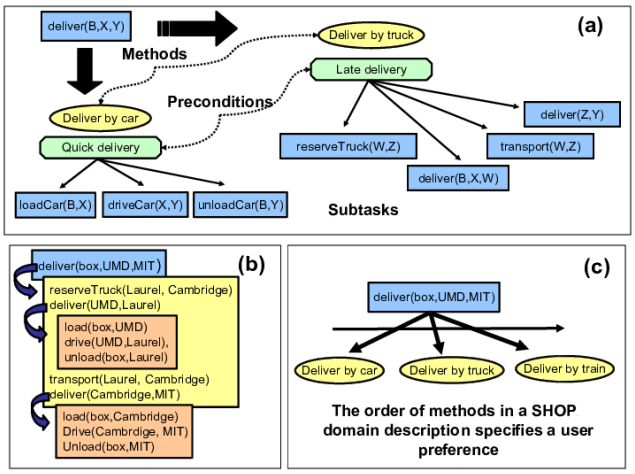

Figure 2: Delivery planning example.

For this work, we use DIVE's **Appraisal** class, which is an **Agent**'s judgment about an activity, entity, or other agent.

For example, we express a DIVE **Appraisal** about a GPS sensor—from which we derive beliefs about the world before planning and during plan execution—with moderate baseline confidence. This baseline confidence in our GPS sensor may affect our confidence of the information it emits, all else being equal, which may ultimately impact our judgment of the success likelihood of our planned actions.

We also use DIVE to express *collection disciplines* such as GEOINT (geospatial), IMINT (image), and other types of information for all relevant information sources, beliefs, and sensors involved in a plan. DIVE is expressed at the meta-level of PROV. DIVE expressions flow through the network to facilitate downstream quality judgments and interpretation, as we demonstrate in this work.

The SHOP3 HTN Planner

SHOP3 (Goldman and Kuter 2019) is the successor to the SHOP2 HTN planner (Nau et al. 2003) developed at the University of Maryland. Unlike a first principles planner, an HTN planner produces a sequence of actions that perform some activity or *task*, instead of finding a path to a goal state. An HTN planning domain includes a set of planning *operators* (actions) and *methods*, each of which is a prescription for how to decompose a task into its *subtasks* (smaller tasks). The description of a planning problem contains an initial state as in classical planning. Instead of a goal formula, however, there is a partially-ordered set of tasks to accomplish. Planning proceeds by decomposing tasks recursively into subtasks, until *primitive tasks*, which can be performed directly using the planning operators, are reached. For each task, the planner chooses an applicable method, instantiates it to decompose the task into subtasks, and then chooses and instantiates other methods to decompose the subtasks even further. If the constraints on the subtasks or the interactions among them prevent the plan from being feasible, the planner will backtrack and try other methods. Figure 2 illustrates how SHOP3 HTN domains are described

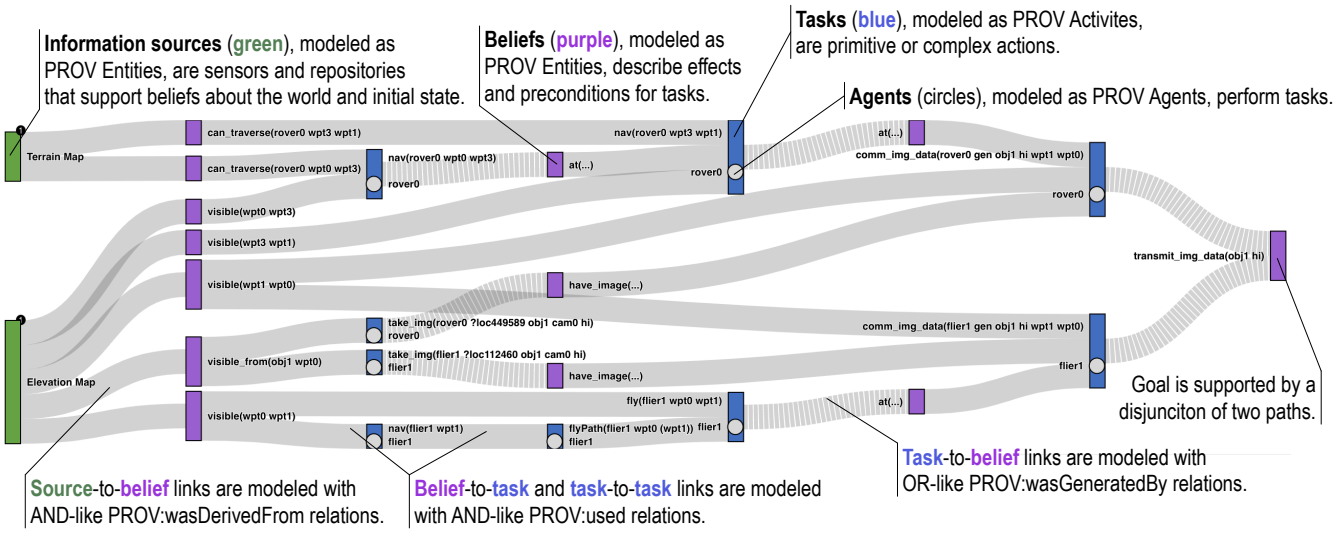

Figure 3: How we represent (with PROV) and display (with D3.js) the provenance of SHOP3 plans.

and used for planning in a Delivery planning example.

SHOP3 is an HTN planner that generates actions in the order they will be executed in the world (hence "hierarchical *ordered* planner" in the name). SHOP2 is relatively efficient and well-tested, and it performed well in the 2002 IPC–the last IPC in which HTN planners competed (Long and Fox 2003). SHOP2 also has been used in a number of planning applications, including recently at SIFT for Air Operations and UAV planning (Musliner et al. 2011; Mueller et al. 2017), cyber security (Burstein et al. 2012), cyber-physical systems (Goldman et al. 2016), planning for synthetic biology experiments (Kuter et al. 2018), and more. For an earlier survey of SHOP2 applications, see Nau, *et al.* (2005). SHOP3 retains the essential features of SHOP2, but has a modernized codebase, is easier to extend (*e.g.* with plan repair capabilities, new input languages, *etc.*), and an alternative, more efficient search engine.

Approach

We first describe how we extended the SHOP3 planner to emit dependency information to support provenance. We then describe our approach with respect to relevant questions from the planning literature (Fox, Long, and Magazzeni 2017) and information analysis literature (Office of the Director of National Intelligence 2015; 2007; Zelik, Patterson, and Woods 2010) that have been proposed as primary targets for integrity and explainability. We describe relevant representations and algorithms in our approach as they apply to these questions.

SHOP3 and Provenance Tracking

In related work on plan repair (Goldman, Kuter, and Freedman 2020), we have augmented SHOP3 so that, when planning, it builds a plan tree that has dependency information (causal links). These links allow the plan repair system to identify the minimally compromised subtree of the plan, as a way to provide *stable*, minimal-perturbation plan repairs. This extension provides much of the provenance information that we need for explainability, because it allows us to trace the choice of methods and primitive actions back to other choices that enabled them. The present approach extends the scope and semantics of these links to (1) trace decisions back to the model components that justify them and (2) trace preconditions back to actions that establish them and information sources that provided them.

Tracing decisions back to model components is straightforward: the SHOP3 planner takes as input `domain` and `problem` data structures, and the `domain` data structures contain the model components, specifically the primitive operator and method definitions. For the moment, we do not track the provenance of components of the planner's model. However, since the domain descriptions are typically maintained in a revision control system, such as subversion or git, it would be relatively simple to extend our provenance tracing back to the person or persons who wrote these model components. For a more sophisticated development environment, one could imagine a traceback that reaches into an integrated development environment or a machine learning system.

Tracing decisions back to information sources is somewhat more difficult. In the base case, a proposition is established in the `problem` data structure – that is, in the initial state. In a larger system that incorporates SHOP3, there is generally a component that builds these `problem` data structures. For example, in a robot planning system, we generally have a component that builds problems programmatically from user input (tasks to achieve) and some source of external information (*e.g.*, a map database, telemetry from robotic platforms, *etc.*). These components can annotate the initial state (and potentially the tasks SHOP3 is asked to plan) with provenance information, using PROV-DM in a way that is appropriate to the application domain. This

provenance information can then be propagated through the causal links in the plan tree.

There is one remaining complication: in the interests of modeling efficiency and expressivity, SHOP3 incorporates a theorem-prover – a backward-chaining engine inspired by Prolog (Warren, Pereira, and Pereira 1977). This is necessary because SHOP3's expressive power is not limited to propositional logic, the way most planners are: it permits state axioms, and non-finite domains of quantification.[2] Thus some preconditions may be established not just causally, but inferentially, through Horn clause ("axiom") deduction. Accordingly, we must extend our theorem-prover so that it also provides traceability. Provenance annotations that traced provenance through axioms back to actions that established antecedents for the axioms were already in place for plan repair. These will now automatically incorporate information source provenance, as well as causal provenance. At the moment, we do not trace the axioms themselves, but this would be a trivial extension.

Mapping SHOP3 plans to PROV

Our system converts the extended SHOP3 plans into the PROV data model, using the PROV-O ontology to represent the elements and relationships between them. Figure 3 illustrates the SHOP-to-PROV mapping in a screenshot of our system displaying SHOP3 planner output (some elements removed for simplicity). The plan content in Figure 3 displays a single goal (at right) to transmit image data of **objective1** in high-resolution, and this goal is supported by two paths of tasks, performed by two separate agents (the aerial unit **flier1** and the land unit **rover0**), with foundational beliefs derived from a **Terrain Map** and an **Elevation Map**. We use the following mapping:

- **Planned Tasks** are specializations of **prov:Activity**. Unlike traditional uses of provenance for tracking *past* events, the PROV Activities from the plan may not yet have–or may never actually–occur.
- **Plan Actors** are specializations of **prov:Agent**. They are the performers of the PROV Activities, related via **prov:wasAssociatedWith** (see Figure 1).
- **Plan Beliefs** are specializations of **prov:Entity**. They support tasks with **prov:used** and they are realized by tasks with **prov:wasGeneratedBy**.
- **Information Sources** are specializations of **prov:Entity**. They represent sensors and repositories that emit information to derive beliefs and measurements used in the plan, and support beliefs via **prov:wasDerivedFrom**.

As shown in the Figure 3 screenshot, the resulting provenance graph incorporates the information sources (at left) with the goals of the plan (at right), and the dependency network between them. We use this SHOP3 plan to acquire **objective1** imagery–with one or more possible planned courses of action–to articulate our approach below.

[2]Note that though critical to SHOP3 applications, these features must be used with care, because they can compromise soundness and completeness.

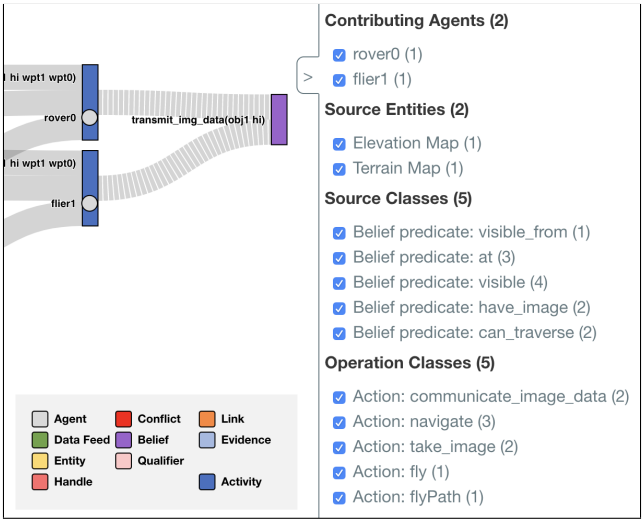

Figure 4: The index of agents, sources, source classes, and operation classes used in a plan.

Indexing the Dimensions of a Plan

Given a plan to assess, our provenance system automatically identifies and catalogs the following dimensions of the plan. These are displayed for user assessment and dynamic interaction, as shown in Figure 4.

1. **Contributing Agents**: Actors in the plan.
2. **Source Entities**: Individual devices or informational resources from which plan-relevant beliefs are derived, such as geolocation, visibility, inventory, and more.
3. **Source Classes**: General categories of information across beliefs and information sources. These may include information sources or belief predicates, as shown in Figure 4.
4. **Operation Classes**: General categories of activities, spanning potentially many planned activities. In Figure 4, we catalog classes of actions.

Cataloging plan nodes along these dimensions allows our approach to instantaneously identify, emphasize, or refute nodes along these dimensions to support explanation. These elements are identified by mining the predicates and sources of the plan, but could also be informed by the planner's model, in future work.

We use an algorithm similar to assumption-based truth-maintenance and explanation-maintenance systems (Forbus and de Kleer 1993; Friedman, Forbus, and Sherin 2018) to compute the *environment* of all nodes (*i.e.*, planned action or belief) in the provenance graph. The algorithm traverses backward exactly once from all sink nodes, so it reaches each node m in the provenance graph and computes its environment $E(m) = \{S_1, ..., S_n\}$, a disjunction of sets (S_i) of *assumptions*, where any $S_i \in E(m)$ is sufficient to derive (*i.e.*, believe, achieve, or enact) m, and where the assumptions correspond to root nodes in the provenance graph. The algorithm attends to the AND- and OR-like links listed in Figure 3 to properly encode disjunctive derivation trees.

This compact index answers questions of necessity and sufficiency in constant time.

The joint indexing of plan nodes by the four above dimensions and by their environments allows the provenance analysis system to identify abstract classes of sources and operations that contribute to it, and that it contributes to. We leverage these indices to help explain the plan in context, as we describe below.

Visualization Environment

Our visualization environment is a graphical display within a larger web-based platform for human-machine collaborative intelligence analysis. At any time, the user may select one or more elements from diagrams or listings and peruse its full provenance.

A web service traverses the knowledge graph to retrieve the full provenance for the desired belief(s) and all relevant appraisals, and then sends it to the client. The client's provenance visualizer uses D3.js, as shown in the Figure 3 and Figure 4 screenshots, to implement the rendering, refutation, emphasis, and propagation effects described below, operating over the PROV and DIVE representations.

Assessing Explainability of our Approach

The majority of prior work on explainable planning focuses on inspecting algorithms (*i.e.*, explicating the decision-making process), synchronizing mental models (*e.g.*, because the user views the problem differently than the planner), and improving usability (*e.g.*, making complex plans more interpretable) (Chakraborti, Sreedharan, and Kambhampati 2020) and assumed fixed background domain knowledge. In contrast, our provenance-based approach treats the plan as a tripartite (**Agents**, **Entities**, and **Activities**) dependency graph. This adds connections among the plan's beliefs and goals (PROV entities), actions (PROV activities), and actors (PROV agents) via type-specific dependency relations. The plan's provenance graph connects to other provenance information (if available), including belief derivations (*e.g.*, describing how initial state beliefs were inferred, as in Figure 3), agent descriptions, and sensor descriptions (*e.g.*, including reliability information), which comprise a larger global provenance graph. This complements previous explainable planning work with additional decision-relevant information and thereby new explanation capabilities.

Explanation in Information Analysis

We review questions from information analysis that are relevant but under-explored for automated planning, especially when a plan's world state is derived and supported by diverse information. These questions stem primarily from directives for integrity in intelligence analysis (Office of the Director of National Intelligence 2015; 2007), and measurements of rigor in analytic workflows (Zelik, Patterson, and Woods 2010). For each question, we briefly note whether our approach addresses it adequately (✓) or partially (∼) or whether it is out of scope (✗).

(✓) How reliable is the information supporting this course of action? We answer this question of information reliability with graph propagation, using all DIVE **Appraisal** instances with numerical confidence ratings and propagating them forward to estimate downstream nodes' confidence. Figure 5 illustrates the **Elevation Map** appraised with moderately high (0.80) confidence and the **Terrain Map** appraised with moderately low (0.20) confidence. We see that the downstream goal (rightmost node) is supported by two paths of varying estimated confidence, where the low confidence begins at the **Terrain Map** and flows through the **rover0** sub-plan. In the present propagation policy, a conjunction is as reliable as the lowest-confidence upstream input and a disjunction is as reliable as the greatest-confidence source upstream, but Bayesian approaches may also apply here (Kuter and Golbeck 2010).

(✓) What information sources, sensors, or actors are pertinent to this [class of] belief or action? Our system answers this question of *information support* using the precomputed environment (described above) to identify all upstream necessary and sufficient nodes in constant time. The Figure 6 screenshot shows the effect of hovering over the **take_image** action in the right-hand panel: the system (1) identifies all nodes catalogued with that action (highlighted with purple glow in Figure 6), and then (2) de-emphasizes all nodes and paths that are not pertinent, so all relevant supporting nodes (upstream of the **take_image** nodes) are available for assessment. We see that all **take_image** actions rely solely on (1) a belief about **objective1** visibility from **waypoint0** and (2) a high-confidence information source.

(✓) How far has this belief/agent/information source influenced my plan? Our approach answers this *impact assessment* question using belief environments: the impact of a belief, agent, or information source m in the provenance graph is the set of elements with m in any subset of their environments. The impact of the **take_image** nodes is shown downstream of those nodes in the Figure 6 screenshot: the **take_image** actions directly impact the communication of image data, in both sub-plans, thereby indirectly impacting the rightmost goal along both avenues.

(✓) How necessary are these sources, beliefs, actions, or actors for an action or goal? This is known as *sensitivity analysis*, and is answerable using environments, as defined above. Given an element m, we can answer whether one or more other elements N are necessary by computing m's environment contracted by N:

$$E(m)/N = \{\, S \in E(m) : N \cap S = \emptyset \,\}$$

If $E(m)/N = \emptyset$, at least one element in N is necessary for m. This allows us to interactively *refute* elements in the provenance graph and observe the downstream effects, answering counter-factual "*what-if*" questions about the necessity of information and actors in the plan.

Our system supports sensitivity analyses via dynamic *refutation* as shown in Figure 7: the user may refute a class of elements (Figure 7, top); information sources (Figure 7, middle); agents (Figure 7, bottom); or any individual node.

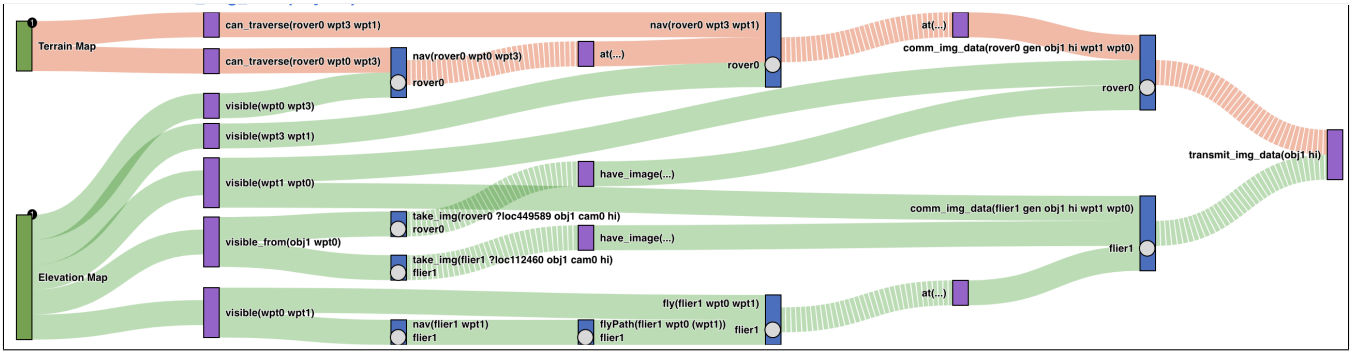

Figure 5: The Figure 3 plan to acquire imagery of **objective1**, with moderately low (0.20) confidence ascribed to the **Terrain Map** and moderately high (0.80) confidence ascribed to the **Elevation Map**.

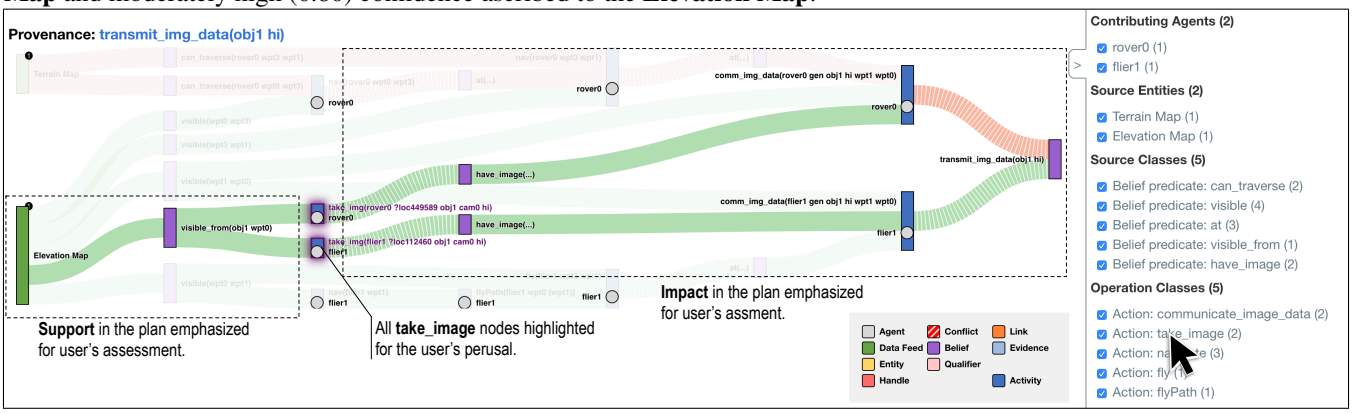

Figure 6: Viewing the relevance and reachability of the **take-image** action, including the sources it relies on, and the impact on downstream actions.

The system contracts nodes' belief environments, as described above, to identify downstream that have lost all support. Note that the downstream goal is still reachable in two of these Figure 7 refutations; however, the confidence of the goal varies depending on which element we refute.

(✓) What assumptions are necessary or sufficient to hold this belief or apply this planned action? Deriving beliefs from information sources often requires making some assumptions. For instance, using a rover's GPS sensor to measure its position assumes that *the GPS sensor is on the rover*. This assumption affects the integrity of all downstream beliefs and planned actions that rely directly or indirectly on positional data.

As with numerical confidence, we express assumptions using DIVE **Appraisal** instances related to the relevant elements (*e.g.*, a GPS sensor). For any node m, we compute the set of necessary and sufficient upstream assumptions as the set of explicit assumptions on the necessary and sufficient nodes in $E(m)$.

Explanation in Automated Planning

We consider explainable planning questions from Fox, Long, and Magazzeni (2017) in relation to our approach:

(✓) Why did you do that? Given any action, our approach uses the provenance structure to identify source nodes (*i.e.*, information sources), sink nodes (*i.e.*, goals), and intermediate nodes (*i.e.*, beliefs and other actions) that explain upstream information support and downstream depenedencies. Through the interface, one can simply hover their mouse over the inquired action and view these relationships from the provenance structure (see Figure 6). The upstream information support indicates a justification for how the decision contributes to the goals, and the downstream dependencies provide reasons for why the decision was possible to make (compared to other potential decisions that could also achieve the goal).

While this can be as simple as visualizing causal links of preconditions and effects, the reliability of information can also play a role. The inquired action may involve entities with greater DIVE **Appraisal**, for example.

(✗) Why can't you do that? This question concerns an action that was *not* included in the plan, and the analysis in this work is limited to analyzing components within the plan (*i.e.*, only actions emitted by the planner). The provenance structures are constructed based on the *outcome of the planning process*, which is an annotated HTN without vestigial structures from actions that were not selected for the returned plan. Therefore, this question is out of scope for our

Figure 7: Counter-factually refutation of: a class of operations **take_image** (top); an information source **Terrain Map** (middle); and a entire agent **flier1** (bottom) from the plan. Refuting elements allows us to see the impact on downstream goals and actions, and the confidence of the information supporting them.

provenance-based analysis.

(✗) Why didn't you do something else? This question frames a planned action against the space of other, unplanned actions. Thus it is also out of scope for similar reasons to the above explainability question. Without the context of an unplanned (novel) action, such comparisons between actions cannot be made.

(∼) Why is that more efficient/safe/cheap than something else? Our provenance-based approach propagates confidence—or alternatively, source reliability or operational risk—downstream through the provenance graph, allowing upstream agents, beliefs, and information sources to color downstream actions and beliefs in the plan. This estimation of downstream confidence and risk (as an inverse of "safe," per the question) allows us to compare alternatives across numerical measures. This does not fully address the question, since propagating confidence does not explain resource costs and efficiency.

(∼) Why do I [not] need to replan or repair the plan at this point? This extends to specific questions about plan robustness such as, "What can go wrong with this plan, and

why?" *e.g.*, "what will happen if this rover breaks down?" Connecting the rover to actions and goals that involve it enable the planning system to explain the overall impacts of such a query, rather than simply identify the chain of broken causal links in a single plan instance (Bercher et al. 2014).

It is trivial to reassign a DIVE **Appraisal** of an entity, since the provenance structures do not change: the new values propagate after updating the confidence and reliability of the remaining plan components. Hence reducing the reliability of a rover that seems likely to break down will downgrade the estimated confidence in the portion of the plan that the rover supports. Similarly, dynamically refuting the unreliable rover, as illustrated in Figure 7, will instantly remove elements of the plan that rely on it.

If there are still sufficient paths to the goal condition—or paths that are of the desired confidence—then the plan is robust enough to address the inquired failure points, and it does not require revision. Alternatively, if the remaining paths to the goal are not of a desired confidence, then these refuted elements (and the degraded paths) explain why revising the plan is necessary.

Related Work

In the AI planning community, it has shown that it is possible to formalize "model synchronization" as a meta-search problem, where traditional search and classical planning algorithms explore explanations with differing properties (Chakraborti et al. 2017; Chakraborti, Sreedharan, and Kambhampati 2018). One of the key insights model synchronization is that explanations are generally needed to identify mismatches in planning models produced by different information sources. This is critically important when the distance between different descriptions of a planning domain cannot capture a cohesive model sufficiently. While explanations can certainly deviate from our actual methods of decision making (Klein 2008) they nevertheless represent how humans are trained and acculturated to providing rationalizations for our decision making. In that sense, we believe this line of work is complementary to our approach in this paper: two approaches can be combined in order to formalize and reason about properties such as social trust, analytic trust, communication frequencies, and others. This approach can balance the trade-offs between explicability and explanations for social interactions. In particular, an "optimal" AI agent might generate an estimate of the state of the world that is inexplicable to humans and model synchronization and provenance tracing will enable an AI agent to choose a less optimal model of the state that would enable an easier explanation to the human users and is close to (but not the same as) the AI agent's actual domain models (Chakraborti, Sreedharan, and Kambhampati 2018).

Provenance-tracking is well-established as a practical tool across source domains (Gehani and Kim 2010; Pasquier et al. 2016) for decision support (Singh, Cobbe, and Norval 2018), complex multi-agent workflows (Toniolo et al. 2015; Friedman et al. 2020), and lineage-tracking for databases (Benjelloun et al. 2008). These previous works have not been applied to the domain of planning, so we believe this work is the first to investigate the explainability of automated planning using provenance.

Label propagation has been used to detect persistent security threats in real time (Han et al. 2020) by propagating information through network flows. We have previously used label propagation for plan recognition with support for refutation, similar to what we demonstrate here (Goldman, Friedman, and Rye 2018), but this previous approach did not use formal provenance notation or operate on forward-generated plans.

Conclusions

This paper presented a provenance-based approach for improving the explainability of plans. Our approach (1) extends the SHOP3 HTN planner to generate dependency information, (2) transforms the dependency information into an established PROV-O representation, and (3) uses graph propagation and TMS-inspired algorithms support dynamic and counter-factual assessment of information flow, confidence, and support.

We qualified our approach's explanatory scope with respect explanation targets from the automated planning literature (Fox, Long, and Magazzeni 2017) and the information analysis literature (Office of the Director of National Intelligence 2015; 2007; Zelik, Patterson, and Woods 2010). We demonstrated that our approach answers questions of pertinence, sensitivity, risk, assumption support, diversity of evidence, and relative confidence. Our approach is limited to explaining elements of the plan itself: it does *not* explain why a given action was not planned or whether an action is plannable via the planner's internal model.

Our provenance approach might be able to help explain "the road not taken" if the planner represents decision points and constraints in the dependency-based plan. This would not enable complete *"what-if"* hypothetical explanations, but it would explain local rationale for planning decisions within context.

This paper used simple example plans. Our underlying graph propagation and TMS algorithms easily scale to larger datasets, and they can execute incrementally when graphs (*i.e.*, automated plans) are revised online (Forbus and de Kleer 1993). However, we face a non-computational scalability problem of user experience: the UI to display the associated provenance cannot intuitively display full plans with hundreds of nodes without graph summarization and graph filtering, which is one avenue of future work.

Future Work

This work demonstrates the explanatory value of provenance for analysis *after* the planning process. We see value in integrating these provenance-based analyses *online*, as well, into a continuous and dynamic planning environment, interleaving provenance analysis and planning. Some possibilities for using provenance while planning include: heuristic guidance (*e.g.*, preferring choices based on higher-confidence information); guiding contingency planning (*e.g.*, prepare for more likely nondeterministic outcomes based on reliability

of sensors, information sources, *etc.*); or to plan repair (*e.g.* triggering the planner to make revisions when provenance changes for the worse).

Furthermore, as a source of explainability to other agents, there is a potential for novel uses in multi-agent planning scenarios such as decentralized planning (*e.g.*, evaluating other agents' performance to assess reliability of their action outcomes and relayed information) and mixed-initiative planning (*e.g.*, using the interface to detail the current plan's provenance and receive iterative changes from the user to parameters such as DIVE **Appraisals**).

Acknowledgments

This work was funded primarily by a SIFT Internal R&D project. We thank Kanna Rajan for his suggestions.

References

Benjelloun, O.; Sarma, A. D.; Halevy, A.; Theobald, M.; and Widom, J. 2008. Databases with uncertainty and lineage. *The VLDB Journal* 17(2):243–264.

Bercher, P.; Biundo, S.; Geier, T.; Hoernle, T.; Nothdurft, F.; Richter, F.; and Schattenberg, B. 2014. Plan, repair, execute, explain - how planning helps to assemble your home theater. In *ICAPS 2014*, 386–394.

Burstein, M.; Goldman, R.; Robertson, P.; Laddaga, R.; Balzer, R.; Goldman, N.; Geib, C.; Kuter, U.; McDonald, D.; Maraist, J.; Keller, P.; and Wile, D. 2012. Stratus: Strategic and tactical resiliency against threats to ubiquitous systems. In *SASO-12*.

Chakraborti, T.; Sreedharan, S.; Zhang, Y.; and Kambhampati, S. 2017. Plan explanations as model reconciliation: Moving beyond explanation as soliloquy. *ICJAI 2017*.

Chakraborti, T.; Sreedharan, S.; and Kambhampati, S. 2018. Explicability versus explanations in human-aware planning. In *AAMAS 2018*, 2180–2182.

Chakraborti, T.; Sreedharan, S.; and Kambhampati, S. 2020. The emerging landscape of explainable automated planning & decision making. In Bessiere, C., ed., *IJCAI 2020*, 4803–4811.

Forbus, K. D., and de Kleer, J. 1993. *Building Problem Solvers*, volume 1. MIT press.

Fox, M.; Long, D.; and Magazzeni, D. 2017. Explainable Planning. In *Proceedings of IJCAI-17 Workshop on Explainable Planning*.

Friedman, S. E.; Rye, J. M.; LaVergne, D.; Thomsen, D.; Allen, M.; and Tunis, K. 2020. Provenance-based interpretation of multi-agent information analysis. In *Proceedings of TaPP 2020*.

Friedman, S.; Forbus, K.; and Sherin, B. 2018. Representing, running, and revising mental models: A computational model. *Cognitive Science* 42(4):1110–1145.

Gehani, A., and Kim, M. 2010. Mendel: Efficiently verifying the lineage of data modified in multiple trust domains. In *Proceedings of the 19th ACM International Symposium on High Performance Distributed Computing*, 227–239.

Goldman, R. P., and Kuter, U. 2019. Hierarchical Task Network Planning in Common Lisp: The case of SHOP3. In *Roceedings of the 12th European Lisp Symposium*.

Goldman, R. P.; Bryce, D.; Pelican, M. J. S.; Musliner, D. J.; and Bae, K. 2016. A Hybrid Architecture for Correct-by-Construction Hybrid Planning and Control. In Rayadurgam, S., and Tkachuk, O., eds., *NASA Formal Methods*, volume 9690. 388–394.

Goldman, R.; Friedman, S.; and Rye, J. 2018. Plan recognition for network analysis: Preliminary report. In *AAAI Workshop on Plan, Activity and Intent Recognition, February 2018*.

Goldman, R. P.; Kuter, U.; and Freedman, R. G. 2020. Stable Plan Repair for State-Space HTN Planning. Unpublished manuscript: under review.

Han, X.; Pasquier, T.; Bates, A.; Mickens, J.; and Seltzer, M. 2020. UNICORN: Runtime Provenance-Based Detector for Advanced Persistent Threats. *arXiv preprint arXiv:2001.01525*.

Klein, G. 2008. Naturalistic decision making. *Human factors* 50(3):456–460.

Kuter, U., and Golbeck, J. 2010. Using probabilistic confidence models for trust inference in web-based social networks. *Transactions on Internet Technology (TOIT)* 7:1377–1382.

Kuter, U.; Goldman, R. P.; Bryce, D.; Beal, J.; DeHaven, M.; Geib, C.; Plotnick, A. F.; Roehner, N.; and Nguyen, T. 2018. Xplan: Experiment planning for synthetic biology. In *Proceedings of the ICAPS-18 Workshop on Hierarchical Planning*.

Lebo, T.; Sahoo, S.; McGuinness, D.; Belhajjame, K.; Cheney, J.; Corsar, D.; Garijo, D.; Soiland-Reyes, S.; Zednik, S.; and Zhao, J. 2013. Prov-o: The prov ontology. *W3C recommendation* 30.

Long, D., and Fox, M. 2003. The 3rd International Planning Competition: Results and Analysis. *JAIR* 20:1–59.

Moreau, L., and Missier, P. 2013. PROV-DM: The PROV data model. W3C Recommendation REC-prov-dm-20130430, W3C.

Mueller, J. B.; Miller, C. A.; Kuter, U.; Rye, J.; and Hamell, J. 2017. A human-system interface with contingency planning for collaborative operations of unmanned aerial vehicles. In *AIAA Information Systems*.

Musliner, D.; Goldman, R. P.; Hamell, J.; and Miller, C. 2011. Priority-based playbook tasking for unmanned system teams. In *Proceedings AIAA*.

Nau, D. S.; Au, T.-C.; Ilghami, O.; Kuter, U.; Murdock, J. W.; Wu, D.; and Yaman, F. 2003. Shop2: An htn planning system. *J. Artif. Intell. Res.(JAIR)* 20:379–404.

Nau, D.; Au, T.-C.; Ilghami, O.; Kuter, U.; Muñoz-Avila, H.; Murdock, J. W.; Wu, D.; and Yaman, F. 2005. Applications of SHOP and SHOP2. *IEEE Intelligent Systems* 20(2):34—41.

Office of the Director of National Intelligence. 2007. Intelligence community directive 206: Sourcing requirements for disseminated analytic products.

Office of the Director of National Intelligence. 2015. Intelligence community directive 203: Analytic standards.

Pasquier, T. F.-M.; Singh, J.; Bacon, J.; and Eyers, D. 2016. Information flow audit for PaaS clouds. In *2016 IEEE International Conference on Cloud Engineering (IC2E)*, 42–51. IEEE.

Singh, J.; Cobbe, J.; and Norval, C. 2018. Decision provenance: Harnessing data flow for accountable systems. *IEEE Access* 7:6562–6574.

Toniolo, A.; Norman, T.; Etuk, A.; Cerutti, F.; Ouyang, R. W.; Srivastava, M.; Oren, N.; Dropps, T.; Allen, J. A.; and Sullivan, P. 2015. Supporting reasoning with different types of evidence in intelligence analysis. *AAMAS 2015*.

Warren, D.; Pereira, L.; and Pereira, F. 1977. PROLOG – The language and its implementation compared with LISP. *Proceedings of the Symposium on Artificial Intelligence and Programming Languages* 12(8).

Zelik, D. J.; Patterson, E. S.; and Woods, D. D. 2010. Measuring attributes of rigor in information analysis. *Macrocognition metrics and scenarios: Design and evaluation for real-world teams* 65–83.