A Web-Based Platform for Iterative Planning with Plan Explanations

Rebecca Eifler and **Jörg Hoffmann**
Saarland University, Saarland Informatics Campus, Germany
{eifler, hoffmann}@cs.uni-saarland.de

Abstract

In a variety of application settings, the user preference for a planning task – the precise optimization objective – is difficult to elicit. One possible remedy, suggested by prior work, is planning as an iterative process, allowing the user to iteratively refine and modify example plans. As part of such an iterative process, it is furthermore useful to provide explanations, answering user questions about the current plan. Here we contribute a platform and user interface for such human-guided iterative planning, instantiating the explanations with plan-property dependencies ("Why does the plan you suggest not satisfy p?" "Because then we would have to forego q.") as suggested by recent work. The platform is Web-based, running in standard browsers and connecting to a server for planning processes. It provides interfaces for developers (setting up case studies) as well as end users (using the iterative planning tool), and it allows to set up user studies where test persons perform iterative planning in a controlled setting.

Introduction & Background

In many real life settings, like space mission control, production planning in Industry 4.0, or robot-aided disaster recovery, typically not all goals, constraints and preferences are known from the beginning. There may even be different user groups with different preferences. Take for instance researchers from different fields in a mission control center, who all have to be satisfied with the plan. Given this setting, the traditional planning workflow – select a set of goals, compute a plan, execute – is not adequate. Instead planning should support the users in making up their minds, exploring plan space until they are satisfied.

A natural framework for the latter is *planning as an iterative process* (Smith 2012). This allows the human users to iteratively refine their preferences and goals based on example plans. In each iteration the user can adjust the preferences and goals that yield the hard goals, resulting in a new plan. Furthermore the possibility to provide explanations, answering user questions about the current plan, is a key feature in this setting. In particular, user questions of the form "Why does the suggested plan not satisfy preference p?" are relevant. Such explanations allow the user to develop a deeper understanding of the space of plans. This in turn enables the user to refine her goals and preferences accordingly.

Eifler et al. (2020a; 2020b), henceforth referred to as *Eif20*, introduced a framework addressing the problem of generating explanations via *plan-property exclusions*, as follows. A plan property p is a Boolean function on plans, here given as LTLf (De Giacomo, De Masellis, and Montali 2014) formulas over actions and facts. Plan property p excludes q if every plan that satisfies p dooes not satisfy q. The above stated question could then be answered with "Because if you achieve p you have to forego q" . As Eif20 argue, such explanations can naturally fit into the iterative planning workflow. Furthermore, the goals, constraints and preferences underlying the iterative process as per (Smith 2012) can be naturally viewed as plan properties as well. This results in an overall approach to iterative planning where users explore plan space in terms of the combinations of plan properties that turn out to be (in)feasible. Here we contribute a Web-based platform implementing this approach and making it readily accessible to different user groups.

Platform

Our platform supports both, performing iterative planning as well as developing user studies. In the former, end users are enabled to perform iterative planning with plan properties representing goals and preferences. It is possible to enforce selected plan properties reflecting changing preferences across planning iterations, one can ask questions about the iteratively refined plans, and the tool allows to add new plan properties to hone in on new issues that become apparent during the iterative planning process. To accommodate test persons of a user study unfamiliar with planning, our tool comprises a simplified version, with a fixed set of plan properties, and with an enriched visualization of the planning task. The tool also supports developers to conduct user studies in an online setting. A user study can be composed of multiple iterative planning tasks, links to external questionnaires and additional descriptions and instructions. During a user study, the steps a test person performs and the time she spends at each of these can be logged.

The platform implementation consists of two parts. The front end runs in all standard Web browsers and provides the interfaces for all user groups. Storing data and computing plans and explanations is realized in the back end. As

a planner we use Fast Downward (Helmert 2006) and as an explanation generator the implementation of Eif20. The translation of plan properties to goal facts is realized using Eif20's implementations for LTL plan properties based on (Edelkamp 2006; Baier and McIlraith 2006).

Interface & Implementation

In the following, we give a short overview of the iterative planning interface, some example explanations in a transport domain and the definition of plan properties and the setup of user studies.

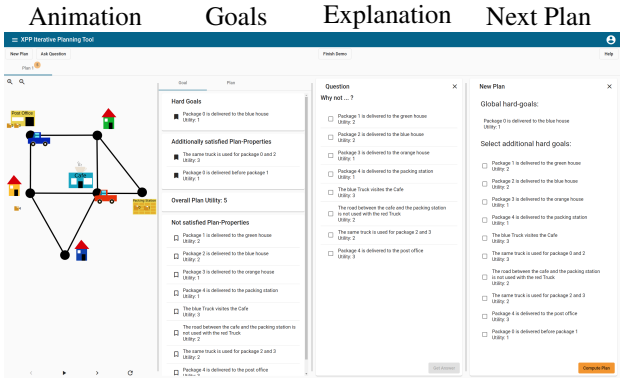


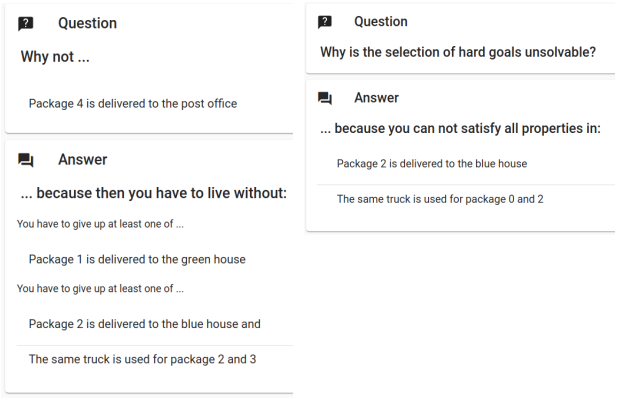
Figure 1: Interface overview: Iterative planning.

Iterative Planning Figure 1 depicts the main interface for iterative planning. It is divided into four columns. The first column contains the graphical representation of the planning task. The hard goals and the additionally satisfied soft goals of the selected plan are listed in the second column. Explanations for the current plan are provided in the third column. The fourth column hosts the interface for selecting the hard goals for the next iteration. This interface is used by end users and test persons for iterative planning. The visual support for the test persons, has to be implemented per domain by a developer.

Explanations To ask a question the user selects a subset P_N of plan properties that are not satisfied by the current plan. The selection is interpreted as the question: "Why are the plan properties in P_N not satisfied".

The answer is provided as depicted in Figure 2a. It is a list of all minimal subsets of plan properties which can not all be satisfied if P_N is added to the hard goals. In the depicted example the answer is: "If you deliver package 4 to the post office, then you cannot deliver package 1, and you can either not deliver package 2 or you cannot use the same truck for packages 2 and 3".

If a selection of hard goals is unsolvable, the user can also ask *why*. The answer is given as a smallest subset of the selected hard goals which can not be satisfied. In the sample explanation, depicted in Figure 2b, the hard goals are unsolvable because: "You can not deliver package 2 if you have to use the same truck for packages 0 and 2".



(a) solvable hard goals (b) unsolvable hard goals

Figure 2: Examples of explanations.

Plan Properties End users can define new plan properties, also during planning, by explicitly stating the corresponding LTL formula. In addition, there exists the possibility to use domain dependent templates for plan properties. These map a predefined natural language representation to the corresponding formula. An example is shown in Figure 3.

Figure 3: Interface for plan property creation.

User Studies For each iterative planning task performed by a test person in a user study, the developer has to define a fixed set of plan properties. Based on this selection a demo with precomputed explanations is generated for each task. User studies can be enriched by adding additional information for test persons and links to external questionnaires. The platform supports conducting user studies with the online recruitment platform Prolific (Palan and Schitter 2018). In many domains only a domain expert has intrinsic preferences for different plan properties or plans. For non-expert test persons, it can therefore make sense to provide artificial preferences to the test person. For that reason, we allow to add a utility to each plan property.

Conclusion

We implemented a Web-based tool for iterative planning as proposed by (Smith 2012), with explanations as suggested by Eif20. One interesting question for the future is the combination with the framework of (Krarup et al. 2019), where user questions are answered through modified example plans, which may naturally become the next plan candidate in the iterative process.

References

Baier, J. A., and McIlraith, S. A. 2006. Planning with first-order temporally extended goals using heuristic search. In *Proc. AAAI*, 788–795.

De Giacomo, G.; De Masellis, R.; and Montali, M. 2014. Reasoning on LTL on finite traces: Insensitivity to infiniteness. In *AAAI*, 1027–1033.

Edelkamp, S. 2006. On the compilation of plan constraints and preferences. In *ICAPS*, 374–377.

Eifler, R.; Cashmore, M.; Hoffmann, J.; Magazzeni, D.; and Steinmetz, M. 2020a. A new approach to plan-space explanation: Analyzing plan-property dependencies in oversubscription planning. In *AAAI*.

Eifler, R.; Steinmetz, M.; Torralba, A.; and Hoffmann, J. 2020b. Plan-space explanation via plan-property dependencies: Faster algorithms & more powerful properties. In *IJCAI*, 4091–4097.

Helmert, M. 2006. The Fast Downward planning system. *JAIR* 26:191–246.

Krarup, B.; Cashmore, M.; Magazzeni, D.; and Miller, T. 2019. Towards model-based contrastive explanations for explainable planning. In *ICAPS XAIP*.

Palan, S., and Schitter, C. 2018. Prolific.ac a subject pool for online experiments. *Journal of Behavioral and Experimental Finance* 17:22–27.

Smith, D. 2012. Planning as an iterative process. In *AAAI*, 2180–2185.