Interactive Explanation for Planning-Based Systems

Ellin Zhao
Dept. of Electrical Engineering and Computer Sciences
University of California, Berkeley
Berkeley, CA, USA
ellinzhao@berkeley.edu

Roykrong Sukkerd
Institute for Software Research
Carnegie Mellon University
Pittsburgh, PA, USA
rsukkerd@cs.cmu.edu

ABSTRACT

As Cyber-Physical Systems (CPSs) become more autonomous, it becomes harder for humans who interact with the CPSs to understand the behavior of the systems. Particularly for CPSs that must perform tasks while optimizing for multiple quality objectives and acting under uncertainty, it can be difficult for humans to understand the system behavior generated by an automated planner. This work-in-progress presents an approach at clarifying system behavior through interactive explanation by allowing end-users to ask *Why* and *Why-Not* questions about specific behaviors of the system, and providing answers in the form of contrastive explanation.

CCS CONCEPTS

• **Human-centered computing** → *User interface management systems.*

KEYWORDS

Interactive Explanation, Explainable Planning, Markov Decision Process Planning

ACM Reference Format:
Ellin Zhao and Roykrong Sukkerd. 2019. Interactive Explanation for Planning-Based Systems. In *ICCPS '19: ACM/IEEE International Conference on Cyber-Physical Systems, April 16–18, 2019, Montreal, CAN*. ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

As the use of artificial intelligence in CPSs increases and systems become more autonomous, it becomes harder for humans who interact with the systems to understand the rationale behind system behavior. Systems that must perform tasks while optimizing for multiple quality objectives and operating under uncertainty can be formulated as MDPs. Many application domains of CPSs have these problem characteristics, and these systems are solvable using automated planners. Our work concerns making the behaviors of these systems more transparent using interactive explanation.

The explanation is directed towards end-users of these autonomous systems, who do not know the technicalities of the systems or planning. The goal of the explanation is to help end-users gain awareness of how the systems make decisions in terms of the domain-specific qualities of concern. Through explanation, end-users should gain confidence that the system is making the best decisions in terms of those concerns.

We allow users to ask *Why* and *Why-Not* questions about the system's behavior and provide answers using a contrastive explanation approach in which we point out the consequence of the counterfactuals in the question [4]. Explanations describe why the system makes certain choices, in terms of the consequences or properties of those choices, and explain how the choices affect the goals and objectives of the system [7].

2 APPROACH

Our method of interactive explanation applies to MDPs with multiple cost functions, characterized by $\langle S, A, Pr, C, s_0 \rangle$ and an optimal policy, π^* [5] [1]. For our purposes, we use linear scalarization to combine the multiple cost functions into one, resulting in $C = k_1 \cdot C_1(s, a, s') + ... + k_n \cdot C_n(s, a, s')$ [2]. Specifically, we demonstrate a tool in the domain of robot navigation planning, in which a self-adaptive robot plans a route from a start state to a goal, while optimizing for various objectives such as travel time, level of intrusiveness and number of collisions.

Explanation generation

When a user expects a system to behave differently than it does in π^*, they will have questions. A user might ask "Why does the system do X?", which means that X occurs in π^*, but the user does not expect X to happen. In this case, we will show the user that not doing X is less preferred to doing X, because of the consequences on the objective values

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
ICCPS '19, April 16–18, 2019, Montreal, CAN
© 2019 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

(e.g. travel time, energy consumed, intrusiveness). Similarly, when a user asks, "Why does the system not do Y?", it is expected that Y occurs in π^*, but it does not occur. We will show the user that doing Y is less preferred to not doing Y in terms of consequences on objective values.

To generate contrastive explanations, we demonstrate what happens if the system behaves as the user expects. We do this by translating the user questions into constraints on the planning problem, resulting in a new MDP formulation that we pass into the planner to get an alternate policy, π. Using probabilistic model checking of π^* and π, we obtain the objective values for both policies and use these to show the trade-off among competing optimization objectives [6] [3]. We justify the behavior of π^* by explaining that the trade-offs made in π are less preferable and the trade-offs in π^* are better than the user initially thought.

User Interface

To capture user questions, we provide question templates for users; this limits the questions to ones that we know how to interpret and handle. We target two types of questions, *Why* and *Why-Not*, to capture user expectations for system behavior. A *Why* question is formulated as "why does the system do X?" and translates to the expectation that the system should not do X. Similarly, a *Why-Not* question, "why does the system not do Y?", reflects the expectation that the system should do Y. In the templates, the user fills in X and Y by specifying an action and optionally, the conditions for that action.

The template takes the form of "why does the robot (not) do a when P?", where a is an action and P is a predicate on the state variables. As such, X and Y target an action in any context or a conditional action. For example, a user can ask "why does the system recharge at location 5 when the energy is greater than 50?" to describe a conditional action or can simply ask "why does the system recharge?". For the former question, $a = recharge$ and $P = (location \in \{5\}) \wedge (energy > 50)$.

"Why" Questions

The process for translating questions into constraints depends on the type of question that is asked. Consider translating "why does the robot do X?", where X is specified by (a, P). To translate this into a constraint, we need to disallow a from occurring when P is satisfied. If the predicate is empty, we can remove a from the action set, resulting in a new action set, A_1. If the predicate is not empty, we add P to the existing preconditions for a. Solving the modified MDP, with either A_1 or the modified preconditions, yields an alternate policy, π, in which X does not occur. π^* and π are then contrasted in terms of their objective values.

For example, if the user asks, "why does the robot use full speed at location 13?", an explanation might be: "The robot plans to use full speed at location 13 because although it would take less travel time (by 0.5 minutes) to get to the goal, it would consume more energy (by 100 mWh). The decrease in travel time is not worth the increase in energy consumed".

"Why-Not" Questions

Handling *Why-Not* questions is still a work-in-progress because we cannot anticipate what steps will be affected by the user's question, and consequently do not know what constraints need to be made to force certain behavior from the planner. However, for certain *Why-Not* questions we have an approach for generating explanations.

When handling "why does the robot not do Y?", where $Y = (a^*, P)$, we need to force a^* to occur. One approach is to create an additional cost function, $C^*(s, a, s')$ that assigns 1 when $a = a^*$ and $P(s) = true$, and create the constraint that $E[C^*] \geq k$. Given a question about an unconditional action, such as "why does the robot not recharge?", we create the constraint that $E[C^*] \geq 1$. However, for a question concerning a conditional action, such as "why does the robot not use full speed at location 5?", k is unknown. Planning with C^* yields π, which we contrast with π^* to highlight the differences between the trade-offs in terms of objective values.

REFERENCES

[1] Dimitri P. Bertsekas and John N. Tsitsiklis. 1991. An Analysis of Stochastic Shortest Path Problems. *Math. Oper. Res.* 16, 3 (Aug. 1991), 580–595. https://doi.org/10.1287/moor.16.3.580

[2] Massimiliano Caramia and Paolo Della-Olmo. 2008. *Multi-objective Management in Freight Logistics: Increasing Capacity, Service Level and Safety with Optimization Algorithms.* https://doi.org/10.1007/978-1-84800-382-8

[3] Vojtěch Forejt, Marta Kwiatkowska, Gethin Norman, and David Parker. 2011. Automated verification techniques for probabilistic systems. In *International School on Formal Methods for the Design of Computer, Communication and Software Systems*. Springer, 53–113.

[4] Tim Miller. 2017. Explanation in Artificial Intelligence: Insights from the Social Sciences. *CoRR* abs/1706.07269 (2017). arXiv:1706.07269 http://arxiv.org/abs/1706.07269

[5] Martin L. Puterman. 1994. *Markov Decision Processes: Discrete Stochastic Dynamic Programming* (1st ed.). John Wiley & Sons, Inc., New York, NY, USA.

[6] Jonathan Sprauel, Florent Teichteil-Königsbuch, and Andrey Kolobov. 2014. Saturated Path-constrained MDP: Planning Under Uncertainty and Deterministic Model-checking Constraints. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence (AAAI'14)*. AAAI Press, 2367–2373. http://dl.acm.org/citation.cfm?id=2892753.2892880

[7] Roykrong Sukkerd, Reid Simmons, and David Garlan. 2018. Towards Explainable Multi-objective Probabilistic Planning. In *Proceedings of the 4th International Workshop on Software Engineering for Smart Cyber-Physical Systems (SEsCPS '18)*. ACM, New York, NY, USA, 19–25. https://doi.org/10.1145/3196478.3196488