Explanations for Human-on-the-loop: A Probabilistic Model Checking Approach

Nianyu Li
EECS, Peking University
li_nianyu@pku.edu.cn

Sridhar Adepu
Singapore University of Technology and Design
adepu_sridhar@mymail.sutd.edu.sg

Eunsuk Kang
Carnegie Mellon University
eunsukk@andrew.cmu.edu

David Garlan
Carnegie Mellon University
garlan@cs.cmu.edu

ABSTRACT

Many self-adaptive systems benefit from human involvement and oversight, where a human operator can provide expertise not available to the system and can detect problems that the system is unaware of. One way of achieving this is by placing the human operator *on the loop* – i.e., providing supervisory oversight and intervening in the case of questionable adaptation decisions. To make such interaction effective, explanation is sometimes helpful to allow the human to understand why the system is making certain decisions and calibrate confidence from the human perspective. However, explanations come with costs in terms of delayed actions and the possibility that a human may make a bad judgement. Hence, it is not always obvious whether explanations will improve overall utility and, if so, what kinds of explanation to provide to the operator. In this work, we define a formal framework for reasoning about explanations of adaptive system behaviors and the conditions under which they are warranted. Specifically, we characterize explanations in terms of explanation *content*, *effect*, and *cost*. We then present a dynamic adaptation approach that leverages a probabilistic reasoning technique to determine when the explanation should be used in order to improve overall system utility.

1 INTRODUCTION

Self-adaptive systems are designed to be capable of dynamically modifying their structure and behavior in response to changes in the environment [1, 2]. Although automation is one desirable characteristic of self-adaptation, certain adaptive systems benefit from human involvement and oversight. For example, a human operator may be able to detect events that are not directly observable by the system, or possess knowledge that is external to those already built into the system. In these cases, the system may be able to respond more effectively to potential anomalies and achieve greater utility when its adaptation decisions are guided by a human input [3–5].

One way to achieve this synergy for the system is by placing an operator *in-the-loop* between the self-adaptation framework and the environment as a deciding authority. A variant of *human-in-the-loop* is *human-on-the-loop*, in which the operator plays a less central role; in this approach, the operator periodically monitors the interaction between the machine and the environment, and intervenes only when deemed necessary (e.g., to avert potentially anomalous behavior) [6]. In this paper, we focus on self-adaptive systems that employ a human-on-the-loop approach. Note that in this context the "system" consists of a machine, human operator, and the environment.

Figure 1 depicts a closed-loop adaptive system in which a human operator is engaged *on-the-loop*. The dynamic behaviors exhibited by the *environment* (which may be an occurrence of certain events or changes in the environmental state) are periodically monitored by a set of *sensors*. Given these sensor readings, the *controller* will perform an analysis of available actions and their potential outcome on the system utility, and plan corresponding adaptation decisions to be enacted by the *actuators*. The role of the human operator on the loop is to observe the adaptation decision made by the controller and determine whether this decision is *appropriate* or potentially *erroneous* (i.e., likely to degrade the overall utility or lead the system into an unsafe state). In the latter case, the operator may *interfere* in this control loop by overriding the commands sent to the actuators or, in the worst case, pausing or shutting down the system.

An important factor behind the operator's decision to interfere or not is their *level of confidence* in the machine; that is, the degree of one's belief that the adaptation decision made by the machine, if followed, will yield a desirable outcome on the system utility. The operator is more likely to interfere if they have lower confidence in the capability of the machine to produce correct decisions; conversely, the greater the level of confidence, the more likely the operator is to allow the machine to carry out its adaptation decisions autonomously.

Prior works have investigated the role of *explanation* as a mechanism to improve an operator's trust in the behavior of an autonomous system [7, 8]. Our conjecture, which we investigate in this paper, is that *in the context of self-adaptive systems, appropriate explanations can be used to aid an operator in dynamically calibrating their level of confidence in adaptation decisions made by a machine.* When an explanation is provided along with a control decision, under the right conditions, the operator may gain a higher level of

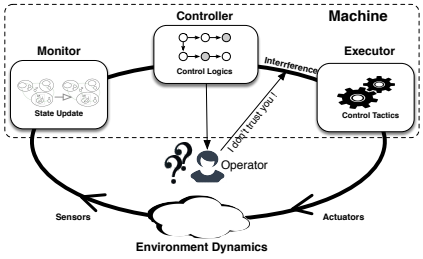

Figure 1: A human-on-the-loop self-adaptive system.

confidence that the machine is following the right course of adaptation decision, and thereby be more likely to allow the machine to continue with its recommended course of decision.

Though explanations might yield positive effects on system outcomes, they also incur costs to system operation. In particular, the operator needs time and mental effort to comprehend this information. This, in turn, may delay actions or in an extreme case, cause overload of information to the operator. Hence, given the space of potential costs and effects of an explanation, it may not always be apparent *when* it is beneficial to provide an explanation (e.g., whether its potential benefit outweighs the cost), or *what* type of information must be provided as part of the explanation. Therefore, a human-on-the-loop system that uses explanations to steer human decisions must consider the trade-offs between the costs and effects of alternative explanations (including not giving one at all) and select one that is *optimal* in the given environmental context.

In this paper, we provide a theoretical framework to specify and reason about the effects of explanations on the human-on-the-loop in self-adaptive systems. In particular, our framework defines an explanation in terms of three major components: (1) explanation *content*, describing the types of information provided as part of an explanation; (2) *effect*, describing how an explanation can impact the operator's level of confidence in supervisory control decisions; and (3) *cost*, specifying the cost involved in comprehending an explanation. Using this, we provide an approach for synthesizing an *explanation strategy* for human-on-the-loop systems based on probabilistic model checking [9]. An explanation strategy describes what explanation (if any) should be provided at a particular point in the execution of a system. The key idea here is to use non-determinism to under-specify the components (i.e., content, effect, and cost) of an explanation candidate, and have the model checker resolve the non-deterministic choices and synthesize an explanation strategy so that the expected system utility is maximized.

Our main contributions are:

- A formal framework for designing human-on-the-loop self-adaptive systems where an explanation can be used to aid the human operator to improve the utility of the overall system;
- The use of probabilistic model checking to perform the synthesis of optimal explanations,

The rest of the paper is structured as follows. Section 2 provides a formal definition of explanations, and Section 3 provides a technique for explanation selection using probabilistic model checking while Section 4 presents the analysis results and discussion. Section 5 discusses related work and Section 6 concludes the paper.

2 EXPLANATIONS: FORMAL FRAMEWORK

In our approach, an explanation is defined as a triple $Exp = \langle\ content, effect, cost\ \rangle$. In the following, we introduce a motivating example and describe how the three components of an explanation can be formally modeled. We also motivate why it is important to consider the trade-offs between the effect and cost of an explanation.

Running example. Consider a self-driving car that is capable of combining a variety of sensors (such as radar, sonar, camera, etc.) to perceive pedestrians and other objects in the environment and move safely with little or no human input. A software controller interprets sensory information and identifies appropriate navigation paths and operations. The driver in this system acts as a human-on-the-loop and may intervene to reduce risks or prevent accidents in dangerous situations.

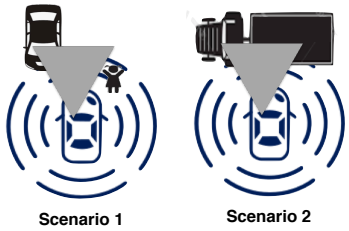

Figure 2: Self-driving vehicle scenarios.

Consider two possible scenarios involving a self-driving car, as shown in Figure 2:

- Scenario 1: Another car is approaching from the opposite direction, and the driver sitting in the ego vehicle decides that it would be safer to move the car to the right to avoid a potential collision. However, the self-driving car makes an adaptation decision to *stop* in this situation because it has detected a child on the right front. For the driver, although he observes the oncoming car, the child is out of sight (grey triangle).
- Scenario 2: A large truck is turning right in front of the ego vehicle. However, the machine makes a decision to go ahead at full speed because it identifies the truck as a highway overpass. Though humans can easily distinguish a truck and an overpass and derive at the safer decision of slowing down or stopping, the machine is not able to do so due to its limited perception capabilities. This scenario is similar to a recent accident involving the autopilot software in a Tesla vehicle [10], where the system failed to recognize the truck in time (which would have been seen by a human driver).

In the remainder of this section, we will revisit these two scenarios in the context of our explanation framework.

2.1 Explanation Content

The *content* of an explanation corresponds to the type of information that the explanation provides to the human operator. In our approach, an explanation is intended to justify why the system has made a decision to behave in a particular way (e.g., perform a particular action or transition to a different state from the current state). To capture this intent, we encode two types of information in the explanation content: (1) the current state, and (2) the transition of the machine that are relevant to the decision being made by the machine. Let us motivate the design of explanation content using the following example.

A well-known class of problems, known as *automation surprises* [11, 12], occur in human involvement when the machine behaves differently than its operator expects. Two reasons are identified as accounting for these problems. One is that the operator may know only a subset of the information that the machine has (e.g., the presence of child in Scenario 1). The other is due to additional information from the environment that is hidden from the machine but known to the operator (for example, the presence of

a truck instead of an overpass in Scenario 2). Both the machine and the operator analyze and plan adaptation decisions for a given situation using their information and reasoning process. But since they have asymmetric information about the environment, there might be differences between their adaptation decisions.

To formally define what constitutes an explanation, we first model the information that the machine and the operator possess. *Machine information* is defined as a tuple $MI = \langle S_M, T_M \rangle$, where S_M represents the set of states while T_M is the transition function. For example, S_M may encode the status of sensors and actuators inside a self-driving car, and T_M may describe how the action of the controller modifies the states of the actuators. Similar to the machine information, *environment information* is defined as a tuple $ENVI = \langle S_E, T_E \rangle$, representing the state of the environment and how this state will change based on the actions of the agents in the environment, respectively.

Then, the information possessed by the human operator is defined as a tuple $HI = \langle S_H, T_H \rangle$, where the state in the operator's mind is the union of *partial* environment state and *partial* machine state, i.e., $S_H = \rho_S(S_M) \cup \rho_S(S_E)$. For example, in Scenario 1, the driver can observe the oncoming car, which is part of machine information since the oncoming car can be detected by the sensors. However, the driver may additionally be able to access part of the environment state (which cannot be observed by the machine), such as the incoming truck. The transition set in the operator's mind, i.e., $T_H = \rho_T(T_M) \cup \rho_T(T_E)$ is also the union of partial environment transition and partial machine transition.

The explanation provided by the machine to the operator contains partial information about the machine state ($\rho_S(S_M)$) and transition ($\rho_T(T_M)$), describing why the machine has decided to perform a particular action. For instance, sensorLeftFront = car& sensorRightFront = child represents the state in which the ego vehicle has detected another car in its front left and a child in the front right. In addition, sensorLeftFront $\neq$ null& sensorRightFront $\neq$ null $\implies$ Stop is a representation of a machine transition, which states that the ego vehicle will stop when it has detected objects in both its front right and left.

2.2 Explanation Effect

In our approach, we model the *effect* of an explanation as calibrating the operator's belief that the system is behaving in a desirable or undesirable way.

There are two cases that we consider. First, an explanation can potentially enable the operator to gain more confidence that the system is making the *right* adaptation decision. Here, the *right* decision is one that would lead the system into a state with a desirable outcome (e.g., a high utility value). With additional information supplemented by an explanation, the operator is more likely to accept the machine decision without interfering in it, especially when the operator has limited observations about the system.

On the other hand, the machine may sometimes make an adaptation decision that is undesirable, in that it leads the system into a state with a low utility. This may occur, for example, due to design faults or security attacks that cause the machine to make a suboptimal decision. In these cases, additional information in an explanation may inform the operator of this undesirable behavior

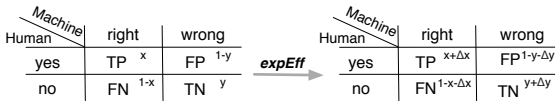

Figure 3: Effect of an explanation as influencing the probabilities that the operator agrees or disagrees with the decision by the machine.

and encourage them to intervene; we capture this as having the effect of *decreasing* the operator's confidence in the machine.

The explanation effect is formally defined as function *expEff*: $\langle Pr, Pr \rangle \rightarrow \langle Pr, Pr \rangle$, mapping a pair of probabilities (i.e., probabilities of true-positive x and true-negative y) to another pair of probabilities. False-positive, denoting the likelihood that the operator approves a wrong adaptation decision by the machine, can be determined by the true-negative (i.e., $1-y$). Similarly, false-negative can be determined by the true-positive (i.e., $1-x$) and describes the situation of unnecessary human interference following a correct adaptation decision from the machine. These are also known as type I and II errors in statistical hypothesis.

Initially, the operator is assigned some true-positive and true-negative probabilities based on their existing view of the system. For example, the driver may equally oscillate between their own adaptation decision and machine adaptation decision if they cannot judge which is more reliable, yielding the true-positive and false-negative values of 0.5 each in Scenario 1 and the true-negative and false-negative of 0.5 in Scenario 2.

The effect of an explanation on the operator is modeled as reducing the probabilities of the operator making false-negative and false-positive errors (i.e., the probabilities of true-positive and true-negative, respectively, will be increased). In Scenario 1, given the information about the presence of the child in front of the vehicle, the driver is more likely to believe that stopping is a better action than turning right, thus decreasing the probability of operator interference. In contrast, the driver may be encouraged to intervene and apply the brake in Scenario 2 if an explanation reveals that the vehicle (mistakenly) assumes the presence of an overpass instead of the truck. Figure 3 summarizes explanation effects as causing changes in false-negative or false-positive probabilities by Δx and Δy, respectively.

2.3 Explanation Cost

Explanation does not come for free; it also incurs costs. In particular, the operator needs time and energy to comprehend this information. In a self-driving system, prompt response from the driver is vital in an emergency, and an explanation might delay the reaction time and distract the driver due to the overload of information. Given this, it is not immediately apparent when to explain; the system needs to consider the trade-offs between the costs and benefits that a particular type of explanation brings. In this work, we simplified the cost as an abstract value that could represent, for example, the human annoyance due to the overload of information, or delays due to the time spent on explanation comprehension. More discussion on explanation cost can be found in subsection 4.1.

Hence, given a pool of explanation candidates, by balancing the effect and cost that the explanation brings for the system, the

explanation with the highest utility will be selected to the operator, or no explanation will be provided if the cost outweighs its benefits.

3 EXPLANATION SELECTION

In this section, we describe an approach to the *explanation selection* problem; i.e., deciding what information to include as part of an explanation to the operator. In the running example, intuitively, a good explanation for Scenario 2 might only point out the mis-identification of an overpass, assuming the driver is experienced. However, for a novice driver, an explanation that includes more details might be more useful, although a more verbose explanation may incur additional operator cost in comprehending the information. Thus, selecting an explanation must take into account potential trade-offs between its potential benefit and cost.

The key idea of solving the explanation selection problem is to leave the explanation under-specified in the model through non-deterministic behavior [13, 14]. In this work, we use the PRISM tool [15], which supports reasoning about well-known behavioral specifications, such as Markov Decision Processes (MDPs) [16] and probabilistic timed automata (PTAs) [17], along with support for non-determinism. In particular, PRISM is used to synthesize a strategy that maximizes the expected utility.

In our approach the human operator and machine are specified as processes that are composed in the MDP model. Processes are abstracted and simplified, containing only the variables that are necessary to compute the value of the utility and to keep track of how the machine and human changes when the explanation is used. In this model, we only focus primarily on whether an explanation is worthwhile to be provided, as the extension to multiple explanations is straightforward.

3.1 Machine Model

The machine is modeled over its evolution of one decision-making. A part of machine behavior is shown in Figure 4. Four steps will be considered in one horizon. First, the machine makes an adaptation decision, which is probabilistic. That decision might be a correct or an incorrect one. (For example, it would be optimal to stop the car in Scenario 1, and incorrect to go ahead in Scenario 2. If it is the right decision, as illustrated in the upper part of the figure (the other option is not shown for simplicity), the machine can provide an explanation to the operator or choose not to, which is

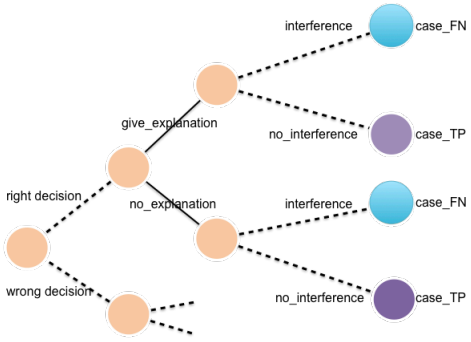

Figure 4: Fragment of machine behaviors.

the explanation strategy the machine can choose to resolve the non-determinism. After that, the final action is executed, such as stopping the car if without human interference. The probability of interference is based on the human operator's capability for making correct oversight decisions (i.e. the probability of true-positive and false-negative). Finally, the successor state after the final action will be assigned an utility value over the look-ahead horizon. Usually, the optimal states, such as two states with annotation "case_TP" will be assigned with higher utility where the machine makes a correct adaptation decision and performs that decision without human interference. In contrast, false-negative states (with the annotation "case_FN") will typically accrue less utility as human erroneously rejects the right system decision.

```
1  module machine
2    macStep:[0..4] init 0;
3    macDecision:[0..2] init 0;
4    macCase:[0..4] init 0;
5
6    []macStep = 0 & macDecision = 0 ->
7          ProMac:    (macStep'=1)&(macDecision'=good)
8     + (1-ProMac):  (macStep'=1)&(macDecision'=bad);
9
10   [give_explanation]    macStep=1 -> (macStep'=2);
11   [ no_explanation ]    macStep=1 -> (macStep'=2);
12
13   []macStep=2 & macDecision=good  ->
14      TP:  (macStep'=3)&(macCase'=case_TP)
15    + FN:  (macStep'=3)&(macCase'=case_FN);
16   []macStep=2 & macDecision=bad   ->
17      TN:  (macStep'=3)&(macCase'=case_TN)
18    + FP:  (macStep'=3)&(macCase'=case_FP);
19
20   [perform] macStep=3 -> (macStep'=4);
21 endmodule
```

Listing 1: Machine model

Generating the PRISM code representing the MDP for the machine behavior is straightforward. Listing 1 shows its specification in PRISM. Three variables represent the state of the machine. The first one is "macStep" encoding the different four steps mentioned previously. The transition out of each step can be encoded directly as commands in PRISM[1]. Variable "macDecision" denotes the adaptation decision that the machine makes. The first command (line 6-8) will advance the step of adaptation decision making, leading to a probabilistic behavior. With the probability of "ProMac" which is defined and initialized as a global variable, the machine makes a correct decision. The action "give_explanation" and "no_explanation" in lines 10-11 are used to synchronize the transitions between the machine and the human. These two commands overlap with the same guard introducing non-determinism in explanation selection. "macCase" records the state the machine will enter after the explanation selection. With the probability of "TP" and "FN", the machine will enter an optimal or suboptimal state with interference when the machine decision is correct in lines 13-15. Meanwhile, the probability will be "TN" and "FP" when the machine decision is wrong.

[1]MDPs are encoded in PRISM with commands like: [action]guard$\rightarrow p_1 : u_1 + \ldots + p_n : u_n$ where guard is a predicate over the model variables. Each update u_i describes a transition that the process can make (by executing action) if the guard is true. An update is specified by giving the new values of the variables and has an assigned probability $p_i \in [0, 1]$. Multiple commands with overlapping guards (and probably, including a single update of unspecified probability) introduce local non-determinism.

Finally, the machine will perform the last step, representing the expected utility the machine will obtain in this decision making, which will be described and calculated in the reward subsection 3.3 below.

3.2 Human Model

The specification of the human module is shown in Listing 2. Lines 7-8 describe two variables "HuYes_MacGood" and "HuNo_MacBad" that capture the human's confidence in machine decisions. They range from 0 to 100, as variables in the processes in PRISM cannot be specified as a decimal. They are initialized with some constants that represent the initial confidence a human has based on his existed information at the time the machine adaptation decision is invoked; that is, at the beginning of the decision horizon. And the probabilities of TP, FP, TN, and FN can be acquired by normalizing these two variables as shown as formula in lines 1-4. For example, the initial four probabilities for a driver novice could be all 50% with random guessing. Another Boolean variable "exp_received" denotes the status of receiving an explanation or not and is initialized with a false value.

```
1  formula TP = HuYes_MacGood/100;
2  formula FN = 1-(HuYes_MacGood/100);
3  formula TN = HuNo_MacBad/100;
4  formula FP = 1-(HuNo_MacBad/100);
5
6  module human
7    HuYes_MacGood:[0..100] init initial_HuYes_MacGood;
8    HuNo_MacBad:[0..100] init initial_HuNo_MacBad;
9    exp_received:bool init false;
10
11   [give_explanation] true ->
12         (HuYes_MacGood'=HuYes_MacGood+Delta_X)
13        &(HuNo_MacBad'=HuNo_MacBad+Delta_Y)
14        &(exp_received'=true);
15   [ no_explanation ] true ->
16         (HuYes_MacGood'=HuYes_MacGood)
17        &(HuNo_MacBad'=HuNo_MacBad);
18 endmodule
19 \vspace{-0.21cm}
```

Listing 2: Human model

Lines 11-14 describe a command that captures how the human confidence can be calibrated and updated with the action "give_explanation" synchronized with machine module, i.e., adding the effect of an explanation "Delta_X" and "Delta_Y" to two variables representing human confidence. Correspondingly, the value of the formula in lines 1-4 will be updated to reflect these changes, which will affect the probabilistic behavior of the machine (line 13-18 in Machine module). The variable "exp_received" will also be set to true. On the contrary, lines 15-17 depict the command where no explanation is received from the machine, and here all the variables will remain the same. So does the confidence in machine decision. Here we assume one decision making is a short period where human's confidence in the machine will not degrade even if machine decision making is different and opaque to the human. However, when the time passes without explanation, the complex analysis, and planning of the machine will probably make human operators lose trust, i.e., reducing the probability of true-positive and true-negative.

Here only one explanation with its effect is shown both in human and machine modules. As described in section 2, a pool of explanation candidates with various effect, i.e., different "Delta_X" and "Delta_Y" values can be specified as commands for possible explanation candidates in the explanation selection problem.

3.3 Explanation Selection

Explanation selection is carried out after the machine model has made an adaptation decision. The input to the probabilistic model checker is the composition of above two modules. Then, we need to specify the property of the model that must hold under the generated strategy. In this case, the desired property is to maximize overall system utility. In PRISM, this property is expressed as

$$R^{sysUtility}_{max=?}[F^c end]$$

where "sysUtility" is the reward structure specified in Listing 3, and *end* is a predicate that indicates the end of the execution in a decision horizon. Such a reward construct in lines 9-12 assigns the value, which is the sum of machine performance and human cost to the transition labeled with action "execute". Machine performance is decided by the state in which the machine will enter in lines 1-5. For example, the utility of "Utility_Case_TP" will be assigned if the machine enters a case "caseTP" where it makes the right decision without human interference. These utility values are specific to different situations – such as in self-driving system, the mistakes of turning right (i.e., false negative) in Scenario 1 or going ahead with the full speed (i.e., false positive) in Scenario 2 is pretty high, and the differences between utility of "case_TP" and "case_FN" and between utility of "case_TN" and "case_FP" will be significant since these are all critical decisions. However, the differences might be minor in non-critical systems. The human cost is an abstract value based on whether the human receives an explanation and translated with a positive shift because PRISM does not allow negative rewards.

```
1  formula machine_performance =
2      (macCase=caseTP? Utility_Case_TP:0)
3    + (macCase=caseFP? Utility_Case_FP:0)
4    + (macCase=caseFN? Utility_Case_FN:0)
5    + (macCase=caseTN? Utility_Case_TN:0);
6  formula human_cost =
7      (exp_received=true? 0: Cost);
8
9  rewards "sysUtility"
10   [execute] true:
11       machine_performance+human_cost;
12 endrewards
```

Listing 3: Reward structure

4 ANALYSIS

To further investigate under what conditions an explanation should be provided, we statically analyze the MDP model described above with a region of the state space, which is projected over three dimensions that correspond to the 1) cost of mistakes; 2) explanation effect; 3) cost of explanation (with values in the range [0,1], [0,100%], [0,1] respectively). To be more specific, cost of mistakes denotes subtracting the high utility value with correct cases ("case_TP" and "case_TN") from the low utility with incorrect cases ("case_FN" and "case_FP") and with normalization; explanation effect averages the value of "Delta_X" and "Delta_Y"; cost of explanation is the

single abstracted value representing the cost explanation brings. We plot two three-dimensional graphs with R [18], as shown in Figure 5. These two cubes encompass all the condition points where it is beneficial to explain, while the remaining part of the three-dimensional state space represents the unnecessary conditions.

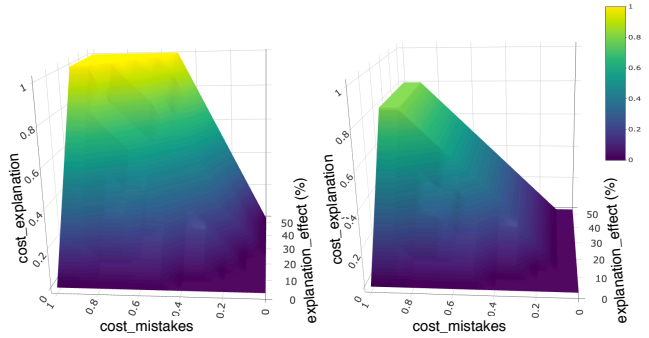

Figure 5: (a) Explanation conditions for a novice (left); (b) explanation conditions for an expert (right).

We can conclude the following from the graphs: 1) when the cost of mistakes is close to zero, there is more space where the explanation will not be provided than those in a high cost of mistakes; 2) when the explanation effect is not obvious (i.e., near zero), which means the human cannot gain much useful information from the explanation to increase the probabilities of true-positive and true-negative, explanation is not necessary for these conditions; 3) when the cost of explanation increases, the chance of explaining will decrease with the gradually decreasing horizontal cross-sectional area of the cube as it is less likely the benefits could outweigh its cost. These conclusions are all consistent with our intuitions.

In addition, graph (a) depicts the conditions for a novice, while (b) is for an expert, who has more information than the machine does and is initialized with higher initial probability of true-positive and true-negative. The differences between two graphs show that the cube volume for a novice is greater than that for an expert as the cube height is around 1 while it reaches 0.8 at most for the expert. Moreover, the area of each horizontal cross-section for an expert is much smaller than each for a novice. This matches our expectation that a novice operator may need to be provided with an explanation more frequently than an expert with more knowledge about the system operation. The interested reader is referred to [19] for more details on an applied case study.

4.1 Limitations and Discussions

Our framework relies on an assumption that the probabilities behind confidence levels as well as explanation effect can be accurately measured. In our group, an ongoing research project with an empirical user case study is exploring how such probabilities may be obtained through experimentation [20]. In addition, the cost of an explanation may not be easy to measure for different operators. One way to overcome this challenge is by assigning the cost based on the complexity of information in the explanation content, e.g., the amount of the information. A qualitative estimate of time for the operator to understand the explanation could be another approach [21].

Another current limitation of our study is that to simplify the explanation selection problem, the overall system utility is computed as a single objective by merging multiple attributes. However, it may not always be appropriate to compare and aggregate certain types of attributes, such as human cost and system performance. In such cases, formulating explanation selection as a multi-objective optimization problem with Pareto-optimal solutions as alternative candidate explanations may be a more suitable approach [22]. In addition, our initial investigation suggests a number of further research questions to be explored, such as how to find the optimal information as an explanation candidate, to maximize the overall utility, and how to take the time delay between decision making and human interference into consideration.

5 RELATED WORK

Research on explanation has surged recently especially in the field of artificial intelligence, with the notion of eXplainable Artificial Intelligence (XAI) [23]. However, over three decades ago, explanation has been investigated with prosperity in expert systems [24–26]. There also exists an extensive literature on explainable agents and robots, with applications in factory environments [27], military missions [28], human players [29], training [30], e-health [31] and recommendation systems [32]. However, in the context of self-adaptive systems, explanations have been relatively little studied. This direction is necessary to support any human-system interaction and confirmed by the ratification of General Data Protection Regulation (GDPR) law which underlines the right to explanations [33].

Several existing works investigate methods to explain why a system produces particular behaviors. The work in [34] describes how the state of the machine is captured in a human's mind. When the behavior of an agent is not explained, the human's understanding may not be consistent with the real system state, which could lead to dangerous situations. Also, lack of a mental model for the human estimating the actions of robots may lead to safety risks [35, 36]. Lin et al. propose an automatic explanation technique for different types of explanations and decision models [21]. Chakraborti et al. introduces the *model reconciliation* problem as aiming to make minimal changes to the human's model to bring it closer to the robot's model [37]. Elizalde et al. propose an approach that identifies factors that are most influential to the decision making with MDP [38]. Khan et al. present an approach for explaining an optimal action in policy by counting the frequency of reaching a goal by taking the action [39]. Sukkerd et al emphasized contrastive justification based on quality attributes and presented a method for generating an argument of how a policy is preferred to other rational alternatives [40]. However, most of their work only focuses on the explanation generation and does not capture the explanation effect nor the cost of an explanation.

6 CONCLUSIONS

Within the context of self-adaptive systems, some human involvement as an operator is crucial. The machine may behave differently than the human operator expects, resulting in the problem known

as automation surprises. In order to calibrate the operator's confidence in machine adaptation decisions, we present a theoretical framework for explanations and a technique for synthesizing explanations based on probabilistic model checking. In our future research, we plan to further elaborate on the theoretical aspects of our framework (e.g., the cost and effect of an explanation) and demonstrate its applicability by applying it to practical scenarios.

7 ACKNOWLEDGMENTS

We would like to thank our anonymous reviewers for their valuable comments. This work is partly supported by China Scholarship Council (201906010177), award N00014172899 from the Office of Naval Research (ONR) and award H9823018D0008 from the NSA. Any views, opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the ONR or NSA.

REFERENCES

[1] B. H. C. Cheng and et al., "Software engineering for self-adaptive systems: A research roadmap," in *Software Engineering for Self-Adaptive Systems [outcome of a Dagstuhl Seminar]*, 2009, pp. 1–26.

[2] R. de Lemos and et al., "Software engineering for self-adaptive systems: A second research roadmap," in *Software Engineering for Self-Adaptive Systems II - International Seminar, Dagstuhl Castle, Germany, October 24-29, 2010 Revised Selected and Invited Papers*, 2010, pp. 1–32.

[3] R. Sukkerd, D. Garlan, and R. G. Simmons, "Task planning of cyber-human systems," in *Software Engineering and Formal Methods - 13th International Conference, SEFM 2015, York, UK, September 7-11, 2015. Proceedings*, 2015, pp. 293–309.

[4] J. Cámara, G. A. Moreno, and D. Garlan, "Reasoning about human participation in self-adaptive systems," in *10th IEEE/ACM International Symposium on Software Engineering for Adaptive and Self-Managing Systems, SEAMS, Florence, Italy, May 18-19, 2015*, 2015, pp. 146–156.

[5] E. Lloyd, S. Huang, and E. Tognoli, "Improving human-in-the-loop adaptive systems using brain-computer interaction," in *12th IEEE/ACM International Symposium on Software Engineering for Adaptive and Self-Managing Systems, SEAMS@ICSE 2017, Buenos Aires, Argentina, May 22-23, 2017*, 2017, pp. 163–174.

[6] J. E. Fischer, C. Greenhalgh, W. Jiang, S. D. Ramchurn, F. Wu, and T. Rodden, "In-the-loop or on-the-loop? interactional arrangements to support team coordination with a planning agent," *Concurrency and Computation: Practice and Experience*, pp. 1–16, 2017.

[7] O. Biran and C. Cotton, "Explanation and justification in machine learning: A survey," in *IJCAI-17 workshop on explainable AI (XAI)*, vol. 8, 2017, p. 1.

[8] T. Nomura and K. Kawakami, "Relationships between robot's self-disclosures and human's anxiety toward robots," in *Proceedings of the 2011 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology-Volume 03*. IEEE Computer Society, 2011, pp. 66–69.

[9] M. Kwiatkowska, G. Norman, and D. Parker, *Probabilistic Model Checking: Advances and Applications*. Cham: Springer International Publishing, 2018, pp. 73–121.

[10] "Tesla's trouble with semi trucks & another shakeup of the autopilot team — is there a connection?" https://cleantechnica.com/2019/05/21/teslas-trouble-with-trucks-and-another-shakeup-of-the-autopilot-team-is-there-a-connection/, accessed: 2019-05-21.

[11] S. Combéfis, D. Giannakopoulou, C. Pecheur, and M. Feary, "Learning system abstractions for human operators," in *MALETS Proceedings of the International Workshop on Machine Learning Technologies in Software Engineering*, 2011, pp. 3–10.

[12] E. Palmer, "Oops, it didn't arm. - a case study of two automation surprises." in *Proceedings of the 8th International Symposium on Aviation Psychology*, 1996, pp. 227–232.

[13] G. A. Moreno, J. Cámara, D. Garlan, and B. R. Schmerl, "Proactive self-adaptation under uncertainty: a probabilistic model checking approach," in *Proceedings of the 2015 10th Joint Meeting on Foundations of Software Engineering, ESEC/FSE 2015, Bergamo, Italy, August 30 - September 4, 2015*, 2015, pp. 1–12.

[14] A. Bianco and L. de Alfaro, "Model checking of probabilistic and nondeterministic systems," in *Foundations of Software Technology and Theoretical Computer Science*, P. S. Thiagarajan, Ed. Springer Berlin Heidelberg, 1995.

[15] M. Z. Kwiatkowska, G. Norman, and D. Parker, "PRISM 4.0: Verification of probabilistic real-time systems," in *Computer Aided Verification - 23rd International Conference, CAV ,July 14-20, 2011. Proceedings*, 2011, pp. 585–591.

[16] M. L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*, ser. Wiley Series in Probability and Statistics. Wiley, 1994.

[17] G. Norman, D. Parker, and J. Sproston, "Model checking for probabilistic timed automata," *Formal Methods in System Design*, vol. 43, no. 2, pp. 164–190, 2013.

[18] K. Soetaert, "plot3d : Tools for plotting 3-d and 2-d data." https://cran.r-project.org/web/packages/plot3D/vignettes/plot3D.pdf, 2018.

[19] N. Li, A. Sridhar, E. Kang, and D. Garlan, "Analysis and synthesis of explanations for a secure industrial control system using probabilistic model checking," Institute for Software Research, Carnegie Mellon University, Tech. Rep. CMU-ISR-20-102, June 2020.

[20] R. Sukkerd, "Improving transparency and understandability of multi- objective probabilistic planning," *Thesis Proposal – School of Computer Science Institute for Software Research Software Engineering, Carnegie Mellon University*, pp. 1–41, 2018.

[21] B. Y. Lim, A. K. Dey, and D. Avrahami, "*Why and why not* explanations improve the intelligibility of context-aware intelligent systems," in *Proceedings of the 27th International Conference on Human Factors in Computing Systems, CHI 2009, Boston, MA, USA, April 4-9, 2009*, 2009, pp. 2119–2128.

[22] S. Mahdavi-Hezavehi, V. H. S. Durelli, D. Weyns, and P. Avgeriou, "A systematic literature review on methods that handle multiple quality attributes in architecture-based self-adaptive systems," *Information & Software Technology*, vol. 90, pp. 1–26, 2017. [Online]. Available: https://doi.org/10.1016/j.infsof.2017.03.013

[23] T. Miller, "Explanation in artificial intelligence: Insights from the social sciences," *Artif. Intell.*, vol. 267, pp. 1–38, 2019.

[24] B. Chandrasekaran, M. C. Tanner, and J. R. Josephson, "Explaining control strategies in problem solving," *IEEE Expert*, vol. 4, no. 1, pp. 9–24, 1989.

[25] T. Fennel and J. D. Johannes, "An architecture for rule based system explanation," 1990.

[26] C. L. Paris, "Generation and explanation: Building an explanation facility for the explainable expert systems framework," in *Natural language generation in artificial intelligence and computational linguistics*. Springer, 1991, pp. 49–82.

[27] B. Hayes and J. A. Shah, "Improving robot controller transparency through autonomous policy explanation," in *2017 12th ACM/IEEE International Conference on Human-Robot Interaction (HRI*. IEEE, 2017, pp. 303–312.

[28] R. W. Wohleber, K. Stowers, J. Y. Chen, and M. Barnes, "Effects of agent transparency and communication framing on human-agent teaming," in *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE, 2017, pp. 3427–3432.

[29] M. Molineaux, D. Dannenhauer, and D. W. Aha, "Towards explainable npcs: a relational exploration learning agent," in *Workshops at the Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[30] M. Harbers, K. Van Den Bosch, and J.-J. Meyer, "A methodology for developing self-explaining agents for virtual training," in *International Workshop on Languages, Methodologies and Development Tools for Multi-Agent Systems*. Springer, 2009, pp. 168–182.

[31] F. Kaptein, J. Broekens, K. Hindriks, and M. Neerincx, "The role of emotion in self-explanations by cognitive agents," in *2017 Seventh International Conference on Affective Computing and Intelligent Interaction Workshops and Demos (ACIIW)*. IEEE, 2017, pp. 88–93.

[32] T. Kulesza, S. Stumpf, M. Burnett, S. Yang, I. Kwan, and W.-K. Wong, "Too much, too little, or just right? ways explanations impact end users' mental models," in *2013 IEEE Symposium on Visual Languages and Human Centric Computing*. IEEE, 2013, pp. 3–10.

[33] P. Carey., *Data protection: a practical guide to UK and EU law*. Oxford University Press, Inc., 2018.

[34] T. Hellström and S. Bensch, "Understandable robots-what, why, and how," *Paladyn, Journal of Behavioral Robotics*, vol. 9, no. 1, pp. 110–123, 2018.

[35] C. L. Bethel, "Robots without faces: non-verbal social human-robot interaction," 2009.

[36] J. Broekens, M. Harbers, K. Hindriks, K. Van Den Bosch, C. Jonker, and J.-J. Meyer, "Do you get it? user-evaluated explainable bdi agents," in *German Conference on Multiagent System Technologies*. Springer, 2010, pp. 28–39.

[37] T. Chakraborti, S. Sreedharan, Y. Zhang, and S. Kambhampati, "Plan explanations as model reconciliation: Moving beyond explanation as soliloquy," in *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017*, 2017, pp. 156–163.

[38] F. Elizalde, L. E. Sucar, M. Luque, J. Diez, and A. Reyes, "Policy explanation in factored markov decision processes." in *In Proc European Workshop on Probabilistic Graphical Models (PGM)*, 2008, pp. 97–104.

[39] O. Z. Khan, P. Poupart, and J. P. Black, "Minimal sufficient explanations for factored markov decision processes," in *Proceedings of the 19th International Conference on Automated Planning and Scheduling, ICAPS 2009, Thessaloniki, Greece, September 19-23, 2009*, 2009.

[40] R. Sukkerd, R. G. Simmons, and D. Garlan, "Towards explainable multi-objective probabilistic planning," in *Proceedings of the 4th International Workshop on Software Engineering for Smart Cyber-Physical Systems, ICSE 2018, Gothenburg, Sweden, May 27, 2018*, 2018, pp. 19–25.