Modeling and Analysis of Explanation for Secure Industrial Control Systems

SRIDHAR ADEPU, Singapore University of Technology and Design, Singapore
NIANYU LI, Peking University, China
EUNSUK KANG, Carnegie Mellon University, USA
DAVID GARLAN, Carnegie Mellon University, USA

Many self-adaptive systems benefit from human involvement and oversight, where a human operator can provide expertise not available to the system and detect problems that the system is unaware of. One way of achieving this synergy 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, an *explanation* can play an important role in allowing the human operator to understand why the system is making certain decisions and improve the level of knowledge that the operator has about the system. This, in turn, may improve the operator's capability to intervene and if necessarily, override the decisions being made by the system. However, explanations may incur costs, in terms of delay in actions and the possibility that a human may make a bad judgement. Hence, it is not always obvious whether an explanation will improve overall utility and, if so, what kind of explanation should be provided 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 system adaptation approach that leverages a probabilistic reasoning technique to determine when an explanation should be used in order to improve overall system utility. We evaluate our explanation framework in the context of a realistic industrial control system with adaptive behaviors.

ACM Reference Format:
Sridhar Adepu, Nianyu Li, Eunsuk Kang, and David Garlan. 2021. Modeling and Analysis of Explanation for Secure Industrial Control Systems . 1, 1 (February 2021), 24 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

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 a major goal 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

Authors' addresses: Sridhar Adepu, Singapore University of Technology and Design, Singapore, adepu_sridhar@mymail.sutd.edu.sg; Nianyu Li, Peking University, China, li_nianyu@pku.edu.cn; Eunsuk Kang, Carnegie Mellon University, USA, eunsukk@andrew.cmu.edu; David Garlan, Carnegie Mellon University, USA, garlan@andrew.cmu.edu.

Manuscript submitted to ACM

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 (i.e. software), 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 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 *intervene* in this control loop by overriding the commands sent to the actuators or, in the worst case, pausing or shutting down the system.

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 knowledge about the system. When an explanation is provided along with a control decision, under the right conditions, the operator may become more certain that the machine is following a desirable (undesirable) course of adaptation decision, and thereby be more likely to allow (disallow) 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 an 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

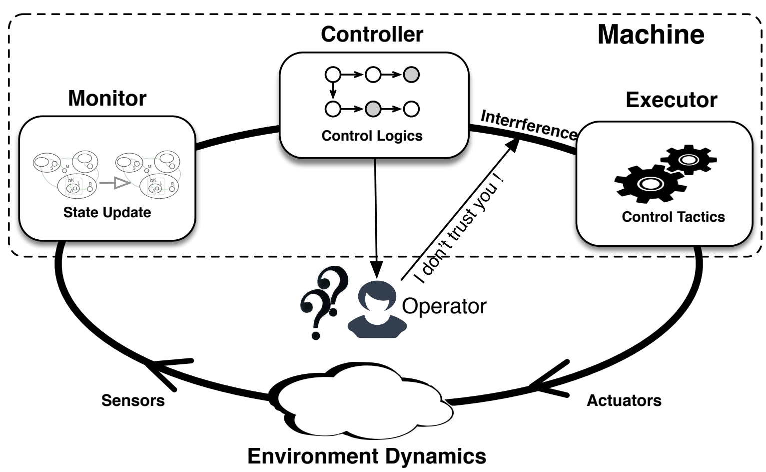

Fig. 1. A human-on-the-loop self-adaptive system.

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 [9], we proposed 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 capability 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 [10]. 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.

In this paper, we demonstrate an application of our proposed approach to a case study involving a real world water treatment system with a human operator who periodically monitors the system for potential undesirable behaviours. We present a case study to demonstrate how our approach can be used to provide explanations that guide the system and the operator towards optimal system utility. We also present a user study that demonstrates the applicability of our approach among human operators who design, build and operate real-world industrial control systems (ICSs). This evaluation aims to investigate how explanation is helpful to improve human operators' knowledge and contributes to their capability to intervene in the system actions.

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;
- An illustration of the applicability of our approach on a case study involving a realistic industrial control system (namely, a secure water treatment plant [11]); and
- A user study validating the applicability of explanations in real-world ICSs.

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 experimental results. The case study on industrial control system is presented in Section 5. The user study is presented in Section 6. Section 7 discusses related work and Section 8 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

Fig. 2. Self-driving vehicle scenarios.

situations.

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 [12], 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* [13, 14], occur in human involvement when the machine behaves differently than its operator expects. Two reasons are identified as accounting for these problems, as shown in Figure 3. One is that the operator may know only a subset of the information that the machine has (e.g., the presence of a 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.

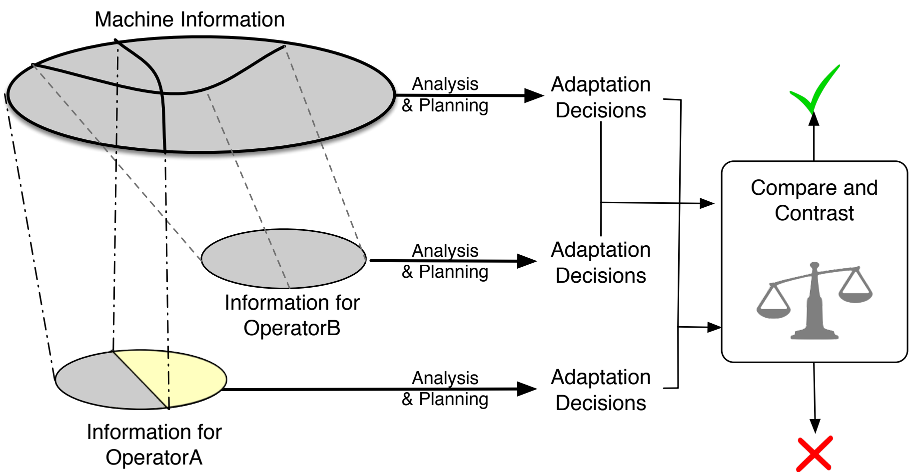

Fig. 3. Automation Surprises.

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, $\mathsf{sensorLeftFront} = \mathsf{car} \& \ \mathsf{sensorRightFront} = \mathsf{child}$ 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, $\mathsf{sensorLeftFront} \neq \mathsf{null} \& \ \mathsf{sensorRightFront} \neq \mathsf{null} \implies \mathsf{Stop}$ 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.

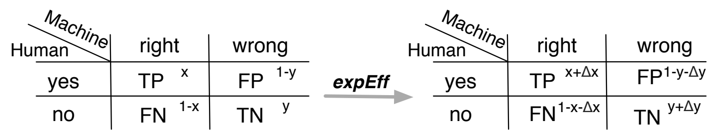

Fig. 4. Effect of an explanation as influencing the probabilities that the operator agrees or disagrees with the decision by the machine.

2.2 Explanation Effect

In our approach, we model the *effect* of an explanation as calibrating the operator's capability to make intervention decisions.

There are two cases that we consider. First, an explanation can potentially enable the operator to gain more capability 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 intervening on 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 and encourage them to intervene; we capture this as having the effect of *decreasing* the operator's capability in the machine.

The explanation effect is formally defined as a 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 intervention 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 intervention. 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 4 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 6.2.

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 by 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 [15, 16]. In this work, we use the PRISM tool [17], which supports reasoning about well-known behavioral specifications, such as Markov Decision Processes (MDPs) [18] and probabilistic timed automata (PTAs) [19], 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 change 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 Probabilistic Model Checking

Probabilistic model checking is a powerful technique for formally modeling and analyzing systems that exhibit stochastic behavior, allowing quantitative reasoning about probability and reward-based properties (e.g., resource usage, time, etc.) [10]. These techniques employ state-transition systems augmented with probabilities to describe stochastic system behavior. Moreover, probabilistic model checking approaches that support specification of non-determinism, such as Markov Decision Processes (MDPs) [18], and probabilistic timed automata (PTAs) [19], also enable the synthesis of strategies guaranteed to achieve optimal expected rewards. Our approach is based on the synthesis of optimal strategies for reward-based properties using PRISM [17].

Definition 3.1. (Markov Decision Process) A Markov decision process (MDP) is a tuple $M = \langle S,\ s_I,\ A,\ \delta,\ r \rangle$, where

- S denotes a finite set of states, and $S \neq \emptyset$;
- $s_I \in S$ is an initial state;
- $A \neq \emptyset$ is a finite set of actions;
- $\delta : S \times A \longrightarrow D(S)$ is a (partial) probabilistic transition function and $D(S)$ denotes the set of discrete probability distributions over finite set S;

- $r : S \rightarrow Q_{\geq 0}$ is a reward structure mapping each state to a non-negative rational reward.

An MDP models how the state of a system can evolve in discrete time steps. In each state $s \in S$, the set of enabled actions is denoted by $A(s)$ (we assume that $A(s) \neq \emptyset$ for all states). Moreover, the choice of which action to take in every state is assumed to be non-deterministic Once an action is selected, the successor state is probabilistically chosen according to probability distribution $\delta(s, a)$. We can reason about the behavior of MDP using strategies (also referred to as policies). A strategy resolves the non-deterministic choices of an MDP, selecting which action to take in every state.

Definition 3.2. (Strategy) A strategy of an MDP, M, is a function $\delta : S \rightarrow D(A)$ s.t., for each state $s \in S$, it selects a probability distribution $\delta(s)$ over $A(S)$.

The strategy in this context is memoryless (i.e., based solely on information about the current state) and deterministic ($\delta(s)$ is a Kronecker function such that $\delta(s)(a) = 1$ if action a is selected, and 0 otherwise).

Reasoning about strategies is a fundamental aspect of model checking an MDP, which enables checking for the existence of a strategy that can optimize an objective expressed as a quantitative property in a subset of probabilistic reward computation-tree logic (PRCTL) [15]. PRCTL extends PCTL [16] to reason about reward-based properties. A PRCTL property can state that an MDP has a strategy that can ensure that the probability of an event's occurrence or an expected reward measure meets some threshold. An extended version of the PRCTL reward operator $R^{r}_{max=?}[F^{*}\phi]$ enables the quantification of the maximum accrued reward r along paths that lead to states satisfying the state formula ϕ.

3.2 Machine Model

The machine is modeled over its evolution of one decision-making. A part of machine behavior is shown in Figure 5. 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 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 intervention. The probability of intervention is based on the human

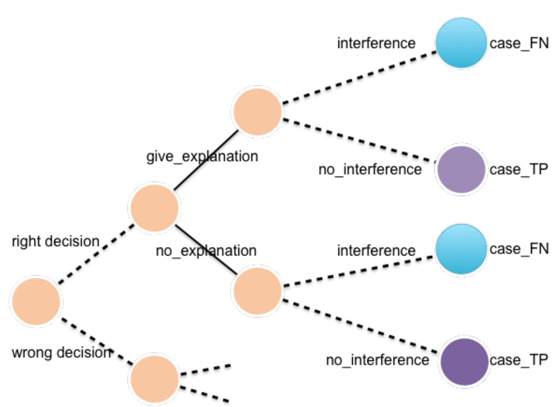

Fig. 5. Fragment of machine behaviors.

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 intervention. 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 four different 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 intervention when the machine decision is correct in lines 13-15. Meanwhile, the probability will be "TN" and "FP" when the machine decision is wrong. 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 3.4 below.

[1]MDPs are encoded in PRISM with commands like: [action]guard$\longrightarrow$ p_1 :u_1 +... + 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.

3.3 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 capability 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 capability a human has based on his existing 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 capability 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 capability. 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 capability in machine decision. Here we assume a decision making is a short period where human's capability 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 effects, i.e., different "Delta_X" and "Delta_Y" values can be specified as commands for possible explanation candidates in the explanation selection problem.

3.4 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 "case_TP" where it makes the right decision without human intervention. 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 EXPERIMENTAL RESULTS

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 [20], as shown in Figure 6. 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.

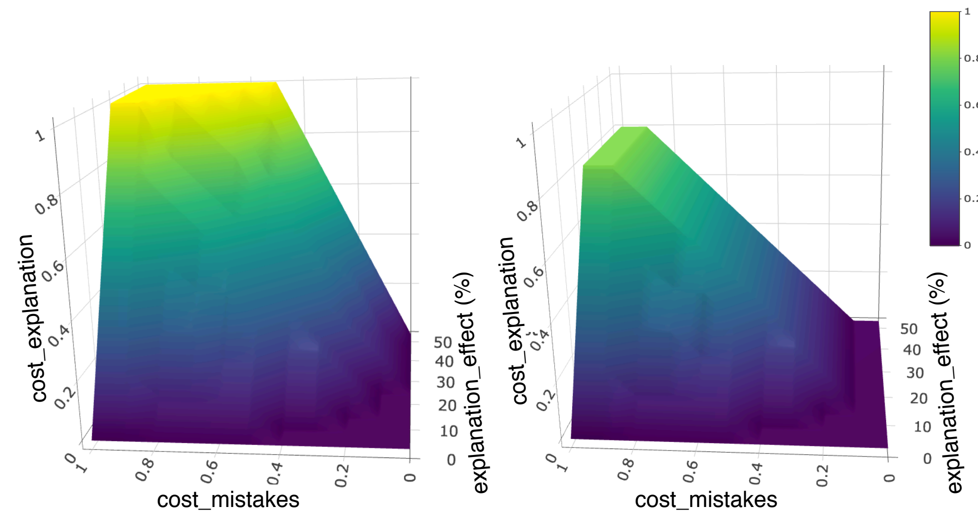

Fig. 6. (a) Explanation conditions for a novice (left); (b) explanation conditions for an expert (right).

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.

5 CASE STUDY: SWAT

In this section, we illustrate an application of our approach to a case study involving a real-world industrial control system (called Secure Water Treatment plant, or SWaT [11]) with a human operator who periodically monitors the system for potentially undesirable behaviors (e.g., faulty components or unexpected environmental inputs). In particular, for this case study, we have constructed PRISM models that describe (1) the behavior of the system, including how the machine makes decisions based on the state of the environment, (2) the behavior of a human operator, including how they may override the machine decisions when provided with an explanation, and (3) a space of candidate explanations and their impact on the operator's decision. We constructed the models based on knowledge collected from SWaT [21] and analysis of SWaT [22, 23]. In the following sections, we describe how our approach can be used to provide explanations that guide the system and the operator towards optimal system utility.

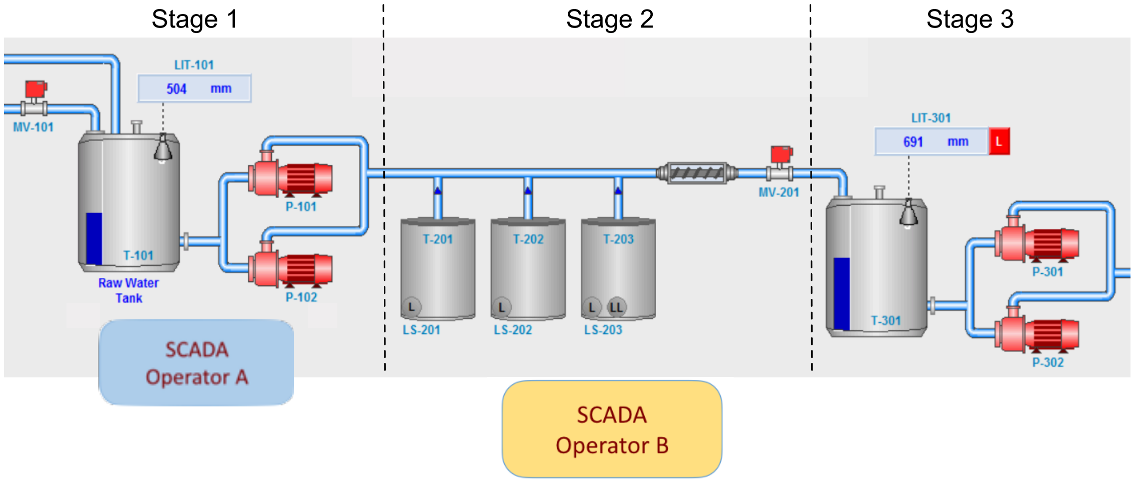

Fig. 7. First three stages of the water treatment process in the SWaT. Each stage is equipped with a set of sensors (e.g., level sensor LIT101) and actuators (valve MV101, pump 101) to monitor and manipulate the states of physical processes (tank T101).

5.1 SWaT: Secure Water Treatment

SWaT [21] is a water treatment testbed deployed at the Singapore University of Technology Design. It is a fully operational plant that produces five gallons/minute of purified water, with a capability to operate non-stop continuously in a fully autonomous mode. The system consists of multiple stages of water treatment processing, with each stage being controlled by a *programmable logic controller* (PLC) that monitors the state of the physical processes (e.g., water tank, pumps) through *sensors* and generates appropriate commands to *actuators* to manipulate those processes. Many of the published articles in ICS security are assessed for their effectiveness in SWaT [24–28]. Figure 7 illustrates the first three stages of the treatment process (which our case study focuses on).

Plant supervision and control: SWaT is equipped with a Supervisory Control And Data Acquisition (SCADA) system that can be used by a human operator to monitor the status of various physical processes and software throughout the stages. In a complex plant like SWaT, multiple operators may observe and manage different parts of the system through multiple SCADA displays. For example, as shown in Figure 7, operator A is in charge of managing the water tank and the two pumps in Stage 1, while a higher-level operator (e.g., operator B) may have access to a global view of the system by monitoring all three stages. (In industrial control systems like SWaT, different levels of operators are responsible for operating different parts of the plant, with a hierarchical relationship between the operators.)

SWaT operation: When initiated by an operator at the SCADA workstation, the system carries out the 6-stage treatment process in an autonomous manner, with the PLCs monitoring the sensors and generating appropriate actuator commands. (E.g., when the water level in the tank is detected as being too high, the PLC for Stage 1 generates a command to activate one of the attached pumps to allow water to flow out of the tank.)

When necessary, the operator can take control of the plant operations through SCADA and override commands generated by the PLCs. There are a number of scenarios in which human intervention may be needed. For example, sensors might fail from time to time and produce incorrect readings; similarly, due to malfunction, actuators may not always respond to the commands from the PLCs in an expected manner. In addition, the system, being connected to the Web and an unencrypted wireless network, may be susceptible to a range of security attacks. In particular, a malicious attacker may attempt to compromise the communication between the PLCs and the SCADA by spoofing network packets and injecting malicious sensor and actuator data[29]. In these cases, an operator is responsible for identifying potential anomalies and mitigating them based on their knowledge of the system.

Certain aspects of the control operation involve cooperation among multiple PLCs. For instance, Pump P101, when its state is set to ON, allows water to flow out of tank T101. Although P101 is controlled by PLC1 (PLC for Stage 1, not shown in Figure 7), the decision to turn it on or off depends on the water level in tank T301. In particular, as soon as PLC2 detects that the water level in tank T301 falls below a predefined value, it opens the motorized valve MV201. In turn, when PLC1 receives information about LIT301 and MV201, it turns on P101 to allow water to flow through MV201 and eventually into T301. This distributed nature of the control logic means that an operator with a partial view of the system may reach a decision that is sub-optimal with respect to overall system utility.

Explanation: In our approach, the information observed by the human operator through SCADA can be augmented with an explanation that is used to calibrate the operator's capability in the behavior of the plant. In particular, for the SWaT, explanations are used to justify two types of decisions made by a PLC: (1) open or close a motorized valve, and (2) turn on or off a water pump. If the operator believes that the machine is performing a wrong action, the operator might temporarily pause part of the plant operation and change the status of a particular actuator manually from the SCADA.

Utility functions: In the SWaT system, the following quality attributes are considered [23]: 1) throughput, measured by the water output from P301 for this study; and 2) safety, denoting that water tank should not overflow and the water properties (i.e., pH) are within the range. Below are the utility functions: 1) water output from T301 in a period of 10 minutes; 2) the risk of water overflow where water level in tank ranges from zero to 1100 millimeters (mm); and 3) pH value of water, where the water property is the best between 6 and 8. For simplicity, we only consider pH values and discard other water properties such as conductivity and oxidation-reduction potential.

$$U_{wateroutput} = \begin{cases} 1 & \text{if } WO \geq 90 \\ 0.5 & \text{if } 40 < WO < 90 \\ 0 & \text{if } WO \leq 40 \end{cases}$$

$$U_{T301_{overflow}} = \begin{cases} 1 & \text{if } T301 \leq 1100 \\ 0.5 & \text{if } 1100 < T301 < 1150 \\ 0 & \text{if } T301 \geq 1150 \end{cases}$$

$$U_{pH} = \begin{cases} 1 & \text{if } 6 \leq pH \leq 8 \\ 0.5 & \text{if } 5 \leq pH < 6 \ \& \ 8 < pH \leq 9 \\ 0 & \text{if } else \end{cases}$$

A utility function $U_{Total} = X * U_{outflow} + Y * U_{overflow} + Z * U_{pH}$ is used to calculate the total utility for the machine. X, Y, and Z are the weights in the equation where $X + Y + Z = 1$ and we assign '1/3' to all three weights for this case study.

5.2 Experimental Scenarios

In self-adaptive systems, adaptation refers to situations in which the environment deviates from its expected behavior. In SWaT, there are two different types of deviations: 1) the level of water in a tank moving into an unsafe state such as an overflow or underflow, 2) water properties, such as pH and conductivity, increasing or decreasing beyond a safe range.

Table 1. A scenario in which the PLC decides to turn OFF pump P101. Operators A and B may decide to intervene and override the PLC command. Since they have only partial information about the system, their overriding action (i.e., turn ON P101) is one that actually results in a state with a lower utility.

Sensors actuators	**Curr. state**	**Machine decision**	**Control logic**	**Info (Op.A)**	**decision (Op.A)**	**Info (Op.B)**	**decision (Op.B)**	**State with higher utility**	**State with lower utility**
P101/ P102	ON	*OFF*	When LIT301 ≥ 1000, turn OFF pump P101	**State** ($\subseteq S_M$): P101, P102, LIT101 **Transition** (T_M): When LIT101>250, turn ON pump P101	Turn P101 ON	**State** ($\subseteq S_M$): P101, P102, LIT101, LIT301, P301, pH **Transition** (T_M): None	Turn P101 ON	P101 == OFF, where water inflow is stopped into T301	P101==ON, where tank T301 will overflow in 10 mins
MV101	CLOSE	CLOSE							
LIT101	750	-							
LIT301	1000	-							
P301	ON	ON							
pH	7	-							

Note: The control logic is shown for the adaptation decision

Scenario 1: In one possible adaptation scenario, the PLC1 makes a decision turning OFF pump P101 as shown in Table 1. Here, the water level in tank T301 has reached the maximum safe threshold value (1000 mm). Based on the current state of sensors and actuators, pump P101 is to be turned off in order to stop the inflow into tank T301 and avoid overflow. In this case, the machine is making the right decision. However, since operator A does not have access to the status of LIT301, it makes a decision that the pump P101 should remain ON, which will eventually lead to an overflow of water in T301. Similarly, even though operator B has access to all of the sensors and actuator's statuses, it does not have knowledge of the control logic that governs how PLC1 and PLC2 manipulate LIT101, LIT301, and P101, and prefers maintaining the status quo (i.e., keep P101 ON). Therefore, both operators may benefit from an explanation to reduce the false-negative error and decrease the probability of operator intervention.

Two of the possible explanation candidates explored by our PRISM method are shown as follows:

Explanation Candidate 1:

Content = {
  Transitions:
    When LIT301 ≥ 1000, turn OFF pump P101;
  States:
    LIT101= 750;
    LIT301 = 1000;
}

Explanation Candidate 2:

Content = {

  Transitions:
    When LIT301 ≥ 1000, turn OFF pump P101;
  States:
    N/A
}

Here, we assign the cost of 0.15 and 0.1 to Candidates 1 and 2, respectively, since the latter contains less amount of information. The utility for the state that results when P101 is turned OFF is assigned a value of '1' (computed as $0.33 \times \text{overflow}(1) + 0.33 \times \text{output}(1) + 0.33 \times \text{pH}(1)$). On the other hand, if P101 remains ON, it eventually leads to an overflow in T301, and thus the utility for the resulting state is assigned a value of 0.66.

For operator A, explanation candidate 1 will promote the probability of true-positive by 0.4 (i.e., $\Delta x = 0.4$), which means this candidate can increase the capability of machine adaptation decision to 0.9 from 0.5 (randomly guessing whether he should trust machine decision without intervention), while explanation candidate 2 only has positive effect of 0.1 as operator A still does not know the status of water level LIT301 from SCADA and this explanation. Therefore, the optimal strategy of the machine to maximize its utility is to provide explanation candidate 1 to operator A. However, for operator B, since both of the candidates could increase the true-positive probabilities by 0.4, explanation candidate 2 should be chosen as its cost is less than that of candidate 1.

Scenario 2: In this scenario, as shown in Table 2, the initial LIT101 is 800, and MV101 is OPEN. Based on the control logic (LIT101 ≥ 800 then CLOSE MV101), MV101 is supposed to be CLOSE. However, an attacker injects the value of 500 to the controller instead of 800; PLC could only access the water level LIT101 as 500. Based on that, the machine adaptation decision is OPEN. Due to the physical environment information about tank T101, Operator A is able to see the real water level LIT101 and plan the adaptation decision CLOSING valve MV101. However, operator B prefer the remaining status quo as he does not have information about the control logic.

Table 2. Scenario for action to overflow tank T101

Sensors actuators	**Curr. state**	**Machine decision**	**Control logic**	**Info (Op.A)**	**decision (Op.A)**	**Info (Op.B)**	**decision (Op.B)**	**State with higher utility**	**State with lower utility**
LIT101	800	-	When LIT101 ≥ 800, CLOSE valve MV101	**State** ($\subseteq S_M$): LIT101 **Transition** (T_M): When LIT101≥250, CLOSE valve MV101	CLOSE valve MV101	**State** ($\subseteq S_M$): LIT101 **Transition** (T_M): None	OPEN valve MV101	MV101== CLOSE, where water inflow is stopped into T101	MV101== OPEN, where tank T101 will overflow in 2 mins
MV101	OPEN	*OPEN*							

Explanation Candidate 1:

Content = {
  Transitions:
    When LIT101 ≥ 800, CLOSE valve MV101;
  States:
    LIT101= 800;

}

We assign a cost of 0.15 to this explanation candidate. The utility for the state with low utility (i.e., OPEN MV101 without operator intervention and after 2 minutes water will overflow tank T101) is '0.66' (=0.33 × overflow(0) + 0.33 × output(1) + 0.33 × water properties(1)). While the utility for the state with high utility (i.e., CLOSE MV101 manually from SCADA) is 1 without overflow in 10 minutes. For operator A, explanation candidate will promote the capability of 0.4 (i.e., from 0.5 probability of true-negative to 0.9 while probability of false-negative will reduce from 0.5 to 0.1) as he is close to the physical environment of stage 1 and very likely to identify the fake value of LIT101 by comparing information the environment has with the one the machine has, and know the good adaptation from the control logic. However, the effect of this candidate for operator B is zero since he only has access to his SCADA screen, which is not located close to the physical environment of stage 1. In this scenario, the explanation candidate will be provided to operator A while operator B will receive no explanation.

To give an indication of the complexity of explanation selection, we note that our prototypical implementation for the SWaT case study consisted of 62 lines of PRISM, and the model checking time was under a few seconds on average.

6 USER STUDY

To validate the applicability of our approach with operators in real-world industrial control systems (ICSs), we performed an evaluation through a survey. This evaluation aims to investigate how explanation is helpful for improving the operators' the capability to make intervention decisions in industrial control systems. We designed a questionnaire that presented participants with multiple, possibly unsafe scenarios that may arise in an ICS (in particular, the SWaT system); the questions were then designed to evaluate how their decision to intervene on the system decision might change after they were presented with an explanation. For recruiting, we invited 100 participants who may prior experience with ICS through emails and the LinkedIn network; out of these, 43 agreed to participate and completed the questionnaire. We discuss the design of the questionnaire and the results of our study in more detail next.

6.1 Study Details

The questionnaire: After introducing the participants to our work, they were required to fill in a questionnaire. Participation was voluntary and the estimated time to complete each survey was around 15 minutes, including 5 minutes of background introduction and 10 minutes for questionnaire completion. Our survey included the 10 questions shown in Figure 8. The first two questions (Q1 and Q2) investigate the familiarity of the SWaT system. Questions 3, 4 and 5 are used to investigate whether an explanation might aid in the operator's decision to intervene in a given scenario—in particular, identical to Scenario 1 presented in Section 5.2. In Q6 and Q7, the participants are asked whether an explanation could be helpful for recognizing that the system is making an erroneous decision in Scenario 2 from Section 5.2. Question 8 asks more generally whether an explanation would be help in improving the operator's knowledge of the system; Q9 and Q10 ask for other types of information beside our notion of explanation that might be helpful in similar scenarios.

Figure 9, Figure 10 and Figure 11 show the results of the survey collected from the 43 participants. We provide a brief summary of the results in the following.

Capability: In Q5, out of 43 participants, 36 responded that an explanation was helpful in aiding their decisions to intervene in Scenario 1 while 7 responded that an explanation was not helpful. As shown in Figure 11, in Q6 (Scenario 2), 18 out of 43 indicated that explanation did not have any effect on improving their knowledge when one could not directly observe the faulty sensor, while 6 responded with 'don't know'. In Q7, 31 out of 43, participants indicated that

Q1. Are you familiar with sensors and actuators in the SWaT? (Yes, No)
Q2. Are you familiar with control actions (control logic) in the SWaT? (Yes, No)
(*Description only*) Figure 7 (cf. 5.1): The first three stages of the water treatment process in the SWaT. Each stage is equipped with a set of sensors (e.g., level sensor LIT101) and actuators (valve MV101, pump 101) to monitor and manipulate the states of physical processes (tank T101).
Q3. *Current state*: P101 = ON, MV101=CLOSE, LIT101=750, LIT301=1000 and P301=ON;
Operator observable state (at SCADA): P101 = ON, MV101=CLOSE, LIT101=750, LIT301=1000 and P301=ON;
Control Action: turn OFF P101;
How likely will you pass the system action given the state?Choose your likelihood to pass the action from below: (No, Not-much, Low, Medium, High)
Q4. *Explanation*: The system has produced action 'turn OFF pump P101' because of the following control logic:
Control logic: When LIT301>= 1000, turn OFF pump P101;
how likely will you pass the systems action given the state after the explanation? Choose your likely to pass the action from below: (No, Not-much, Low, Medium, High)
Q5. Does an explanation above change your likelihood on intervening the control action associated with P101.? (Yes, No)
Q6. Let us assume, we have an operator who can directly observe the physical water level of Tank T101.
Current state: LIT101=800, MV101=OPEN
Operator observable state (at SCADA): LIT101=500, MV101=OPEN. Here, due to a glitch on LIT101 (level sensor), the controller reports 500 instead of 800 to the SCADA.
Control Action: OPEN MV101
Explanation:
Control logic: When LIT101>= 800, CLOSE MV101
Current state: LIT101=800, MV101=OPEN
Does an explanation have any effect on the probability of intervention decision of the operator who cannot directly observe the faulty sensor? (Yes, No, Don't Know)
Q7. Do you think the explanation above can help the operator recognize that the system is making an erroneous decision with MV101, so that increase operator's probability to intervene? (Yes, No, Don't Know)
Q8. If you are an operator, would you be interested in using such an explanation to calibrate your capability that the system is making an appropriate or erroneous decision? (Yes, No)
Q9. Do you think you need any further assistance as an operator other than an explanation? (Yes, No, Maybe)
Q10. If yes or maybe to the above question, what other types of information would you like to have?

Fig. 8. Survey questionnaire

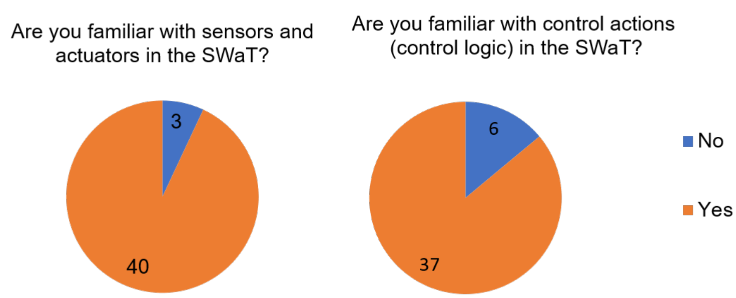

Fig. 9. Familiarity of participants with the SWaT (Q1, Q2)

an explanation was helpful in recognizing whether the system was making an erroneous decision with MV101 (in scenario 2). In Q8, 38 out of 43 expressed interest in using an explanation as an aid for making intervention decisions.

Suggestions for explanation: In total, 28 out of 43 would like to have other types of assistance an operator apart from the explanation (Q9). We have combined these suggestions in Table 3 (Q10).

Conclusions: Our study shows that 1) an operator's capability to intervene with machine decisions can be improved by explanations; 2) When the system is making an erroneous decision, the explanation is helpful for the operator to recognize it; and 3) operators are interested to use an explanation as an aid in determining whether that the system is making an appropriate or erroneous decision in real plants.

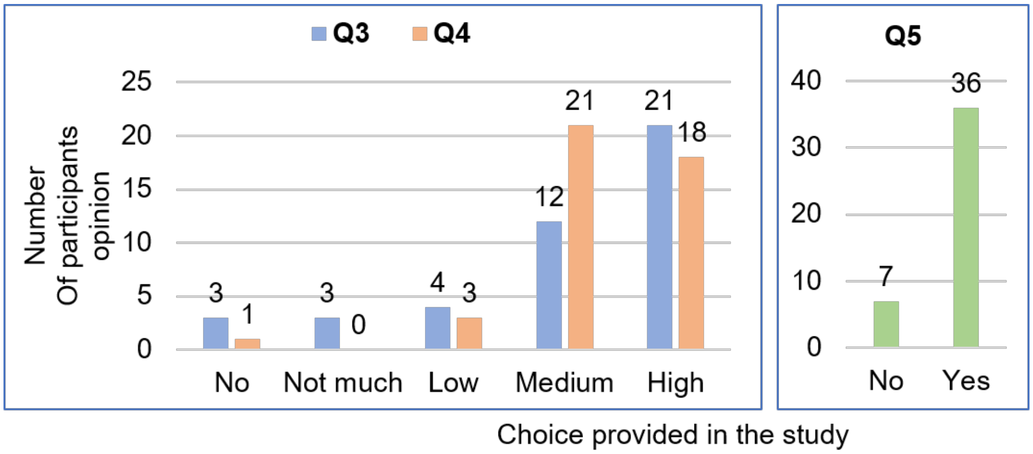

Fig. 10. Number of participants opinion on whether the explanation is improving the capability of the operator in scenario 1 (Q3, Q4, Q5).

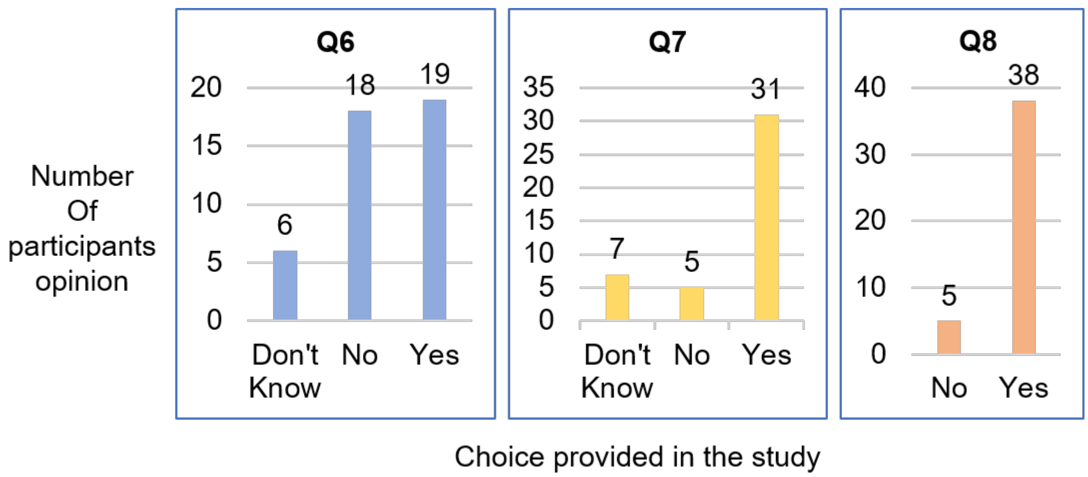

Fig. 11. Number of participants opinion on whether the explanation is improving the capability of the operator in scenario 2 (Q6, Q7, Q8).

In total, 7 out of 43 participants indicate that the explanation may not help to improve operator accuracy of information in control action associated with P101 (Q5). It indicates that experts may not need any explanation to understand the control action associated with P101. Experts already have enough capability in this particular scenario to pass the actuator action. It is also possible that some of the operators may not want to change the system decision irrespective of the input information. In Q6, the majority (19) of the participant's responses state that explanation does affect improving the accuracy of the operator who cannot directly observe the faulty sensor.

Threats to Validity: The study is focused on explanation effect, and not considered about explanation cost (i.e., human annoyance or human satisfaction). Although our framework includes explanation cost as an important component, this

Table 3. Responses for the Q10. If yes or maybe to the above question, what other types of information would you like to have?

S.No	Q8 response
1	An alert message showing malfunctioning of MV101
2	I need to know the status of the other sensors and actuators dependent on stage 1 and flow rate of the water to determine and take appropriate action
3	Explanation is only helpful if an operator can directly observe the physical state but it would not be possible for a huge number of physical states in a city-scale plant
4	It would be great to pinpoint the anomalies and potential faults in the system based on the control logic to help the operator adjust their capability
5	It should be able to detect sensor failures instead and inform the operator there is a fault. Alert him to check and rectify.
6	A more user-friendly explanation (or as a simple message) as an operator may not have a logic level or debug experience.
7	Most SCADA Operators do not have the academic background to appreciate SCADA control systems using state mechanics. It would be more helpful to simply provide a conclusion of a glitch affecting readings will sometimes happen
8	The explanation may also be faulty as the explanation is based on the sensor glitch. So I do not think it will affect capability.
9	Possible resolutions, and automatic incidence response
10	explanation with graphical representations.

Note: This table presents a non-exhaustive list of suggestions.

study does not consider it. We are not considering different explanations for different abilities of operators in the study. The explanation effect for different operators with different training levels could be different. A more accurate study can be conducted to handle different abilities of operators by collecting the amount of time and annoyance levels to read and performing actions after understanding the explanation.

6.2 Discussion

Our framework relies on an assumption that the probabilities behind intervention decision levels, as well as explanation effect, can be accurately measured. In our group, an ongoing research project with a user study is exploring how such probabilities may be obtained through experimentation [30]. 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 [31].

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 [32]. 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 overall utility, and how to take the time delay between decision making and human intervention into consideration.

The results from our study show that our explanation approach significantly improves the operators' capability to correctly determine actuator status. The operators who are given explanations are, on average, more likely to gain more capability than those who are not. The explanations provide an improvement in the operators' capability. The results also show that, when one or more sensors are malfunctioning, it is more difficult for the participant to recognize the actuator state. In that situation, the operators are 50% of less likely to be correct in this type of scenario. This poses many challenges to improve explanation for operators and other types of assistance during plant operation.

Two participants suggested less capability with the explanation than without explanation for the scenario 1. This indicates that the two participants' capability was reduced with the explanation. Two more participants mentioned the accuracy of information level of "Not sure" with and without explanation while participants mentioned that they are aware of sensors, actuators and control strategy. In our approach, we model the explanation effect as the Δ in true positive and true negative probability. The Δ could be a positive or negative value. So, if the explanation is confusing, it could have no or even detriment effect on the operator. Several further directions Include exploring building models, reason model, and conduct empirical studies on important aspects collected in Q10 such as possible resolutions and automated incident response, and explanations with graphical representations.

7 RELATED WORK

The explanation has surged recently especially in the field of artificial intelligence, with the notion of eXplainable Artificial Intelligence (XAI) [33]. However, over three decades ago, explanation has been investigated with prosperity in expert systems [34–36]. Also, there exists literature on explainable agents and robots; in applications on factory environments [37], military missions [38], human players [39], training [40], e-health [41] and recommendation systems [42]. And in the fields of philosophy, social psychology, and cognitive psychology, there are vast questions such as what constitutes an explanation, what is the function of explanation and what are their structures. However, despite the fact that self-adaptive systems are becoming a trend for several applications such as self-driving, smart office and e-health, research work on explanation is still in its infancy stage. 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 [43].

Explaining self-adaptive systems' behaviors and reasoning mechanisms have been studied in different ways across different disciplines. The authors in [44] distinguish three explanation phases: explanation generation, explanation communication, and explanation reception. Our formal definition of explanation touches all three phases, generating the explanation content, presenting the content to the human operator, and denoting how well the operator understands the explanation with the explanation effect. Explanation generation is aiming to generate two categories [30]: 1)"what-explanation", a description of the solution of a planning problem; 2)"why-explanation", a justification of why the policy is selected as the solution. In our work, we focus on the second category – justifying why the adaptation decision is chosen or why the machine behaves in a particular way.

Several existing works are aiming to explain systems that produce particular behaviors. The work in [45] describes how the state of the machine is captured in a human's mind. When the behavior of an agent is not explained, the state in mind may not be consistent with the real state, which could lead to dangerous situations. Also, lack of mental model for the human estimating the actions of robots may lead to safety risks [46, 47]. Lin et al. contributed an automatic explanation for the different explanation types and decision model types [31]. Chakraborti et al. treated the explanation as the model reconciliation problem aiming to make minimal changes to the human's model to bring it closer to the robot's model [48]. Elizalde et al. contributed an approach that identifies factors that are most influential to the decision making with MDP [49]. Khan et al. presented an approach for explaining an optimal action in policy by counting the frequency of reaching a goal by taking the action [50]. 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 [51]. However, most of their work only focuses on the explanation generation and does not capture the explanation effect nor the cost explanation brings.

Secure industrial control systems: As we know there is no related work exists related to explanation in secure industrial control systems. However, in this part of the related work, we cover related work in ICS security. A large body of research has investigated impact of cyber attacks on industrial control systems measurement and control signals[29, 52]. These works consider different attack models that result in different kinds of cyber attacks from false data injection to denial of service attacks. The false data injection attacks on industrial control systems are also investigated in various instances ([53–55]). To protect against cyber attacks, the works in [56–58] to develop various methods for monitoring, detecting and preventing cyber attacks. When there is an alert from these monitoring systems for the human operator, it is necessary to analyse and determine how the operator is receiving the alerts information. Close work in these aspects studied about explainable software for cyber physical systems in [59]. Mentioned the importance of self-explainable software systems within CPS in run-time and how explainability useful for safety and security. They are briefly discussed *'security explain-ability by design'* in evolving security mechanisms and explain-ability of provably safe distributed autonomous cars.

8 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, known as automation surprises. We present an explanation selection framework with a formal definition of explanation in three components and synthesize explanation strategy based on probabilistic model checking. This proposed framework is applied in a water treatment plant and evaluated in a real-world human-on-the-loop system. We have conducted a user study to determine the applicability of our approach among human operators who design, build and operate ICSs.

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, 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*. IEEE Computer Society, 2011, pp. 66–69.

[9] N. Li, S. Adepu, E. Kang, and D. Garlan, "Explanations for human-on-the-loop: A probabilistic model checking approach," in *Proceedings of the 15th International Symposium on Software Engineering for Adaptive and Self-managing Systems (SEAMS)*.

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

[11] Singapore University of Technology and Design, "Secure water treatment (SWaT)," https://itrust.sutd.edu.sg/itrust-labs-home/itrust-labs_swat/.

[12] "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.

[13] 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.
[14] 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.
[15] 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, pp. 1–12.
[16] 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.
[17] 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.
[18] M. L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*, ser. Wiley Series in Probability and Statistics. Wiley, 1994.
[19] 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.
[20] K. Soetaert, "plot3d : Tools for plotting 3-d and 2-d data." https://cran.r-project.org/web/packages/plot3D/vignettes/plot3D.pdf, 2018.
[21] A. P. Mathur and N. O. Tippenhauer, "SWaT: A water treatment testbed for research and training on ICS security," in *2016 International Workshop on Cyber-physical Systems for Smart Water Networks (CySWater)*, April 2016, pp. 31–36.
[22] E. Kang, S. Adepu, D. Jackson, and A. P. Mathur, "Model-based security analysis of a water treatment system," in *2016 IEEE/ACM 2nd International Workshop on Software Engineering for Smart Cyber-Physical Systems (SEsCPS).* IEEE, 2016, pp. 22–28.
[23] S. Adepu and A. Mathur, "An investigation into the response of a water treatment system to cyber attacks," in *2016 IEEE 17th International Symposium on High Assurance Systems Engineering (HASE).* IEEE, 2016, pp. 141–148.
[24] A. Maw, S. Adepu, and A. Mathur, "ICS-BlockOpS: blockchain for operational data security in industrial control system," *Pervasive and Mobile Computing*, vol. 59, p. 101048, 2019.
[25] Y. Chen, C. M. Poskitt, J. Sun, S. Adepu, and F. Zhang, "Learning-guided network fuzzing for testing cyber-physical system defences," in *2019 34th IEEE/ACM International Conference on Automated Software Engineering (ASE).* IEEE, 2019, pp. 962–973.
[26] S. Adepu, F. Brasser, L. Garcia, M. Rodler, L. Davi, A.-R. Sadeghi, and S. Zonouz, "Control behavior integrity for distributed cyber-physical systems," in *2020 ACM/IEEE 11th International Conference on Cyber-Physical Systems (ICCPS).* IEEE, 2020, pp. 30–40.
[27] T. K. Das, S. Adepu, and J. Zhou, "Anomaly detection in industrial control systems using logical analysis of data," *Computers & Security*, p. 101935, 2020.
[28] M. A. Umer, A. Mathur, K. N. Junejo, and S. Adepu, "Integrating design and data centric approaches to generate invariants for distributed attack detection," in *Proceedings of the 2017 Workshop on Cyber-Physical Systems Security and PrivaCy*, 2017, pp. 131–136.
[29] S. Adepu and A. Mathur, "Assessing the effectiveness of attack detection at a hackfest on industrial control systems," *IEEE Transactions on Sustainable Computing*, 2018.
[30] 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.
[31] 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 USA, April 4-9*, 2009, pp. 2119–2128.
[32] 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.
[33] T. Miller, "Explanation in artificial intelligence: Insights from the social sciences," *Artif. Intell.*, vol. 267, pp. 1–38, 2019.
[34] 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.
[35] T. Fennel and J. D. Johannes, "An architecture for rule based system explanation," 1990.
[36] 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.
[37] 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.
[38] 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.
[39] M. Molineaux, D. Dannenhauer, and D. W. Aha, "Towards explainable NPCs: a relational exploration learning agent," in *Workshops at the 32nd AAAI Conference on Artificial Intelligence*, 2018.
[40] 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.
[41] 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.
[42] 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.
[43] P. Carey., *Data protection: a practical guide to UK and EU law.* Oxford University Press, Inc., 2018.

[44] M. A. Neerincx, J. van der Waa, F. Kaptein, and J. van Diggelen, “Using perceptual and cognitive explanations for enhanced human-agent team performance,” in *Engineering Psychology and Cognitive Ergonomics - 15th International Conference, EPCE 2018, Held as Part of HCI International 2018, Las Vegas, NV, USA, July 15-20, 2018, Proceedings*, 2018, pp. 204–214.

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

[46] C. L. Bethel, “Robots without faces: non-verbal social human-robot interaction,” 2009.

[47] 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.

[48] 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, Australia*, 2017, pp. 156–163.

[49] 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.

[50] 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.

[51] 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.

[52] A. A. Cárdenas, S. Amin, Z.-S. Lin, Y.-L. Huang, C.-Y. Huang, and S. Sastry, “Attacks against process control systems: risk assessment, detection, and response,” in *Proceedings of the 6th ACM symposium on information, computer and communications security*, 2011, pp. 355–366.

[53] Y. Liu, P. Ning, and M. K. Reiter, “False data injection attacks against state estimation in electric power grids,” *ACM Transactions on Information and System Security (TISSEC)*, vol. 14, no. 1, pp. 1–33, 2011.

[54] N. Trcka, M. Moulin, S. Bopardikar, and A. Speranzon, “A formal verification approach to revealing stealth attacks on networked control systems,” in *Proceedings of the 3rd international conference on High confidence networked systems*, 2014, pp. 67–76.

[55] S. Adepu, N. K. Kandasamy, and A. Mathur, “EPIC: An electric power testbed for research and training in cyber physical systems security,” in *Computer Security.* Springer, 2018, pp. 37–52.

[56] Y. Z. Lun, A. D’Innocenzo, F. Smarra, I. Malavolta, and M. D. Di Benedetto, “State of the art of cyber-physical systems security: An automatic control perspective,” *Journal of Systems and Software*, vol. 149, pp. 174–216, 2019.

[57] G. Sabaliauskaite and S. Adepu, “Integrating six-step model with information flow diagrams for comprehensive analysis of cyber-physical system safety and security,” in *2017 IEEE 18th International Symposium on High Assurance Systems Engineering (HASE).* IEEE, 2017, pp. 41–48.

[58] S. Adepu, E. Kang, and A. P. Mathur, “Challenges in secure engineering of critical infrastructure systems,” in *2019 34th IEEE/ACM International Conference on Automated Software Engineering Workshop (ASEW).* IEEE, 2019, pp. 61–64.

[59] J. Greenyer, M. Lochau, and T. Vogel, “Explainable software for cyber-physical systems (ES4CPS): Report from the GI Dagstuhl seminar 19023, january 06-11 2019, schloss dagstuhl,” *arXiv preprint arXiv:1904.11851*, 2019.