Hey! Preparing Humans to do Tasks in Self-adaptive Systems.

Nianyu Li
Peking University
li_nianyu@pku.edu.cn

Javier Cámara
University of York
javier.camaramoreno@york.ac.uk

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

Bradley Schmerl
Carnegie Mellon University
schmerl@cs.cmu.edu

Zhi Jin
Peking University
zhijin@pku.edu.cn

***Abstract*—Many self-adaptive systems benefit from human involvement, where human operators can complement the capabilities of systems (e.g., by supervising decisions, or performing adaptations and tasks involving physical changes that cannot be automated). However, insufficient preparation (e.g., lack of task context comprehension) may hinder the effectiveness of human involvement, especially when operators are unexpectedly interrupted to perform a new task. Preparatory notification of a task provided in advance can sometimes help human operators focus their attention on the forthcoming task and understand its context before task execution, hence improving effectiveness. Nevertheless, deciding when to use preparatory notification as a tactic is not obvious and entails considering different factors that include uncertainties induced by human operator behavior (who might ignore the notice message), human attributes (e.g., operator training level), and other information that refers to the state of the system and its environment. In this paper, informed by work in cognitive science on human attention and context management, we introduce a formal framework to reason about the usage of preparatory notifications in self-adaptive systems involving human operators. Our framework characterizes the effects of managing attention via task notification in terms of task context comprehension. We also build on our framework to develop an automated probabilistic reasoning technique able to determine when and in what form a preparatory notification tactic should be used to optimize system goals. We illustrate our approach in a representative scenario of human-robot collaborative goods delivery.**

I. Introduction

A self-adaptive system is designed to be capable of modifying its structure and behavior at run time in response to changes in its operational environment and the system itself (e.g., faults, changing requirements and attacks) [1], [2]. Although automation is one desirable characteristic of self-adaptation, many adaptive systems also benefit from human involvement (e.g., when a human operator supervises decisions, performs actions that are difficult to automate, or provides information that is difficult to acquire). In these cases, the system may be able to adapt more effectively when its adaptation strategies can exploit human actions [3], [4], [5], [6], [7], [8]. For example, in a security setting, a system can combine automated actions for intrusion detection and remediation with human-guided decision making to defend the system against potential attackers [6].

Enabling effective human machine collaboration, however, is challenging because humans may be immersed in other things and when they are unexpectedly interrupted to perform a new task, they do not only work less efficiently, but also make more mistakes. Thus, a lack of preparation may hamper human effectiveness at carrying out the task assigned by the system. Prior work has investigated how attention management [9], [10], [11], [12] and context awareness [13] affect a human's capability to cooperate with the system. Attention management allows human operators to be more proactive and controls the focus of their attention to reduce the chances of task performance degradation due to distraction and lack of timeliness. Context awareness entails assimilating the information about the context in which a task is conducted, and integrating that information in responses or actions [14], [15]. A preparatory notification for human operators is helpful to manage their attention and allow them to understand the context of a forthcoming task in advance. Human preparation can be regarded as an example of proactive self-adaptation [16], [17] with human-involvement.

Despite the fact that preparatory notification can yield positive effects on the outcome of a forthcoming task, it may also interfere with a human's current activities, including the execution of other existing tasks. When there is little time between the instant in which the need of a human-involved task is predicted and the start of that task, the urgency of the notification might need to be promoted so that the human can divert their attention in time. Moreover, the inherent uncertainty in the behavior of human operators, who might even ignore the notification altogether, further increases the difficulty of proactive adaptation. Hence, it is nontrivial and important to reason about when and in what form to use preparatory notification in accordance with the level of human involvement required: too late or too little urgency of the notification, and the human may not be effectively prepared to complete the assigned task; too early or too much urgency, and the human operator's level of attention may progressively degrade prior to the arrival of the new task.

In this paper we apply a formal framework following our previous work [6], [4], [7], [8] for reasoning about when and in what form to use preparatory notification. Specifically, we see preparatory notification as a proactive adaptation action (or *tactic*) that the system can use judiciously in strategies involving human participation. Such tactics are associated with benefits and costs that affect overall system utility, and allow a probabilistic planner to determine the optimal use of preparatory notification under an uncertain environment. Such uncertainty is influenced by human attributes (e.g., training level) and other external conditions – like the characteristics of the forthcoming task. Key to this framework is the explicit use of human attention to refine a human attributes model (i.e., OWC model [7], [6], [18]), which positively affects their task preparation, as well as the notion of context complexity negatively affecting context awareness and ultimately, the effectiveness of human operators.

As we elaborate, using our framework allows an on-line planner to determine when and in what form preparatory notification will be most effective. Important features of the framework are that it can be tuned as needed to accommodate: (a) different training levels of the human operator, (b) the impact that preparatory notification will have on operator's attention depending on the relative task priority, and (c) different task context complexity to understand the context that arises from the characteristics of the task. The main contributions in this paper are: 1) A formal framework for designing self-adaptive systems where preparatory notification can be used as a tactic to proactively aid the human operator in improving overall system utility; and 2) The use of probabilistic model checking to analyze when and in what forms to use a preparatory tactic in an adaptation strategy. We illustrate the applicability and benefits of our approach in a collaborative delivery scenario between human and robot.

II. Background and Related Work

This section introduces a motivating scenario to illustrate our approach, background on model checking of stochastic multi-player games (SMG), human involvement in self-adaptation, and related work on attention management and context awareness for human involvement in software systems.

A. Motivating Scenario

Semi-autonomous mobile robots such as Amazon Scouts [19] are becoming increasingly popular to deliver goods in selected urban areas. These are mini-tank-like delivery vehicles that can be supported by a human operator who can assist the system by generating delivery plans for the robots. These plans have to take into consideration updated conditions of the mission's execution context, such as the priority of delivery or the distance to the points in which the packages have to be delivered. Once a plan is provided, the robot will follow the plan without any human intervention, avoiding obstacles automatically to arrive safely to its delivery points. During delivery, human operators are free to do other tasks while the system monitors the robot's progress and tries to attract the operator's attention back to the delivery task when the robot is near completion. Once the delivery is completed, the operator will be asked to produce a new plan based on the updated information provided by the system.

In this scenario, the human operator and the system collaborate to complete the delivery task as fast as possible. Without preparatory notification, the time required for an operator to generate a plan may be more because she may need to shift her attention to the new task and understand the new context.

B. Model Checking Stochastic Multiplayer Games

Probabilistic model checking is a 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.) [20]. Our approach to reasoning about preparatory notification for human involvement in proactive adaptation builds upon a technique for modeling and analyzing turn-based stochastic multiplayer games [21] (abbr. SMG). In each state of the model, only one player (e.g., operator or system) can choose between several actions, the outcome of which can be probabilistic. Players can follow strategies to either cooperate to achieve the same goal, or compete to achieve their own (possibly conflicting) goals.

Reasoning about strategies is a fundamental aspect of model checking SMG, which enables checking for the existence of a strategy that is able to optimize an objective expressed as a quantitative property in a logic called rPATL [22]. Properties written in rPATL can state that a coalition of players has a strategy which can ensure that the probability of an event's occurrence or an expected reward measure meet some threshold. Moreover, reward-extended versions of the rPATL enable the quantification of the maximum and minimum accrued reward that can be guaranteed by players in a coalition, independently of the strategies followed by the rest of players. Model checking of rPATL properties supports optimal strategy synthesis for a given property.

C. Human Involvement in Self-Adaptation

Self-adaptive systems were developed to autonomously adapt to changing circumstances. Dynamics (which results from the occurrence of certain events or changes in the environment state or target system state) is periodically *monitored* with a set of sensors. Given these sensor readings, the *Analyzer* performs an analysis of available actions and their potential impact on the satisfaction of system goals based on the information available. The *Planner* plans corresponding adaptation decisions to be enacted by the *Executor*.

The different activities in the feedback loop can benefit from human involvement in a variety of ways [7], [8]: *Monitors* can receive information from humans (acting as sophisticated sensors) that would be otherwise difficult to automatically monitor or analyze (e.g., humans can indicate whether there is an ongoing anomaly based on context information that is not captured by the models included in the knowledge base). *Planners* can incorporate inputs (e.g., recommendations, validation) into the decision-making process from application domain experts who can have additional insight about the best way of adapting the system. Execution can employ humans as system-level actuators to execute adaptations when changes to the system cannot be fully automated, or as a fallback mechanism. Beyond that, the role of the human can be supervisory, observing the activities carried out by the system and determining whether they are *appropriate* or potentially *erroneous* (e.g., likely to lead the system into an unsafe state, or degrade the satisfaction of system goals).

To capture the attributes of human operators that might affect human involvement with the system, the OWC [18], [6] model has been employed and categorizes human attributes into Opportunity, Willingness, and Capability. Specifically, capability captures the ability to carry out a particular task, which is usually determined by fixed attributes of the human actor, such as training level. OWC model is an abstract and customizable model that can be instantiated in different ways. As such, OWC has been used in several prior works [11], [6], [4], [7] that study human involvement in self-adaptation. However, human operators are very complicated and the simplified modeling of human attributes in previous instantiations of OWC may not match reality. Transient factors such as attention, which is not explicitly considered in OWC and varies in relatively short periods of time, can have a remarkable impact on the effectiveness of human operators when performing tasks. Thus, OWC that defines three broad categories needs to be further refined to deal with attention management.

D. Attention Management and Context Awareness

Attention is a neurobiological conception related to the capacity for information processing. In cognitive science, human attention can be divided into the following four categories [23]:

- *Sustained attention* indicates the ability to remain highly alert and focus on one specific task for a continuous period of time. Examples may include reading a book, or watching a video.
- *Divided attention* describes the ability to attend to two or more sources of information simultaneously, often referred to as multi-tasking. Examples include listening in a meeting while checking email.
- *Selective attention* shows the ability to select from various sources that are present and to focus on only one of them while the others are intentionally blocked out to some degree. For instance, a person can selectively choose to focus on the voice of the person she is talking to in a noisy room.
- *Alternating attention* denotes the ability to change the focus of attention and switch between different tasks. A typical example is preparing the different ingredients at different times to prepare a complicated recipe.

All these categories can be explained by the definition of attention, where e.g., alternating attention means that the information contained in the capacity of attention will change with tasks.

Human attention is a valuable but limited resource [24], and should be carefully managed when designing human-involved systems. Gil et al. [11] identify three design principles to design human-involved systems and achieve seamless and solid participation, especially when considering situations where human mental resources matter. These principles include: 1) Complement functionality with human involvement – humans are integrated in a close collaboration with the system to assist in the execution of certain tasks; 2) Achieve understandability – humans must understand how the system operates and what is happening at the current moment or even the moment before to trust the system; and 3) Manage user attention – the system must perform effective actions in order to mentally get humans involved.

A suitable level of attention is required for operators in different roles to perform a task. This is partially highlighted by Kahneman, who introduces two systems of thinking [25]. The first one (System 1) allows us to think rapidly without putting forth much mental effort, and is usually composed of initial impressions, reactions, and emotions toward an idea, person, or event. The other one (System 2) demands a higher level of mental effort and attention to analyze complex ideas. In most scenarios, the execution of nontrivial human tasks requires employing System 2 to some extent. Human attention processes also play a major role in the optimization of human-robot interaction (HRI) systems [13], such as the teleoperation requiring a great deal of direct human attention [26]. To this end, human attention should be considered as an explicit factor to calibrate human attributes modeling in human-involved adaptive systems.

It has been observed that when a human is unexpectedly interrupted to perform a task without much attention, they not only work less efficiently but also make more mistakes [27]. Apart from attention management, comprehending the context of a task before its execution would also help human operators prepare and adjust the direction of their efforts, thus reducing inefficiency and avoiding potential errors [28]. This has been illustrated in Crew Resource Management in air carriers where pilots could make disastrous mistakes just because the context of the task is temporarily collected without sufficient comprehension [29]. Thus, comprehending the task context also needs to be considered in a task preparation. The term *context* is a concept that spans many different domains, lacking a universally acknowledged definition and often defined in the way that is most suitable for a particular domain [14]. In this work, we adopt the definition of the context for a human task characterizing all related information with which a task can be completed more efficiently with less mistakes [14], [15]. Context awareness of human involvement in psychological studies has been investigated in several application areas, such as air traffic control, driver analysis, and military operations [13].

Attention management is about how much mental concentration an operator has on a task, whereas context awareness is about how much understanding the operator has of that task for efficient task performing. Preparatory notification is a proactive tactic we are using in this work for both attention management and context awareness to provide timely preparation for operators so that they can best complete the tasks when required. However, The system should also avoid disturbing or overwhelming operators with actions that require too much attention and context comprehension, which might lead to undesirable results or bad user experiences [11] by wasting their vigour and time. Hence, preparatory notification as a tactic requires formal reasoning in accordance with the level human involvement required, in the right time and in the right form. Although prior work has emphasized the importance of attention management and context awareness, none of them, to the best of our knowledge, has proposed to quantitatively model the influence of notification on attention and task context, and no one has reasoned about the usage of preparatory notification.

Our previous work [6], [4], [8], [7] applies an analysis technique based on model checking of SMGs to reason about human involvement without any explicit consideration about the role of attention or context complexity of a task in such involvement. In this work, we build on this analysis technique, which is particularly suitable for our purposes since it enables reasoning quantitatively under uncertainty about "when" and in "what" form preparatory tactics should be provided. In addition, prior work has investigated the role of explanation as a mechanism to improve the understanding of human operators and whether to provide explanation based on the trade-off [8], [7]. Although explanation, which could be regarded as the content of preparatory notification, has been adopted to allow operators to efficiently comprehend the context, prior work has not considered human attention as an explicit factor to refine capability in OWC model nor reasoned about when to notify human operators in advance.

III. Preparatory Notification: Formal Framework

Preparation for an operator is essential to efficiently perform a task. If operators are too immersed in the task at hand, the sudden arrival of a new task request from the system will require some time to divert their attention to the new task [27]. Such delays due to the shift of attention lead to a higher task latency. However, the speed of attentional shift varies for different tasks [30]. In critical scenarios like semi-autonomous driving, driver's attentional shift can be instant, taking over the control from the automatic system following a notification in an emergency. Even if a considerable amount of mental effort is put into the control, limited time for the operator assimilating the driving context information (e.g., road shape, position and trajectory of other cars) many still result in a catastrophic accident. Thus, these two activities – attention management and context awareness – together affect human task effectiveness, further influencing system utility. Preparatory notification can be used as a tactic to simultaneously manage human attention and allow operator to comprehend the context. To this end, deciding when and how to use preparatory notification has to take these two activities and their relation into account so that operators can complete a certain task effectively and in a timely manner.

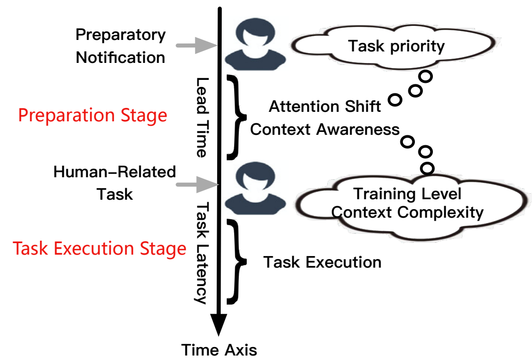

Fig. 1: Time Axis for Human Involvement.

Fig. 1 shows a high level view of the time axis for human-involved adaptation strategies. Our preparatory framework will proactively notify the operator (the first stage) before assigning the task (the second stage), which is a typical example of proactive self-adaptation for human-involved systems taking into account an arbitrary look-ahead horizon [17], [31], [16], [32]. *Lead Time* denotes the interval between the instant in which the preparatory notification is received by the operator and the start of task execution. The optimal lead time will be the minimum time required for an operator to make a full preparation (i.e., understanding the task context with a required level of attention). The task execution latency would be minimal with a full preparation, given the fixed training level of an operator in a short period. Figuring out the optimal lead time is particularly important: 1) when the time between the instant in which the need of a human-involved task is predicted and the start of that task is more than enough (i.e., greater than the optimal lead time), notification shall be delayed to minimize the disruption of the task execution at hand; 2) if it is insufficient, notification might be issued in a more urgent form such as sounding an alarm to accelerate human preparation.

In the remainder of this section, we elaborate on the modelling of how attentional shift is affected by task priority and how context awareness is influenced by task context complexity and operator training level.

a) Task Priority Affecting Attentional Shift: The speed of attentional shift is greatly influenced by the type [30] and urgency [33] of the forthcoming task as illustrated in empirical studies from the field of psychology. We combine the above two factors by introducing "task priority" where high priority denotes a type of important task with high urgency. Fig. 2 illustrates two curves that exemplify how human attention diverts from the task at hand to the forthcoming task for different priorities, with the assumption of no attention at the beginning of receiving notification and full attention required for the new one. The left plot depicts a quick startup for a high priority task, as operators promote their attention to its maximum capacity in a very short time. An example of such situations might be an emergency in semi-autonomous driving scenarios with instant human concentration required to mitigate life-threatening dangers. The right plot describes a slow startup where a human operator takes some time to gradually focus on a task. This is probably because the operator is very focused on the task at hand while the forthcoming one has comparatively low priority. Note that task priority is a relative concept, depending on the relative importance of the task at hand and the forthcoming one. Moreover, different forms of preparatory notification for the upcoming task can be employed to change its priority. For example, sounding an alarm or sending a text message might be associated with different priorities and also affect differently operator's attentional shifts.

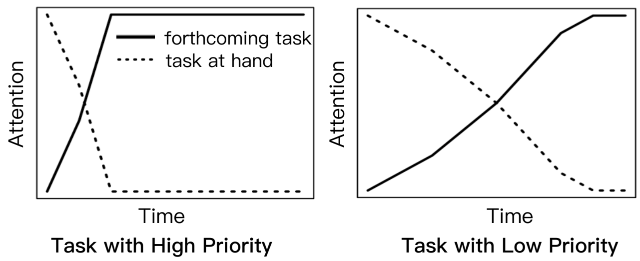

Fig. 2: Modeling on Attentional Shift.

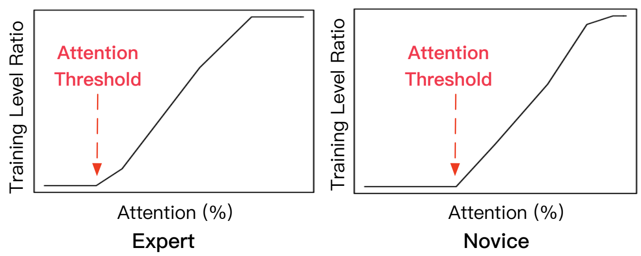

Fig. 3: Modeling on How Attention Affects Capability.

b) Training Level and Context Complexity Affecting Context Awareness: Context complexity denotes the amount of information required to perform a task in an efficient manner. Context awareness is the ability to comprehend and use context information to the benefit of performing a task. Thus, time to achieve context awareness is proportional to the complexity of a task's context. Operator training level, which is typically considered as one of the main capability factors in OWC, is developed over a long period of time and contributes to the ability of assimilating context information faster. Time to achieve context awareness is intuitively inversely proportional to the training level.

c) Attention Affecting Context Awareness: Capability, which is typically determined by the training level in prior work [7], [6], is also affected by attention. It is intuitive that when an operator is not paying much attention to a task, he cannot efficiently comprehend the context even with a high training level (more details can be referred to the relationship between attention and comprehension [34], [35]). Thus, we introduce *training level ratio* referring to the ratio of training level that can be exerted under a certain attention level. For example, a 60% attention level for a reading comprehension task may reduce efficiency, resulting in an actual capacity to process, e.g., one page per minute, for a person who is able to process two pages per minute with a 100% attention level (i.e., training level ratio 0.5). Fig. 3 illustrates how the attention on a task can affect this ratio, thus influencing capability. In the left plot, under the attention threshold, the operator is not able to logically process the context information no matter how well trained she is. With more attention, she can gradually increase her potential to understand the context. However, attention threshold to logically process the information and attention-ratio curves could be very different for each person. For example, a novice with less training might have a higher attention threshold as shown in the right plot. This phenomenon can be partially explained by the model proposed by Kahneman [25] where System 1 is fast and intuitive while System 2 is slower, deliberative and logical, requiring more attention. An expert has more experience in comprehending and performing a task, thus some part might be handled by System 1. On the contrary, a novice may need to employ System 2 to a greater extent to complete this task. As a result, his attention threshold is usually higher than that of an expert and will typically have a lower training level ratio even with the same attention level. Capability to comprehend the context is then determined by both the training level and the attention which affects *training level ratio* (i.e., *capability* = *training level* $\times$ *ratio*)[1].

To this end, we take three factors into account to synthesize adaptation strategy involving preparatory notification as a tactic in human preparation: 1) task priority affecting attentional shifts which in turn affects the capability to understand the context; 2) context complexity denoting the amount of context information of the task; 3) training level; by referring to the factors affecting driver's task switching behavior proposed by Lee et al. [36]. Optimal lead time of preparatory notification will be figured out so that operator will be informed with the right time and form of the notification to make proper preparations.

IV. Reasoning about Preparatory Notification for Human-Involved Adaptation

Deciding when preparatory notification should be provided and in what form is not trivial because its influence on human operator behavior can be affected by factors such as task priority, task context complexity, and training level. These factors build as additional sources of uncertainty affecting the behavior of the self-adaptive system [37]. Moreover, the extent to which the system places demand on the preparation of a new task might reduce the operator's ability to concentrate on the tasks at hand. To notify the operator in accordance with the level of human involvement required and at the right time, we introduce attention as an explicit factor in the model of human operator attributes, refining the categories of capability in OWC. Concretely, we want to answer two questions: **(Q1)** How can the outcome of adaptation with preparatory notification be predicted for human-involved adaptive systems? and

[1]Although we exploit attention as a factor affecting capability in OWC model in this work, attention might contribute to opportunity as mental availability (i.e., a required level of attention) is a complement to physical availability

(Q2) How can a self-adaptive system determine when and in what form preparatory notification as a tactic should be provided to a human operator in a given situation?

The key idea of our approach to enable automated reasoning about preparatory notification is to: (i) consider it as one of the possible actions or *tactics* that the system can enact as part of an adaptation strategy, and (ii) leave the choice of using such preparatory notification under-specified in the SMG model (encoded as a non-deterministic choice). The probabilistic model checker PRISM-games [38] is then used to analyze the model, resolving the nondeterminism to produce a strategy that maximizes the expected system utility by selecting when notification should be given and in what form. Our preparatory framework can be easily extended to any human-involved scenarios by customizing the part annotated with "human task execution" on the task execution stage for human operator module while requiring the retention of human preparation modeling (possibly intertwined with other module, e.g., controller).

Our motivating example is modeled as a SMG with four players, illustrated in Listing 1. Player *env* is in control of all the (asynchronous) actions that the environment can take out of system's control (including controller, operator and robot). Player *conl* specifies the actions controlled by the controller as well as *preparatory_tactic* and *human_related_tactic*, representing the tactic of preparatory notification and human-involved task respectively. Player robot's behavior is encoded in the process *rob* while player *ope* controls the actions that belong to the operator. The global variable *turn* (line 6) is used to explicitly encode alternating turns between the environment, controller, operator, and robot players. The following subsections present the models of the four players.

```
1  player env environment endplayer
2  player conl
         controller, [prepartory_tactic], [human_related_tactic] endplayer
3  player rob robot, [delivery_almost_done], [delivery_done] endplayer
4  player ope operator, [deliver_start], [wasting] endplayer

5  const ENVT=0; const CONL=1; const OPER =2; const ROBT = 3;
6  global turn:[ENVT..ROBT] init ENVT;
```

Listing 1: Player Definition.

```
1  module environment
2      t : [0..MAX_TIME] init 0;
3      [        ] (turn=ENVT) & (t<MAX_TIME) -> (t'=t+1) & (turn'=CONL);
4  endmodule
```

Listing 2: Environment Model.

A. Environment Model

The environment process *env* (Listing 2) models the potential evolution of environment variables. For simplicity, we assume our environment model only keeps track of time, although additional behavior controlling other elements (e.g, distractions) can be encoded (please refer to [31] for further details illustrating the modeling of adversarial environments in turn-based SMGs). Variable *t* (line 2) keeps track of execution time (the time frame for the system's execution is determined by [0, *MAX_TIME*]). During its turn, the environment checks that the end of the time frame for the execution has not been reached yet, and if that is the case, it increments the value of *t* one unit, yielding the turn to the controller player (i.e., turn'=CONL in line 3).

B. Robot Model

Listing 3 shows the encoding of behaviors corresponding to the robot in the collaborative delivery task. This module simply models the progress of delivery planned by the human operator. Variable *lead_time* (line 1) represents how much in advance before the start of the human task (i.e., the end of the current delivery in this scenario) the controller should inform the human operator. Variable *delivery_state* (line 3) is the counter used to keep track of the latency of delivery (i.e., the time needed for a robot to automatically deliver the good). Moreover, the module includes commands that model the progress of delivery as updates on its variables. In particular, there are four different commands for delivery in the module.

- Delivery trigger (line 4). Delivery triggers (synchronisation action *deliver_start* is initiated by the human operator) when the latency counter for delivery is zero, meaning that delivery is not being executed. As a consequence, the latency counter is activated (i.e., *deliver_state'=1*);
- Delivery latency counter update (line 5). If the delivery counter is active, but still has not reached the delivery's latency value, nor reached the time to proactively inform the human operator (i.e., *delivery_latency - lead_time*), the counter is incremented one unit;
- Delivery close to completion (line 6). If the counter is still active and reaching the time *lead_time* time units before completion, the robot will send a message *delivery_almost_done* to the controller;
- Delivery completion (line 7). When the delivery's latency counter expires, the latency counter is reset and the synchronization action *task_completion* will be triggered to inform the controller.

Every command in this module, except the one with a synchronized action (line 4) which is triggered by the operator, includes a predicate in the guard to ensure that the command is triggered only during the robot player's turn (*turn=ROBT*), and an additional update in the post state that yields the turn to the environment player (*turn'=ENVT*). Moreover, an additional command (line 8) lets the process progress without any variable updates when the counter state for the delivery is inactive.

C. Controller Model

Listing 4 shows the encoding of behaviors corresponding to the controller. Variables *prepOpe* and *planOpe* indicate the need of tactic execution corresponding to the preparatory notification tactic and human-involved task tactic (i.e., operator delivery plan generation in our scenario), respectively (as shown in Fig. 1). Once the controller receives the message *delivery_almost_done* from the robot (line 4) and if the

```
1  const lead_time; //human preparation
2  module robot
3     delivery_state : [0..delivery_latency+1] init 0;
4     [deliver_start] delivery_state = 0 -> (delivery_state'=1);
5     [       ] turn=ROBT & delivery_state!=0 &
            delivery_state <= delivery_latency &
            delivery_state != delivery_latency - lead_time
        -> (delivery_state'= delivery_state +1) & (turn'=ENVT);
6     [ delivery_almost_done ] turn=ROBT & delivery_state!=0 &
            delivery_state <= delivery_latency &
            delivery_state = delivery_latency - lead_time
        -> (delivery_state'= delivery_state+1) & (turn'=ENVT);
7     [task_complete] turn=ROBT & delivery_state= delivery_latency +1
        -> (delivery_state'= 0) & (turn'=ENVT);              Robot Delivery
8     [       ] turn=ROBT & delivery_state = 0 -> (turn'=ENVT);
9  endmodule
```

Listing 3: Robot Model.

```
1  module controller
2     prepOpe : bool init false;  //human preparation
3     planOpe : bool init false;  //human task execution
4     [delivery_almost_done] prepOpe = false -> (prepOpe'= true);
5     [preparatory_tactic] turn = CONL  & ! planOpe & prepOpe
          -> (prepOpe'= false) & (turn'=OPER);          Preparation Stage
6     [delivery_done] planOpe = false -> (planOpe'= true);
7     [human_related_tactic] turn = CONL & planOpe
          -> (planOpe'= false) & (turn'=OPER);       Task Execution Stage
8     [       ] turn=CONL & prepOpe = false & planOpe = false
          -> (turn'=OPER);
9  endmodule
```

Listing 4: Controller Model.

controller does not need to assign an operator to generate a delivery plan yet (i.e., *!planOpe* in line 5), it will try to notify the operator via synchronization on *preparatory_tactic* to attract the operator's attention and allow her to prepare on the preparation stage. On the task execution stage, the duty of controller is to ask the operator to generate a new delivery plan (line 6) once the controller receives the task completion message from the robot (line 7). The command on line 8 lets the process progress without any variable updates when *prepOpe* and *planOpe* are inactive.

D. Operator Model

The specification of the human module is shown in Listing 5. Variable *receive* (line 7) indicates whether the operator has received the preparatory notification while *shift_time* (line 8) keeps track of how long ago the operator has received such notification. Human attention, ranging from 0 to 100, is determined by *shift_time* and a typical relation is shown in formula (line 1) (i.e., attentional shift for task with high priority in left plot of Fig. 2). Note that attentional shift can be encoded in different formula with changing task priority, which are ignored in this description for the sake of clarity. Variable *accrued_context* (line 9) describes the accrued context information an operator already assimilated after receiving the notification. Variable *plan_state* (line 10) is the counter used to keep track of the latency of delivery plan making (i.e., human-related task on the second stage in framework 1).

On the preparation stage, the first command (line 11), which synchronizes with action *preparatory_tactic* from the Robot module, triggers when the counter for *shift_time* is zero, meaning that preparation has not started yet. As a consequence, the *receive* is set to true and *shift_time* is activated with probability (i.e., *prResponse*) as operator may ignore the notice message, leading to a probabilitisc behavior with probability (i.e., 1-*prResponse*) not receiving the notification.

Once the operator receives the notification and while the accrued context is less than the task complexity (i.e., task context awareness has not been fully achieved), *shift_time* will be incremented by one, and this in turn increments human attention and the accrued context (line 13). The capability to understand the context in a time unit (i.e., formula *unit_context* in line 5) is calculated by the training level (i.e., *training_level_unit*) multiplied by the training level ratio, which is affected by the operator's current attention level. A typical ratio is defined (line 2) by an explicit set of value pairs (with intermediate points linearly interpolated in line 2) as shown in the left plot Fig. 3. The command on line 12 denotes the situation in which the operator has already achieved context awareness (i.e., *accrued_context ¿= context_complexity*) but has not started performing the task yet. If context awareness is achieved way before the robot completes its delivery, there is an economic cost (*wasting*, captured in the reward structure described in Sec. IV-E) that corresponds to operator idle time that could have been spent performing other tasks assigned by the system.

On the task execution stage, the commands in lines 14 to 17 model the effect of human-involved task performing (delivery plan making in this scenario). These four commands are modeled:

- Planning trigger (line 14). Delivery planning triggers (synchronisation action *human_related_plan* is initiated by controller) when the counter for planning is zero, meaning that delivery planning is not being executed. As a consequence, the planning counter is activated (i.e., *plan_state'=1*);
- Follow up of preparation (line 15). If the planning counter is active, but the preparation has not yet finished (i.e., *accrued_context ¡ context_complexity*), the command update variable *shift_time* with one increment and *accrued_context* according to the encoding of the impact of context comprehension based on training level and ratio, leading to a higher task latency than expected (i.e., planning time plus additional time to comprehend the context).
- Planning counter update (line 16). If the context comprehension is done and the planning counter is still active, the planning counter is incremented one unit, representing the progress of plan making;
- Planning completion (line 17). When the planning counter expires, the planning counter and accrued context are reset. Variable *shift_time* is reset, denoting that the human operator is free to complete other tasks. Synchronization action *delivery_start* will be triggered to inform the robot to start delivering.

```
1  formula attention =  //human preparation
          (shift_time <= 0 ? 0 : 0)
          +(shift_time = 1 ? 40 : 0)
          +(shift_time >= 2 ? 100 : 0);
2  formula ratio =      //human preparation
          (attention<20? 0:0)
          +(attention>=20 & attention<30 ? attention-20:0)
          +(attention>=30 & attention<40 ? 10 + 20*(attention-30)/10:0)
          +(attention>=40 & attention<60 ? 30 + 40*(attention-40)/20:0)
          +(attention>=60 & attention<80 ? 70 + 30*(attention-60)/20:0)
          +(attention>=80 ? 100:0);

3  const context_complexity; //human preparation
4  const training_level;     //human preparation
5  formula unit_context = training_level * (ratio/100); //human preparation

6  module operator
7    receive : bool init false;  //human preparation
8    shift_time : [0..lead_time+1] init 0; //human preparation
9    accrued_context: [0..context_complexity+1] init 0; //human preparation
10   plan_state : [0..plan_latency+1] init 0; //human task execution
                                                        Preparation Stage
11   [preparatory_tactic] shift_time=0
       -> prResponse: (receive'= true) & (shift_time'=1)
             //operator receives the notification
          1-prResponse: (receive'= false) & (shift_time'=0);
             //operator misses the notification
12   [wasting] turn = OPER & shift_time>=1 & receive &
                accrued_context >= context_complexity
       -> (shift_time'= shift_time +1) & (turn'=ROBT) ;

13   [       ] turn = OPER & shift_time>=1 & receive &
                accrued_context < context_complexity
       -> (shift_time'=shift_time+1) & (turn'=ROBT) &
          (accrued_context' =
           min(context_complexity , accrued_context+floor(unit_context)));
                                                        Task Execution Stage
14   [human_related_tactic] plan_state=0
       -> (plan_state'=1) & (receive'= false);
15   [       ] turn = OPER & (plan_state!=0) &
           (accrued_context <context_complexity)
       -> (turn'=ROBT) & (shift_time'= shift_time +1) &
           (accrued_context' =
           min(context_complexity , accrued_context+floor(unit_context)));
16   [       ] turn = OPER & plan_state!=0 & plan_state<=plan_latency &
           accrued_context>= context_complexity
       -> (plan_state'=plan_state+1) & (turn'=ROBT);
17   [deliver_start] turn = OPER & plan_state=planning_time+1
       -> (plan_state'= 0) & (accrued_context'=0) &
           (turn'=ROBT) & (shift_time'=0);

18   [       ] turn = OPER & plan_state=0 & shift_time = 0 -> (turn'=ROBT);
19 endmodule
```

Listing 5: Operator Model.

Every command in this module, except the two with synchronized actions (line 11 and 14) initialized by the operator, include a predicate in the guard to ensure that the command is triggered only during the operator player's turn (*turn=OPER*), and an additional update in the post state that yields the turn to the environment player (*turn'=ROBT*). Moreover, an additional command (line 18) lets the process progress without any variable updates when none of the latency periods for preparation or task are active. Note that in our model, we assume human operator is willing to perform the tasks assigned by the system (possibly due to payment).

```
rewards "rGR"
  [delivery_done] true: reward;
endrewards

rewards "rGC"
  [wasting] true: cost;
endrewards
```

Listing 6: Utility Reward Structure

E. Utility Profile

Utility functions are encoded using reward structures that enable the quantification of the utility of a given game state. Listing 6 illustrates 1) how a reward structure labeled with rGR can be defined to assign a reward value for any state with successful delivery (i.e., synchronization *delivery_done*), which has to be collaborated by human operator planning and robot delivering, and 2) how a cost structure can be defined with a value if an human operator is idle, thus wasting his working time (i.e., synchronization *wasting*). Note that the cost is independently defined with a positive value because PRISM does not allow negative rewards.

V. Analysis Results

In this section, we illustrate how our approach can produce decisions about when and in what form to proactively provide preparatory notification in our motivating scenario. In particular, we exploit our SMG model of human-involved adaptation (described in Section IV) to determine: (i) the optimal *lead time* to initiate notification as a preparatory tactic in adaptation, and (ii) the form under which notification as a tactic could optimize the expected outcome of human involvement within a limited time horizon.

To explore our scenario, we statically analyze a discretized region of the state space of our problem. Each state of the discrete set requires two runs of the model checker per adaptation alternative (i.e., different modes of notification). Each run of the model checker quantifies one of the two terms of the utility formula, which is calculated as the subtraction of the cost due to operator idle time c_{idle} from the utility accrued for successful deliveries u_{del}. That is:

$$u_{mau} \equiv u_{del} - c_{idle}$$

where the terms of the expression are encoded in the following rPATL temporal logic formulas:

$$u_{del} \equiv \langle\langle \text{ROBT, OPER, CONT} \rangle\rangle\ R^{rGU}_{max=?}[F\ end]$$
$$c_{idle} \equiv \langle\langle \text{ROBT, OPER, CONT} \rangle\rangle\ R^{rGC}_{max=?}[F\ end]$$

In these properties, operator $<<\ >>$ contains a coalition of players, "rGR" and "rGC" are the reward structures specified in listing 6, and *end* is a predicate that indicates the end of the adaptation decision-execution period. To ensure the consistency of our results, the quantification of the second property c_{idle} in every point of the discretized space is done subject to the fixed policy computed during the first run of the model checker for u_{del} in that same point of the discrete space.

Once we have quantified the utility of every alternative in each state of the discrete space S, we rank the different adaptation alternatives (i.e., various notification formats) and select the one that maximizes the expected accrued utility. Hence the selected notification tactic for a state $s \in S$ from a set of notification tactic alternatives Γ can be determined according to: $\gamma_{\uparrow}(s, \Gamma) \triangleq arg \max_{\gamma \in \Gamma} u_{mau}\ (s, \gamma)$ where $u_{mau}(s, \gamma)$ is the value of property u_{mau} evaluated in a model instantiated with with an initial state s and the adaptation logic of tactic γ.

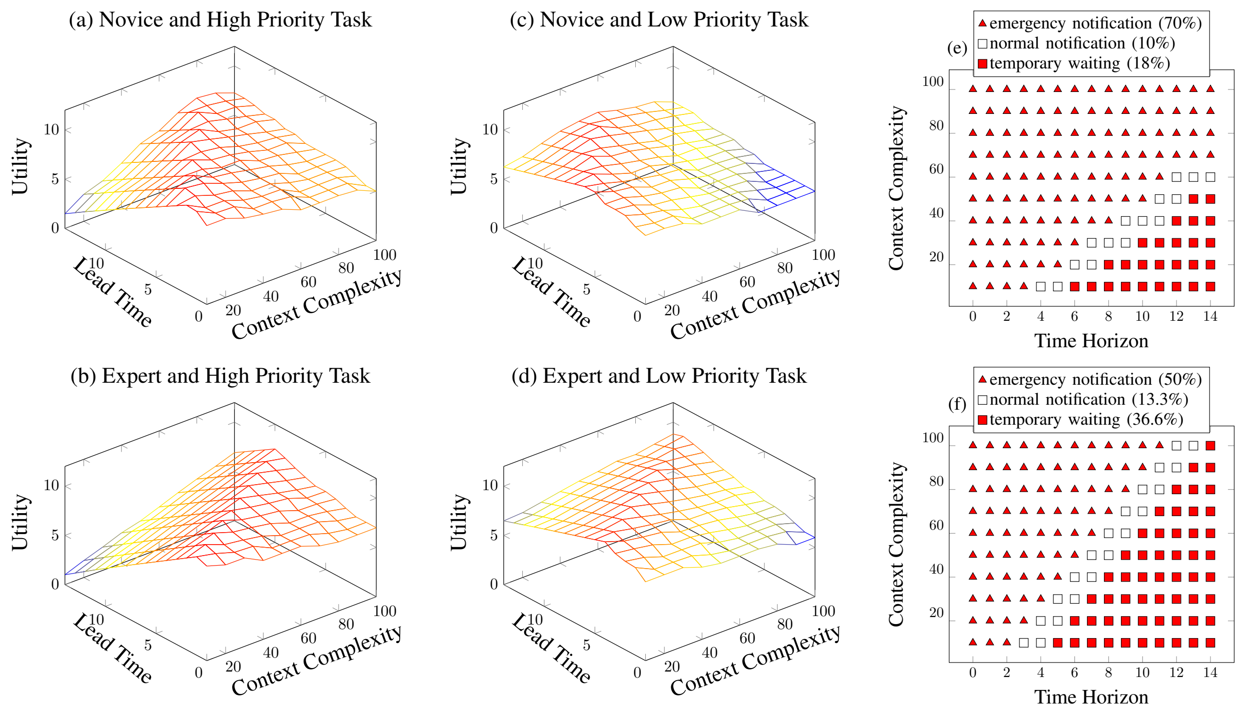

Fig. 4: Analysis Results

To set up the experiment, we consider two types of operator (expert and novice) with different training levels. The relationship between how attention will affect the capability for these two operators is shown in the plots shown on Fig. 3. We also consider two types of tasks: (task with high priority and with low priority), corresponding to different ways in which attentional shift happens (illustrated in Fig. 2).

Plots 4 (a-d) depict utility analysis results for different adaptation conditions (i.e., various *lead time* for preparatory notification and context complexity for a task). The experiments employ fixed values for unit comprehension (i.e., training level) of 6 for the novice and 10 for the expert, and planning time 5 for the novice and 3 for the expert. The overall time frame for the experimentation is 200 minutes. Note that the planning time is grounded on the assumption that human operator has completed the preparation with required attention level (i.e., 100 in this experimentation) and context awareness (i.e., assimilating all task context information). The probability of operator receiving the notification is set as 0.85. The reward for each delivery is 1 while the cost for each idle time unit is 0.08 for an expert and 0.04 for a novice.

Plot 4 (a) illustrates the maximum accrued utility the system together with human operator and robot can achieve for varying context complexity and lead time. We can observe that when the context of the task is very complicated (i.e., context complexity near 100), the utility increases progressively with the increasing lead time. Even though the speed of attentional shift is fast for the task with high priority, more lead time yields better results for tasks with complicated context since the novice can understand the context as much as possible so that he can make a plan without much delay. In contrast, we can observe how utility increases first and then reduces when the context complexity is intermediate or less (less than 80 as shown in this plot). The optimal lead time (i.e, the state with the maximum accrued utility) generally increases with the increasing context complexity. Longer lead time does not always bring better results and may weaken system utility. This is due to the cost of waiting after the operator has been able to understand task context by employing only a fraction of the *lead time*, while the rest is just wasted time.

Plot 4 (c) depicts the utility for a task whose priority is comparatively low. In particular, we observe that this plot is flatter than (a). The trend of utility going up first and then going down happens only when the complexity is less than 40. Otherwise, the utility is positively correlated with the lead time. The peak (i.e., maximum accrued utility with the optimal lead time) occurs with a higher lead time compared to (a) with the same context complexity. This is because attentional shift for a task with low priority is slower and managing human attention to a required level takes a longer time, demanding an earlier preparation to focus the operator's attention on the upcoming task.

Plot 4 (b) shows the utility for an expert who has higher training level to assimilate more context information per time unit. The optimal lead time for a given context complexity is in general less than that in plot (a). This is because a well-trained operator can act quickly with less preparation time in terms of fully achieving context awareness. The utility in the left part of plot (b) where complexity is near 0 while lead time near 15 is, however, less than the same area in plot (a). This is due to a higher cost as an expert could quickly finish his preparation thus wasting more time in waiting, which he could have used to complete his task at hand. Plot 4 (d) is analogous to (c), but with generally comparative smaller optimal lead time.

Plots 4 (e) and (f) present the results of tactic selection among *emergency notification*, *normal notification* and *temporary waiting*, given the strict time horizon constraints on human involvement. An emergency notification leads to faster attentional shift and thus quicker preparation, but it can lead to more cost due to longer waiting time especially when the time constraint is relaxed. The states in which *emergency notification* is chosen are indicated by a triangle, whereas the *normal notification* is indicated by a rectangle, among which mark filled with red color represents tactic of postponed normal notification (i.e., *temporary waiting*). Such tactic happens when the predicted future time horizon for task involvement is more than sufficient, exceeding the required preparation time. These two plots reveal that states with *emergency notification* are mainly distributed in the upper left. In particular, all states within 4 time horizon are marked with triangles in Plots 4 (e) with various context complexity. This is because the time is too limited for a novice to be fully prepared with a normal notification corresponding to the innate context complexity of upcoming task. An emergency notification allows the operator be more alert and shift his attention faster so that she can understand the context as much as possible within a time frame. When the time horizon is gradually relaxed, a human operator could naturally divert his attention and make preparation by following a normal notification instead of an urgent one. Also, a normal preparatory notification should be postponed if the future horizon is even greater than the optimal lead time for a task with less complicated context to reduce operator waiting. This can be observed in the lower right area with rectangle marks.

Plot (f) denotes the results of strategy selection for an expert. The proportion of choosing *emergency notification* is 50%, which is 20% less than for a novice. Besides, the states with *temporary waiting* are twice as much than those in plot (e). These results are also explained by the fact an expert is more likely to start preparations later (i.e., *temporary waiting*) or make orderly preparations (i.e., *normal notification*) without hurrying up (i.e., *emergency notification*) even within a limited time due to a high training level.

In summary, the results of analyzing operator-robot collaborative delivery scenario have shown that: (i) the outcome of adaptation with preparatory notification is predicted by automatic reasoning with SMG model (i.e., **(Q1)**) and preparatory notification (i.e., lead time > 0) can enhance the overall utility when involving human operators especially with a proper lead time, (ii) the optimal lead time (i.e., "when" this preparatory tactic should be used in **(Q2)**) to optimize system utility varies under different conditions: typically smaller for an expert with a high priority task whereas longer for a novice with low priority task, (iii) time interval constraints for an operator to start the task will lead to different strategies involving emergency notification or normal notification (i.e., "what" form this preparatory tactic should be adopted in **(Q2)**): more urgency with stricter time constraints.

VI. Conclusions and Discussion

Attention management and preparation have been considered as one fundamental principle to design human-involved self-adaptive system [11]. To this end, we presented an general framework based on probabilistic model checking to determine when and in what forms a notification should be used as an adaptation tactic to attract human attention and allow them to comprehend the context. Although one limitation of our evaluation is that we did not directly correlate our analytical results with actual systems through an empirical study, our findings are supported by and consistent with those obtained by Braithwaite et al [39], who conducted an empirical user study to determine if preparation promotes attention to the relations underlying concepts, increasing the benefits of learning task. Experimentation from Sukkerd et al [40] also supports our model, which assumes that awareness of a task situation by explanation increases the capability to cooperate with the system. Our previous work [7] where explanation as a potential tactic can be integrated within preparatory framework so that notification with context explanation can further facilitate task preparation and improve the overall system utility.

A second limitation is the lack of a direct procedure for obtaining the task priority nor the process of attentional shift. While this is an area for future work, the priority of a task might be determined by ranking the accrued system utility with probabilistic model checking if such task is performed [17], [7]. Attention level can be measured by posture measurement combined with neuroimaging techniques such as electroencephalogram (EEG) [41] or expression with a humanoid face [42] while attentional shift can be traced through saccadic eye movement and historical behaviors [43]. Another limitation, and also a topic for future work, is that context comprehension is sometimes intertwined with task performing and context complexity is not easy to quantify across different tasks nor training level across different operators. However, sensitivity analysis can be useful when they cannot be determined with precision but lies within a known range. Uncertain transition probabilities for operator missing the notification can be handled by quantitative verification with confidence intervals [44]. Besides, qualitative estimates of time for the operator to understand context information could also be explored [45].

Our initial investigation suggests a number of additional research directions, such as formulating the problem as a multi-objective optimization with Pareto-optimal solutions (in contrast with a single utility function which is a linear combination of different contributions, as in this paper); and finally, analyzing and planning strategies by explicitly considering preparatory notification and automated tactics as potentially concurrent adaptation tactics. This framework could be further adjusted and extended in any human-involved systems by customizing the human task execution part in the formal model. Scenarios with multiple operators involvement can also be considered in future work with additional task delegation module [3].

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. Lecture Notes in Computer Science, vol 5525. Springer, Berlin, Heidelberg*.

[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. Lecture Notes in Computer Science, vol 7475. Springer, Berlin, Heidelberg*.

[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. Cámara, G. A. Moreno, D. Garlan, and B. Schmerl, "Chapter 7 - evaluating trade-offs of human involvement in self-adaptive systems," in *Managing Trade-Offs in Adaptable Software Architectures*, I. Mistrik, N. Ali, R. Kazman, J. Grundy, and B. Schmerl, Eds. Boston: Morgan Kaufmann, 2017, pp. 155 – 180.

[7] N. Li, J. Cámara, D. Garlan, and B. R. Schmerl, "Reasoning about when to provide explanation for human-involved self-adaptive systems," in *IEEE International Conference on Autonomic Computing and Self-Organizing Systems, ACSOS 2020, Washington, DC, USA, August 17-21, 2020*, 2020, pp. 195–204.

[8] 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)*, 2020, to appear.

[9] M. Gil, P. Giner, and V. Pelechano, "Personalization for unobtrusive service interaction," *Pers. Ubiquitous Comput.*, vol. 16, no. 5, pp. 543–561, 2012.

[10] M. Gil and V. Pelechano, "Self-adaptive unobtrusive interactions of mobile computing systems," *J. Ambient Intell. Smart Environ.*, vol. 9, no. 6, pp. 659–688, 2017.

[11] M. Gil, M. Albert, J. Fons, and V. Pelechano, "Designing human-in-the-loop autonomous cyber-physical systems," *Int. J. Hum. Comput. Stud.*, vol. 130, pp. 21–39, 2019.

[12] E. Horvitz, C. M. Kadie, T. Paek, and D. Hovel, "Models of attention in computing and communication: from principles to applications," *Commun. ACM*, vol. 46, no. 3, pp. 52–59, 2003.

[13] A. Dini, C. Murko, S. Yahyanejad, U. H. Augsdörfer, M. W. Hofbaur, and L. Paletta, "Measurement and prediction of situation awareness in human-robot interaction based on a framework of probabilistic attention," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2017, Vancouver, BC, Canada, September 24-28, 2017*, 2017, pp. 4354–4361.

[14] P. Brézillon and A. J. Gonzalez, Eds., *Context in Computing - A Cross-Disciplinary Approach for Modeling the Real World*. Springer, 2014.

[15] *Context Management and Reasoning about Situations in Pervasive Computing*. Caulfield School of Information Technology, Monash University, Australia., 2006.

[16] 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.

[17] G. A. Moreno, J. Camara, D. Garlan, and B. R. Schmerl, "Flexible and efficient decision-making for proactive latency-aware self-adaptation," *ACM Trans. Auton. Adapt. Syst.*, vol. 13, no. 1, pp. 3:1–3:36, 2018.

[18] D. Eskins and W. H. Sanders, "The multiple-asymmetric-utility system model: A framework for modeling cyber-human systems," in *Eighth International Conference on Quantitative Evaluation of Systems, QEST 2011, Aachen, Germany, 5-8 September, 2011*, 2011, pp. 233–242.

[19] "Amazon's autonomous robots have started delivering packages in a new location: Southern california," https://www.washingtonpost.com/technology/2019/08/12/amazons-autonomous-robots-have-started-delivering-packages-new-location-southern-california/, accessed: 2019-08-13.

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

[21] A. Simaitis, "Automatic verification of competitive stochastic systems," Ph.D. dissertation, University of Oxford, UK, 2014.

[22] T. Chen and J. Lu, "Probabilistic alternating-time temporal logic and model checking algorithm," in *Fourth International Conference on Fuzzy Systems and Knowledge Discovery, FSKD 2007, 24-27 August 2007, Haikou, Hainan, China, Proceedings, Volume 2*, 2007, pp. 35–39.

[23] R. A. Barkley, "Behavioral inhibition, sustained attention, and executive functions: Constructing a unifying theory of adhd." *Psychological Bulletin*, Vol 121(1), Jan 1997, 65-94.

[24] Gibbs and W. Wayt, "Considerate computing," *Scientific American*, vol. 292, no. 1, pp. 54–61, 2005.

[25] D. Kahneman, "Thinking, fast and slow," *Farrar, Straus and Giroux;*, vol. 172, 01 2011.

[26] M. A. Goodrich and A. C. Schultz, "Human-robot interaction: A survey," *Found. Trends Hum. Comput. Interact.*, vol. 1, no. 3, pp. 203–275, 2007.

[27] J. Laarni, H. Karvonen, S. Pakarinen, and J. Torniainen, "Multitasking and interruption management in control room operator work during simulated accidents," in *Engineering Psychology and Cognitive Ergonomics - 13th International Conference, EPCE 2016, Held as Part of HCI International 2016, Toronto, ON, Canada, July 17-22, 2016, Proceedings*, ser. Lecture Notes in Computer Science, D. Harris, Ed., vol. 9736. Springer, 2016, pp. 301–310.

[28] A. F. Abdelrahman, A. Mitrevski, and P. G. Plöger, "Context-aware task execution using apprenticeship learning," in *2020 IEEE International Conference on Robotics and Automation, ICRA 2020, Paris, France, May 31 - August 31, 2020*. IEEE, 2020, pp. 1329–1335.

[29] A. Degani and E. L. Wiener, "Human factors of flight-deck checklists: The normal checklist," Work of the US Gov. Public Use Permitted., Tech. Rep. 19910017830, September 2013.

[30] K. Nakayama and M. Mackeben, "Sustained and transient components of focal visual attention," *Vision Research*, vol. 29, pp. 1631–1647, 1989.

[31] J. Cámara, G. A. Moreno, D. Garlan, and B. Schmerl, "Analyzing latency-aware self-adaptation using stochastic games and simulations," *ACM Trans. Auton. Adapt. Syst.*, vol. 10, no. 4, Jan. 2016.

[32] G. A. Moreno, J. Cámara, D. Garlan, and B. R. Schmerl, "Efficient decision-making under uncertainty for proactive self-adaptation," in *2016 IEEE International Conference on Autonomic Computing, ICAC 2016, Wuerzburg, Germany, July 17-22, 2016*, 2016, pp. 147–156.

[33] K. Anan, K. Fujiwara, C. Yaguchi, and N. Kiyota, "Effect of time pressure on attentional shift and anticipatory postural control during unilateral shoulder abduction reactions in an oddball-like paradigm," *Journal of physiological anthropology*, vol. 33, pp. 21–39, 2014.

[34] B. Jennings and A. Daniel., *Children's Understanding of Television: Research on Attention and Comprehension.*, ser. ISBN-0-12-138160-9. Academic Press,, 1983.

[35] M. Yildiz and E. Çetinkaya, "The relationship between good readers' attention, reading fluency and reading comprehension," *Universal Journal of Educational Research.*, vol. 5, no. 3, pp. 366–371, 2017.

[36] J. Y. Lee and J. D. Lee, "Modeling microstructure of drivers' task switching behavior," *Int. J. Hum. Comput. Stud.*, vol. 125, pp. 104–117, 2019.

[37] N. Esfahani and S. Malek, "Uncertainty in self-adaptive software systems," in *Software Engineering for Self-Adaptive Systems II - International Seminar, Dagstuhl Castle, Germany, October 24-29, 2010 Revised Selected and Invited Papers*, 2010, pp. 214–238.

[38] 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.

[39] D. W. Braithwaite and R. L. Goldstone, "Benefits of variation increase with preparation," in *Proceedings of the 36th Annual Meeting of the Cognitive Science Society, CogSci 2014, Quebec City, Canada, July 23-26, 2014*, P. Bello, M. Guarini, M. McShane, and B. Scassellati, Eds. cognitivesciencesociety.org, 2014.

[40] R. Sukkerd, R. G. Simmons, and D. Garlan, "Tradeoff-focused contrastive explanation for MDP planning," in *29th IEEE International Conference on Robot and Human Interactive Communication, RO-MAN*

2020, Naples, Italy, August 31 - September 4, 2020. IEEE, 2020, pp. 1041–1048.

[41] R. Furutani, Y. Seino, T. Tezuka, and T. Satoh, "Monitoring the level of attention by posture measurement and EEG," in *Proceedings of the 38th Annual Meeting of the Cognitive Science Society, Recogbizing and Representing Events, CogSci 2016, Philadelphia, PA, USA, August 10-13, 2016*, A. Papafragou, D. Grodner, D. Mirman, and J. C. Trueswell, Eds. cognitivesciencesociety.org, 2016.

[42] A. Bruce, I. R. Nourbakhsh, and R. G. Simmons, "The role of expressiveness and attention in human-robot interaction," in *Proceedings of the 2002 IEEE International Conference on Robotics and Automation, ICRA 2002, May 11-15, 2002, Washington, DC, USA*. IEEE, 2002, pp. 4138–4142.

[43] B. B., "Eye movements. in: V.s. ramachandran (ed.)," *The Encyclopedia of Human Behavior. Academic Press.*, vol. 2, pp. 160–166, 01 2012.

[44] R. Calinescu, C. Ghezzi, K. Johnson, M. Pezzè, Y. Rafiq, and G. Tamburrelli, "Formal verification with confidence intervals to establish quality of service properties of software systems," *IEEE Trans. Reliab.*, vol. 65, no. 1, pp. 107–125, 2016.

[45] 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.