Proceedings of the 27th IEEE International Symposium
on Robot and Human Interactive Communication,
Nanjing, China, August 27-31, 2018

Situation Awareness for Autonomous Agents

Nikolas Dahn[1], Stefan Fuchs[2], Horst-Michael Gross[1]

Abstract— **Situation Awareness is a prominent concept in the human factors community. It is used to analyze and eliminate common sources of human errors in complex tasks and has seen wide-spread use in many fields, such as aviation, health care or ergonomics. Humans who are situation aware are able to reliably generate competent performance, a skill that is also highly desired for other autonomous agents. Yet, the concept has seen only limited use in robotics. We attest this to a lack of clear definitions which would allow assessing an artificial agents capacity for Situation Awareness. Our major contribution is an application-agnostic definition of the terminology and the processes involved in acquiring Situation Awareness. By integrating our definitions into the perception-action-cycle we provide a connection to the agent's observable behavior. Our second major contribution is a way to estimate, whether an agent has lost Situation Awareness based on surprise. This measure can be used online and does not require explicit or implicit knowledge of the task. We evaluate our concept on a physical workspace built for abstract Human-Robot-Cooperation scenarios.**

I. INTRODUCTION

Intelligent agents have become an indispensable part of modern everyday life. Drones deliver packages, factory robots assemble and paint cars, and (somewhat) smart home devices play music or tell jokes on demand. Still, these systems are fundamentally limited in their capability to act within the real world. Delivery drones will casually drop their package right on your new ant colony, assembly robots will fervently weld on while the factory is on fire, and Alexa will happily play the Star Wars theme at your Star Trek evening. It is apparent that these systems lack the ability to take into account parts of the environment their designers did not consider.

This is especially detrimental in situations, where such a system has the potential of posing a risk to human safety. This becomes obvious when considering Human-Robot-Cooperation (HRC). Regarding the robot, there are two basic requirements for successful HRC: adequate robot hardware and intelligent control software [4]. As of today shared workspaces for humans and robots only exist in constrained research scenarios. To ensure the human workers' safety the robots are separated through spatial and/or temporal separation (e.g. cages). But even if the safety issues could be solved, actually being helpful to a human is notoriously hard. This can be attributed to a lack of software that is able to control a robot while taking human intentions into account. Thus, the potential benefits of shared workspaces stay unutilized.

In order to enable goal-driven, sensible behavior in unknown situations, an agent has to answer the following question: *which parts of the environment are actually relevant to me right now?* To find an answer the agent will usually build a partial model of the environment and the effects actions will have. In the presence of other agents, their intentions, beliefs and capabilities have to be included as well. Furthermore, if a goal requires cooperation, the participants have to synchronize their actions and beliefs. This is one of the reasons why communication plays a major role in teams [1] and is a major challenge in HRC.

In this work we present a novel approach to make the concept of Situation Awareness (SA) available for autonomous agents. SA plays a prominent role in human factors research and is aimed at identifying parts of the environment relevant to a problem. Depending on the definition it might describe a process or a product. It has been used to improve human performance in many areas, e.g. aviation. In a nutshell, having SA allows to find an apropriate answer to the question ”what should I do?”. However, we find that - since SA was designed with humans in mind - the definitions available are not directly applicable to a technical system. To alleviate this, we construct a general concept based on the work by Smith & Hancock [2] and integrate Dey's definition of Context Awareness [3] as well. Based on our concept, we also propose an online way to estimate whether an agent has SA by measuring their surprise. Our approach follows a top-down design, and thus is not bound to a specific application.

To evaluate our concept we have built a physical HRC setup with high task abstraction. To this end, we employ task graphs and placeholder actions in the form of button presses. This has the added benefit of keeping manipulation complexity and action recognition simple.

II. RELATED WORK

There exists plenty of research into how humans evaluate situations and make their decisions. One highly influential term in human factors research is the concept of Situation Awareness (SA). Coined in the 1980s, SA was initially used to optimize user interfaces and has seen adoption in a host of different domains, e.g. aviation and health care [5]. SA is a concept tailored to humans: it describes a complex set of mental capabilities, which enable beneficial decisions. It can be said that having SA leads to good decisions, while bad decisions can be a sign of lacking SA (the reverse is not necessarily true, i.e. even without SA good decisions can be made). Since its rise to fame, many definitions have been

[1] Neuroinformatics and Cognitive Robotics Lab, Ilmenau University of Technology, 98693 Ilmenau, Germany {nikolas.dahn, horst-michael.gross}@tu-ilmenau.de

[2] Honda Research Institute, 63073 Offenbach, Germany, stefan.fuchs@honda-ri.de

proposed [5]. While most works refer to the definition by Endsley [6], we base our approach on the definition by Smith and Hancock [2]. Although less known, this model has the advantage that it integrates with the well-known perception-action cycle, which we will expand upon as well. They define SA as follows:

> *[Situation awareness] is the capacity to direct consciousness to generate* competent performance *given a particular situation as it unfolds.*

This definition also shows why SA is interesting for autonomous agents in the first place: it is a means to evoke repeatable, optimal behavior. For more details on competence and performance we refer the reader to [2].

While some work exists where SA is applied in robotics, it is usually the designer who strives to become situation-aware and then transfers their model to the implementation. Naturally, since the system will lack the capability to exhibit this process, it will not be able to cope with situations not considered by the designer.

In robotics and human-computer-interaction, a frequently occuring and quite similar concept is Context Awareness. In fact, existing literature seems to suggest that these two terms are used for the same underlying concept, but in different contexts. Working on context-aware human-computer-interaction, Dey provides the following definition [3]:

> *Context is any information that can be used to characterise the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and application themselves.*

The author also defines a situation as *" a description of the states of relevant entities"*. We agree, but also note that these definitions fall short several steps by failing to provide a clear guidelines to base an implementation upon. They also don't provide an answer to the question of how an agent can achieve Context/Situation Awareness.

As mentioned above, SA has already been applied in robotics, albeit mostly as part of the design process (what does the robot need to know?).

In [7] the authors apply the concept of SA to analyze problems human operators had at the DARPA Robotics Challenge. Albeit robotics-related, this marks yet another application of SA to analyze human factors. The application to the participating robots stays unconsidered.

[8] identifies various capabilities robots would need in order to achieve SA. These include reasoning about unknowns, deliberative sensing and spatiotemporal reasoning. While this approach is not application-specific, the relation between the individual components is left open. This makes it difficult to derive a concrete system from it.

In [9] commonsense reasoning is proposed as a way to enable robots to achieve SA. The authors propose to use default assumptions to fill gaps in the robot's representation. This would allow the robot to fulfill tasks similar to already solvable ones. There is no indication in this paper of how SA itself can be acquired.

The Situation Aware FEar Learning (SAFEL) [10] model is a human-brain inspired system to classify and react to "aversive" and "safe" situations. The system is planned to be able to learn, which aspects of the environment are relevant (indicative of aversive situations). This would allow it to more easily achieve SA, improving the reliability of its behavior over time. Currently, the situation assessment module still seems to be missing.

In [11] the authors present a system, which extracts features and relations from its environment and connects these through an ontology. HRC is facilitated by explicitly taking the human's capabilities and even their perspective and beliefs into account. This is demonstrated in simple cooperative tasks like cleaning up a table or getting a bottle from a shelve. While the system appears to reach a high degree of agency, the lack of an overarching concept makes it difficult to transfer it to other contexts (e.g. is perspective taking always required?).

[12] proposes a framework for creating situation-aware systems using a layered structure for data abstraction. The framework heavily relies on experts, which have to provide situation decompositions, sensor placement and -abstraction as well as an application-specific ontology. The level of SA thus depends on the expertise of the system designer.

An application of this work can be found in [13], where a shared workspace for a human and a robot is presented. By applying the concept of SA the authors classify situations and identify relevant features. This information is then used to adapt the robot's behavior.

Instead of designing for specific applications, we wanted to go one step further and look into the underlying cognitive requirements. In the following section we present a novel conceptual framework to apply SA to robotics. Our approach is not tailored to a specific application and thus does not rely on expert knowledge. Instead we see expert knowledge as a means to fill gaps in a system's cognitive abilities to enable its use in a specific application.

III. DEFINING SITUATION AWARENESS

A. Definition

Although there are numerous definitions of SA [5], none of them are directly applicable in a technical context. This is due to the fact that at least the general meaning of SA is easily understood by humans without ever requiring a clear definition. This makes it about as tangible as intelligence or creativity. As an example, we found that most theories of SA lack a definition of what a situation actually is (e.g. [6][2][14][15]). For an implementation this is insufficient.

To provide a comprehensive definition of SA, we will first define our core terms. Since our definitions are compatible to those by Dey [3], our concept not only covers Situation Awareness, but also Context Awareness. These terms will be used as defined here throughout the rest of the paper.

1) Aspect: a logical statement describing part of the environment at a certain point in time. Aspects may describe properties (”A is yellow”) as well as relations (”A is on top of B”). What an aspect describes (”the color of A”) is distinct from its instance value (”yellow”).

2) Representation: the set of aspects an agent is aware of. We sometimes use the term ”internal representation” to underline that it’s usually not accessible from the outside.

3) Goal: a set of aspects desired by an agent. This may include constraints, e.g. a time by which the goal has to be reached or the maximum number of steps in the solution.

4) Solution: a sequence of actions, which change aspects in order to reach a goal.

5) Decision-Making-Process (DMP): generates a solution for a goal from the representation passed to it. Crucially, this involves a model of how aspects change and relate to predict future situations.

6) Relevant Aspects: an aspect is relevant if including it in the DMP changes the solution (selected actions, quality, confidence…).

7) Situation: a set of relevant aspects given a goal. For a situation-aware agent this will be a subset of the representation.

8) Evolution: given a goal, a situation is said to have evolved if at least one of its relevant aspects have changed.

9) Completeness: given a goal, a representation is complete if it includes all relevant aspects.

10) Minimal: given a goal, a representation is minimal if it includes only the relevant aspects.

The question remains how a robot (or any agent really) can acquire SA. Complementing the definition of SA by Smith and Hancock above, we state that:

> *Given a goal and a situation, an agent is situation-aware, if it can build a complete representation before the situation evolves.*

Akin to [2], we stress that SA only makes sense in relation to a goal. An agent also only ever has access to their own internal representation, so they can never know that they are situation-aware: there may always be a relevant aspect unknown to the agent. On the other hand, an agent may know that they are not situation-aware, e.g. by knowing that some relevant aspects are missing. That is, the agent cannot know that it knows everything, but it can know that it knows not enough. SA thus has to be assessed either from the outside or in hindsight.

We stated that to become situation-aware, an agent must be able to take into account all relevant aspects. Since the agent’s capabilities matter in this context, this also limits the goals for which an agent can achieve SA. While taking more relevant aspects into account usually improves the solution, missing just one relevant aspect may lead to complete failure. This implies that a solution’s quality is not directly linked to the number of relevant aspects correctly represented. As such, for us SA is binary: one either has it or one doesn’t. Saying that e.g. an agent is ”90% situation-aware” is an invalid statement. We found that this alleviates many problems when talking about SA and does not weaken the definition.

Smith and Hancock define SA as ”externally directed consciousness” [2]. The process pushing consciousness towards an agent’s inner state is called introspection and creates self-awareness. In a similar fashion, the process pushing consciousness outwards could be called extrospection and creates Situation Awareness. This is depicted in Fig. 1a.

The terms defined above integrate with the well-known perception-action-cycle, as shown in Fig. 1b. A similar application to this cycle has been shown in [2], although we employ a more detailed configuration. Our SA-cycle includes four nodes internal to the agent: perception of the environment, cognition of relevant aspects, decision making, and execution and monitoring of actions. In addition, we can determine the types of data passed between these nodes using the terms defined above. The perception node applies a filter to the aspects present in the environment. This filtered perception is passed to the cognition node which assembles a representation using the goal. The solution generated by the DMP from the representation is passed to the execution node, which in turn alters the environment through actions. It is worth pointing out that the world (and thus the situation) may evolve even when no action is executed by the agent.

Since the representation is the output of the cognition node and the input to the DMP, these nodes are of special interest to us. To achieve SA, the cognition node must provide the relevant aspects through means of abstraction, inference, embedding and so on. To act situation-aware the DMP must be able to take all relevant aspects into account. Ideally, the cognition node would provide a minimal representation and the DMP would ignore all non-relevant aspects.

B. Estimating Situation Awareness

In [16] various techniques to measure SA in human subjects are analyzed. While a subject will always try to stay situation-aware, in the face of challenging tasks, periods of SA will typically be mixed with non-situation-aware intervals. Based on how this temporal dynamic is evaluated we identify three fundamental measurement techniques.

The simplest one, which is employed by e.g. SART, uses a post-trial survey to assess the subjects’ SA. While this does not influence subjects during the trial, subjects might have difficulty remembering individual situations and their reaction to them. It is also not possible to estimate the subjects’ SA online.

A more involved approach is to administer queries at specific points during the trial. For this the task is frozen and the subject is asked targeted questions. This has the advantage that SA can be checked online and is used by one of the best known techniques, SAGAT. On the other hand, interrupting the trial has the potential to influence the subjects’ performance. Furthermore, freezing a task requires a strictly controlled environment which limits the possible use cases.

The third approach is to make use of a subject matter expert. This expert has to rate the subjects’ SA, which

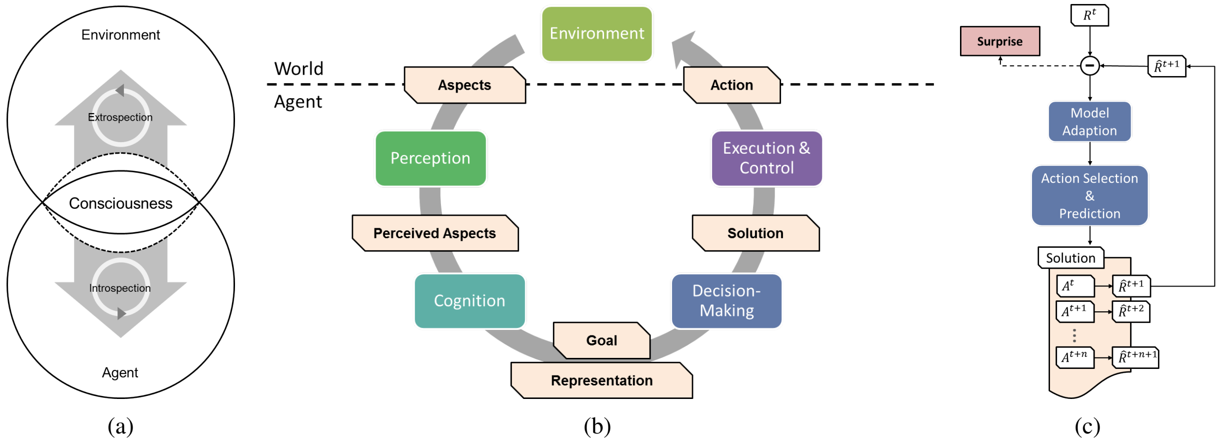

Fig. 1: (a) Situation Awareness depicted as outwards extending consciousness as described in [2]. (b) Extension of the perception-action-cycle using our terms. This process allows the agent to expand their consciousness into the environment and act within it. (c) Solution generation and model adaption inside the Decision-Making Process. The difference between representation and prediction is the system's surprise.

can be done at arbitrary points mid- or post-trial. Although formalized techniques are rare (SABARAS being one example), this approach can be recognized in most task-specific implementations, e.g. [12][13]. While subject bias is not an issue with this technique, the quality of the measure is limited by the expert's expertise. It is also questionable to what extend an external observer can rate a subject's internal SA construct.

All measurement techniques require extensive knowledge of the task at hand. Furthermore, the application of surveys requires human-like mental capabilities, which disqualifies them for use in present-day robots. While using a subject matter expert alleviates this requirement, this introduces another highly task-specific element.

In order to assess a technical system's SA, we propose a new way to do so based on surprise. While surprise has already been used in machine learning (e.g. [17]), we are the first to use it to assess SA. Surprise has been defined in [18] as a stimulus that disagrees with one or more expectations. For agents which predict future states (e.g. those using autoamted planning algorithms), we can assess the agent's surprise as the difference between predicted and actual situation (remember that a situation is the set of aspects relevant to a goal at a certain point in time). Not correctly predicting a situation can have two reasons: the agent's prediction model was flawed and/or the representation was not complete. In either case the agent knows that they were not situation-aware when they encounter an unforeseen situation. As a result the agent could schedule actions to attain SA again (e.g. acquiring additional information). A learning agent could also take this as a cue to improve their internal routines.

Since we defined a situation as a set of relevant aspects, we can use set operations to verify predictions. We say that an agent was situation-aware in $t-1$, if the minimal prediction $\widehat{R}^t_{min}$ made in $t-1$ is a subset of the actual complete representation R^t_{com} made in t, as shown in equation (1) and depicted in Fig. 1c.

$$SA^{t-1} = \widehat{R}^t_{min} \subseteq R^t_{complete} \tag{1}$$

In implementations the prediction may of course include non-relevant aspects (it would be complete, but not minimal). However, this would also mean that these aspects have to be predicted correctly in order to keep the surprise strictly related to the goal.

This method of estimating SA through surprise is limited in that it can only assess whether an agent is *not* situation-aware. If some relevant aspects are not included in the representation, they will neither be part of the prediction, and thus will not lead to surprise. Although one might assume that an incomplete representation will lead to flawed predictions and changes in the solution, this can not be expected in general since the effects of (erroneous) decisions can be delayed. On the other hand, evaluating (non-)SA through surprise has the benefit of being applicable independent of the actual application. Since situation prediction is already an integral part in many systems doing automated planning, this measure can be made explicit in these with very little effort.

In chapter V we will give a first look at our approach to utilize surprise in order to allow a system to acquire SA in the face of novel situations.

IV. Experimental Setup

A. Workspace

Since most of today's robot arms are non-compliant, humans put themselves at risk if they occupy the same spatial and temporal space as a robot. But even if robots can move collision-free, this is not enough for a successful cooperation; the robot must also consider that other agents will take part in the task and alter the world around them. We believe that SA can act as an enabler for these scenarios.

In order to evaluate our concept, we have built a physical workspace for generic HRC tasks. We split the workspace into dedicated areas, which are assigned to either the robot or the human. In the future, we will also introduce a shared workspace area where either agent may act. The human's cooperative partner is a UR5 robot. All systems have been integrated into the Robot Operating System (ROS). By relying on data being provided by the subsystems (Perception) and leveraging the ROS/MoveIt! library for trajectory generation and execution (Execution & Control), we are able to focus on the Cognition and the DMP.

We employ a multitude of subsystems, which already do major data abstraction, producing high-level data-streams like gaze vectors or hand positions. In order to extract the human's pose and skeleton, we use a markerless multi-camera motion tracking system called CapturyLive[1]. To better understand the human's actions and intentions, we added a Pupil Labs eye tracking headset[2]. Since CapturyLive does not provide a reliable body part orientation we needed a way to get the human's head orientation so we could map their gaze into the scene. To solve this we added numerous augmented reality markers to our workspace. Using a perspective-n-point algorithm, we are able to calculate the position of the front facing scene camera of the eye tracking gear in relation to these markers. For recordings and future research two Microsoft Kinect 2 depth cameras have been added as well. A photo of our setup can be seen in Fig. 2.

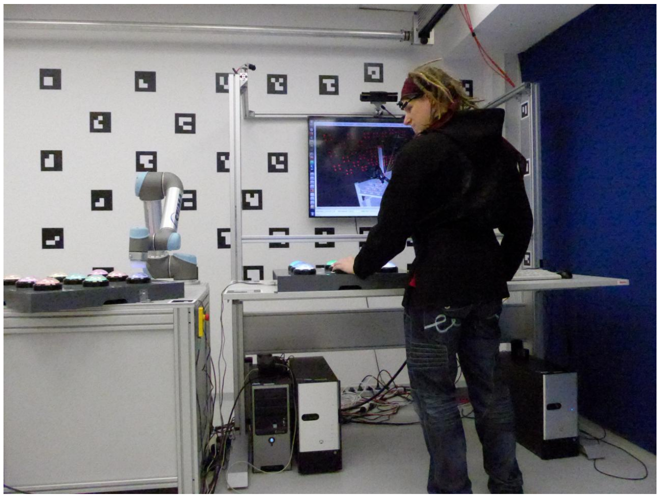

Fig. 2: Photo of our experimental setup. To the left the UR5 is about to press one of its buttons. Meanwhile, one of the authors is waiting for the robot to finish, wearing the eye tracker. In the background several augmented reality markers and a screen showing a visualization of our scene are visible.

B. Task Graphs

In order to keep the system generic and improve traceability we wanted to avoid physical object manipulation. Instead, we represent actions within our tasks by pressing large color- and position-coded buttons. Pressing a button can represent complex actions like tightening a screw or assembling a controller, but does not make any assumptions about the actual level of detail. By representing our tasks as abstract branching sequences of button presses, we are able to exchange the underlying generative model without changing our implementation. This has the added benefit that control of the robot, as well as human action recognition, are significantly simplified. This comes in handy in a related work, where we estimate the human's SA from their actions [19].

Since we don't have a shared workspace area yet, all task steps are assigned to either the robot *or* the human. Each of the agents has access to a board with nine large buttons. Depending on the length and complexity of the task not all of them might be needed. A visualization of a sample task can be seen in Fig. 3.

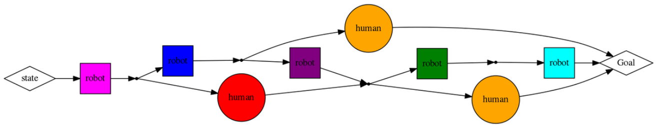

Fig. 3: Example of a branching task as used in our setup. The steps have been color-coded according to the corresponding button. Circles code for steps the human has to execute, squares depict steps assigned to the robot.

To keep our tasks simple, we generate sequential total-ordered tasks, with no dead ends and a single terminal state. There are also no concurrent or synchronized actions. As our system becomes more complex, so will our tasks, removing these constraints one by one.

V. Utilizing Surprise

We designed a simple HRI experiment to showcase the current state of our system and outline where we want to go. In this experiment our system is confronted with tasks as seen in Fig. 3. The tasks always have a path which can be executed by the robot without human help, but also offer several shortcuts which only the human can execute. The DMP is designed to always choose the option with the smallest distance to the goal from those available. We visualized one instance of our experiment in Fig. 4.

Initially our system does not know about the human's capabilities and so always follows its own path, predicting state transitions accordingly. This is implemented as a rule ("<action.type == robot>") which is applied in the Cognition to provide the available options to the DMP. Now, whenever the human triggers a shortcut for the first time, the system observes an unexpected state transition - it was not situation-aware. In order to adapt, it adds a new rule which explains the transition from the previous situation to the current one. For this experiment we provide a pool of predefined rules specific to human actions (e.g. "<action.name == r[ed]>") from which the system can choose. The new rule allows the system to take additional human actions into account when deciding what to do. It will thus experience less surprise and can be considered situation-aware more often.

[1] http://thecaptury.com/
[2] https://pupil-labs.com/

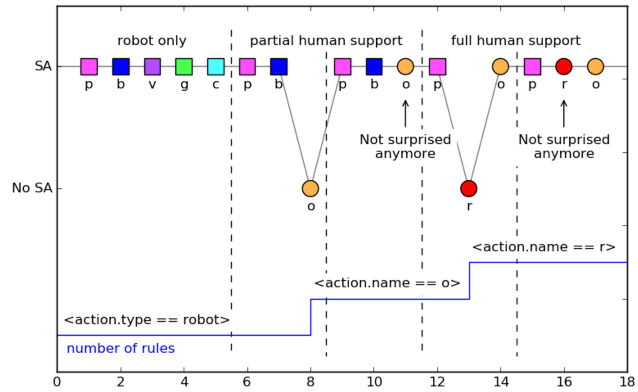

Fig. 4: Actions taken by the robot and the human for the example task shown in Fig. 3. Squares and circles correspond to robot and human actions, respectively. Whenever the system is surprised by a human action it adds a new rule to explain it, allowing the system to account for this action in the future and thus improve its SA.

Obviously this experiment is still very limited in its scope. The point we want to make is that explicit surprise can be used to drive behavior adaption, allowing a system to acquire SA in novel situations. In the future we plan to cover more complex tasks and improve the system's capabilities to reason about situations.

VI. Summary

To reliably evoke competent performance, we proposed a novel way of implementing autonomous agents based on Situation Awareness. Since the application of this human-centered concept is difficult in technical applications, we provided clear definitions of the terms we identified as essential. This also allowed us to define the process through which an agent can become situation-aware. Our concept integrates Context Awareness with Situation Awareness and provides a link to the perception-action-cycle often used in robotics. Since we followed a top-down approach, our definitions are not bound to a specific application or use-case.

Based on our concept, we proposed a new way to find out when an agent was *not* situation-aware by measuring the agent's surprise. This can be done online and even by the agent themselves, albeit only in hindsight. We find that a direct comparison between surprise and other techniques for evaluating SA (such as SART, SAGAT or SABARAS) is misleading due to fundamentally different purposes. While the mentioned techniques aim to make an objective statement over a subject's SA, surprise only triggers when the subject notices that they are *not* situation-aware, and thus is subjective. This also means that a false assumption of SA can be detected by SART and cohorts, but not by surprise. However, its ease of use and applicability to technical systems allows suprise to be used in different areas and with much higher evaluation frequency.

In a simple human-robot-cooperation experiment, we gave a first look on how we plan to utilize surprise to improve the robot's behavior. We plan to expand on the ideas presented here in the future. Specifically, we aim to make our system more general by improving its ability to explain and reason about situations. We would also like to evaluate our concepts in more complex scenarios that closer resemble real-world applications.

Acknowledgment

We would like to thank Thomas Weisswange for his continuous support and feedback.

References

[1] P. R. Cohen, H. J. Levesque, and I. Smith, "On Team Formation," *Contemporary Action Theory, Synthese*, pp. 87–114, 1997.

[2] K. Smith and P. A. Hancock, "Situation Awareness Is Adaptive, Externally Directed Consciousness," *Human Factors: The Journal of the Human Factors and Ergonomics Society*, vol. 37, no. 1, pp. 137–148, 1995.

[3] A. K. Dey, "Understanding and Using Context," *Personal and Ubiquitous Computing*, vol. 5, pp. 4–7, 2001.

[4] S. Robla-Gomez, V. M. Becerra, J. R. Llata, E. Gonzalez-Sarabia, C. Torre-Ferrero, and J. Perez-Oria, "Working Together: A Review on Safe Human-Robot Collaboration in Industrial Environments," *IEEE Access*, vol. 5, pp. 26 754–26 773, 2017.

[5] P. M. Salmon, N. A. Stanton, G. H. Walker, C. Baber, D. P. Jenkins, R. McMaster, and M. S. Young, "What really is going on? Review of situation awareness models for individuals and teams," *Theoretical Issues in Ergonomics Science*, vol. 9, no. 4, pp. 297–323, 2008.

[6] M. R. Endsley, "Toward a Theory of Situation Awareness in Dynamic Systems," *Human Factors: The Journal of the Human Factors and Ergonomics Society*, vol. 37, no. 1, pp. 32–64, 1995.

[7] A. Norton, W. Ober, L. Baraniecki, E. McCann, J. Scholtz, D. Shane, A. Skinner, R. Watson, and H. Yanco, "Analysis of humanrobot interaction at the DARPA Robotics Challenge Finals," *The International Journal of Robotics Research*, p. 027836491668825, 2017.

[8] J. D. Gehrke, "Evaluating situation awareness of autonomous systems," *Performance Evaluation and Benchmarking of Intelligent Systems*, pp. 93–111, 2009.

[9] S. T. Freedman and J. A. Adams, "Improving robot situational awareness through commonsense: Side-stepping incompleteness and unsoundness," Tech. Rep., 2009.

[10] C. Rizzi, C. G. Johnson, F. Fabris, and P. A. Vargas, "A Situation-Aware Fear Learning (SAFEL) model for robots," *Neurocomputing*, vol. 221, no. September, pp. 32–47, 2017.

[11] S. Lemaignan, M. Warnier, E. A. Sisbot, and R. Alami, "Human-Robot Interaction: Tackling the AI Challenges," *Artificial Intelligence*, 2014.

[12] T. Springer, P. Wustmann, I. Braun, W. Dargie, and M. Berger, "A comprehensive approach for situation-awareness based on sensing and reasoning about context," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 5061 LNCS, pp. 143–157, 2008.

[13] Puls, Stephan, J. Graf, and and Heinz Wörn, "Cognitive Robotics in Industrial Environments," *Advanced Biomedical Engineering*, 2011.

[14] G. Bedny and D. Meister, "Theory of Activity and Situation Awareness," *International Journal of Cognitive Ergonomics*, vol. 3, no. 1, pp. 63–72, jan 1999.

[15] J. Lundberg, "Situation awareness systems, states and processes: a holistic framework," *Theoretical Issues in Ergonomics Science*, vol. 16, no. 5, pp. 447–473, 2015.

[16] P. Salmon, P. N. Stanton, G. Walker, and D. Green, "Situation Awareness Measurement: A review of applicability for C4i environments," *Applied Ergonomics*, vol. 37, no. 2, pp. 225–238, 2006.

[17] M. Faraji, K. Preuschoff, and W. Gerstner, "Balancing New against Old Information: The Role of Puzzlement Surprise in Learning," *Neural Competition*, vol. 30, no. 1, pp. 34–83, 2017.

[18] D. E. Berlyne, *Conflict, arousal, and curiosity.* New York, NY, US: McGraw-Hill Book Company, 1960.

[19] M. C. Buehler and T. H. Weisswange, "Online inference of human belief for cooperative robots," 2018, submitted for publication.