Taxonomy of Trust-Relevant Failures and Mitigation Strategies

Tolmeijer, Suzanne ; Weiss, Astrid ; Hanheide, Marc ; Lindner, Felix ; Powers, Thomas M.; Dixon, Clare ; Tielman, Myrthe

DOI
10.1145/3319502.3374793

Publication date
2020

Document Version
Accepted author manuscript

Published in
HRI 2020 - Proceedings of the 2020 ACM/IEEE International Conference on Human-Robot Interaction

Taxonomy of Trust-Relevant Failures and Mitigation Strategies

Suzanne Tolmeijer
University of Zurich
tolmeijer@ifi.uzh.ch

Astrid Weiss
Vienna University of Technology
astrid.weiss@tuwien.ac.at

Marc Hanheide
University of Lincoln
mhanheide@lincoln.ac.uk

Felix Lindner
Ulm University
felix.lindner@uni-ulm.de

Thomas M Powers
University of Delaware
tpowers@udel.edu

Clare Dixon
University of Liverpool
cldixon@liverpool.ac.uk

Myrthe L. Tielman
Delft University of Technology
m.l.tielman@tudelft.nl

ABSTRACT

We develop a taxonomy that categorizes HRI failure types and their impact on trust to structure the broad range of knowledge contributions. We further identify research gaps in order to support fellow researchers in the development of trustworthy robots. Studying trust repair in HRI has only recently been given more interest and we propose a taxonomy of potential trust violations and suitable repair strategies to support researchers during the development of interaction scenarios. The taxonomy distinguishes four failure types: Design, System, Expectation, and User failures and outlines potential mitigation strategies. Based on these failures, strategies for autonomous failure detection and repair are presented, employing explanation, verification and validation techniques. Finally, a research agenda for HRI is outlined, discussing identified gaps related to the relation of failures and HR-trust.

CCS CONCEPTS

• **Human-centered computing** → **User models**; **Interaction design theory, concepts and paradigms**; *HCI theory, concepts and models*; Empirical studies in HCI; • **Computer systems organization** → **Robotics**.

KEYWORDS

Trust Violation, Trust Repair

ACM Reference Format:
Suzanne Tolmeijer, Astrid Weiss, Marc Hanheide, Felix Lindner, Thomas M Powers, Clare Dixon, and Myrthe L. Tielman. 2020. Taxonomy of Trust-Relevant Failures and Mitigation Strategies. In *Proceedings of the 2020 ACM/IEEE International Conference on Human-Robot Interaction (HRI '20), March 23–26, 2020, Cambridge, United Kingdom.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3319502.3374793

HRI '20, March 23–26, 2020, Cambridge, United Kingdom

ACM ISBN 978-1-4503-6746-2/20/03.
https://doi.org/10.1145/3319502.3374793

1 INTRODUCTION

Trust is an important component to ensure successful diffusion and uptake of human-robotic systems interaction in society. Trust in and trustworthiness of these systems have been considered important for long-term interaction, collaboration, and acceptance [50]. However, how should we design and implement trustworthy systems? Software engineering techniques such as verification and validation can be used to ensure that the system conforms to its requirements (verification) and the system meets the need of the stakeholder (validation). This improves reliability, safety and trustworthiness of the systems (see for example [23, 77]) and will help mitigate some of the failures leading to loss of trust.

Does the HRI community currently have sufficient knowledge of what makes a system trustworthy to be able to design robots as such? Human responses towards robotic systems are very complex in their nature and depend on many factors, such as the morphology and behavior of the system and the context in which they are deployed. Therefore, in order to design trustworthy robots, we have to base our design decision on detailed knowledge of (1) how humans react towards robots and (2) how robot features might foster or harm trust.

The challenge becomes more complex as trust has both static and dynamic components in human-robot interaction. Static components such as gender do not change, but dynamic components related to the system can be influenced [39]. We need to systematically structure the knowledge on trust that has been gained so far; it influences our design choices, also when an interaction is unsuccessful and possible negative effects need to be mitigated. The aspects of trust repair and trust violations have been understudied in the field of HRI [4]. Trust repair can be understood as the activity of rebuilding trust after one party breaks the trust of the other, i.e. after a trust violation. But what causes these trust violations and how can trust be repaired after they occur?

In this paper, we present a taxonomy of trust-relevant failures and mitigation strategies, based on literature as well as empirical data from known real-world use cases. Becoming aware of the fundamental need to structure our knowledge on how to build trustworthy systems, the discussion of this taxonomy started during a seminar where the authors met. The authors of this paper, who all have different disciplinary backgrounds ranging from philosophy and AI to mathematics and logic, analyzed the state of the art on

trust research with respect to their disciplinary background. We propose a *taxonomy* that enables fellow researchers to incorporate mitigation strategies into their systems to recover from failure situations that potentially harm trust.

The inspiration for the taxonomy stems from so-called risk tables [6]. By definition, a risk equals uncertainty plus damage, in our case damage of trust [65]. In analyzing risks, one is attempting to envision how a scenario will play out if a certain course of action (or inaction) is undertaken. Therefore, a risk analysis always starts from three basic questions: (i) What can happen? (i.e., What can go wrong), (ii) How likely is it that that will happen?, and (iii) If it does happen, what are the consequences? [78] Classical risk tables visualize this information, e.g. the risk of getting a specific disease. We present an overview for failure situations in HRI that can harm trust in the robotic system, and offer robot designers mitigation strategies to (1) avoid or (2) recover from failure and reestablish trust.

We also outline explanation-based approaches, as well as validation and verification techniques that can be used to formalize our taxonomy in order to build trustworthy human-robot interactions.

2 RELATED WORK

Trust is a valued feature of individual human relationships which also enables social cohesion. Its dimensions have been studied by several disciplines, yielding results that both guide and limit the extent to which robot trust may be developed.

2.1 Approaches to Trust

The *psychology of trust* focuses on interpersonal relationships. The development of trust between persons typically follows familiarity, is concomitant with dependence, and in close personal relationships is associated with both behavioral predictability and the attribution of beneficent motives [62]. Risk regulation [58] allows the trusting agent to temper the degree of vulnerability to the party being trusted. Different kinds of trust attach to agents, depending on expectations and expertise. While the neurochemistry of trust is not well understood, it is assumed that trust can be understood both as a brain process and an emotional process [72].

The *ethics of trust* has been analyzed as necessary to economic exchange [2], friendships [73], and even the Hobbesian civil state itself [38]. Rusbult et al. [66] identified accommodation processes that allow close relationships to survive otherwise trust-breaking failures of expectations, e.g., via charitable interpretations of motives. Actions (commissions) that fail expectations and thus damage trust are, according to some [19], worse than failures to act (omissions). However, psychologists Tversky and Kahneman [74] as well as other ethicists find in this to be an omission bias, since the consequences of (not) acting can be the same. Hence from a consequentialist perspective, the damage to trust in the case of human failures ought to be similar. However, Malle et al. have shown that for robot failures, there is an asymmetry of blame–that humans blame robots more for failures of inaction than of action [53, 54].

Turning to *trust in robots*, we see the potential for overlap and contrast with the psychology, ethics, and pragmatics of trust between humans. Prior to the development of complex behaviors in robots, many philosophers would have insisted that trusting robots is more like trusting a tool than another person.With some conceptual flexibility, we can see that trusting robots has elements of both sorts. Studies on trust within robotics have mainly been motivated by the literature on trust in automation [40, 46, 47], which operates with a conceptualization of trust as mere reliance. According to this stance, trust is a domain-specific relation between the human and the robotic system involved. We follow this stance for our proposed trust taxonomy and define trust in robotic systems, in accordance to Lewis and colleagues [50], as a predictive belief or assumption about what will occur given the performance, process, or purpose of the robot. The definition of trust as appropriate reliance also stresses the importance of trust in situations involving risk and uncertainty. Humans who misplace trust, understood as both under- and over- reliance, might be exposed to serious danger, which is the reason safety concerns are of high consideration. In our understanding of trust as reliance, we consider the robotic system as tools intended for accomplishing certain ends. Other dimensions of trust, such as institutional trust are intentionally excluded, as our taxonomy should serve as robot-centered knowledge base.

2.2 Modeling Trust

The aim of defining/modeling trust in HRI is nothing new. Billings proposed a three-factor model of trust in robots, including *human* characteristics such as ability and personality, *environmental* characteristics such as task and team, and *robot* characteristics such as performance and attributes [5]. These three factors have also been identified in a meta-analysis on trust [29], where the authors stressed that too few studies have yet been conducted on environmental and human-related factors, although robot-related factors have been shown to affect trust the most.

Similarly, modeling trust from the perspective of risk has been considered before. Drawing on the model from organizational contexts by Mayer et al. [56] and the model on trust in automation by Lee and See [47], Wagner et al. [75] propose a trust model based on risk. They define trust as "a belief, held by the trustor, that the trustee will act in a manner that mitigates the trustor's risk in a situation in which the trustor has put its outcomes at risk" [75, p.26:4]. Trust is modeled in game-theoretic terms and, similar to what Hancock et al.[29] proposed, they highlight three important factors that influence trust-based decisions, namely the trustee, the trustor, and the situation. The model is also tested in an emergency experiment by Robinette et al. [64], where people tended to overtrust the robot despite half of them observing the same robot performing poorly in a navigation guidance task minutes before.

Based on the three-factor model by Hancock et al. [29], Hoff and Bashir [39] have also suggested a three-layered model in which these factors contribute to *dispositional*, *situational* and *learned* trust. They point out that age, gender, culture and personality are components of dispositional trust. Situational trust is shaped by internal and external variability, such as self-confidence and task difficulty. Learned trust consists of initial learned trust (e.g. expectations of the system) and dynamic learned trust. The latter is influenced by design features and system performance and influences the user's reliance on the system.

2.3 Trust, Failure, and Repair

The concepts of trust repair and trust violations have been understudied in the HRI literature so far. The need for research on trust in artificial agents in cases of inevitable failure has been highlighted as well [5]. Baker [4] surveys trust with a focus on trust violation and repair of human-robot interaction. For a successful recovery of trust, (perceived) shared intentions have shown to be important (cf. [16]). Even though from a scientific and engineering perspective we know that robots do not intend their behaviors in the same way as humans do, taking robots as intentional agents may aid users (psychologically) in attributing sufficient beneficence to their "motives" — at least insofar as this is necessary to engage with them. Following errors of automation, information related to limitations further aid in trust recovery. Hence, perceived benevolence may promote acceptance of a robot's changing behaviour [52], as with human interpersonal relationships [62].

In ongoing studies, several actions of trust repair have been proposed, including apologies, promises, internal or external attribution, and the showing of consistent series of trustworthy actions [4, 14]. In an emergency setting, where an apology right after violated trust has not recovered trust, an apology right before the next trust decision point has repaired trust. Promises lead to a better trust recovery than apologies, and in general, the message timing and exact content was shown to be crucial [63].

Thus, studies show that trust harmed by untrustworthy behaviour of a robot can be restored when people encounter a consistent series of trustworthy actions. However, trust harmed by deception and the same untrustworthy actions never fully recovers, even with actions of trust repair [68]. Additionally, a promise to change behavior can significantly speed the trust recovery process, but prior deception harms the effectiveness of a promise.

Studies on trust violation and repair take into account the evolving nature of trust, where trust is seen as something that changes over time. For example, Desai et al. [17] and Sebo et al. [69] researched robot failure and its influence on dynamic trust during one interaction. However, it has been outlined that long-term studies exploring the transient nature of trust are missing in the literature [50]. For example, how does trust change with increasing familiarity of the user with robots? Also due to their little employment in society, long-term studies have not been conducted so far.

Nordqvist and Lindblom [59] analyze trustworthiness of industrial robots with an operators' experience framework. The evaluation framework consists of the factors ability, benevolence, integrity, perceived safety, time on task and errors, where in total 12 user experience (UX) goals were characterized, 2 for each component. For each UX goal, data collection methods were selected and mixed, including observations, video recordings interviews, and Likert scales. Interestingly, major identified reasons for limited trust were communication problems during collaboration resulting in participant's uncertainty of their own ability to collaborate with the robot. The communication problems were strongly linked to the interface design. Further, the participants initially had confidence in the robot itself, but were insecure of their own ability to collaborate due to their inability to predict the robot's intentions and instructions.

In an online survey, Brooks et al. [7] explored people's reactions to failures in autonomous robots, namely a vacuum cleaner and a self-driving taxi, by manipulating four variables: context risk, failure severity, task support and human support. Participants' perceptions of an erroneous robot became less negative when it deployed a mitigation strategy, either by prompting task support, human support or both. However, the authors reported an interesting but non-significant tendency showing a preference for both task and human support in high severity situations, and a preference for only task support in low severity situations.

3 PROPOSAL

We propose a taxonomy of failure types that can influence trust during Human-Robot Interaction. For each failure type, different mitigation strategies are suggested. While De Visser et al. [14] stress the importance of trust repair and list possible mitigation strategies, these strategies have not been linked to different failure types before. As mentioned by Baker et al. [4], models of human-automation and human-human trust are a helpful starting point, but do not account for the complexities of building and maintaining trust in HRI. A taxonomy for trust repair in HRI does not exist, but a framework for rebuilding trust in automation has been proposed by Marinaccio et al. [55]. It follows a similar intention: providing recommended trust repair strategies depending on the violation committed. However, they base their framework on the error taxonomy of Reason [61] which does not account for the interactive nature of HRI. Furthermore, human error taxonomies such as [70, 71] focus mostly on human error, while our taxonomy takes a holistic approach by including errors by other actors such as the system('s designer).

3.1 The Taxonomy

As a starting point for our discussions, we defined trust as "a person's willingness to rely on a robot to carry out its duties". As HRI involves two different actors, namely the robotic system and the human interacting with it, we base our taxonomy on a first fundamental distinction: who performed a type of action which caused a break of trust, (1) the system or (2) the user. Next, we distinguished the failure type (i.e. categorization of the actions into different types of failure). We differentiate four different failure types with respect to their impact on trust and the related mitigation strategies: (1) Design, (2) System, (3) Expectation, and (4) User (see Table 1 for condensed failure type descriptions).

Design. Imagine you have designed a robotic system in a specific way (in terms of behaviour, appearance, dialogue and so on) to the best of your knowledge. While in the real world the system behaves exactly the way you intended it to, it turns out that you made design choices that were not ideal for the HRI. For example, a specific function that you added to the robot is not used as often because the command is not as intuitive for the user as you thought, which influences the trust the user has in the system. A user misinterpreting the system's output because of its design; not understanding the interaction or not knowing about certain functionality when they should have are all considered Design failures. These failures are limited to the target audience of the system, as for Design failures the system's behaviour *should* be different in retrospect.

Table 1: Types of actions which cause a loss of trust: we call these failures

Failure type:	Action by	Meant to act this way	In retrospect, should actor behave this way?	Description
Design	System	Yes	No	System does what it's been made to do, but in retrospect the system should not actually behave this way
System	System	No	No	System doesn't do what it's been made to do
Expectation	System	Yes	Yes	System does what it's been made to do, but user expected something different to happen. In retrospect system should still behave this way
User	User	No	If design fail: yes If expectation fail: no	User behaves in a way they are not supposed to. (Only a problem if leading to other type of failure)

Table 2: Risk Analysis of Failure leading to loss of Trust (cf. Sec. 3.2)

Failure	Probability	Impact on trust	Risk score	Mitigation strategy
Design failure	3	2	6	ID, E, A
System failure				
Hardware	1*	3	3	E, A, F, Alt
Software	3*	3	9	E, A, F, Alt
Expectation failure				
Commission failure	2	4	8	E, A, ID, T
Omission failure	3	2	6	E, A, ID, T
User failure				
Intentional	2*	1	2	J, ID, Emo, Auth
Unintentional	2	3	6	T, ID

Probability scores: 1 = 1 occurrences in about 1000 interactions, 2 = 1 in 100, 3 = 1 in 10, 4 = likely in every interaction episode.
Impact scores: 1 = minor impact (negligible) to 4 = fatal impact (potential loss of trust and further use).
Mitigation strategies: ID = Interaction design; E = Explanation; A = Apology; F = Fix; J = Ask for justification; Emo = show emotion; Auth = Involve authority figure; Alt = Propose alternative; T = Training

System. When a System failure occurs, the system does not act as intended. For example, the robot stops in the middle of a room during a navigation task without a reason, or stops a scanning task because its scanner malfunctions. In other words, the system does not do what it should, e.g. because of a system crash. The distinction can be made between a hardware and software failure.

Expectations. Trust in technological systems is typically concerned with the human's expectations of the system. With an Expectation failure, the system acts as intended, but defies the user's expectation. For example, when the user expects a robot to turn while observing a room, but the robot does not need to do so, the systems performs as it should but confuses the user. This is an example of an omission failure: the robot does not act when the user expects that it will. The opposite of this is a commission failure: the robot does something the user does not expect, e.g. start moving in the middle of an interaction because it needs to charge its battery. Expectation failures are different from Design failures in that for an Expectation failure the system should in retrospect still behave the same, while in case of a Design failure it should not. In case of the robot turning, the turning is an Expectation failure. However, there is probably a related Design failure as the robot does not explain its actions to the user properly. In this example, the Design failure is what leads to the Expectation failure.

User. In this last category the user interacts with the system in a way that he/she was not supposed to do, e.g. disturbing or sabotaging the robot (intentional) or standing in the robot's way so it cannot move (unintentional). This type of failure can be caused either by a Design failure or an Expectation failure which influences its impact on trust and potential mitigation strategies. While an Expectation failure deals with what the user expects the robot to do, a User failure is about what the users themselves do. Of course, Expectation failures could lead to unintentional User failures.

Combining all these failure types gives our foundation of the taxonomy shown in Table 2, including mitigation strategies that potentially repair the broken trust. This table is designed to resemble risk tables [6], also aiming to quantify the *Probability* of a failure occurring and the estimated *Impact* it will have on trust. In line with risk assessment practice, a *Risk score* is computed by multiplying the probability and impact scores, providing an indication of the priority for suitable mitigation strategies. These scores are system-

and scenario-specific. To show how such a risk table can be used in a HRI context, the scores in Table 2 are from a real world use case.

3.2 Trust loss as a risk: A Case-Study

We present the following interactive system as a case-study in this paper, to show how our proposed taxonomy can be used for a real-world use case. In this example case [31, 32], an autonomous mobile robot has been deployed in a care home for a total of just over a year, in the context of the STRANDS project[1]. This experiment was split over three individual deployments, following an iterative design principle, spread over a duration of three years. Here, the robot served as a mobile info-terminal and was also engaged in occupational therapy sessions. It was left without any technician or researcher on site, interaction with visitors and residents in the care home was without explicit solicitation by any experimenter.

Rich data sets, comprising task and error logs [32], user demographics [36], and navigation failures [15] have been obtained from these deployments, and analysed for the case study for this paper.

Tab. 2 presents the results of this case study analysis, in terms of *Probability* and *Impact* scores derived from the retrospective analysis of the data sets from the deployments. It shall be noted that this constitutes merely a case study, based on available data, allowing only some scores to be robustly computed from logs, while others have to be informed guesses, based on the authors' experience. For transparency, we have marked scores that are estimated from available data sets with an asterisk (*).

Probability and Detection. In the specific instance, a variety of problems were detected automatically, such as navigation issues [15], forceful pushes to the robot, and hardware failures. Consequently, many failure types can be detected from system logs and from dedicated anomaly or failure detection modules that allow to estimate the probability of them occurring. In our case study of the STRANDS system, we analysed logs covering a cumulative deployment of over a year and employ some "Back-of-the-envelope" (BoE) calculations to derive the probability score. Given that the probability score is only intended to give an indication of the magnitude of a specific failure class, a BoE is most adequate for this assessment. The system data in [36] indicated that there were about 3.5 interactions per operational hour (i.e. time the robot is not resting or charging) with users that are actively using the robot. We shall take this estimate as the baseline for our BoE approximation.

An analysis of software failures, in particular navigation failures (which account for more than 99% of all software-related issues in this particular use case) in [15] reveals that in 1605 instances the robot had to ask for help as it could not recover from a navigation problem, making its failure obvious to the interacting humans, and hence potentially having an impact on trust. Thus, we observed such failure about every 2 hours of autonomous operation, leading to a ratio of $7 : 1$ for Software failures to interactions, leading to a *Probability* score of "3" in Tab. 2. Most scores in Tab. 2 were calculated in a similar fashion: hardware failures were counted (e.g. snapper drive belt, failed encoder) as well as intentional User failures. In the case of the latter, by counting the number of forceful robot pushes and deliberate tampering, "intentional User failure" was observed in about 1 out of 200 interactions, scoring "2".

The other probability scores are much harder to obtain in a *post-mortem* analysis of long-term deployment, and require more focused studies, e.g. [33], involving the users directly. For instance, [33] revealed some of the Design failures that lead to the iterative improvements between annual deployments.

Impact. To assess the impact of individual failures, we base our assessment of a qualitative analysis in the context of the care deployment within the STRANDS project [24, 33, 34]. The assessment is not an exact science; within this case study we do not aim for a comprehensive analysis of this STRANDS system, but rather present the concepts of considering trust loss as a risk open to a systematic analysis. For instance, feedback from on-site interviews showed that commission failures have a very high impact on trust. As an example, we quote a participant, who complained that the robot appears "stupid", because it would "start talking to a wall", a consequence of misclassification leading to a commission failure. However, establishing a robust scoring system for impact of trust that has wider applicability is one of the areas of future research.

3.3 Mitigation Strategies

Depending on the type of failure that has taken place, there are different possible mitigation strategies that can help regain the trust of the user. Given the interaction between different failure types, mitigation strategies for the initial failure type should be applied first. For example, if an Expectation failure was caused by a Design failure, the Design failure should be considered first. For Design, System and Expectation failures the following mitigation strategies can be used:

Fix. When a System failure occurs, be it hardware or software, the problem needs to be fixed. This is a very practical mitigation strategy to ensure the issue does not occur again and only applies to System failures.

Interaction Design. While it is intuitive that Interaction Design is important to foster trust, it can also be a tool in reestablishing trust. However, we can assume that once trust is broken due to a Design failure, the redesign of the system becomes even more challenging. As Lewicki and Wiethoff [49] explain restoring trust after a violation is a three-step process: (1) exchanging information about the perceived trust violation, (2) willingness to forgive the violator, and (3) reaffirm their commitment. Implicitly communicating all of these aspects to the same user with a change in interaction design will be hardly possible. However, improving trust through the interaction design for other prospective users will still be a viable way to go. Proper design allows for smooth interactions and substantial research is available in HRI on understanding robot-related factors affecting trust in the interaction design, such as social skills [35], robot role [27], and communication style [60]. Hancock et al. [29] provide a detailed overview on HRI studies on the impact of robot design features on trust in HRI.

Explanations. Explanations for the end user can be a suitable mean to repair trust. Methods, such as plan-based explanations related to previous decisions can be used. However, the correct level

[1] http://strands-project.eu/

of detail of abstractions and human-comprehensible explanations are challenging. Explanations to end users do not necessarily need to be in natural language, but can use cues such as closed eyes, blinking lights, nodding head etc. Overall, the aim of explanations should be to increase transparency and understandability in order to repair trust in a failure situation.

Apology. Once a trust failure occurs, it is essential to recognize that trust has been broken and acknowledge that the failure that occurred was unpleasant for the user. Apologies are effective for trust violations related to the violator's competences (e.g. an error in planning or judgement) [45]. In human-human interaction, they are more effective than shifting the blame elsewhere. Once the human understands the effect was not intended and is not intended to happen in the future, the trust repair can start. Lee et al. showed that the apology strategy was most effective to mitigate perceptions of competence, closeness and likeability of a service robot [48].

Propose Alternative. In case of a system breakdown, the trust lost in the system can be minimized when alternatives are available. If possible, the system can propose a workaround the user can employ to still get the intended task done despite a System failure.

Our discussions on User Failures revealed that there is little to no research on how to mitigate this type of failure. We consider the following strategies as promising:

Ask the Human for Justification. When a user misbehaves, the response the system gives will influence future behavior of the user towards the system. If the user was not aware of any misbehavior, asking the user for justification of their actions can create awareness of their mistakes. We assume that unintentional negative behavior will not be repeated once the user becomes aware of it. Intentional misbehavior is harder to address, since the user acted purposefully. Asking a justification is intended to help the user realize the negative consequences of their actions.

Show Emotion. It is in our nature to anthropomorphize robots, for example by projecting a personality onto the robot or reading emotions into its output. When a user misbehaves, emotion can be a powerful tool to persuade the user to behave better. However, the impact of negative emotions displayed by a robot is understudied [42]. The only study we are aware of, in which a robot shows negative emotions - namely an aggressive movement pattern - could show that this was enough to reduce robot abuse [67].

Involve Authority Figure. Using authority is a persuasion mechanism [12] that can be useful to make sure users behave properly towards the robot. An example can be to alert the owner of the robot or authorities. Research on children's abusive behaviour towards robots in shopping malls revealed, that children typically did not stop such misbehavior until their parents (their authority figure) stopped them or they got bored [8].

Training. For unintentional User failures, training can be a potential mitigation strategy to avoid repeated future failure situations. So far, little research has been done on how users can be trained in HRI (since research mostly focuses on how users can train robots [1]), but existing work shows that "training is essential" [9].

4 AUTONOMOUS TRUST REPAIR

What we want to achieve in HRI at some point is autonomous trust repair, which implies both failure detection and failure mitigation is managed without human assistance. The first step towards this goal is failure detection: is something wrong with the system? Related to this is failure classification: once it is established something is wrong, the system needs to assess what is wrong. Finally, using this classification and the detected deviation from the plan the system had, an explanation can be presented in an attempt to repair the lost trust. In our opinion, this is a fundamental prerequisite: a robot needs to detect that a failure happened and an explanation to the end user should be the starting point for any mitigation strategy.

4.1 Failure Detection

Robots that interact with humans in the wild will at some point face failure situations, which can either be inflicted by the robot, the human, or by unexpected environmental events. However, dealing properly with failure situations from a robot-centered perspective is a challenging endeavour. Firstly, the robot has to detect that an error situation has occurred; secondly, it needs to analyze what kind of error situation occurred; thirdly, it needs to apply an error recovery strategy to get back into a safe interaction state.

What can be detected? Looking at our taxonomy in Table 1, the question arises which of those failure types can be detected by a robot itself (self-awareness) without further involvement of the user. The common definition of failure usually requires the exact knowledge and definition of a *failure case*, i.e., a formal definition of what constitutes a failure. In other words, the failure detection problem is considered a classification problem, where a model of the failure itself can either be defined or learned.

One way to do this is by using verification and validation techniques. Formal verification is a mathematical analysis of all behaviours of the robot or system using logics, and tools such as theorem provers or model checkers (see for example [13, 21]). Using model checking, a desirable property encoded in some logic is checked over a model, often a finite state transition system, to ensure that it holds on all paths through the system from an initial state. Theorem proving involves a mathematical proof to show that the property expressed in some logic is a logical consequence of the system also expressed in logic. Simulation based testing utilises simulations of the robots and the environment, possibly including hardware in the loop, to facilitate large numbers of tests that may not be possible in the real world. Tools are used to automate the testing and analyse the coverage of the tests. End user experiments can be used to test aspects such as trustworthiness. Formal verification, simulation based testing and end user experiments can help improve the safety, reliability and trust in robotic systems [23, 77], as well as help mitigate as system failure (all), design failure and expectation failure (end user experiments).

However, this approach limits the ability to detect failures to properties that have been identified in the specification. A complementary approach relates to *anomaly detection* (e.g. recently surveyed in [28]). It aims to detect any deviation from a normal behaviour of a system, without necessarily classifying a problem. The identification of a potentially known problem can then be deferred to approaches to generation explanations, utilising domain

knowledge as formally defined in the following section and also explored in [30].

4.2 Offering Explanations

Once a failure is detected and possibly classified, we consider explanations as one possibility for failure mitigation (see Tab. 2). Therefore, it is desirable to investigate how a robot can automatically generate explanations based on its perception and deliberation modules. According to Miller [57], explanations should be *contrastive*, *selective*, and *social*. Contrastive explanations (implicitly or explicitly) refer to situations different to the one to be explained. For instance, *Why does the robot do X?* should be understood as *Why does the robot do X rather than Y?* One way to generate contrastive explanations is by counterfactual analysis: the occurrence of some phenomenon X in situation S can be explained by a sufficiently altered situation S' where X does not occur (but Y does). Counterfactual explanations have recently been applied to generating explanations for plan failures [25], for explaining why an action plan contains a specific action [22], and to explain why an action plan does (not) adhere to moral principles [51]. These approaches only partially fulfill Miller's criteria of selectivity, though: although minimality criteria are considered, there are generally many possible explanations and it is not necessarily clear how to pick the most appropriate ones. Wang and colleagues [76] circumvent this challenge by generating explanations from Partially Observable Markov Decision Processes using a template-based approach. The downside of this approach is its being less generic and its requiring hand-crafted template modeling. Finally, Miller requires explanations to be social, that is, explanations should take the user's mental state (beliefs, desires etc.) into account. This requirement is a big challenge to the current state of the art of explanation generation.

4.3 Formalism for representing plans

A procedure for explaining failures can be based on the STRIPS formalism for planning [20]. STRIPS and its derivatives are widely used to describe a robot's deliberate actions and external events. A STRIPS model is a tuple $\langle P, s_0, s_g, O, pre, del, add\rangle$ with a set of propositions P, an initial state $s_0 \subseteq P$, a partial state $s_g \subseteq P$ called goal description, a set of operators O (actions and events), a function $pre\colon O \mapsto 2^P$ mapping each operator to a set of preconditions that must hold for the operator to be executable, a function $del\colon O \mapsto 2^P$ mapping each operator to a set of propositions to be deleted from the current world state as an effect of the operator's execution, and a function $add\colon O \mapsto 2^P$ mapping each operator to a set of propositions to be added to the current world state. The execution of operators thus triggers transitions from current world states to successor world states, where world states are sets of propositions. An operator o is *applicable* in a state s iff $pre(o) \subseteq s$. The successor state $s' = (s \setminus del(o)) \cup add(o)$ results from applying o in s. A state s is a goal state if $s_g \subseteq s$. We assume the existence of the empty action $\epsilon \in O$, which has an empty precondition, an empty delete list, and an empty add list.

As an example, consider a robot currently situated in the kitchen. It wants to move to the dining room. The applicable action operator $move(kitchen, diningroom)$ has precondition $\{in(kitchen)\}$. The action's effect is given by the delete list $\{in(kitchen)\}$ and the add list $\{in(diningroom)\}$. Hence, by performing the action $move(kitchen, diningroom)$ in state $s_0 = \{in(kitchen)\}$, the world state transitions from state s_0 to state $s_1 = (s_0 \setminus \{in(kitchen)\}) \cup \{in(diningroom)\} = \{in(diningroom)\}$.

4.4 Explaining failures from plans

Let $\pi = s_0 \rightarrow_{o_0} s_1 \rightarrow_{o_1} \ldots \rightarrow_{o_{n-1}} s_n$ be a course of actions and events o_i—also called a *plan*—originating from the initial state s_0 and terminating in some state s_n, which may (or may not) qualify as a *failure state* in the sense of the conceptualization outlined in Subsect. 3.1 and Tab. 2. In case of failure, we want to understand why the failure occurs by answering Why-questions about facts and actions:

(1) Why does fact p (not) hold at time point t?
(2) Why does the robot (not) perform action a at time point t?

As an example, consider the following case which involves an *expectation failure* of type *commission* and requires generating an answer to a question of type (2): *After the robot receives a navigation goal from the user, the robot suddenly starts turning to get a precise estimate of its current location via its front-mounted laser rangefinder.* The user expects the robot to immediately start moving towards the specified destination and thus wants to understand *Why does the robot start turning?* To see how an answer can be generated, first consider the robot's plan $\pi = s_0 \rightarrow_{tl} s_1 \rightarrow_{nd} s_2$, i.e., the robot plans to first make a turn to improve its localization (action tl) and then to navigate to the destination (action nd). Initially, the robot's pose estimate is poor (fact pe) and the robot is not at the destination, i.e., $s_0 = \{pe\}$. The goal $s_g = \{d\}$ is to be at the destination. The precondition of the navigation action nd is that the robot has a good pose estimate (fact ge). Performing nd adds d to the state. The turn action tl has delete list $del(tl) = \{pe\}$ and add list $add(tl) = \{ge\}$. To explain why the robot is turning, counterfactual analysis is used: an inclusion-wise minimal subset $x \subseteq add(tl)$ of the add list of action tl is identified, such that if the facts in x were removed from $add(tl)$, then the final state of plan π would be no goal state. This is to say that x is a necessary means to the goal d. Clearly, removing fact ge from $add(tl)$ would make action nd inapplicable and thus fact d would be missing from the final state. Accordingly, the robot can explain *Turning around results in knowing where I am, and this is necessary for finally reaching the destination.*

4.5 Logics for Trust Loss Detection

One way to recognize whether trust was lost because of a failure, is by using logics to model and reason about trust loss. Logics for trust have been developed. In [37] the authors formalise the work of [11, 18]. In [11, 18], i (truster) trusts j (trustee) to do α (an action) with respect to φ (a goal) if and only if (1) i has the goal φ; (2) i believes that (a) j is capable to do α; (b) j, by doing α, will ensure φ; and (c) j intends to do α.

In [37] the notion of trust is reduced to more primitive concepts of belief, goal, capability and opportunity which is formalised in a logic of time, action, beliefs and chosen goal. Two kinds of trust are considered. Firstly, the truster believes that the trustee is going to act here and now (termed occurrent trust). Secondly, the truster believes that the trustee is going to act whenever some conditions

are satisfied (dispositional trust). Only occurrent trust and qualitative aspects of trust are considered in [11, 18]. Two dynamic logic operators $After_{i:\alpha}$ and $Does_{i:\alpha}$ are proposed. The former gives the result of agent i's performing action α (its capabilites) and the latter about what holds after agent i does action α (what an agent does and intends to do). The modal operators Bel_i (agent i believes) and $Choice_i$ (agent i has chosen the goal) and the temporal operators G (always in the future) and F (at some future moment) are also used. Occurrent trust $OccTrust(i, j, \alpha, \varphi)$ is defined as follows: $OccTrust(i, j, \alpha, \varphi) = Choice_i F\varphi \wedge Bel_i(Does_{j:\alpha} \top \wedge After_{j:\alpha}\varphi)$. That is i trusts j to do α with respect to φ if and only if, i wants φ to be true at some point in the future and believes that j will ensure φ by doing action α. The authors argue that this may be too strong as j is going to do α immediately. This leads to the definition and formalisation of dispositional trust which is weaker than this. A complete axiomatisation is provided but complexity and decidability are not considered.

In [43, 44] the authors consider automated quantitative reasoning about trust via stochastic multi-agent systems. They formulate probabilistic rational temporal logic (PRTL*) as a combination of the probabilistic computation tree logic (PCTL*) with cognitive attitude operators (belief, goal, intention) and trust operators (competence, disposition and dependence). The resulting logic is, in general, undecidable but decidable fragments are identified. The work has again been inspired by [18] and, as with our work, the focus is on trust between humans and robots/autonomous systems.

These logics could be used to model robotic trust scenarios to identify when and how the system is not trusted or trust is lost. The belief aspects from [11, 18] and modelled in the logics mentioned above seem to match the expectation failure type discussed above. However, they do not match the more complex models of trust as introduced in Section 2.2.

5 FUTURE WORK

Reflecting on existing HRI research on trust repair and the introduced taxonomy, as well as autonomous failure handling through explanation generation, verification and validation techniques lead us to identify research gaps we consider crucial to be further explored for successful trust failure classification and mitigation.

Mitigation of User Failures. Our discussions identified the category of intentional and unintentional *User Failures* as up-to-now understudied with respect to mitigation strategies [41]. Mainly how robots could react if people intentionally cause errors, e.g. by covering sensors, giving wrong information or other ways of intentionally bullying the robot. We gave potential examples of mitigation strategies, namely calling an authority, showing emotions, and ask the person for justification. However, effects of robots showing negative emotions are in general understudied [42], and no systematic studies of the other strategies exist so far.

Impact of Failure Repetition. Similarly, the impact of failure repetition is understudied, above all with respect to how it affects trust. Some studies on people's willingness to help robots after repeated failure indicate that repeatedly helping robots in need when the suggested repair strategy is successful does not reduce likability [3]. However, this does not give insights into how much overall trust is harmed. It will need long-term studies outside of laboratory experiments to get an ecologically valid grasp on how failure repetition affects trust. Subsequently, long-term in-the-wild studies, lasting several weeks to out-rule novelty effects [26], will be needed to assess the impact of familiarity with the robot. Studies on gracefully failing robots will substantially inform trustworthy HRI design.

Severity Rankings. Failure classifications often come with severity rankings, such as the failure classification by Carlson and Murphy [10]. They classified physical failures according to severity (terminal failure: terminates the system's current mission; non-terminal failures: degrades its ability to perform its mission) and repairability (field repairable: repairable with tools that accompany the system in the field; nonfield repairable: cannot be repaired with tools that accompany the system in the field). For our approach we would like to extend our taxonomy with a severity ranking with respect to the loss of trust. Similarly, to the impact of repetition, data from long-term field trials will be needed in order to add empirical evidence to our taxonomy.

Automated recognition of Trust Loss. As mentioned before, the current logics that allow trust (loss) modeling are fairly simplistic. Furthermore, different cues in user behavior need to be distinguishable to detect trust loss of the user in the system. While automated detection of a failure is the first necessary step in failure mitigation, the next goal should be automated trust loss detection to be able to respond appropriately. As the STRANDS use case has shown, proper recognition and standardized scoring of trust loss could greatly benefit trust research in HRI.

6 CONCLUSION

In this paper, we aimed at consolidating the knowledge we have on trust and trust repair in HRI in a taxonomy with the aim to help fellow researchers developing trustworthy robots according to the state of the art. We aimed at specifically structuring potential failure situations from the robot as well as from the user perspective. Our efforts revealed that empirical research in HRI tries to more and more identify suitable mitigation strategies, but hardly considers the type of failure that caused the trust violation. We argue that a framing of failure situations will have an impact on trust repair and needs to be considered in future studies, but above all in future interaction designs. Moreover, we tried to outline how failure detection could be improved for future HRI, as well as the logics of verification of failure states. Future work in these areas will be essential to actual enable autonomous trust repair in HRI including autonomously generated suitable explanation strategies.

ACKNOWLEDGMENTS

We thank Schloss Dagstuhl — Leibniz Center for Informatics as well as the organizers of Dagstuhl Seminar 19171 on "Ethics and Trust: Principles, Verification and Validation" for bringing us together to produce this work. C. Dixon was partially supported by UKRI Hubs for Robotics and AI in Hazardous Environments EP/R026092 (FAIR-SPACE), EP/R026084 (RAIN), and M. Hanheide by EP/R02572X (NCNR). We thank the Center for Science, Ethics & Public Policy at the University of Delaware for their Open Access funding support.

REFERENCES

[1] Brenna D Argall, Sonia Chernova, Manuela Veloso, and Brett Browning. 2009. A survey of robot learning from demonstration. *Robotics and autonomous systems* 57, 5 (2009), 469–483.

[2] Kenneth J Arrow. 1972. Gifts and exchanges. *Philosophy & Public Affairs* 1, 4 (1972), 343–362.

[3] Markus Bajones, Astrid Weiss, and Markus Vincze. 2016. Help, anyone? a user study for modeling robotic behavior to mitigate malfunctions with the help of the user. *arXiv preprint arXiv:1606.02547* (2016).

[4] Anthony L. Baker, Elizabeth K. Phillips, Daniel Ullman, and Joseph R. Keebler. 2018. Toward an Understanding of Trust Repair in Human-Robot Interaction: Current Research and Future Directions. *ACM Transactions on Interactive Intelligent Systems* 8, 4 (Nov. 2018), 1–30. https://doi.org/10.1145/3181671

[5] Deborah R. Billings, Kristin E. Schaefer, Jessie Y.C. Chen, and Peter A. Hancock. 2012. Human-robot interaction: developing trust in robots. In *Proceedings of the seventh annual ACM/IEEE international conference on Human-Robot Interaction - HRI '12*. ACM Press, Boston, Massachusetts, USA, 109. https://doi.org/10.1145/2157689.2157709

[6] Barry W. Boehm. 1991. Software risk management: principles and practices. *IEEE software* 8, 1 (1991), 32–41.

[7] Daniel J Brooks, Momotaz Begum, and Holly A Yanco. 2016. Analysis of reactions towards failures and recovery strategies for autonomous robots. In *2016 25th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*. IEEE, New York, NY,USA, 487–492.

[8] Drazen Brscić, Hiroyuki Kidokoro, Yoshitaka Suehiro, and Takayuki Kanda. 2015. Escaping from Children's Abuse of Social Robots. In *Proceedings of the Tenth Annual ACM/IEEE International Conference on Human-Robot Interaction (HRI '15)*. ACM, New York, NY, USA, 59–66. https://doi.org/10.1145/2696454.2696468

[9] Maya Cakmak and Leila Takayama. 2014. Teaching people how to teach robots: The effect of instructional materials and dialog design. In *Proceedings of the 2014 ACM/IEEE international conference on Human-robot interaction*. ACM, Toronto, Canada, 431–438.

[10] Jennifer Carlson and Robin R Murphy. 2005. How UGVs physically fail in the field. *IEEE Transactions on robotics* 21, 3 (2005), 423–437.

[11] Cristiano Castelfranchi and Rino Falcone. 1998. Principles of Trust for MAS: Cognitive Anatomy, Social Importance, and Quantification. In *Proceedings of the Third International Conference on Multiagent Systems, ICMAS 1998, Paris, France, July 3-7, 1998*, Yves Demazeau (Ed.). IEEE, New York, NY,USA, 72–79. http://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=5659

[12] Robert B Cialdini. 1993. *Influence: The psychology of persuasion*. Morrow New York, New York, NY, USA.

[13] E. Clarke, O. Grumberg, and D. A. Peled. 2000. *Model Checking*. MIT Press, Cambridge,MA,USA.

[14] Ewart J de Visser, Richard Pak, and Tyler H Shaw. 2018. From 'automation' to 'autonomy': the importance of trust repair in human–machine interaction. *Ergonomics* 61, 10 (2018), 1409–1427.

[15] Francesco Del Duchetto, Ayse Kucukyilmaz, Luca Iocchi, Marc Hanheide, Francesco Del Duchetto, Ayse Kucukyilmaz, Luca Iocchi, and Marc Hanheide. 2018. Do Not Make the Same Mistakes Again and Again: Learning Local Recovery Policies for Navigation From Human Demonstrations. *IEEE Robotics and Automation Letters* 3, 4 (oct 2018), 4084–4091. https://doi.org/10.1109/LRA.2018.2861080

[16] Daniel Clement Dennett. 1989. *The intentional stance*. MIT press, Cambridge,MA,USA.

[17] Munjal Desai, Poornima Kaniarasu, Mikhail Medvedev, Aaron Steinfeld, and Holly Yanco. 2013. Impact of robot failures and feedback on real-time trust. In *Proceedings of the 8th ACM/IEEE international conference on Human-robot interaction*. IEEE Press, Tokyo, Japan, 251–258.

[18] Rino Falcone and Cristiano Castelfranchi. 2001. Social Trust: A Cognitive Approach. In *Trust and Deception in Virtual Societies*, Cristiano Castelfranchi and Yao-Hua Tan (Eds.). Springer, Dordrecht, 55–90. https://doi.org/10.1007/978-94-017-3614-5_3

[19] Joel Feinberg. 1987. *The moral limits of the criminal law. 1, Harm to others*. Oxford University Press, New York, NY,USA.

[20] Richard E. Fikes and Nils J. Nilsson. 1971. STRIPS: A New Approach to the Application of Theorem Proving to Problem Solving. *Artificial Intelligence* 2 (1971), 189–208. Issue 3–4.

[21] M. Fisher. 2011. *An Introduction to Practical Formal Methods Using Temporal Logic*. Wiley, New York, USA.

[22] Maria Fox, Derek Long, and Daniele Magazzeni. 2017. Explainable Planning. In *IJCAI-17 Workshop on Explainable AI*. IJCAI, Melbourne, Australia.

[23] Paul Gainer, Clare Dixon, Kerstin Dautenhahn, Michael Fisher, Ullrich Hustadt, Joe Saunders, and Matt Webster. 2017. CRutoN: Automatic Verification of a Robotic Assistant's Behaviours. In *Critical Systems: Formal Methods and Automated Verification - Joint 22nd International Workshop on Formal Methods for Industrial Critical Systems - and - 17th International Workshop on Automated Verification of Critical Systems, FMICS-AVoCS 2017, Turin, Italy, September 18-20, 2017, Proceedings (Lecture Notes in Computer Science)*, Laure Petrucci, Cristina Seceleanu, and Ana Cavalcanti (Eds.), Vol. 10471. Springer, Turin, Italy, 119–133. https://doi.org/10.1007/978-3-319-67113-0_8

[24] Kathrin Gerling, Denise Hebesberger, Christian Dondrup, Tobias Körtner, and Marc Hanheide. 2016. Robot deployment in long-term care. *Zeitschrift für Gerontologie und Geriatrie* 49, 4 (jun 2016), 288–297. https://doi.org/10.1007/s00391-016-1065-6

[25] Moritz Göbelbecker, Thomas Keller, Patrick Eyerich, Michael Brenner, and Bernhard Nebel. 2010. Coming Up with Good Excuses: What To Do When No Plan Can be Found. In *Proceedings of the 20th International Conference on Automated Planning and Scheduling (ICAPS 2010)*. AAAI Press, Toronto, Canada, 81–88.

[26] Rachel Gockley, Allison Bruce, Jodi Forlizzi, Marek Michalowski, Anne Mundell, Stephanie Rosenthal, Brennan Sellner, Reid Simmons, Kevin Snipes, Alan C Schultz, et al. 2005. Designing robots for long-term social interaction. In *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, New York, NY,USA, 1338–1343.

[27] Victoria Groom, Vasant Srinivasan, Cindy L Bethel, Robin Murphy, Lorin Dole, and Clifford Nass. 2011. Responses to robot social roles and social role framing. In *2011 International Conference on Collaboration Technologies and Systems (CTS)*. IEEE, New York, NY,USA, 194–203.

[28] Ritwik Gupta, Zachary T Kurtz, Sebastian Scherer, and Jonathon M Smereka. 2018. Open Problems in Robotic Anomaly Detection. *CoRR* abs/1809.0 (sep 2018). arXiv:1809.03565 http://arxiv.org/abs/1809.03565

[29] Peter A. Hancock, Deborah R. Billings, Kristin E. Schaefer, Jessie Y. C. Chen, Ewart J. de Visser, and Raja Parasuraman. 2011. A Meta-Analysis of Factors Affecting Trust in Human-Robot Interaction. *Human Factors: The Journal of the Human Factors and Ergonomics Society* 53, 5 (Oct. 2011), 517–527. https://doi.org/10.1177/0018720811417254

[30] Marc Hanheide, Moritz Göbelbecker, Graham S. Horn, Andrzej Pronobis, Kristoffer Sjöö, Alper Aydemir, Patric Jensfelt, Charles Gretton, Richard Dearden, Miroslav Janicek, Hendrik Zender, Geert-Jan Kruijff, Nick Hawes, and Jeremy L. Wyatt. 2017. Robot task planning and explanation in open and uncertain worlds. *Artificial Intelligence* 247 (jun 2017), 119–150. https://doi.org/10.1016/j.artint.2015.08.008

[31] Marc Hanheide, Denise Hebesberger, Tomáš Krajník, Tomas Krajnik, and Others. 2017. The When, Where, and How: An Adaptive Robotic Info-Terminal for Care Home Residents. In *Proceedings of the 2017 ACM/IEEE International Conference on Human-Robot Interaction - HRI '17*. ACM/IEEE, ACM Press, New York, New York, USA, 341–349. https://doi.org/10.1145/2909824.3020228

[32] Nick Hawes, Chris Burbridge, Ferdian Jovan, Lars Kunze, Bruno Lacerda, Lenka Mudrová, Jay Young, Jeremy Wyatt, Denise Hebesberger, Tobias Körtner, others, Rares Ambrus, Nils Bore, John Folkesson, Patric Jensfelt, Lucas Beyer, Alexander Hermans, Bastian Leibe, Aitor Aldoma, Thomas Fäulhammer, Michael Zillich, Markus Vincze, Muhannad Al-Omari, Eris Chinellato, Paul Duckworth, Yiannis Gatsoulis, David C. Hogg, Anthony G. Cohn, Christian Dondrup, Jaime Pulido Fentanes, Tomas Krajník, JoÃčo M. Santos, Tom Duckett, and Marc Hanheide. 2017. The STRANDS Project: Long-Term Autonomy in Everyday Environments. *Robotics and Automation Magazine* (4 2017). http://arxiv.org/abs/1604.04384

[33] Denise Hebesberger, Tobias Koertner, Christoph Gisinger, Juergen Pripfl, and Christian Dondrup. 2016. Lessons learned from the deployment of a long-term autonomous robot as companion in physical therapy for older adults with dementia a mixed methods study. In *2016 11th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. IEEE, Christchurch, New Zealand, 27–34. https://doi.org/10.1109/HRI.2016.7451730

[34] Denise Viktoria Hebesberger, Christian Dondrup, Christoph Gisinger, and Marc Hanheide. 2017. Patterns of Use: How Older Adults with Progressed Dementia Interact with a Robot. In *Proceedings of the Companion of the 2017 ACM/IEEE International Conference on Human-Robot Interaction - HRI '17*. ACM, ACM Press, New York, New York, USA, 131–132. https://doi.org/10.1145/3029798.3038388

[35] Marcel Heerink, Ben Krose, Vanessa Evers, and Bob Wielinga. 2006. The influence of a robot's social abilities on acceptance by elderly users. In *ROMAN 2006-The 15th IEEE International Symposium on Robot and Human Interactive Communication*. IEEE, New York, NY,USA, 521–526.

[36] Roberto Pinillos Herrero, Jaime Pulido Fentanes, and Marc Hanheide. 2018. Getting to Know Your Robot Customers: Automated Analysis of User Identity and Demographics for Robots in the Wild. *IEEE Robotics and Automation Letters* 3, 4 (oct 2018), 3733–3740. https://doi.org/10.1109/LRA.2018.2856264

[37] Andreas Herzig, Emiliano Lorini, Jomi Fred Hübner, and Laurent Vercouter. 2010. A logic of trust and reputation. *Logic Journal of the IGPL* 18, 1 (2010), 214–244. https://doi.org/10.1093/jigpal/jzp077

[38] Thomas Hobbes. 1980. Leviathan (1651). *Glasgow 1974* (1980).

[39] Kevin Anthony Hoff and Masooda Bashir. 2015. Trust in Automation: Integrating Empirical Evidence on Factors That Influence Trust. *Human Factors* 57, 3 (2015), 407–434. https://doi.org/10.1177/0018720814547570 PMID: 25875432.

[40] Robert R Hoffman, Matthew Johnson, Jeffrey M Bradshaw, and Al Underbrink. 2013. Trust in automation. *IEEE Intelligent Systems* 28, 1 (2013), 84–88.

[41] Shanee Honig and Tal Oron-Gilad. 2018. Understanding and resolving failures in human-robot interaction: Literature review and model development. *Frontiers in psychology* 9 (2018), 861.

[42] Ruud Hortensius, Felix Hekele, and Emily S Cross. 2018. The perception of emotion in artificial agents. *IEEE Transactions on Cognitive and Developmental Systems* 10, 4 (2018), 852–864.

[43] Xiaowei Huang, Marta Kwiatkowska, and Maciej Olejnik. 2019. Reasoning About Cognitive Trust in Stochastic Multiagent Systems. *ACM Trans. Comput. Logic* 20, 4, Article 21 (July 2019), 64 pages. https://doi.org/10.1145/3329123

[44] Xiaowei Huang and Marta Zofia Kwiatkowska. 2017. Reasoning about Cognitive Trust in Stochastic Multiagent Systems. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA.*, Satinder P. Singh and Shaul Markovitch (Eds.). AAAI Press, 3768–3774. http://aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/14566

[45] Peter H Kim, Donald L Ferrin, Cecily D Cooper, and Kurt T Dirks. 2004. Removing the shadow of suspicion: the effects of apology versus denial for repairing competence-versus integrity-based trust violations. *Journal of applied psychology* 89, 1 (2004), 104.

[46] Morteza Lahijanian and Marta Kwiatkowska. 2016. Social trust: a major challenge for the future of autonomous systems. In *2016 AAAI Fall Symposium Series*. AAAI Press.

[47] John D. Lee and Katrina A. See. 2004. Trust in Automation: Designing for Appropriate Reliance. *Human Factors* 46, 1 (2004), 50–80.

[48] Min Kyung Lee, Sara Kiesler, Jodi Forlizzi, Siddhartha Srinivasa, and Paul Rybski. 2010. Gracefully mitigating breakdowns in robotic services. In *2010 5th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. IEEE, New York, NY,USA, 203–210.

[49] Roy J Lewicki and Carolyn Wiethoff. 2000. Trust, trust development, and trust repair. *The handbook of conflict resolution: Theory and practice* 1, 1 (2000), 86–107.

[50] Michael Lewis, Katia Sycara, and Phillip Walker. 2018. The Role of Trust in Human-Robot Interaction. In *Foundations of Trusted Autonomy*, Hussein A. Abbass, Jason Scholz, and Darryn J. Reid (Eds.). Springer International Publishing, Cham, 135–159. https://doi.org/10.1007/978-3-319-64816-3_8

[51] Felix Lindner, Robert Mattmüller, and Bernhard Nebel. 2019. Moral Permissibility of Action Plans. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence (AAAI-19)*. AAAI Press, 7635–7642.

[52] Joseph B. Lyons and Paul R. Havig. 2014. Transparency in a Human-Machine Context: Approaches for Fostering Shared Awareness/Intent. In *Virtual, Augmented and Mixed Reality. Designing and Developing Virtual and Augmented Environments*, Randall Shumaker and Stephanie Lackey (Eds.). Springer International Publishing, Cham, 181–190.

[53] Bertram F Malle, Matthias Scheutz, Thomas Arnold, John Voiklis, and Corey Cusimano. 2015. Sacrifice one for the good of many?: People apply different moral norms to human and robot agents. In *Proceedings of the tenth annual ACM/IEEE international conference on human-robot interaction*. ACM, 117–124.

[54] Bertram F Malle, Matthias Scheutz, Jodi Forlizzi, and John Voiklis. 2016. Which robot am i thinking about?: The impact of action and appearance on people's evaluations of a moral robot. In *The Eleventh ACM/IEEE International Conference on Human Robot Interaction*. IEEE Press, 125–132.

[55] Kaitlyn Marinaccio, Spencer Kohn, Raja Parasuraman, and Ewart J De Visser. 2015. A framework for rebuilding trust in social automation across health-care domains. In *Proceedings of the International Symposium on Human Factors and Ergonomics in Health Care*, Vol. 4. SAGE Publications Sage India, New Delhi, India, 201–205.

[56] Roger C. Mayer, James H. Davis, and F. David Schoorman. 1995. An Integrative Model of Organizational Trust. *The Academy of Management Review* 20, 3 (July 1995), 709. https://doi.org/10.2307/258792

[57] Tim Miller. 2019. Explanation in artificial intelligence: Insights from the social sciences. *Artificial Intelligence* 267 (2019), 1–38.

[58] Sandra L Murray, John G Holmes, and Nancy L Collins. 2006. Optimizing assurance: The risk regulation system in relationships. *Psychological bulletin* 132, 5 (2006), 641.

[59] Malin Nordqvist and Jessica Lindblom. 2018. Operators' Experience of Trust in Manual Assembly with a Collaborative Robot. In *Proceedings of the 6th International Conference on Human-Agent Interaction*. ACM, ACM, New York, NY, USA, 341–343.

[60] PL Patrick Rau, Ye Li, and Dingjun Li. 2009. Effects of communication style and culture on ability to accept recommendations from robots. *Computers in Human Behavior* 25, 2 (2009), 587–595.

[61] James Reason. 1990. *Human error*. Cambridge university press.

[62] John K Rempel, John G Holmes, and Mark P Zanna. 1985. Trust in close relationships. *Journal of personality and social psychology* 49, 1 (1985), 95.

[63] Paul Robinette, Ayanna M Howard, and Alan R Wagner. 2015. Timing is key for robot trust repair. In *International Conference on Social Robotics*. Springer, 574–583.

[64] Paul Robinette, Wenchen Li, Robert Allen, Ayanna M. Howard, and Alan R. Wagner. 2016. Overtrust of Robots in Emergency Evacuation Scenarios. In *The Eleventh ACM/IEEE International Conference on Human Robot Interaction (HRI '16)*. IEEE Press, Piscataway, NJ, USA, 101–108. http://dl.acm.org/citation.cfm?id=2906831.2906851

[65] William D Rowe. 1975. *An" Anatomy" of risk*. Environmental Protection Agency.

[66] Caryl E Rusbult, Julie Verette, Gregory A Whitney, Linda F Slovik, and Isaac Lipkus. 1991. Accommodation processes in close relationships: Theory and preliminary empirical evidence. *Journal of Personality and social Psychology* 60, 1 (1991), 53.

[67] Mark Scheeff, John Pinto, Kris Rahardja, Scott Snibbe, and Robert Tow. 2002. Experiences with Sparky, a social robot. In *Socially intelligent agents*. Springer, 173–180.

[68] Maurice E. Schweitzer, John C. Hershey, and Eric T. Bradlow. 2006. Promises and lies: Restoring violated trust. *Organizational Behavior and Human Decision Processes* 101, 1 (Sept. 2006), 1–19. https://doi.org/10.1016/j.obhdp.2006.05.005

[69] Sarah Strohkorb Sebo, Priyanka Krishnamurthi, and Brian Scassellati. 2019. "I Don't Believe You": Investigating the Effects of Robot Trust Violation and Repair. In *2019 14th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. IEEE, 57–65.

[70] Scott A Shappell and Douglas A Wiegmann. 2000. The human factors analysis and classification system–HFACS. (2000).

[71] Neville A Stanton and Paul M Salmon. 2009. Human error taxonomies applied to driving: A generic driver error taxonomy and its implications for intelligent transport systems. *Safety Science* 47, 2 (2009), 227–237.

[72] Paul Thagard. 2019. *Mind-Society: From Brains to Social Sciences and Professions (Treatise on Mind and Society)*. Oxford University Press.

[73] Laurence Thomas. 1987. Friendship. *Synthese* 72, 2 (1987), 217–236.

[74] Amos Tversky and Daniel Kahneman. 1974. Judgment under uncertainty: Heuristics and biases. *science* 185, 4157 (1974), 1124–1131.

[75] Alan R. Wagner, Paul Robinette, and Ayanna Howard. 2018. Modeling the Human-Robot Trust Phenomenon: A Conceptual Framework Based on Risk. *ACM Trans. Interact. Intell. Syst.* 8, 4, Article 26 (Nov. 2018), 24 pages. https://doi.org/10.1145/3152890

[76] Ning Wang, David V. Pynadath, and Susan G. Hill. 2016. Trust Calibration within a Human-Robot Team: Comparing Automatically Generated Explanations. In *11th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. IEEE, New York, NY,USA, 109–116.

[77] M. Webster, D. Western, D. Araiza-Illan, C. Dixon, K. Eder, M. Fisher, and A. Pipe. 2020. A Corroborative Approach to Verification and Validation of Human–Robot Teams. *International Journal of Robotics Research* 39 (2020), 73–99.

[78] Charles Yoe. 2019. *Principles of risk analysis: decision making under uncertainty*. CRC press, Abingdon, UK.