

DOI:10.1145/3433637

**As robots begin to interact closely with humans, we need to build systems worthy of trust regarding the safety and quality of the interaction.**

**BY HADAS KRESS-GAZIT, KERSTIN EDER, GUY HOFFMAN, HENNY ADMONI, BRENNAN ARGALL, RÜDIGER EHLERS, CHRISTOFFER HECKMAN, NILS JANSEN, ROSS KNEPPER, JAN KŘETÍNSKÝ, SHELLY LEVY-TZEDEK, JAMY LI, TODD MURPHEY, LAUREL RIEK, AND DORSA SADIGH**

# Formalizing and Guaranteeing Human-Robot Interaction

ROBOT CAPABILITIES ARE maturing across domains, from self-driving cars to bipeds to drones. As a result, robots will soon no longer be confined to safety-controlled industrial settings; instead, they will directly interact with the general public. The growing field of human-robot interaction (HRI) studies various aspects of this scenario, from social norms to collaborative manipulation to human-robot teaming, and more.

Researchers in HRI have made great strides in developing models, methods, and algorithms for robots acting with and around humans,<sup>29</sup>

but these “computational HRI” models and algorithms do not generally come with formal guarantees and constraints on their operation. To enable human-interactive robots to move from the lab to real-world deployments, we must address this gap.

Demonstrating trustworthiness in various forms of automation through formal guarantees has been the focus of validation, verification, and synthesis efforts for years. For instance, aircraft control systems must meet guarantees—such as correctly handling transitions between discrete modes of operation (take-off, cruise, landing)—while simultaneously providing a guarantee on *safety* (for example, not being in both take-off and landing modes at the same time) and *liveness*, the ability to eventually achieve a desirable state (for instance, reaching cruise altitude).

Formal methods, such as model checking and theorem proving, play a central role in helping us understand when we can rely on automation to do what we have asked of it. Formal methods can be used to create correct-by-construction systems, provide proofs that properties hold, or find counterexamples that show when automation might fail.

Formalizing HRI can enable the creation of trustworthy systems and, just as importantly, support explicit reasoning about the context of guarantees. First, writing formal models of aspects of HRI would enable verification, validation, and synthesis, thus

## » key insights

- **To obtain formal guarantees, specifications that capture the requirements and constraints of “good” HRI are essential.**
- **Our analysis of different HRI domains highlights the importance of understanding human behavior for successful HRI.**
- **HRI brings unique challenges and opportunities for formal methods, from defining specifications to dealing with adaptation and variability in humans.**

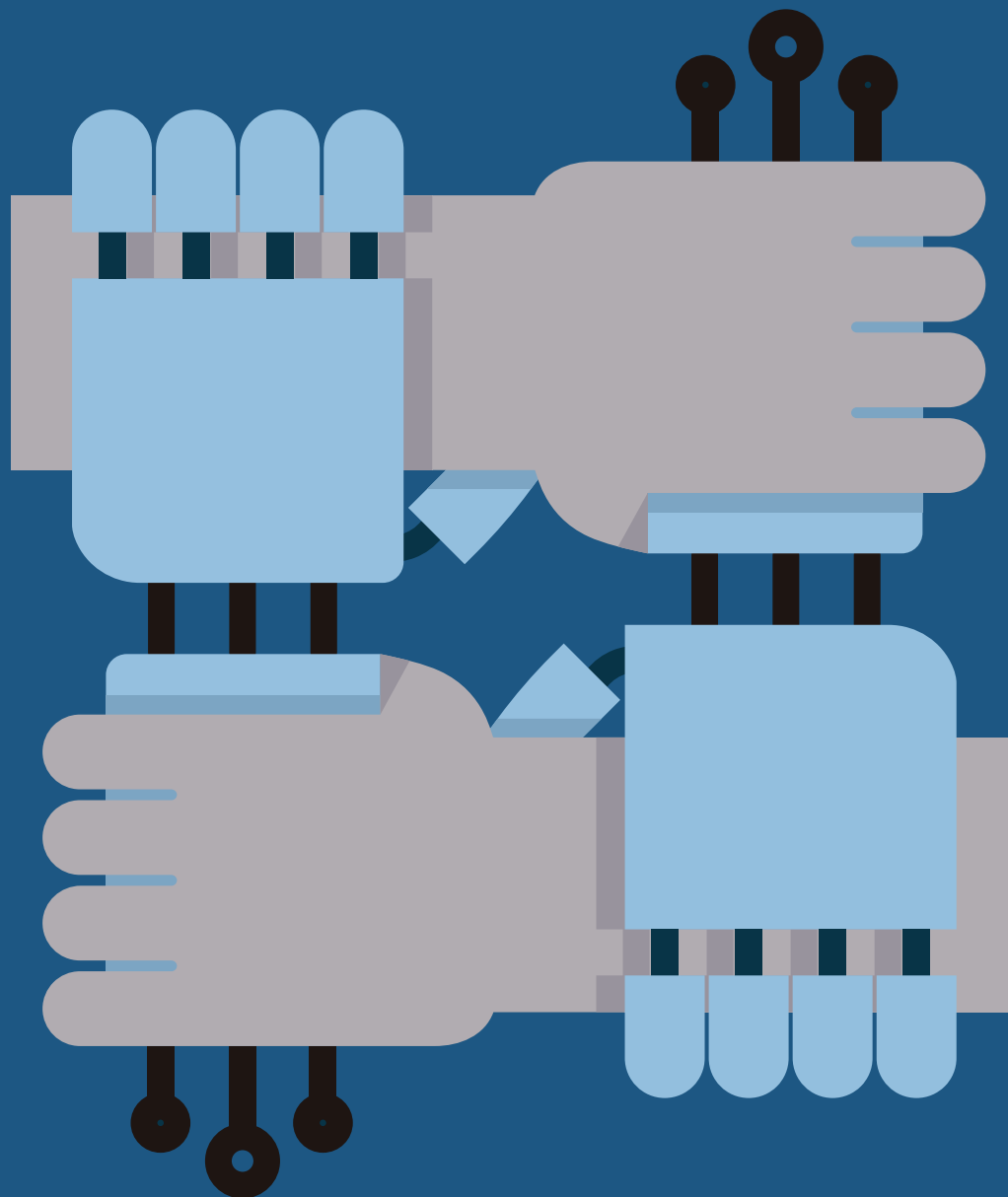


IMAGE BY PEDRO NEVES DESIGN

providing some guarantees on the interaction. Second, it is unrealistic to verify a complete human-robot system due to the inherent uncertainty in physical systems, the unique characteristics and behaviors of people, and the interaction between systems and people. Thus, a formal model requires us to articulate explicit assumptions regarding the system, including the human, the robot, and the

environments in which they are operating. Doing so exposes the limits of the provided guarantees and helps in designing systems that degrade gracefully when assumptions are violated.

In this article, we discuss approaches for creating trustworthy systems and identify their potential uses in a variety of HRI domains. We conclude with a set of research challenges for the community.

### **Techniques for Demonstrably Trustworthy Systems**

We divide the techniques for gaining confidence in the correctness of a system into four approaches: synthesis of correct-by-construction systems, formal verification at design time, runtime verification or monitoring, and test-based methods. Common to all of these approaches is the need to articulate specifications—descriptions of


what the system should and should not do. Specifications typically include both safety and liveness properties and are defined in a formal language, for example temporal logics over discrete and/or continuous states, or satisfiability modulo theory (SMT) formulas (for example in Clarke et al.<sup>8</sup>).

The four approaches outlined below are listed in decreasing order of exhaustiveness and, as a result, computational complexity. Less exhaustive approaches can typically handle more complex systems at a greater level of realism. Synthesis is the most computationally expensive approach and requires the coarsest abstraction, but it can automatically create a system with guarantees. Test-based methods, however, can handle the most complex systems but do not provide formal guarantees regarding the satisfaction of the specifications. In practice, a combination of techniques is required, as no single technique can be relied upon on its own.<sup>31</sup>


**Synthesis** is the process of automatically generating a system from the specifications. In the context of robotics, there are different techniques for doing so,<sup>14</sup> including offline generation of state machines or policies satisfying discrete and probabilistic temporal logic specifications, online receding horizon optimization for continuous temporal logics, and online optimization with SMT solvers.

**Formal verification** techniques span methods that exhaustively explore the system (model checking, reachability analysis<sup>8</sup>) to those that reason about the system using axioms and proof systems (theorem proving<sup>10</sup>). Techniques vary from deterministic, worst-case analysis to probabilistic reasoning and guarantees, and from discrete spaces to continuous ones. Such methods are typically applied at design time and either determine that the specification is met in every possible system behavior or provide a counterexample—a system execution that violates the specification—which may then be used to further refine the design or the specification.

**Runtime monitoring** is the process of continuously checking system accuracy during execution using *monitors* that check specifications, either



**When the interaction involves shared human-robot control, equally important to the idea of humans trusting the robot is the notion of whether and to what extent the robot can trust the human.**



created manually or automatically through synthesis.<sup>20</sup> This type of verification is, in a sense, the most lightweight way to integrate formal methods into a design. It does not alter the design; it enables the detection of failures or deviations from expected/formalized behavior, allowing the robot to be shut down or switched into a safe mode. An additional benefit of runtime-checkable specifications is that they allow us to “probe” the system at design time using methods such as statistical model checking.<sup>15</sup>

**Test-based** methods complement formal methods during verification and validation. In particular, simulation-based testing<sup>2</sup> can expose the system under test to more realistic stimuli than the often highly abstract scenarios that can be verified formally. From a performance point of view, simulation-based testing can reach verification goals faster and with less effort than conventional testing in the real world. *Coverage* is a measurement of verification progress, allowing engineers to track the variety of tests used and determine their effectiveness in achieving verification goals. Assertion monitors act as test oracles, much like the monitors used for runtime verification. Model-based testing is a specific technique that asserts the conformance of a system under test to a given formal model of that system.<sup>30</sup> This is particularly important when guarantees or code generation rely on the correctness of a model.

Validation, verification, and synthesis techniques are always related to given specifications. These specifications can never cover the full behavior of a physical system in the world; rather, they include assumptions and abstractions to make the problem tractable. Therefore, guarantees are provided with respect to the specification, enabling us to gain confidence in the system’s overall correctness, narrow down the sources of problems, and understand the constraints that limit deployment.

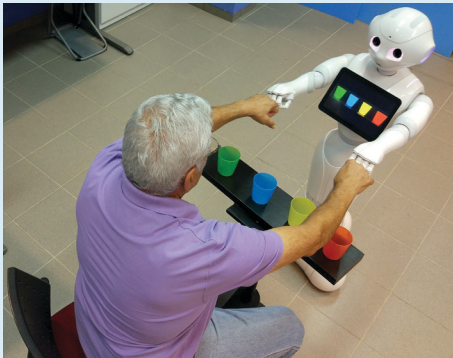
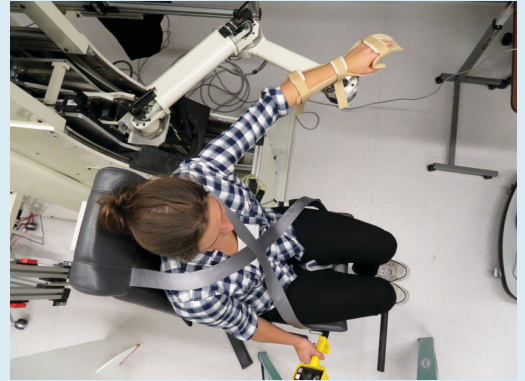
### **HRI Domains and Their Unique Challenges**

Many HRI domains could benefit from formal methods, and each domain brings about unique challenges:

**Domains of HRI that could benefit from formal methods.**

Clockwise from top left, physical HRI (construction), physical HRI in healthcare (rehabilitation), autonomous driving, social HRI, and cognitive healthcare.

Images courtesy of (clockwise from top left): Michael Suguitan, Kathleen Fitzsimons, Wendy Ju, Yaxin Hu, and The Levy-Tzedek Lab.



**Physical HRI** involves systems in which the physical states of an automation interact with the physical states of a human;<sup>4</sup> for example, a robotic wheelchair carrying a person or a construction-assistant robot and a person carrying a heavy load together. In addition to physical states interacting, their *internal* states interact, since both the robot and human often have a model of the task they are working on to achieve as well as a model of each other.

For example, in a setting where rehabilitation robots assist an individual with motion, the robot may be responsible for physical safety (keeping someone upright) while simultaneously maximizing therapy benefit, requiring it to stay out of the way as much as possible. Thus, the system is tasked to assist, but not to over-assist. This fundamental tension between the two purposes of the automation with respect to the human leads to challenging questions in terms of specification (for example, how does one articulate the notion of safety while avoiding over-assisting?) and verification (for instance, how does one prove that the control methods satisfy the specification?).

**Healthcare Robotics:** There are a variety of robots being developed to assist people with activities of daily living, including physical mobility, manipulation, medicine reminders, and cognitive support. Robots are also developed to support clinicians, caregivers, and other stakeholders in healthcare contexts.<sup>27</sup>

For example, physically assistive robots, such as exoskeletons and robotic prostheses, can help individuals perform movements, such as walking and reaching. Socially assistive robots can help individuals engage in positive health behaviors, such as exercise and wellness.<sup>23</sup> People have different abilities and disabilities that may change over short- and long-term horizons. Therefore, modeling a person's ability and personalizing the system is crucial for creating successful HRI systems in the healthcare domain.

**Autonomous Driving:** Recent years have seen significant advances in autonomous driving. As these fully or semi-autonomous vehicles appear on the road, challenges arise due to interactions with humans. Humans, in the context of autonomous driving, fall into three main categories: drivers or riders in the autonomous vehicle,

drivers of other vehicles around the autonomous car, and pedestrians or bicyclists interacting with autonomous vehicles on roads.

An obvious specification in this domain is safety—no collisions. However, that specification is not enough. When merging onto a highway, the safest course of action is to wait until no other vehicles are nearby. On busy roads this is not a reasonable course of action. Therefore, the specification needs to go beyond addressing the challenges of driving a single vehicle by formalizing desirable behavior when cars interact with other vehicles and road users.<sup>28</sup> The challenges of this domain are to model and validate acceptable road behavior; reason about the expected and unexpected behavior of people in the above categories; and provide certification, diagnosis, and repair techniques that will enable autonomous vehicles to drive on our roads.

**Social Collaboration:** In addition to the contexts listed above, there are many instances in which humans and robots will engage in predominantly social, rather than physical, interactions.<sup>5</sup> For example, information-kiosk robots at an airport might engage

people in conversations to get them to their destinations.

Social collaborations across many domains can be characterized by the use of social actions, such as verbal and nonverbal communication, to achieve a shared goal. Social collaboration typically requires the agents involved to maintain a *theory of mind* about their partners, identifying what each agent believes, desires, and aims to achieve. In social collaboration, it is important that the robot follows social norms and effective collaboration practices, for example not interrupting the speaker and providing only true and relevant information.<sup>9</sup> If a robot fails to follow such conventions, it risks failing at the collaboration due to lack of trust or other social effects. One major challenge in formalizing social collaborations is how to encode social norms and other behavior limitations as formal constraints. Researchers interested in verifying or synthesizing social collaborations will have to identify which social behaviors and which elements of the task are important for the collaboration to succeed.

### Work in Formalizing HRI

Researchers in computational HRI<sup>29</sup> have developed models for human behavior, for human-robot collaboration and interaction, and algorithms that have been demonstrated in various HRI domains. Whereas these approaches are evaluated qualitatively and quantitatively, the HRI research community has not often formalized what constitutes correct behavior. Generally speaking, there are very few examples of formal specifications, or algorithms that can verify or synthesize such specifications.

In the past few years, collaborations between HRI researchers and researchers studying formal methods, verification, and validation have begun to address the challenge of formalizing specifications and creating demonstrably trustworthy HRI systems. Some efforts have explored linear temporal logic as a formalism to capture and verify norms in an interaction<sup>25</sup> and to synthesize human-in-the-loop cyber-physical systems.<sup>21</sup> Other examples include using satisfiability modulo theories for encoding social navigation specifications,<sup>7</sup> signal temporal logic for handover behavior,<sup>16</sup> and automata-based

assertion monitors for robot-to-human handover tasks.<sup>3</sup>

Other researchers have focused on socio-cyber physical systems, for instance by including human factors—ranging from specific roles of humans, their intentions, legal issues, and levels of expertise—into cyber-physical systems.<sup>6</sup> Other work models an assisted-living scenario as a Markov decision process,<sup>22</sup> making use of the probabilistic model-checker PRISM.<sup>17</sup>

### Challenges for the Research Community

Work described above suggests the promise of introducing formal methods and techniques into HRI domains. That said, creating and reasoning about trustworthy HRI requires addressing HRI's unique aspects and rethinking current approaches to system verification, validation, and synthesis. In this section, we distill three unique aspects of HRI research posing a challenge for formal methods: designing useful HRI specifications, dealing with expected human adaptation to the automated system, and handling the inherent variability of humans. For each challenge domain, we identify high-priority research directions that could drive progress toward creating trustworthy HRI systems.

**Designing formal HRI specifications:** Whenever verifying, testing, or synthesizing a system, one needs to formalize the system by defining the state space of the model and the specification of interest. For example, in the context of autonomous cars obeying the law and social conventions, the state space may include the position and velocity of the car and any other cars in the environment. The specification may represent a requirement of the form, ‘the car never exceeds the speed limit and always maintains a safe distance from all other cars.’ In the context of HRI, designing useful specifications raises several research questions:

► **What should be the space of specifications?** In HRI, simply modeling the physical state of the robot and the human is usually not enough. The physical state does not capture requirements such as avoiding over-assisting a person or maintaining social and cultural norms. We need to create richer spaces that enable the writing of such

specifications while balancing the complexity of the algorithms that will be used for verification and synthesis in these spaces.

► **How do we write specifications that capture trust?** A human will only trust a robot to react in a safe way if it obviously and demonstrably does so. Hence, the robot needs to not only be safe but also be *perceived* as safe, which may require a considerable safety margin. On the other hand, when the interaction involves shared human-robot control, equally important to the idea of humans trusting the robot is the notion of whether and to what extent the robot can trust the human. This plays a role in determining under what circumstances the robot should step in and in what manner. Particularly in safety-critical scenarios, and when the robot is filling a gap in the human's own capabilities, reasoning about trust in the human is key. Critical factors are to measurably assess the human's ability to actually perform the task, and the human's current state, for instance accounting for levels of fatigue. These notions of trust go beyond typical safety and liveness specifications and require specification formalisms that can capture them.

► **What should be the definition of failure?** Beyond failure with respect to physical safety, which is well studied in the literature, interaction failures may have varying impacts. A small social gaffe, such as intruding on personal space, may not be an issue, but a large mistake, such as dropping a jointly manipulated object, might have a long-term effect on interaction. We need to be able to define specifications that capture the notion of social failure and develop metrics or partial orders on such failures, so that the systems can fail gracefully.

► **How can we formalize the human's behavior during an interaction?** A common technique in verification is assume-guarantee reasoning, where a system's behavior is verified only under the assumption that its input satisfies a well-defined specification. If the input violates the assumption, the system behavior is no longer guaranteed. Given our understanding and observations of human-human and human-robot interactions, a challenge for synthesizing and verifying HRI is to


formalize assumptions on the behavior of the human—who provides the input of the HRI system—in a way that supports verification, is computationally tractable, and captures the unique characteristics of humans.

**Adapting to human adaptation:** During interaction, humans and robots will engage in mutual adaptation processes.<sup>12</sup> For example, people become less cautious operators of machines (cutting corners, giving a narrower berth to obstacles) as they become more familiar with them. Therefore, any models used to represent the interaction and reason about it must capture this adaptation. To complicate matters, the temporal adaptation may occur at different time scales: short time scales, such as morning vs. evening fatigue, and longer time scales, such as functional ability improvement or deterioration over months.<sup>11,12</sup> Changing models in and of themselves makes formalizing HRI more complicated, but it is the diversity of the ways humans adapt to a task and a teammate that makes their accurate modeling even more challenging. This property brings up the following research challenges:


► **Which mathematical structures can capture non-stationary models?** Mutual adaptations are common in human-human interaction. For example, humans build conventions when communicating with each other through repeated interactions using language or sketches.<sup>32</sup> Studying these interactions and formalizing them can form the basis for new HRI models. When developing such models, an important consideration is how to capture the different time scales of adaptation.

► **How can the robot detect and reason about the human's adaptation?** As humans adapt to the interaction, their behavior (and thus the input to the interaction) may change. For example, people may become less emotionally expressive as the novelty of the interaction wears off, or they may give less control input as they trust the autonomy of the system more. This creates a challenge at runtime when a robot is attempting to ascertain how the human adapted. We need to develop runtime verification algorithms that can detect such adaptation and influence the interaction.

► **How to model feedback loops?** As the robot and the human adapt to each



## Modeling a person's ability and personalizing the system is crucial for creating successful HRI systems in the healthcare domain.



other, it is important to reason about the positive and negative feedback loops that emerge and their effect on the resulting interaction. These feedback loops can take the human-robot systems to desirable or undesirable equilibria. For example, the difference between driving cultures around the world may be explained by repeated interactions between drivers causing behavioral feedback loops, leading to emergent locally distinct conventions. We need to study the long-term behavior of repeated interactions and adaptations and verify the safety of the resulting emergent behaviors.

**Variability among human interactants:** While we can reasonably assume that the model of a particular type of robot is the same for all robots of that type, there does not exist a model of a “typical” human—one size does not fit all. Even identifying the proper parameters or family of parameters that encapsulate the types of variability in people is a seemingly impossible task. People differ across backgrounds, ages, and abilities, which raises the important question of how much to personalize the model and specification to a specific individual or population:

► **Can we identify general specifications for which one, simple human model is enough?** Is it possible to create a basic, human-centric and application-agnostic model of human behavior that indicates a basic specification, such as loss of engagement of a human in the interaction? Such a generic model can detect behavior outside the expected, for example distraction or lack of attention, and could be used to trigger safety measures irrespective of the specific application area. A current example for such a model is used in driver assist systems; they measure where the driver is looking, suggesting the driver take a break if they detect staring or lack of eye movement—universal signs of sleepiness.

► **What levels of personalization are needed?** Refining the research question above, it is important to study not only the formalisms that allow models and specifications to be personalized but also to what extent personalization is required for smooth interaction, what are the trade-offs between the complexity of the model and improved interaction, and what metrics enable

reasoning about the trade-offs. For this purpose, models of mental representations (for example, levels of cognitive control for error-free decision-making<sup>26</sup>) could be useful.


► **How can we model the human’s ability level?** The interaction should be appropriate for the ability level of the person. When humans are better off completing a task on their own, too much assistance is not desirable; for example, in therapeutic and educational settings. In other cases, too little assistance can be frustrating and lead to disengagement. It is important to model both the ability and the modes of interactions that are most appropriate for each task.

► **How do we formalize experiential considerations?** People from different backgrounds may have different assumptions<sup>24</sup> and expectations<sup>19</sup> from robots and may perceive the interaction with the robot differently. Since meeting user expectations is important for fostering trust between the human and the robot,<sup>13,18</sup> the personalization of the interaction should consider the experiential background of the user, who may expect the robot to be, for example, more assertive and active, or more reserved and passive.

**Conclusion**

As robots begin to interact closely with humans, we need to build systems worthy of trust regarding both the safety and the quality of the interaction. To do so, we have to be able to formalize what a “good” interaction is, and we need algorithms that can check that a given system produces good interactions or can even synthesize such systems.

To make progress, we must first acknowledge that a human is not another dynamic physical element in the environment, but has beliefs, goals, social norms, desires, and preferences. To address these complexities, we must develop models, specifications, and algorithms that use our knowledge about human behavior to create demonstrably trustworthy systems. In this article, we identified a number of promising research directions and we encourage the HRI and formal methods communities to create strong collaborations to tackle these and other questions toward the goal of trustworthy HRI.

**Acknowledgment.** This article is a result of fruitful discussions at the Dagstuhl seminar on Verification and Synthesis of Human-Robot Interaction.<sup>1</sup> The authors thank all fellow participants and the Schloss Dagstuhl—Leibniz Center for Informatics, for their support. 

**References**

1. Alami, R. et al. Verification and synthesis of human-robot interaction (Dagstuhl Seminar 19081). *Dagstuhl Reports* 9, 2 (2019), 91–110. <https://doi.org/10.4230/DagRep.9.2.91>.
2. Araiza-Illan, D. et al. Coverage-driven verification—An approach to verify code for robots that directly interact with humans. *Hardware and Software: Verification and Testing* (2015), 69–84.
3. Araiza-Illan, D. et al. Systematic and realistic testing in simulation of control code for robots in collaborative human-robot interactions. *Towards Autonomous Robotic Systems Conference* (2016), 20–32.
4. Argall, B.D. and Billard, A.G. A survey of tacitile human-robot interactions. *Robotics and Autonomous Systems* 58, 10 (Oct. 2010), 1159–1176. <https://doi.org/10.1016/j.robot.2010.07.002>.
5. Breazeal, C. et al. Social robotics. *Springer Handbook of Robotics*, Springer, (2016), 1935–1972.
6. Calinescu, R. et al. Socio-cyber-physical systems: Models, opportunities, open challenges. *2019 IEEE/ACM 5th Intern. Wkshp. Soft. Eng. for Smart Cyber-Physical Systems* (2019), 2–6.
7. Campos, T. et al. SMT-based control and feedback for social navigation. *Intern. Conf. Robotics and Automation, 2019, Montreal, QC, Canada*, 5005–5011.
8. Clarke, E.M. et al. eds. *Handbook of Model Checking*, Springer (2018).
9. Grice, H.P. Logic and conversation. *Syntax and Semantics, Vol. 3: Speech Acts*. P. Cole and J.L. Morgan, eds. Academic Press. (1975) 41–58.
10. Hoare, C.A.R. An axiomatic basis for computer programming. *Commun. ACM* 12, 10 (Oct. 1969), 576–580. <https://doi.org/10.1145/363235.363259>.
11. Hoffman, G. Evaluating fluency in human–robot collaboration. *IEEE Transactions on Human-Machine Systems* 49, 3 (2019), 209–218.
12. Iqbal, T. and Riek, L.D. Human-robot teaming: Approaches from joint action and dynamical systems. *Humanoid Robotics: A Reference* (2019), 2293–2312.
13. Kellmeyer, P. et al. Social robots in rehabilitation: A question of trust. *Science Robotics* 3, 21 (2018). <https://doi.org/10.1126/scirobotics.aat1587>.
14. Kress-Gazit, H. et al. Synthesis for robots: Guarantees and feedback for robot behavior. *Ann. Review of Control, Robotics, and Auton. Systems* 1, 1 (2018). <https://doi.org/10.1146/annurev-control-060117-104838>.
15. Kretínský, J. Survey of statistical verification of linear unbounded properties: Model checking and distances. *Leveraging Applications of Formal Methods, Verification and Validation: Foundational Techniques* (2016), 27–45.
16. Kshirsagar, A. et al. Specifying and synthesizing human-robot handovers. In *Proc. of the IEEE/RSJ Intern. Conf. on Intelligent Robots and Systems* (2019).
17. Kwiatkowska, M. et al. PRISM 4.0: Verification of probabilistic real-time systems. In *Proc. of the 23rd Intern. Conf. on Computer-Aided Verification* (Berlin, Heidelberg, 2011), 585–591.
18. Langer, A. et al. Trust in socially assistive robots: Considerations for use in rehabilitation. *Neuroscience & Biobehavioral Reviews* 104 (2019), 231–239. <https://doi.org/https://doi.org/10.1016/j.neubiorev.2019.07.014>.
19. Lee, H.R. et al. Cultural design of domestic robots: A study of user expectations in Korea and the United States. *2012 IEEE RO-MAN: The 21st IEEE Intern. Symp. on Robot and Human Interactive Communication* (2012), 803–808.
20. Leucker, M. and Schallhart, C. A brief account of runtime verification. *The J. Logic and Algebraic Programming* 78, 5 (2009), 293–303. <https://doi.org/http://dx.doi.org/10.1016/j.jlap.2008.08.004>.
21. Li, W. et al. Synthesis for human-in-the-loop control systems. *Tools and Algorithms for the Construction and Analysis of Systems—20th Intern. Conf.* (2014), 470–484.
22. Mason, G. et al. Assurance in reinforcement learning using quantitative verification. *Advances in*

*Hybridization of Intelligent Methods: Models, Systems and Applications*. I. Hatzilygeroudis and V. Palade, eds. Springer International Publishing, 71–96.

23. Mataric, M.J. and Scassellati, B. Socially assistive robotics. *Springer Handbook of Robotics*. Springer (2016), 1973–1994.
24. Nomura, T. Cultural differences in social acceptance of robots. *26th IEEE Intern. Symp. on Robot and Human Interactive Communication (RO-MAN)* (Aug. 2017), 534–538.
25. Porfiro, D. et al. Authoring and verifying human-robot interactions. In *Proc. of the 31st Ann. ACM Symp. on User Interface Software and Tech.* (2018), 75–86.
26. Rasmussen, J. Mental models and the control of action in complex environments. *Selected Papers of the 6th Interdisciplinary Wkshp. on Informatics and Psychology: Mental Models and Human-Computer Interaction 1* (NLD, 1987), 41–69.
27. Riek, L.D. Healthcare robotics. *Commun. ACM* 60, 11 (2017), 68–78. <https://doi.org/10.1145/3127874>.
28. Sadigh, D. et al. Planning for cars that coordinate with people: Leveraging effects on human actions for planning and active information gathering over human internal state. *Autonomous Robots (AURO)* 42, 7 (Oct. 2018), 1405–1426.
29. Thomaz, A. et al. Computational human-robot interaction. *Found. Trends Robotics* 4, 2–3 (Dec. 2016), 105–223. <https://doi.org/10.1561/23000000049>.
30. Tretmans, G.J. Test generation with inputs, outputs and repetitive quiescence. *Centre for Telematics and Information Technology (CTIT)*.
31. Webster, M. et al. A corroborative approach to verification and validation of human–robot teams. *The Intern. J. Robotics Research* 39, 1 (2020), 73–99. <https://doi.org/10.1177/0278364919883338>.
32. Wilkes-Gibbs, D. and Clark, H.H. Coordinating beliefs in conversation. *J. Memory and Language* 31, 2 (1992), 183–194.

**Hadas Kress-Gazit** is the Geoffrey S.M. Hedrick Sr. Professor in the Sibley School of Mechanical and Aerospace Engineering at Cornell University, Ithaca, NY, USA.

**Kerstin Eder** is a professor of Computer Science at the University of Bristol, UK.

**Guy Hoffman** is an associate professor and the Mills Family Faculty Fellow in the Sibley School of Mechanical and Aerospace Engineering at Cornell University, Ithaca, NY, USA.

**Henny Admoni** is the A. Nico Habermann Assistant Professor of Robotics at Carnegie Mellon University, Pittsburgh, PA, USA.

**Brenna Argall** is an associate professor of Mechanical Engineering, Computer Science, and Physical Medicine & Rehabilitation at Northwestern University, Evanston, IL, USA.

**Rüdiger Ehlers** is a professor for Embedded Systems at Clausthal University of Technology in Clausthal-Zellerfeld, Germany.

**Christoffer Heckman** is an assistant professor and the Pankove Faculty Fellow of Computer Science at the University of Colorado, Boulder, CO, USA.

**Nils Jansen** is an assistant professor with the Institute for Computing and Information Sciences at Radboud University Nijmegen, The Netherlands.

**Ross Knepper** is a roboticist and computer scientist specializing in the algorithmic aspects of autonomous robots and human-robot systems.

**Jan Křetínský** is a professor of Computer Science at the Technical University of Munich, Germany.

**Shelly Levy-Tzedek** is an associate professor at Ben-Gurion University, Beersheba, Israel.

**Jamy Li** is an assistant professor in Industrial Engineering at Ryerson University in Toronto, Canada.

**Todd Murphey** is a professor of Mechanical Engineering at Northwestern University in Evanston, IL, USA.

**Laurel Riek** is associate professor of Computer Science and Engineering at the University of California San Diego, USA.

**Dorsa Sadigh** is an assistant professor of Computer Science at Stanford University, Stanford, CA, USA.