Courtesy, John Carff, IHMC

No AI Is an Island: The Case for Teaming Intelligence

Matthew Johnson, Alonso H. Vera

■ *The purpose of this article is to draw attention to an aspect of intelligence that has not yet received significant attention from the AI community, but that plays a crucial role in a technology's effectiveness in the world, namely teaming intelligence. We propose that AI will reach its full potential only if, as part of its intelligence, it also has enough teaming intelligence to work well with people. Although seemingly counterintuitive, the more intelligent the technological system, the greater the need for collaborative skills. This paper will argue why teaming intelligence is important to AI, provide a general structure for AI researchers to use in developing intelligent systems that team well, assess the current state of the art and, in doing so, suggest a path forward for future AI systems. This is not a call to develop a new capability, but rather, an approach to what AI capabilities should be built, and how, so as to imbue intelligent systems with teaming competence.*

The purpose of this article is to draw attention to an aspect of intelligence that has not yet received significant attention from the AI community, but that plays a crucial role in a technology's (and a person's) effectiveness in the world, namely teaming intelligence. Over the past decade, there have been many successful attempts to apply technology to increasingly complex domains — domains once reserved almost exclusively for human effort. It is rare that an evening goes by without a news report of some event in the field of autonomous drones or self-driving cars or even the dangers of AI. Newly refined techniques and improved computing power have propelled AI into the forefront of both the media and human imagination once again. Today's excitement about AI is largely driven by demonstrations of AI making inroads into areas heretofore untouched by machine intelligence. Some examples include the Google (now Waymo) self-driving cars driving the streets of Mountain View, Watson defeating human opponents in the game show Jeopardy, DeepMind's AlphaGo beating the world champion at the ancient game of Go, and digital assistants like Siri and Alexa becoming ubiquitous. Additionally, a host of less conspicuous examples of AI success are making an impact in finance, commerce, marketing, and customer experience. On the heels of many success stories, it is important for researchers and developers to be identifying what gaps and hurdles remain to the successful development of intelligent technologies. These gaps and hurdles will characterize the nature of foundational and applied research that needs to evolve in order to develop advanced intelligent technologies that deliver on the promises of reduced cost, enhanced performance, and improved safety. This article will argue that a lack of teaming intelligence is one of the most prominent gaps in intelligent systems.

The gap in teaming intelligence is an important one to emphasize because it can impact system performance, resilience, and even viability. Understanding the challenges faced by AI, or any technology, is not just about making better AI, but it is also about enabling AI to achieve its full potential and deliver on its promised benefits. This seemingly counterintuitive effect is well characterized in the autonomy paradox (Blackhurst, Gresham, and Stone 2011), which describes the experience of the Department of Defense in which investments in autonomy have resulted in increased rather than decreased operational costs. Without adequately addressing teaming intelligence, technologies can end up making the job more difficult — requiring more humans, more training, and more expertise.

As we pursue more advanced intelligent technologies, it is important to remember that *no AI is an island*. Technology does not work in isolation from people. In fact, technology thrives most when successfully woven into human work practice. Managing this integration requires teaming intelligence. What is common knowledge for most human-centered design specialists, but not always for those outside this discipline, is that the growth of sophistication in machine capabilities must go hand in hand with the growth of sophistication in human-machine interaction capabilities. Machines do not automatically get simpler to use because they have gotten smarter. Indeed, just the opposite is usually true. This correlation can be seen in more advanced commercial airliners, which typically require longer training times for type ratings than their predecessors did. More intelligent capabilities inevitably require correlated teaming capability enhancements. As such, we propose that AI will reach its full potential only if, as part of its intelligence, it also has enough teaming intelligence to work well with people. This is not simply a call to develop a new capability. Teaming itself is not an isolatable, unitary capability that needs to be developed as an add-on to systems. Rather, it should be viewed as an approach to what AI capabilities should be built, and how, so as to imbue intelligent systems with teaming competence.

What Is Teaming Intelligence?

Modern theories of general intelligence provide various categories of intelligence, but none adequately capture the knowledge, skills, and strategies necessary to team effectively. For example, one of Gardner's nine types of intelligence is interpersonal (Gardner 1998) and Adam's list of 89 general intelligence competency areas includes social interaction (Adams et.al. 2012). However, most general intelligence categories have an aspect related to teaming intelligence and can be assessed with respect to how the particular competency applies to oneself, the environment, and team members. Thus teaming intelligence permeates general intelligence.

The literature on the study of human teams provides more sophisticated models of teamwork. These models tend to be lists of characteristics, properties, or behaviors. For example, one set of competencies from Baker, Day, and Salas (2006) includes team leadership, backup behavior, mutual performance monitoring, communication, adaptability, shared mental models, mutual trust, and team orientation. Teamwork categories, characteristics, and properties vary from model to model, but the one concept that is consistent throughout is the importance of *interdependence*. This truth is both invariant across all domains and fundamental to teaming. As such, we propose that "teaming intelligence" involves knowledge, skills, and strategies with respect to managing interdependence. The knowledge is an understanding of interdependencies within the work and among the team members. The skills are having the supporting mechanisms to participate in interdependent activity, such as being capable of observing one another's state, sharing information, or requesting assistance. The strategy is about using the knowledge to exploit the existing skills in order to intelligently manage the interdependencies with the purpose of producing effective teaming, such as knowing what information to share and when to request assistance.

To better intuit the concept of interdependence, consider the example of playing the same sheet of music as a solo versus playing it as a duet. Although the music is the same, the processes involved are very different (Clark 1996). The difference is that the process of playing music as a duet requires ways to support the interdependence between the players. Understanding, supporting, and exploiting interdependence is what teaming intelligence is all about.

Success in a duet requires not only execution of the musical score (that is, individual competency), but also the extra work of coordinating with someone else. Such work includes a knowledge of the coordination needs and possession of the mechanisms by which to achieve coordination, as well as the reasoning to perform the necessary coordination. Intelligently managing the interdependencies of this extra work is enabled by what is being called *teaming intelligence*. Many human activities (perhaps most) are more like a duet than a solo. AI must be competent with this type of activity because failure of either party in the duet will hinder or prevent success of the duet.

How AI Is (Not) at Odds with Teaming

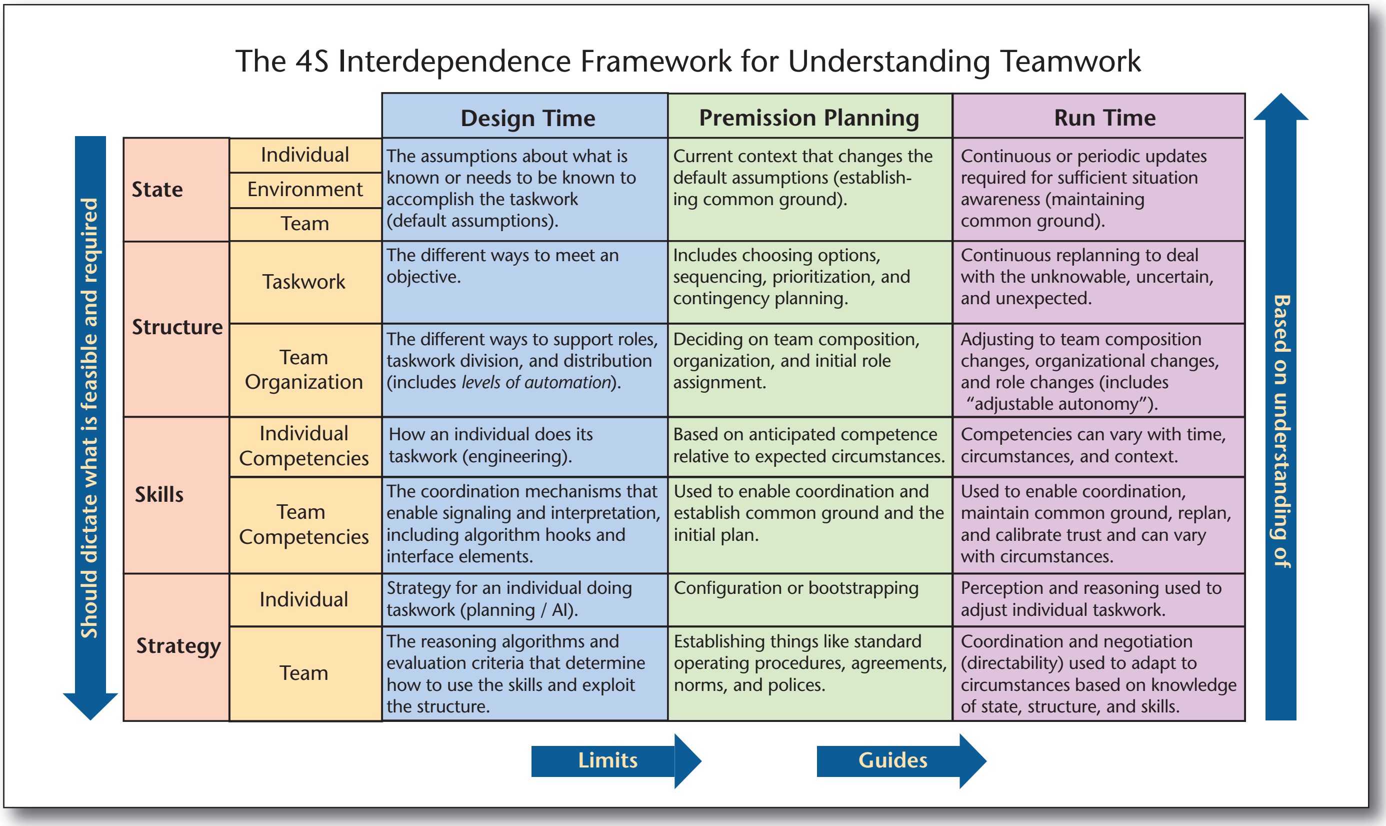
One of the main drivers promoting AI and automation in general is reduction in cost either directly or through improvements in efficiency such as speed, reliability, or economy of scale. Many of the challenges Amazon is looking for AI to solve focus on cost and efficiency. Safety is another driver for AI. Technologies like antilock brakes, collision warnings, and lane-departure alerts help save lives every day. A main argument for the adoption of self-driving vehicles is that, as with the earlier technology successes, AI will save lives. Besides cost and safety, there is another important driver: the potential to exceed human abilities and open new capabilities. Today's computers have access to an enormous amount of data and they have the speed to pour over that data far faster than any human could. This is just one example of how AI can open, and is opening, doors that people will never be able to open themselves.

At the heart of all of these arguments is the concept that humans are the limiting component of many systems. Humans can be a big cost and limit efficiency, they can negatively impact safety, and their natural capabilities will limit what they can achieve in some areas. It is natural and appropriate to consider AI or technology in general as a means to compensate for human limitations, and indeed AI can play a crucial role. However, a key misconception is in how AI should be compensating for human limitations. Very often, AI is seen as replacing the human, the argument being that if people are the source of problems, eliminating them is the solution. However, this is a very narrow perspective that can cause designers to miss or ignore the potential benefits of teaming.

One problem with the replacement perspective is that replacement frames the problem as AI versus human, the goal being to determine which is better. This perspective is fraught with pitfalls. Often human failure is compared against imagined AI perfection, for example, comparing the 35,000 driving deaths annually against the promise of flawless autonomous cars. Nothing ever measures up in comparison to perfection. Another common issue is extrapolation of performance to competence. Brooks warns against the difference, stating: "People hear that some robot or some AI system has performed some task. They then generalize from that performance to a competence that a person performing the same task could be expected to have" (Brooks 2017, under "Performance versus competence"). An example is a statement like "[Google's] cars have driven in autonomous mode for more than one million miles since 2009. In all that time, they've been involved in 16 accidents through August — none of which were caused by the self-driving car."[1] This is truly amazing performance, but care must be taken not to project unwarranted competence. In this case, some of the important points that have been left out are the type of driving conditions in which the system was tested, that the self-driving cars always had a human safety driver ready to take control, and that those safety drivers actually took control relatively frequently. Partially in response to some of the earlier ambitious claims of perfection, autonomous car companies are now required to track disengagements, in other words, the number of times a human needed to take over. A recent report[2] shows an average of 695 miles per disengagement, with the best company posting 5,128 miles per disengagement. The vast majority of the time, there would not have been an accident when the safety driver took over — drivers are directed to be risk averse. However, companies are also asked by the state of California to report "critical disengagements," those cases where there would have been an accident if the safety driver had not taken control. As of 2017, the best performing company has about one critical disengagement every 50,000 miles, a remarkable accomplishment for these new robotic technologies.

What is the rate of serious accidents for human drivers, one might ask? The accident statistics can be a little hard to parse (for example, the number of people injured or killed in an accident does not reflect the number of drivers or vehicles involved), but the number is roughly one injury accident every 1.5 million miles driven.[3] This means self-driving cars are still performing one to two orders of magnitude worse than humans on this metric. More troubling is the fact that the rate of disengagements per miles driven seems to have plateaued for self-driving vehicles since 2015, averaging just under 20 disengagements per 100,000 miles.[4] This data is from the leading company in self-driving cars. This company is greatly outperforming all other companies in this area. They reported more autonomous miles than all other companies combined, seven times more than the next leading company, and they also have the lowest disengagement rate.

These numbers do not tell the whole story, but they certainly temper the predictions about the perfect performance of self-driving cars. Autonomous



The 4S Interdependence Framework for Understanding Teamwork

		Design Time	Premission Planning	Run Time
State	Individual Environment Team	The assumptions about what is known or needs to be known to accomplish the taskwork (default assumptions).	Current context that changes the default assumptions (establishing common ground).	Continuous or periodic updates required for sufficient situation awareness (maintaining common ground).
Structure	Taskwork	The different ways to meet an objective.	Includes choosing options, sequencing, prioritization, and contingency planning.	Continuous replanning to deal with the unknowable, uncertain, and unexpected.
	Team Organization	The different ways to support roles, taskwork division, and distribution (includes *levels of automation*).	Deciding on team composition, organization, and initial role assignment.	Adjusting to team composition changes, organizational changes, and role changes (includes "adjustable autonomy").
Skills	Individual Competencies	How an individual does its taskwork (engineering).	Based on anticipated competence relative to expected circumstances.	Competencies can vary with time, circumstances, and context.
	Team Competencies	The coordination mechanisms that enable signaling and interpretation, including algorithm hooks and interface elements.	Used to enable coordination and establish common ground and the initial plan.	Used to enable coordination, maintain common ground, replan, and calibrate trust and can vary with circumstances.
Strategy	Individual	Strategy for an individual doing taskwork (planning / AI).	Configuration or bootstrapping	Perception and reasoning used to adjust individual taskwork.
	Team	The reasoning algorithms and evaluation criteria that determine how to use the skills and exploit the structure.	Establishing things like standard operating procedures, agreements, norms, and polices.	Coordination and negotiation (directability) used to adapt to circumstances based on knowledge of state, structure, and skills.

Should dictate what is feasible and required
Based on understanding of
Limits
Guides

Figure 1. The 4S Interdependence Framework for Understanding Teamwork.

vehicles undoubtedly have advantages over human drivers in aspects of performance such as reaction time and vigilance; however, they fall short in perception, judgment, and dealing with novelty. Good driving requires the best of both skill sets, and both parties — the human and the autonomous vehicle — fall short in some areas. Each will be a "better driver" in different circumstances.

Another problem with the replacement perspective is that replacement is rarely what is actually happening. When the problem is viewed as replacement, it fosters "the idea that new technology can be introduced as a simple substitution of machines for people — preserving the basic system while improving it on some output measure (lower workload, better economy, fewer errors, higher accuracy)" (Dekker and Woods 2002). This naive viewpoint is one of the myths of autonomous systems (Bradshaw et al. 2013) and it can lead to a host of well-known consequences, including clumsy automation (Wiener 1989) and automation surprises (Sarter, Woods, and Billings 1997).

Additionally, humans are often the enabling components of many technologies. For even the most sophisticated technology, people are usually setting the goals and parameters, monitoring for anomalous situations, and acting as the de facto backup in case of automation failure. Today's commercial aircraft provide an excellent example. These are highly automated systems that can fly the majority of the flight without assistance — some are even capable of landing themselves. However, it is the human pilots who dictate the flight plan, who decide whether automatic landing is appropriate, and who handle the takeoff. Some small drones have demonstrated automatic takeoff, but no aircraft automatically taxi to or from the runway. Not to mention, human pilots are essential for interacting with Air Traffic Control throughout the flight. It is often pointed out that human error is causal in over 80 percent of commercial airline accidents.[5] However, this statistic hides the Bayesian fact that air travel is incredibly safe at this point in history and that accidents often arise from those situations in which the pilots could not resolve a teaming problem with onboard automation. Perhaps most significantly, a 2013 study by the National Transportation Safety Board (Flight Deck Automation Working Group, 2013) found that commercial pilots report addressing and resolving a safety critical situation one in every five flights. If even one in ten thousand of these had the potential of being an actual accident, there would be airplane accidents every day. Human pilots help enable aviation's remarkable safety record.

By focusing on compensating for human limitation by replacement, another important consideration is often neglected — that replacement is not the only, or even the best, way to compensate for human limitations. Instead of replacing a human, AI could enhance the human's ability. AI could support or improve the human's ability like an orthotic or extend it like a prosthetic (Ford et al. 2015). People have an amazing set of abilities that should not be discarded simply because they also have some limitations. In many ways, people and machines can complement one another to potentially yield better performance than either could achieve separately. This sentiment is echoed throughout *Digital Apollo*, David Mindell's historical account of past technological achievements surrounding the Apollo missions. Throughout the recounting, there is friction between the astronauts and engineers on the proper design philosophy with respect to getting to the moon. The astronauts, who were all pilots, took the viewpoint that NASA was designing a human-piloted flying machine. The engineering perspective was that NASA was designing an autonomous rocket that would happen to carry human passengers. The debate was sometimes characterized as "black boxes vs. gray matter." Though the same arguments about whether people or machines were better suited for a given task existed back then, the real story with all of these systems is how humans and automated technology worked together to achieve these great accomplishments — neither was likely to succeed alone.

AI is often regarded as at odds with people, promoting the misconception that the solution is either exclusively AI or exclusively people. People will continue to be a vital part of all but the least complex systems for the foreseeable future, and this reality must influence system design. The success of AI capabilities will depend not only on technical competence, but also on how well an AI system functions with people. Achieving both competencies is a fundamental problem with which designers must wrestle as they improve the capabilities of the technology, and although this has been true for most technological advances throughout human history, the challenge is particularly acute for the sophisticated goals of today's intelligent systems. At the heart of this problem is the issue that even with advances in capability, machines and people remain interdependent. Without adequate support for the interdependencies — which we call *teaming* — problems are inevitable. Most people have probably come across a very capable person who did not work well with others, making it difficult to leverage that individual's talents. Unfortunately, many technologies are built this way. They are designed to perform a function, and it is up to people to work around the technology's lack of social competence. For any intelligent agent, human or machine, to leverage their talents within a larger group outside themselves, having the knowledge, skills, and strategies to effectively team, which we are terming *teaming intelligence*, will be essential.

Interdependence Framework for Understanding Teaming Intelligence

This section is intended to provide a general structure for AI researchers to use in developing intelligent systems that team well. It focuses on interdependence as the central organizing principle for understanding teaming and designing team intelligence. Interdependence is often simply equated to mutual dependence, where entities rely on one another usually because they lack some capacity. However, this definition of the concept is too simplistic to capture the nuances observed in interdependence relationships between humans and machines engaged in joint activity, as the duet example highlights. Interdependence is the set of relationships used to manage dependencies (Johnson et al. 2014). Interdependence relationships must be complementary among the parties involved. Simply stated, one can only give if the other can take and vice versa. These relationships can be required (that is, hard constraints) or opportunistic (that is, soft constraints). Required forms of interdependence include things like use of shared resources and producer or consumer relationships. Examples of opportunistic forms of interdependence include progress appraisals, warnings, helpful adjuncts, and observations about relevant unexpected events.

Understanding the nature of the interdependencies between groups of humans and machines provides insight into the kinds of coordination or teaming that will be required. Indeed, coordination mechanisms in skilled teams arise largely because of such interdependencies (Johnson et al. 2014). Since interdependence is the essence of joint activity, it should not be a surprise that different kinds of joint activity can be distinguished according to the types of interdependencies involved. As such, developing teaming intelligence will require developing support for managing interdependent relationships.

A significant body of work exists on teaming. Some contribute important concepts like joint activity and common ground (Clark 1996; Klein et al. 2004) and situation awareness (Endsley and Kiris 1995). Others have pointed out the requirements and challenges of being an effective team player (Sarter, Woods, and Billings 1997; Klein et al. 2004). Taking a fresh look at this work through the lens of interdependence, figure 1 depicts a framework for organizing many of the important concepts associated with teaming. The focus on interdependence is key to making this framework more actionable than the high-level concepts alone could. The framework is based on four facets: state, structure, skills, and strategy. State and structure are both related to knowledge. Differentiating them provides a clearer understanding of interdependencies. Skills, as the term is being used here, are the building blocks of team behavior. Strategy is the ability to reason over and leverage those building blocks to produce effective team behavior. These facets are explained across design time, pre-mission planning, and run time.

State

The need to understand state leads to informational interdependence, which in turn motivates the need for common ground. Clark and Brennan define common ground as “mutual knowledge, mutual beliefs, and mutual assumptions” (1991, 127). Teamwork is built on common ground. Design-time choices dictate default assumptions about what is known or what needs to be known to accomplish the work. Premission planning can be used to alter the default assumptions based on current context and to establish common ground. Continuous or periodic updates are required to maintain common ground during the course of the activity. Common ground is also related to situation awareness (Endsley and Kiris 1995). It includes aspects of state about the individual, the environment, and the team. Awareness of the team state is essential for team intelligence and often lacking in today’s technology. State captures the past and present situation within the relevant structure.

Structure

For any activity, there is structural interdependence. Structural interdependence can be a result of the taskwork (for example, the need for shared resources) or the team’s organizational structure (for example, role B depends on the output of role A). Whether the structure is inherent in the taskwork or designed in through organizational choices, it can create different types of interdependencies. Teaming intelligence needs to understand where interdependencies exist and with whom. It also must understand the nature of those interdependencies. In addition to understanding structure, agents might need to be able to create or manipulate structure for different situations.

Taskwork can create interdependencies that determine the different ways the individual contributions of members of a group can be combined. Design choices determine the options, pre-mission planning can sequence and prioritize the taskwork, and run time requires continuous replanning to deal with uncertainty and unexpected events, which might require changes to the structure. Steiner (1972) identified five task types that dictate different interdependencies: additive, compensatory, disjunctive, conjunctive, and discretionary. Additive tasks allow members to each contribute individually, and those individual contributions then add together. An example of an additive task is picking up trash. Compensatory tasks allow group members to average their individual contributions. An example would be when a contestant “asks the audience” on the game show *Who Wants to Be a Millionaire,* and they receive a summary of all of the audience members’ opinions. Disjunctive tasks require group members’ contributions to be combined into a single solution. An example is when the contestant on *Who Wants to Be a Millionaire* must take the audience input and make a single choice. Conjunctive tasks require contributions from all group members. An example would be a platoon needing all members to cross the bridge for the team to cross the bridge. Discretionary tasks allow flexibility on how group members combine contributions. Though there can be flexibility in how contributions between people and machines could be combined, this flexibility is often inhibited by design choices.

In addition to the taskwork structure, design choices about team structure can equally create interdependencies. The importance of this point is manifest by much of the work associated with organization theory. Thompson’s work (1967) in particular identified three types of interdependencies: pooled, sequential, and reciprocal. Organizational structures that permit pooled interdependence allow all units to contribute independently, similar to Steiner’s additive task type. Sequential structures arise when the output from one organization unit is needed as the input to another unit, for example, as in an assembly line. Reciprocal structures are similar to sequential, except that they are cyclical. Team structure is not only about the human organizational choices, but the engineering design choices as well. It includes what are commonly referred to as *levels of automation* (Sheridan and Verplank 1978; Parasuraman, Sheridan, and Wickens 2000; Kaber and Endsley 2004), which really describe team structure options. Team structure applies to design-time choices as well as to runtime operational decisions, which are sometimes referred to as *adjustable autonomy*.

These examples of taskwork structure and team organizational structure are only the beginning of understanding how structure creates interdependencies. In general, identifying teaming structure involves understanding the work, how it can be distributed and, most importantly, the interdependencies created by such distribution. It also involves understanding how changes to the team composition, organization, roles, and environment impact the potential interdependencies.

Though people usually develop some basic intuition about structural interdependence, this intuition becomes insufficient as complexity increases, hence the need for fields of study such as organizational theory. Unfortunately, intelligent systems in general do not currently reason over structural interdependence. The “intelligence” about such things is hardcoded into behaviors and algorithms as design-time choices, making it rigid and brittle. Without an understanding of structural interdependence, intelligent systems will remain poor team members because they will not be able to adjust their behavior to comply with and exploit the structural interdependence, especially in dynamic domains with high uncertainty.

Skills

While knowing what structural interdependencies exist is important, teaming intelligence also requires having appropriate coordination mechanisms to deal with those interdependencies. As noted by Thompson: "If there are different types of interdependence, we would expect to find different devices for achieving coordination" ([1967] 2017, 55–56). People naturally develop capabilities to coordinate in different ways. In fact, many of our technologies — such as the telephone, email, texting, and Slack messaging — are based on the need to coordinate. While people can easily learn and adapt to new coordination mechanisms, machines do not come equipped with this capability. They require the development of interfaces and supporting algorithms instrumented with appropriate hooks to enable interaction. Design choices can actually inhibit or prevent coordination, as demonstrated by the need for programs like the Defense Advanced Research Projects Agency (DARPA) Explainable Artificial Intelligence (XAI) program, which notes that "the effectiveness of these systems is limited by the machine's current inability to explain their decisions and actions to human users."[6] The solution DARPA envisions is not just an interface, but new machine learning techniques that produce more explainable models. For intelligent systems to be successful in joint activity, they will need teaming skills. An accumulating body of theory and research points to the key elements that comprise an effective teaming skill set (Christoffersen and Woods 2002; Klein et al. 2004; Johnson et al. 2011). Observability, predictability, and directability are properties commonly emphasized in the literature. We use these terms as a shorthand to refer both to instilling one's own behavior with these properties and to interpreting the behavior of others with respect to these properties.

Observability refers to how clearly pertinent aspects of one's status — as well as one's knowledge of the team, task, and environment — are observable to others. The importance of this capability, often referred to as *transparency*, can be found in many references (for example, Gao and Lee 2006; Klein et al. 2004; Wiener 1989). However, effective teaming often requires complementary relations, so observability also involves intelligent systems that are capable of observing the status of their human counterparts, something they are not particularly good at. Consider how much more capable our phones are with just a little human context, such as GPS position, GPS heading, and IMU. All of these provide context about the person and enable new and more useful AI capabilities. Implied in observability are the communication skills, both verbal and nonverbal, necessary to observe and interpret pertinent signals about status. Observability is about mutually sharing knowledge about past and present state to establish and maintain common ground.

Predictability, in contrast, involves the future. Predictability refers to whether one's actions and intentions are predictable enough that others can reasonably rely on them when considering their own actions, p. 195. Its importance is also made abundantly clear throughout the literature on teamwork (for example, Billings 1997; Klein et al. 2004; Wiener 1989). Mutual predictability involves the capability to receive information about the intentions of others, to be able to predict future states, and to take those future states into account when making decisions. One of the big challenges with technology as it takes on more sophisticated roles is supporting human situation awareness. Situation awareness, as defined by Endsley and Kiris (1995), is about perception (that is, observability) and projection (that is, predictability), in addition to awareness of the literal "situation" (for example, "We are now flying over Wyoming"). Situation awareness is often applied to the human's awareness of the machine, but effective teaming might require the machine to have sufficient awareness of the human as well.

Directability refers to one's ability to direct or influence the behavior of others and complementarily to be directed and influenced by others. While the importance of observability and predictability are well supported throughout literature, directability has had much less attention (for example, Christoffersen and Woods 2002; Klein et al. 2004). Directability includes explicit commands such as task allocation and role assignment, as well as subtler influences. Examples of subtler influences include providing guidance, suggestions, or even salient information that is anticipated to alter behavior, such as a warning.

One of the topics that has gained prominence over the past few years with respect to technology is trust. People tend to live on the extremes of trust: blind trust of automation or open contempt and distrust of automation. This dwelling in the extremes, this lack of gradation, is largely because technology often lacks the mechanisms to help people calibrate their trust. Instead, "a majority of empirical research treats factors of trust as correlates rather than as processes, neglecting interdependent aspects of trust. Trust evolves over time, particularly between interdependent team members" (Chiou and Lee 2015, p. 195). People calibrate their trust through their teaming skills. Observability and predictability provide clues about the competence or lack thereof of teammates.

Unfortunately, technology often has invisible incompetence that prevents such calibration. Examples of this invisible incompetence are numerous, ranging from historic examples of pilots being unaware of the competency envelope of the onboard automation (Wiener 1989; Norman 1990) to more contemporary examples of people being unaware of system brittleness, such as the fact that modern machine learning perceptual models can be fooled with very small data changes (Goodfellow, Shlens, and Szegedy 2014). Directability also plays a role by providing the affordances of control that also contribute to trust. Being able to bound or quickly alter a teammate's actions increases confidence in team behavior, while lack of such controls often leads to mistrust and overly conservative behavior.

The properties of observability, predictability, and directability themselves express interdependence relationships, and the three of them together help define design requirements for both algorithms and interfaces. These relationships are used to navigate structural interdependencies, and as such, they are the mechanisms — skills, in fact — for establishing and maintaining common ground. People naturally acquire these skills throughout life, but such skills are all too often lacking in technology.

Strategy

Teaming strategy is about having the competency to discern how and when to exploit interdependence appropriately. As Peter Drucker noted long ago, "when it comes to the job itself, however, the problem is not to dissect it into parts or motions but *to put together an integrated whole*" ([1954] 2006, 295). This is the challenge for AI — integrating into human workflow. It involves things like understanding informational relevance, discerning appropriate communication frequency and modality, managing attention, and estimating the value of information. These elements change with task context, environment/situation context, and team context. People generally develop some skill at competently sharing information at appropriate moments, though most people also know examples of those who coordinate too often or not often enough. Again, machines lack this social competency. They provide only the information decided on a priori by the engineer. This information is often streamed regardless of relevance, alerts annoyingly interrupt without normal social timing or prioritization, and "communication" regularly occurs without regard to or understanding of the recipient's attention or comprehension. Much of human factors research is about identifying and categorizing the ways machines fail to have adequate teaming strategies and skills.

In summary, strategy is composed of design-time choices and preactivity agreements, as well as realtime negotiation and replanning (directability). It is based on reasoning over knowledge (common ground) that can be known a priori (initial state and structure) and updated in real time via skills (observability, predictability, and directability) to maintain common ground in support of teamwork. As figure 1 indicates, strategy is based on an understanding of skills, structure, and state. It is common today for individual task competencies, that is, the automation, to dictate the interaction, but the 4S interdependence framework suggests that this approach should be inverted and that interdependencies (from state and structure) should dictate the design of the automation (skills), as argued by Johnson (2011). Support for the various forms of interdependence must be built in at design time, because design-time choices will limit runtime options.

How Team Intelligent Are Today's AI Systems?

While today's systems demonstrate strides in many dimensions of intelligence, AI in general has only rudimentary teaming intelligence. Team awareness remains impoverished. Many systems are opaque to their human teammates, and few systems have any knowledge or awareness of the people they work with. Dogs, who lack the natural language skills found in today's technologies like Alexa, are much more capable teammates than the most sophisticated technologies fielded to date. Even a child of two, who has not fully mastered body control, who has only limited knowledge and immature reasoning and decision-making skills, possesses amazing teaming abilities. One could argue that the way children reach mature competency in the other intelligence areas is through the early development of a teaming competency that allows them to ask questions, to take instruction, and to give and receive explanations — skills uncommon in the AI world.

We can further understand the limitations of current teaming intelligence by considering the competencies with respect to the 4S interdependence framework, that is, by considering the teaming state, structure, skills, and strategy. Today's technology collects a massive amount of data about state. This data has improved the self-awareness and environmental awareness of technology, but often teammember awareness is overlooked. Most systems also have limited or no knowledge of structural interdependence, with the exception of rigidly defined task sequencing and authorization requests. Task interdependence is determined at design time, often based solely on technological capability. Team organization is rarely designed, but is instead often the result of requiring people to fill the gaps left by the technology. When the interdependence arises, it is up to the people in the system to recognize the need and to compensate. Systems also have narrow teaming skills, usually confined to diagnostics focused on current state (basic observability). Few provide predictive information and most have extremely limited directability. By and large, there is no team strategy support. Humans again are the default enabler of strategy and must manage both sides of the interaction, using whatever constrained and potentially poorly designed teaming skills are provided by the technology.

Technologies Targeting Full Automation

The goal of many technology development efforts is to eliminate the need for people. Some popular examples are the Roomba vacuum, self-driving cars, and many of the current deep learning efforts.

While teaming is often ignored, especially in the early stages of a particular technology, the need for teaming always arises as the technology matures. This connection between mature technology and the need for teaming can be seen by noting the difference between the original Roomba and the newer 980 model. The original model relied on a simple random bump-and-go exploration strategy. The 980 is indeed more intelligent, using sensors and algorithms to build a map of the room to generate a more effective and shorter cleaning route. This enhanced intelligence also allows the robot to pause whenever it is low on battery, go back to the charging station, and then resume where it left off. Besides these individual competences, the company also added better teaming intelligence to enable the "fully autonomous vacuum" to better work with its owner (the teammate). The owner can receive detailed cleaning maps of their Roomba through an app. The maps show exact areas of clean and dirty spots in the home (observability). The Roomba can also notify the owner when it is finished cleaning (observability).The owner can see how long a task will take (predictability) and can schedule the operational time of the Roomba (directability). This pattern of aiming at full automation and ending up with human collaboration is the true story of AI. Even when the task is so simple that it can be done independently, like vacuuming, there remain interdependencies with people.

Addressing these interdependencies often requires completely redesigning the underlying AI algorithm. In the case of the Roomba, the cleaning algorithm changed from a random bump-and-go approach to a more sophisticated path-planning approach. This more advanced capability made the AI more efficient at its work, but also enabled better human interaction through support for observability, predictability, and directability. Today's machine learning advances are facing a similar circumstance. Machine learning has demonstrated more sophisticated capabilities that have raised interest in the technology, while at the same time the DARPA XAI project is examining how to redesign these types of technologies to be more explainable so they can be more useful to the humans with whom they will need to work.

Another interesting comparison is the Google / Waymo self-driving car and the Tesla S, which advertises an autopilot. Both technologies have demonstrated astounding abilities over the last few years. Early on, Google made the ambitious choice to eliminate the role of the human driver. Thus, the Waymo car has little teaming intelligence since people are seen as inert passengers instead of involved drivers. Tesla began as a normally driven car and has been offering increasingly more advanced driver-assist options. This design stance forced consideration of the driver from the beginning. Despite discussions of its future as a fully autonomous vehicle, as of the version 8 autopilot, the Tesla is not a self-driving car, but a semiautonomous car meant for highway driving.[7] This approach has actually made Tesla more attuned to teaming intelligence and more capable of teaming with a driver. As one reviewer put it, "The Tesla felt more like a giant iPhone than a car,"[8] presumably complementing Tesla's attention to the user. In the Tesla, drivers have profiles, they can adjust how far behind other cars the autopilot will follow, they can influence when lane changes occur, and they get warned if they remove their hands from the steering wheel. The biggest statement about team intelligence learned from the comparison of the Google/Waymo self-driving cars and Tesla's semiautonomous cars is that Tesla's cars have been commercial products for years and, as of the writing of this article, the Google/Waymo cars still are not.

Technologies Targeting Teaming

The commercialization of digital assistants, as they are often referred to, provides another interesting look into the state of teaming intelligence. These systems, like Siri and Alexa, have seemingly broken through the teaming barrier and work with millions of people every day. Yet Ward notes, "Given Siri's broad deployment and popular salience, one might imagine that it solved the problems of interacting in dialogue: we often meet people who are unaware how cleverly Siri and her sisters avoid dialogue. While they do use speech, their preferred interaction style is to map one user input to one system output, avoiding any of that messy interaction stuff" (Ward and DeVault 2016, 8). AI in general has been avoiding the messy stuff involved with teaming — addressing it will play a vital role in AI's usefulness.

There are certainly examples of systems that have focused on partnering people with machines and demonstrated the efficacy of this mindset. One such example comes from Freestyle chess, which allows players to use other resources (for example, books, people, or machines) to help during the game. An approach called *centaur chess* (Cassidy 2014) marries human and machine capabilities to produce a better player than either alone. Centaur teams have been able to occasionally punch above their weight, allowing average players to beat grand masters. The centaur model is also an illustration of how AI will best work in complex domains such as healthcare. This perspective of leveraging the abilities of both AI and people to accomplish something beyond the reach of either is the story of teaming intelligence.

The Risks of Ignoring Teaming

Failure to develop the teaming intelligence necessary to support the human-machine interdependence in the work can result in systems that make the job more difficult, requiring more humans, more training, and more expertise (Blackhurst, Gresham, and Stone 2011). A current example of this principle is the navy's experience with the littoral combat ship (LCS). The LCS was designed to be a smart ship, with a great deal of embedded automation, the idea being that the ships could basically "sail themselves," leaving the small crew to perform mission-related tasks rather than ship-related tasks. At the time of this writing, the ships require about 55 sailors to operated them — close to the original target of 45 and about one-third the crew of a frigate, which is slightly larger. So the program was successful in reducing crew size. However, LCS crews require three times the training time of other crews — 18 months instead of six. The average age of the crew on navy ships is 21, whereas on the LCS it is 30. The seniority of enlisted sailors is designated from E-1, the lowest, to E-9. Typical navy ships have a lot of E-1s and E-2s and only a handful of E-5s and above, while the LCS can be sailed only by E-5s and higher. Most significantly, an LCS can achieve about one-third of operational/mission goals as compared to a frigate, directly proportional to crew size.

The fundamental difficulty with the LCS is not just that the designers of the shipboard intelligent systems failed to understand how crew roles would be changed on the ship, but, additionally, that the nature of today's intelligent technologies often requires human participation for the technologies as a whole to perform safely and effectively. The human crew was the adaptive problem-solving component of the LCS. This is an important lesson for designers of future complex technologies. While not having data on the amount of effort focused on teaming issues, it is safe to say that it was probably relatively small. In contrast, consider the amount of effort Apple has focused on how their iPhone technology works with people. A phone is a much simpler piece of technology that works with a single user. Why would one expect that less effort would be needed to address the teaming requirements of something as complex as the LCS, with a crew of 55 sailors? The challenges with the LCS were not because the automation failed to perform as promised, but rather because the requirements for automated technologies were not based on a system-level view — one that includes people as part of the system. In the end, the littoral combat ship works because the crews have enabled it through great effort and hard-earned human expertise.

From Surface to Deep Teaming

One last example comes from NASA's work on an AI-based activity planner. It demonstrates both getting teaming wrong and eventually getting it right. In 2002, the NASA Ames Research Center fielded an AI-based activity planning and scheduling system for the Mars Exploration Rovers mission. The system added an optimizing scheduling engine to JPL's existing activity planning system. The mission engineer simply needed to input all the planned activities, along with their associated constraints (typically 500 to 700 of each), and then press a "Plan All" button. This process would produce an optimal plan for the day's activities, meaning as much science as possible given the resources available. But the system was almost thrown off the mission before the rovers even landed on Mars. The optimizing engine had basically scrambled everything, from a human perspective, making the plan extremely difficult to modify, manipulate, and validate. More significantly, since it was not possible to represent every constraint in just the way that the science teams envisioned them, the system was often too rigid. The engineering team at JPL called the system *the blender.*

A human-computer interaction (HCI) team was brought in to help. The first request from the mission engineers was that the system allow users to pin activities (that is, directability). The change was made, but the HCI team quickly realized that this was not the right solution — or, more correctly, that they were not solving the right problem. Users were asking for this feature, because they just wanted to stop the system from moving everything around, but, from observation of the mission planning process, it was clear that the better solution would be to have the system flexibly respect users' positioning of activities as much as possible (that is, mutual directability). The HCI team worked with the AI team to see whether an algorithm could be developed that would minimally perturb users' placement of activities while still optimizing the overall plan.

The minimal perturbation algorithm plan was implemented and that change dramatically increased the effectiveness of the rover activity planning tool, allowing the JPL engineering teams to generate plans much faster than anticipated. This planning system was then used for the 2007 Phoenix Mars Lander mission, the 2013 Mars Science Laboratory Mission, and crew activity planning for the International Space Station. In 2017, it was used on a mobile device by an astronaut on the space station to test, for the first time, crew self-scheduling. It will also be used for the upcoming Mars 2020 rover mission. This track record of success has been due to an AI team working closely with an HCI team to shape, not just the interfaces of the tool, but core functionality of the AI capability in order to make it effectively support the mission engineering processes involved. The initial solution was a nonteaming AI system. The first attempt to fix it, by allowing pinning, was a surface, interface-level solution. The eventual system required deeper teaming in both the AI algorithms and the interfaces, yielding a highly functional tool that actually enhanced activity planning productivity.

Conclusion

While AI continues to demonstrate remarkable achievements, the future lies in its ability to work well with people. Highly publicized AI examples such as Watson showcase individual competence, but the future of such systems lies in their teaming competence. Jonas Nwuke, an IBM Watson ecosystem manager, emphasizes, "Our perspective is that you can't take the humanity out of it. There are a ton of opportunities that can be met by [technology]. There are a ton of challenges that can be overcome by it. But, at the end of the day it is that partnership between man and machine that matters most."[9] This sentiment is consistent with our case for teaming intelligence. Our recommended pathway forward is for AI to pursue areas that emphasize teaming competencies. Such an approach would not require a complete redirection of current AI work, but an adjustment and broadening of that work to include how the new AI capability will team with people. If done well, the result should be that the combination of AI and people exceeds the performance of either alone.

Current intelligent systems technologies are fundamentally different from human intelligence, and, more importantly, there is no reason to believe they are on a convergent path with human intelligence in the longer term either. This diversity has the potential to provide strength and resilience — but only if machine and human can work together effectively. Designing for such mutual cooperation suggests a radical shift from the traditional divide-and-conquer approach based on the allocation of function to a more sophisticated strategy based on supplementation or enhancement, instead of replacement — in other words, teaming. Interdependence is the essence of teamwork. AI will only become an effective team player if it has an understanding of interdependence and is designed to support management of interdependencies with people.

One of the key challenges AI faces with respect to teaming is that AI and teaming are often seen as opposites. This is not the case and in fact development of increasingly sophisticated AI capabilities must go hand-in-hand with increasingly sophisticated human-machine interaction. Intelligent systems must be designed from the outset to team with human capabilities, providing assistance where human intelligence has limits and leveraging that intelligence where it is uniquely powerful. Instead of viewing AI and teaming as opposites, we should view them as complements, always remembering that no AI is an island.

Acknowledgments

We thank Pat Hayes, Ken Ford, Peter Pirolli, Daniel Duran, Micael Vignati, and Dorrit Billman for insightful reviews of this work.

Notes

[1] www.theatlantic.com/technology/archive/2015/09/self-driving-cars-could-save-300000-lives-per-decade-in-america/407956.

[2] www.wired.com/2017/02/california-dmv-autonomous-car-disengagement.

[3] crashstats.nhtsa.dot.gov/Api/Public/ViewPublication/812451.

[4] www.dmv.ca.gov/portal/wcm/connect/946b3502-c959-4e3b-b119-91319c27788f/GoogleAutoWaymo_disengage_report_2016.pdf?MOD=AJPERES.

[5] www.boeing.com/commercial/aeromagazine/articles/qtr_2_07/article_03_2.html.

[6] www.darpa.mil/program/explainable-artificial-intelligence.

[7] www.techrepublic.com/article/tesla-autopilot-behind-the-wheel-of-the-worlds-first-ai-powered-driving-experience.

[8] www.techrepublic.com/article/tesla-autopilot-behind-the-wheel-of-the-worlds-first-ai-powered-driving-experience.

[9] www.inverse.com/article/13630-what-has-ibm-watson-been-up-to-since-winning-jeopardy-5-years-ago.

References

Adams, S., Arel, I.; Bach, J.; Coop, R; Furlan, R.; Goertzel, B.; Storrs Hall, J.; Samsonovich, A.; Scheutz, M.; schlesinger, M.; Shapiro, S. C.; Sowa, J. 2012. Mapping the Landscape of Human-Level Artificial General Intelligence. *AI Magazine* 33(1): 25–42. doi.org/10.1609/aimag.v33i1.2322

Baker, D. P.; Day R.; and Salas, E. 2006. Teamwork as an Essential Component of High-Reliability Organizations. *Health Services Research* 41(4 pt 2): 1576–98. London: Blackwell Publishing. doi.org/10.1111/j.1475-6773.2006.00566. x.

Billings, C. E. 1997. *Aviation Automation: The Search for a Human-Centered Approach*. Mahwah, NJ: Lawrence Erlbaum Associates.

Blackhurst, J. L.; Gresham, J. S.; and Stone, M. O. 2011. The Autonomy Paradox. *Armed Forces Journal*, October 1, 2011. www.armedforcesjournal.com/2011/10/.

Bradshaw, J. M.; Hoffman, R. R.; Johnson, M.; and Woods, D. D. 2013. The Seven Deadly Myths of Autonomous Systems. *IEEE Intelligent Systems* 28(3): 54–61. doi.org/10.1109/MIS.2013.70.

Brooks, Rodney. 2017. The Seven Deadly Sins of AI Predictions. *MIT Technology Review*, October 6, 2017. www.technologyreview.com/s/609048/the-seven-deadly-sins-of-ai-predictions/.

Cassidy, Mike. 2014. Centaur Chess Shows Power of Teaming Human and Machine. *Huffington Post*, December 30,

2014. www.huffingtonpost.com/mike-cassidy/centaur-chess-shows-power_b_6383606.html.

Chiou, E. K., and Lee, J. D. 2015. Beyond Reliance and Compliance: Human-Automation Coordination and Cooperation. *Proceedings of the Human Factors and Ergonomics Society*, 2015(January): 195–99. doi.org/10.1177/1541931215591040.

Christoffersen, K., and Woods, D. D. 2002. How to Make Automated Systems Team Players. *Advances in Human Performance and Cognitive Engineering Research* 2: 1–12. doi.org/10.1016/S1479-3601(02)02003-9.

Clark, H. H. 1996. *Using Language*. New York: Cambridge University Press. doi.org/10.1017/CBO9780511620539.

Clark, H. H., and Brennan, S. E. 1991. Grounding in Communication. In *Perspectives on Socially Shared Cognition*, edited by L. B. Resnick, J. M. Levine, and S. D. Teasley, 127–49. Washington DC: American Psychological Association. doi.org/10.1037/10096-006.

Dekker, S. W. A.; and Woods, D. D. 200). MABA-MABA or Abracadabra? Progress on Human-Automation Co-Ordination. *Cognition, Technology, & Work* 4(4): 240–244. doi.org/10.1007/s101110200022

Drucker, P. F. (1954) 2006. *The Practice of Management*. New York: Harper and Row. Reprint, New York: HarperBusiness. Citations refer to the reprint edition.

Endsley, M. R., and Kiris, E. O. 1995. The Out-of-the-Loop Performance Problem and Level of Control in Automation. *Human Factors and Ergonomics Society* 37(2): 381–94. doi.org/10.1518/001872095779064555.

Ford, K. M.; Hayes, P. J.; Glymour, C.; and Allen, J. 2015. Cognitive Orthoses: Toward Human-Centered AI. *AI Magazine* 36(4): 5–8. doi.org/10.1609/aimag.v36i4.2629.

Gao, J. G. J., and Lee, J. D. 2006. Extending the Decision Field Theory to Model Operators' Reliance on Automation in Supervisory Control Situations. *IEEE Transactions on Systems, Man, and Cybernetics – Part A: Systems and Humans* 36(5): 943–59. doi.org/10.1109/TSMCA.2005.855783.

Gardner, Howard. 1998. A Multiplicity of Intelligences. *Scientific American Presents Exploring Intelligence*. 9(4): 19-23.

Goodfellow, I. J.; Shlens, J.; and Szegedy, C. 2014. Explaining and Harnessing Adversarial Examples. In *2015 IEEE Conference on Computer Vision and Pattern Recognition*. Piscataway, NJ: Institute for Electrical and Electronics Engineers. doi.org/10.1109/CVPR.2015.7298594.

Johnson, M.; Bradshaw, J. M.; Feltovich, P. J.; Jonker, C. M.; van Riemsdijk, B.; and Sierhuis, M. 2011. The Fundamental Principle of Coactive Design: Interdependence Must Shape Autonomy. In *Coordination, Organizations, Institutions, and Norms in Agent Systems VI*, edited by M. De Vos, N. Fornara, J. Pitt, and G. Vouros, 172–91. Lecture Notes in Computer Science 6541. Berlin: Springer. doi.org/10.1007/978-3-642-21268-0_10.

Johnson, M.; Bradshaw, J. M.; Feltovich, P. J.; Jonker, C. M.; van Riemsdijk, M. B.; and Sierhuis, M. 2014. Coactive Design: Designing Support for Interdependence in Joint Activity. *Journal of Human-Robot Interaction* 3(1): 43–69. doi.org/10.5898/JHRI.3.1.Johnson.

Kaber, D. B., and Endsley, M. R. 2004. The Effects of Level of Automation and Adaptive Automation on Human Performance, Situation Awareness and Workload in a Dynamic Control Task. *Theoretical Issues in Ergonomics Science* 5(2): 113–53. doi.org/10.1080/1463922021000054335.

Klein, G.; Woods, D. D.; Bradshaw, J. M.; Hoffman, R. R.; and Feltovich, P. J. 2004. Ten Challenges for Making Automation a Team Player in Joint Human-Agent Activity. *IEEE Intelligent Systems* 19(6): 91–95. doi.org/10.1109/MIS.2004.74.

Norman, D. A. 1990. The Problem of Automation: Inappropriate Feedback and Interaction, Not Over-Automation. In *Human Factors in Hazardous Situations*, edited by D. E. Broadbent, A. Baddeley, and J. T. Reason, 585–93. Oxford, UK: Oxford University Press. doi.org/10.1093/acprof:oso/9780198521914.003.0014.

Parasuraman, R.; Sheridan, T. B.; and Wickens, C. D. 2000. A Model for Types and Levels of Human Interaction with Automation. *IEEE Transactions on Systems, Man, and Cybernetics – Part A: Systems and Humans* 30(3): 286–97. doi.org/10.1109/3468.844354.

Sarter, N. B.; Woods, D. D.; Billings, C. E. 1997. Automation Surprises. In *Handbook of Human Factors and Ergonomics*, 2nd ed., edited by G. Salvendy. New York: Wiley.

Sheridan, T. B, and Verplank, W. 1978. Human and Computer Control of Undersea Teleoperators. Technical Report N79-15612. Cambridge, MA: Massachusetts Institute of Technology. doi.org/10.21236/ADA057655.

Steiner, I. D. 1972. *Group Process and Productivity*. New York: Academic Press.

Thompson, J. D. (1967) 2017. *Organizations in Action: Social Science Bases of Administrative Theory*. New York: McGraw-Hill. Reprint, New York: Routledge. Citations refer to the reprint edition.

Ward, N. G., and DeVault, D. 2016. Challenges in Building Highly Interactive Dialogue Systems. *AI Magazine* 37(4): 7–18. doi.org/10.1609/aimag.v37i4.2687.

Wiener, E. L. 1989. Human Factors of Advanced Technology (Glass Cockpit) Transport Aircraft. Technical Report NASA-CR-177528). Coral Gables, FL: Miami University.

Matthew Johnson is a research scientist at the Florida Institute for Human and Machine Cognition. His research focuses on human-machine teaming for technologies such as robotics, software agents, and autonomous vehicles, in a variety of domains including disaster response, space applications, aviation, and military operations. Johnson received a BS from the University of Notre Dame, an MS from Texas A&M – Corpus Christi, and a PhD from Delft University. Johnson is also a retired naval aviator with both fixed-wing and helicopter experience.

Alonso Vera is chief of the Human Systems Integration Division at NASA Ames Research Center. His expertise is in human-computer interaction, information systems, artificial intelligence, and computational human performance modeling. He has led the design, development, and deployment of mission software systems across NASA robotic and human space flight missions, including Mars Exploration Rovers, Phoenix Mars Lander, ISS, Constellation, and Exploration Systems. Vera received a BS with first class honors from McGill University and a PhD from Cornell University. He went on to a post-doctoral fellowship in the School of Computer Science at Carnegie Mellon University.