Editor: **Daniel Zeng**, University of Arizona and Chinese Academy of Sciences, zengdaniel@gmail.com

# Man versus Machine or Man + Machine?

**Mary (Missy) Cummings,** *Duke University and MIT*

In developing any complex system that involves the integration of human decision making and an automated system, the question often arises as to where, when, and how much humans and automation should be in the decision-making loop. Allocating roles and functions between the human and computer is critical in defining efficient and effective system architectures. However, despite the recognition of this problem more than 60 years ago, in this case by NASA (see Figure 1), little progress has been made in balancing role and function allocation across humans and computers.

The problem of human-automation role allocation isn't an academic exercise or limited to a few highly specialized domains such as NASA. The rise of drones (or unmanned aerial vehicles) and the problems with remote human supervision are an extension of well-documented human-automation interaction problems in fly-by-wire systems in commercial aviation. Mining industries increasingly use automation to augment and in some cases outright replace humans, and robots that require human interaction are on the battlefield and in surgical settings. While these applications might seem far from everyday life, Google's recent announcement to introduce driverless cars to the mass market in 2017 and the race to develop in-home robots will make the human-automation allocation issue and associated computing demands ubiquitous.

The predominant engineering viewpoint across these systems is to automate as much as possible, and minimize the amount of human interaction. Indeed, many controls engineers see the human as a mere disturbance in the system that can and should be designed out. Others may begrudgingly recognize that humans must play a role in such systems, either for regulatory requirements or low probability event intervention (such as problems in nuclear reactors).

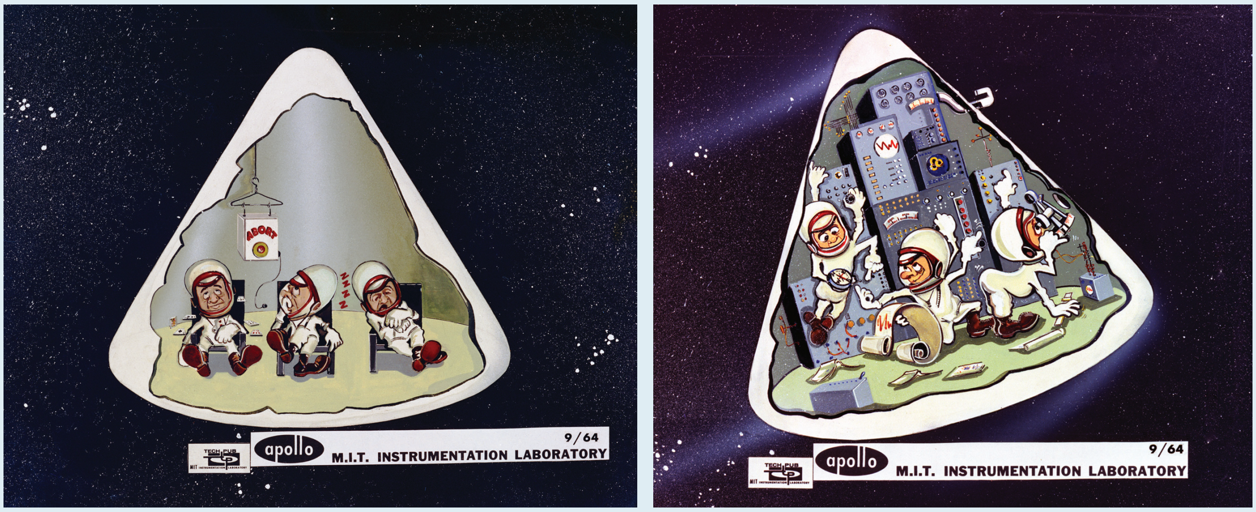
But how do we know what's the right balance between humans and computers in these complex systems? Engineers and computer scientists often seek clear design criteria, preferably quantitative and directive. Most engineers and computer scientists have little to no training in human interaction with complex systems and don't know how to address the inherent variability that accompanies all human performance. Thus, they desire a set of rules and criteria that reduce the ambiguity in the design space, which for them typically means reducing the role of humans or at least constraining human behavior.

## A Brief Historical Perspective

In 1951, a National Research Council committee attempted to characterize human-computer interaction (then called human-machine interaction) prior to developing a national air traffic control system.[1] The result was a set of heuristics about the relative strengths and limitations of humans and computers (see Table 1), sometimes referred to as "men are better at" and what "machines are better at" (MABA-MABA).

The heuristic role allocation approach, exemplified in Table 1, has been criticized as attempting to determine points of substitution—because, for example, such approaches provide engineers with justification (possibly erroneously) for how to replace the human with automation.[2] For traditional engineers with no training in human-automation interaction, this is exactly what they're trained to do—reduce disturbances and variability in a system and make it more predictable. Indeed, they're trying to "capitalize on the strengths [of automation] while eliminating or compensating for the weaknesses,"[2] and this is an important piece of ethnographic information critical for understanding why traditional engineers and computer scientists are so attracted by such representations.


1541-1672/14/$31.00 © 2014 IEEE
*Published by the IEEE Computer Society*




(a) (b)

**Figure 1. The role allocation conundrum for the Apollo missions. (Photos provided courtesy of The Charles Stark Draper Laboratory, Inc.)**

**Table 1. Fitts' list,[1] which characterizes human-computer interaction.**

| Attribute | Machine | Human |
|---|---|---|
| Speed | Superior | Comparatively slow |
| Power Output | Superior in level in consistency | Comparatively weak |
| Consistency | Ideal for consistent, repetitive action | Unreliable learning and fatigue are factors |
| Information capacity | Multichannel | Primarily single channel |
| Memory | Ideal for literal reproduction, access restricted, and formal | Better for principles and strategies, access is versatile and innovative |
| Reasoning computation | Deductive, tedious to program, fast and accurate, poor error correction | Inductive, easier to program, slow, accurate, and good error correction |
| Sensing | Good at quantitative assessment, poor at pattern recognition | Wide ranges, multifunction, judgment |
| Perceiving | Copes with variation poorly, susceptible to noise | Copes with variation better, susceptible to noise |

In part to help traditional engineers and computer scientists understand the nuances of how humans could interact with a complex system in a decision-making capacity, Levels of Automation (LOAs) were proposed. LOAs generally refer to the role allocation between automation and the human, particularly in the analysis and decision phases of a simplified information processing model of acquisition, analysis, decision, and action phases.[3,4] Such LOAs can range from a fully manual system with no computer intervention to a fully automated system where the human is kept completely out of the loop, and this framework was later expanded to include 10 LOAs (see Table 2).

For LOA scales like that exemplified in Table 2, at the lower levels the human is typically actively involved in the decision-making process. As the levels increase, the automation plays a more active role in decisions, increasingly removing the human from the decision-making loop. This scale addresses authority allocation—for example, who has the authority to make the final decision, and to a much smaller degree, it addresses types of collaborative interaction between the human and computer. Raja Parasuraman and his colleagues later clarified that the LOAs could be applied across the primary information processing functions perception, cognition, and action, and not strictly to the act of deciding but again using the same 10 levels.[4]

Other taxonomies have proposed alternate heuristic-based LOAs, attempting to highlight less rigid and more dynamic allocation structures,[5] as well as address the ability to humans and computers to coach and

Table 2. Levels of automation.[4]

| Automation level | Automation description |
|---|---|
| 1 | The computer offers no assistance: human must take all decision and actions. |
| 2 | The computer offers a complete set of decision/action alternatives, or |
| 3 | Narrows the selection down to a few, or |
| 4 | Suggests one alternative, and |
| 5 | Executes that suggestion if the human approves, or |
| 6 | Allows the human a restricted time to veto before automatic execution, or |
| 7 | Executes automatically, then necessarily informs humans, and |
| 8 | Informs the human only if asked, or |
| 9 | Informs the human only if it, the computer, decides to. |
| 10 | The computer decides everything and acts autonomously, ignoring the human. |

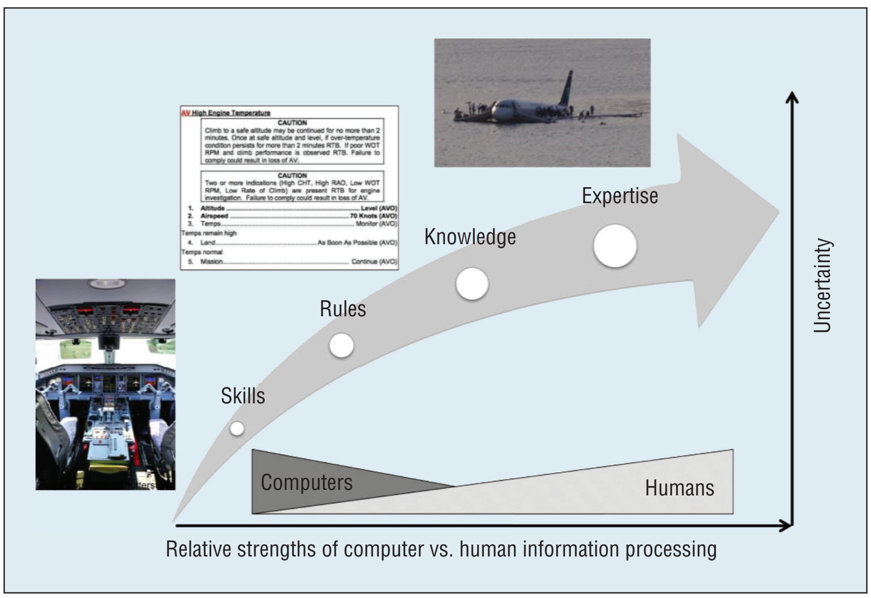



Figure 2. Role allocation for information processing behaviors (skill, rule, knowledge, and expertise) and the relationship to uncertainty.

guide one another. For example, Mica Endsley[6] incorporated artificial intelligence into a five-point LOA scale.

Such LOA scales have been criticized for their primary focus on an exclusive role and function allocation between humans and computers, and less on the collaborative possibilities between the two.[7] However, as noted previously, engineers and designers of such systems desire some way to determine just when and how to design either exclusive or shared functions between humans and computers, and while imperfect and never intended to be rigid design criteria,[8] the notion of LOAs helped such professionals conceptualize a design space, as well as give them a language to discuss competing design philosophies.

## A New Look at an Old Problem

After more than a decade of attempting to train traditional engineers and computer scientists to consider the human early in the design process, in addition to exposing the students to the previously discussed lists and debates surrounding them, the most useful representation I've found that elicits the "aha" moment most educators are looking for is depicted in Figure 2.

First, a map is needed that links information processing behaviors and cognition to increasingly complex tasks, which is best exemplified through Jens Rasmussen's taxonomy of skills, rules, and knowledge-based behaviors[9] (SRK; see Figure 2). One addition to the SRK taxonomy is my representation of uncertainty via the *y* axis. Uncertainty occurs when a situation can't precisely be determined, often due to a lack of or degraded information with potentially many unknown variables. Both external (environmental) and internal (operator performance variability and the use of stochastic algorithms) sources of uncertainty can drive system uncertainty higher.

For Rasumussen,[9] skill-based behaviors are sensory-motor actions that are highly automatic, typically acquired after some period of training. Indeed, he says, "motor output is a response to the observation of an error signal representing the difference between the actual state and the intended state in a time-space environment" (p. 259). This is exactly what controls engineers are taught in basic control theory.

In Figure 2, an example of skill-based control for humans is the act of flying an aircraft. Student pilots spend the bulk of training learning to scan instruments so they can instantly recognize the state of an aircraft and adjust if the intended state isn't the same as the actual state (which is the error signal controls engineers are attempting to minimize.) Once this set of skills is acquired, pilots can then turn their attention (which is a scarce resource, particularly under high workload), to higher cognitive tasks.

Up the cognitive continuum in Figure 2 are rule-based behaviors, which are effectively those human actions guided by subroutines, stored rules, or procedures. Rasmussen likens rule-based behavior to following a cookbook recipe (p. 261).[9] Difficulties for humans in rule-based environments often come from not recognizing the correct goal in order to select the correct procedure or set of rules.

In Figure 2, in the aviation example, pilots spend significant amounts of time learning to follow procedures. For example, when an engine light illuminates, pilots recognize that they should consult a manual to determine the correct procedure (since there are far too many procedures to be committed to memory), and then follow the steps to completion. Some interpretation is required, particularly for multiple system problems, which is common during a catastrophic failure such as the loss of thrust in one engine. Recognizing which procedure to follow isn't always obvious, particularly in warning systems where one aural alert can indicate different failure modes.

For Rasmussen, the highest level of cognitive control is that of knowledge-based behaviors, where mental models built over time aid in the formulation and selection of plans for an explicit goal.[9] The landing of USAIR 1549 in 2009 in the Hudson River, as Figure 2 shows, is an example of a knowledge-based behavior in that the captain had to decide whether to ditch the aircraft or attempt to land it at a nearby airport. Given his mental model, the environment, and the state of the aircraft, his quick mental simulation made him choose the ditching option.

However, this same accident highlights the importance of the need for a collaborative approach between the human and the machine in that when a complete engine failure occurs in the Airbus 320, the fly-by-wire system automatically trims the plane, computes the ideal glide speed, and readjusts pitch position for landing, which is difficult for pilots to maintain. A single press of the DITCHING button seals the aircraft for water entry. This mutually supportive flight control environment was critical to the successful outcome of this potentially catastrophic event.

I added a fourth behavior to the SRK taxonomy, that of expertise, to demonstrate that knowledge-based behaviors are a prerequisite for gaining expertise in a particular field, and this can't be achieved without significant experience in the presence of uncertainty. So while a person can be knowledgeable about a task through repetition, they become experts when they must exercise their knowledge under vastly different conditions. For example, one pilot who has flown thousands of hours with no system failures isn't as much of an expert as one who has had to respond to many system failures over the same time period. Moreover, judgment and intuition, concepts that often make traditional engineers uncomfortable since they lack a mathematical formal representation, are the key behaviors that allow experts to quickly assess a situation in a fast and frugal method,[10] without necessarily and laboriously comparing all possible plan outcomes.

Figure 2 depicts role and function allocation between computers/automation/machines. Such assignments aren't just a function of the type of behavior, but also the degree of uncertainty in the system. It should be noted that these behaviors don't occur in discrete stages with clear thresholds, but rather are on a continuum.

For complex systems with embedded automation, uncertainty can arise from exogenous sources such as the environment—for example, birds in the general vicinity of an airport that *might*, on rare occasion, be ingested in an engine. However, uncertainty can also be introduced from endogenous sources, either from human behaviors or computer/automation behaviors. As evidenced by the Air France 447 crash in 2009 where the pitot-static system gave erroneous information to the pilots due to icing, sensors can degrade or outright fail, introducing possibly unknown uncertainty into a situation. In this case, where the plane crashed because of pilot error, the pilots couldn't cope with the uncertainty since they hadn't gained the appropriate knowledge or expertise.

## Skill-Based Tasks

When considering role allocation between humans and computers, it's useful to consider who or what can perform the skill, rule, knowledge, and expertise-based behaviors required for a given objective and associated set of tasks. For many skill-based tasks, like flying an aircraft, automation in general outperforms humans easily. By flying, I mean the act of keeping the aircraft on heading, altitude, and airspeed—that is, keeping the plane in balanced flight on a stable trajectory.

Ever since the introduction of autopilots and more recently, digital fly-by-wire control, computers are far more capable of keeping planes in stable flight for much longer periods of times than if flown manually by humans. Vigilance research is quite clear in this regard, in that it's very difficult for humans to sustain focused attention for more than 20–30 minutes, and sustained attention is precisely what's needed for flying, particularly for long-duration flights.

There are other domains where the superiority of automation skill-based control is evident, such as autonomous trucks in mining industries. These trucks are designed to shuttle between pickup and drop off points and can operate 24/7 in all weather conditions, since they aren't hampered by reduced vision at night and in bad weather. These trucks are so predictable in their operations that some uncertainty must be programmed into them, or else they repeatedly drive over the same tracks, creating ruts in the road that make it difficult for manned vehicles to negotiate.

For many domains and tasks, automation is superior in skill-based tasks because, given Rasmussen's earlier definition, such tasks are reduced to motor memory with a clear feedback loop to correct errors between a desired outcome and the observed state of the world. In flying and driving, the bulk of the work is a set of motor responses that become routine and nearly effortless with practice. The automaticity that humans can achieve in such tasks can, and arguably should, be replaced with automation, especially given human limitations such as vigilance, fatigue, and the ~ 0.5 second neuromuscular lag present in every human.

The possibility of automating skill-based behaviors (and as we will later see, all behaviors) depends on the ability of the automation to sense the environment, which for a human happens typically through sight, hearing, and touch. This isn't trivial for computers, but for aircraft, through the use of accelerometers and gyroscopes, inertial and satellite navigation systems, and engine sensors, the computer can use its sensors to determine with far greater precision and reliability whether the plane is in stable flight and how to correct in microseconds if there's an anomaly.

This ability is why military and commercial planes have been landing themselves for years far more precisely and smoothly than humans. The act of landing requires the precise control of many dynamic variables, which the computer can do repeatedly without any influence from a lack of sleep or reduced visibility. The same is true for cars that can parallel park by themselves.

However, as previously mentioned, the ability to automate a skill-based task is highly dependent on the ability of the sensors to sense the environment and make adjustments accordingly, correcting for error as it arises. For many skill-based tasks, like driving, vision (both foveal and peripheral) is critical for correct environment assessment. Unfortunately, computer vision still lags far behind human capabilities in many respects, although there's significant research underway in this area. Ultimately, this means that for a skill-based task to be a good candidate for automation, uncertainty should be low and sensor reliability high, which is difficult for many computer vision applications in dynamic environments.

This is why even the most advanced forms of robotic surgery are still just teleoperation, where the doctor is remotely guiding instruments, but still in direct control. Currently robotic surgical tools don't have mature sensors that allow for the closure of the control feedback loop with a high degree of reliability, like those of autopilots. And while some tasks in the driving domain can be automated because of their skill-based nature (like parallel parking), seemingly simple tasks like following the gestures of a traffic cop for a driverless car are extremely difficult due to immature computer vision systems, which don't cope well with uncertainty.

## Rule-Based Tasks

As depicted in Figure 2, skill-based behaviors and tasks are the easiest to automate, since by definition they're highly rehearsed and automatic behaviors with inherent feedback loops. Rule-based behaviors for humans, however, require higher levels of cognition since interpretation must occur to determine that, given some stimulus, which set of rules or procedures must be applied to attain the desired goal state.

By the very nature of their if-then-else structures, rule-based behaviors are also potentially good candidates for automation—but again, uncertainty management is key. Significant aspects of process control plants, including nuclear reactors, are highly automated because the rules for making changes are well-established and based on first principles, with highly reliable sensors that accurately represent the physical plant's state.

Path planning is also very rule-based in that given rules about traffic flow (either in the air or on the road), the most efficient path can be constructed. However, uncertainty in such domains makes path planning a less ideal candidate for complete automation. When an automated path planner is given a start and end goal, for the most part the route generated is the best path in terms of the least time (if that is the operator's goal). However, many possibilities exist that automation may not have information about that cause such a path to be either suboptimal or even infeasible, such as in the case of accidents or bad weather.

It is at this rule-based level where there's significant opportunity for humans to collaborate with automation to achieve a better solution than either could alone. While fast and able to handle complex computation far better than humans, computer optimization algorithms, which work primarily at the rule-based level, are notoriously brittle in that they can only take into account those quantifiable variables identified in the design stages that were deemed to be critical. In complex systems with inherent uncertainties (such as weather impacts or enemy movement), it isn't possible to include a priori every single variable that could impact the final solution.

Moreover, it's not clear exactly what characterizes an optimal solution in such uncertain scenarios. Often, in these domains, the need to generate an optimal solution should be weighed against a satisficing[11] solution. Because constraints and variables are often dynamic in complex environments, the definition of optimal is also a constantly changing concept. In those cases of time pressure, having a solution that's good enough, robust, and quickly reached is often preferable to one that requires complex computation and extended periods of times, which might not be accurate due to incorrect assumptions.

Another problem for automation of rule-based behaviors is similar to one for human selection of the right rule or procedure for a given set of stimuli. Automation will reliably execute a procedure more consistently than any human, but the assumption is that the computer selects the correct procedure, which is highly dependent on the sensing aspect. This is where obstacle detection and avoidance, particularly for driverless cars, is critical. If the automated sensors detect an obstacle, then procedures will be executed for avoidance or braking or both. Indeed, it has been shown that cars equipped with radar can automatically brake much more effectively than a human can.[12] However, the sensing aspect is a significant problem for this futuristic technology, which isn't as reliable in bad weather with precipitation and standing water on roadways.

## Knowledge-Based Tasks and Expertise

The most advanced form of cognitive reasoning occurs in domains where knowledge-based behaviors and expertise are required. Coincidentally, these settings are also typically where uncertainty is highest, as Figure 2 shows. While rules may assist decision makers (whether human or computer) in aspects of knowledge-based decisions, such situations are by definition vague and ambiguous and mathematically optimal solutions are unavailable.

It's precisely in these situations where the human power of induction is critical. Judgment and intuition are critical in these situations, as these are the weapons needed to combat uncertainty. Because of the aforementioned brittleness problems in the programming of computer algorithms and the inability to replicate the intangible concept of intuition, knowledge-based reasoning, and especially true expertise, for now, are outside the realm of computers. However, there's currently significant research underway to change this, particularly in the machine learning community—but progress is slow.

IBM's Watson, 90 servers each with a 3.5-gigahertz core processor, is often touted as a computer with knowledge-based reasoning, but people confuse the ability of a computer to search vast databases to generate formulaic responses with knowledge. For Watson, which leverages natural language processing and pattern matching through machine learning, uncertainty is low. Indeed, because Watson leverages statistical reasoning, it can bound answers with confidence intervals.

A more near-term example of human-computer collaboration for knowledge-based medical decision making is the Athena Decision Support System that implements guidelines for hypertension and opioid therapies.[13] This system harnesses the power of computer search and filtering but also allows doctors the ability to guide the computer based on their own experiences.

A limitation of pattern-matching approaches is the overreliance on supervised learning, in that labels must be assigned (typically by humans) for a computer to recognize a pattern. Not only is it possible for humans to introduce error in this process, it raises the question of whether a computer can detect a pattern or event it has never seen before, or that's slightly different than a pattern it has seen before.

There has been increasing interest in using semisupervised and unsupervised machine learning algorithms that don't use labels, and thus generate groups of patterns in absence of such bias. However, with regard to replicating human learning in terms of object recognition, unsupervised machine learning for computers is still quite immature. In a recent major "breakthrough," an unsupervised algorithm was able to cluster and successfully recognize cats in unlabeled images with only 15.8 percent accuracy, which was reported to be an improvement of 70 percent over the current state of the art.[14] For computer vision applications, robust, fast, and efficient perception will be needed before computers can reliably be trusted in perception-based tasks.

With such brittleness, it will be some time before computers can truly begin to approach the expertise of humans, especially in situations

**Table 3. Degree of automation as a function of a desired behavior.**

| Cognitive behavior/task | Degree of automation |
|---|---|
| Skill-based | Best candidate for automation, assuming reliable sensors for state and error feedback |
| Rule-based | Possible candidate for automation, if rule set is well-established and tested |
| Knowledge-based | Some automation can be used to help organize, filter, and synthesize data |
| Expertise | Human reasoning is superior, but can be aided by automation as a teammate |

of high uncertainty. But this isn't to say there's no role for computers in knowledge-based reasoning. Again, this area is ripe for more development in human-computer collaboration. IBM's first commercial application of Watson will be *aiding* nurses and doctors in diagnoses, which falls squarely in the domain of expert decision makers.

While Paul Fitts and his colleagues were perhaps overly focused on mutually exclusive assignment of human and machine roles, their basic premise more than 60 years ago should be interpreted through the lens of collaborative systems and the behaviors that need to be supported. The modified SRK taxonomy presented here isn't meant to be a replacement for earlier role and function efforts, but rather a different lens through which to think about system design. The intent is to provide engineers and computer scientists with a principled framework by which to formulate critical questions, such as the following:

- Can my sensors provide all the data I need at a high enough reliability to approximate trained human skill sets?
- Is there a high degree of uncertainty in either my environment or my sensors, which would necessitate human supervision?
- Can humans augment and improve either sensor or reasoning deficiencies, and how would this occur without overloading the human?
- Can automation reasoning be improved through human guidance and coaching?
- Can automation be leveraged to help the human reduce uncertainty, particularly when knowledge and expertise is needed? The reverse should also be explored in that the human may be able to reduce uncertainty for the automation.

As Table 3 shows, skill-based behaviors are the best candidates for automation, assuming significant sensor performance assumptions can be met, but rule- and knowledge-based reasoning are better suited for human-computer collaboration. Systems should be designed so that humans harness the raw computational and search power of computers for state-space reduction, but also allow them the latitude to apply inductive reasoning for potentially creative, out-of-the-box thinking. As a team, the human and computer are far more powerful than either alone, especially under uncertainty.

In a 2005 competition against the Hydra chess computer, two novices with three computers beat the computer and other grandmasters aided by single computers. Arguably chess is an environment of low uncertainty (particularly for computers that can search a large but finite set of possible outcomes.) However, in a real-world and highly uncertain command-and-control environment of one operator controlling multiple robots in a search-and-find task, it has been shown that allowing the human to coach a highly automated system produces results up to 50 percent better than if the automation were left to its own devices.[15] Collaboration between humans and computers, particularly in knowledge-based domains where complementary strengths can be leveraged, hold much future potential.

Last, role and function allocation is as much art as science. The complexity of systems with embedded autonomy supporting dynamic human goals suffers from the "curse of dimensionality."[8] As a result, these systems will never have closed-form solutions and will be intractable from a mathematical perspective. But because of the necessary mix of art and science in designing such systems, both industry and academia should recognize the need for a new breed of engineer/computer scientist. Such a person should have an appreciation for human psychology and performance characteristics, but at the same time understand control theory, Bayesian reasoning, and stochastic processes.

## References


1. P.M. Fitts, "Human Engineering for an Effective Air Navigation and Traffic Control System," tech. report, *Nat'l Research Council*, 1951.
2. S.W.A. Dekker and D.D. Woods, "MABA-MABA or Abracadabra? Progress on Human-Automation Coordination," *J. Cognition, Technology and Work*, vol. 4, 2002, pp. 240–244.
3. R. Parasuraman, "Designing Automation for Human Use: Empirical Studies and Quantitative Models," *Ergonomics*, vol. 43, 2000, pp. 931–951.
4. R. Parasuraman, T.B. Sheridan, and C.D. Wickens, "A Model for Types and Levels of Human Interaction with Automation," *IEEE Trans. Systems, Man, and Cybernetics—Part A: Systems and Humans*, vol. 30, 2000, pp. 286–297.

5. D.B. Kaber et al., "Adaptive Automation of Human-Machine System Information-Processing Functions," *Human Factors*, vol. 47, Winter 2005, pp. 730–741.
6. M. Endsley, "The Application of Human Factors to the Development of Expert Systems for Advanced Cockpits," *Proc. Human Factors and Ergonomic Soc. 31st Ann. Meeting*, 1987, pp. 1388–1392.
7. Defense Science Board, *The Role of Autonomy in DoD Systems*, US Dept. of Defense, 2012.
8. T.B. Sheridan, "Function Allocation: Algorithm, Alchemy, or Apostasy?" *Intl J. Human-Computer Studies*, vol. 52, 2000, pp. 203–216.
9. J. Rasmussen, "Skills, Rules, and Knowledge: Signals, Signs, and Symbols, and Other Distinctions in Human Performance Models," *IEEE Trans. Systems, Man, and Cybernetics*, vol. 13, no. 3, 1983, pp. 257–266.
10. G. Gigerenzer, P.M. Todd, and T.A.R. Group, *Simple Heuristics That Make Us Smart*, Oxford Univ. Press, 1999.
11. H.A. Simon et al., *Decision Making and Problem Solving*, Nat'l Academy Press, 1986.
12. Insurance Institute for Highway Safety, "More Good News about Crash Avoidance: Volvo City Safety Reduces Crashes," *Status Report*, vol. 48, no. 3, 2013; www.iihs.org/iihs/news/desktopnews/more-good-news-about-crash-avoidance-volvo-city-safety-reduces-crashes.
13. A. Advani et al., "Developing Quality Indicators and Auditing Protocols from Formal Guideline Models: Knowledge Representation and Transformations," *Proc. AMIA Ann. Symp.*, 2003, pp. 11–15.
14. Q.V. Le et al., "Building High-level Features Using Large Scale Unsupervised Learning," *Proc. 29th Int'l Conf. Machine Learning*, 2012; http://icml.cc/2012/papers/73.pdf.
15. M.L. Cummings et al., "The Impact of Human-Automation Collaboration in Decentralized Multiple Unmanned Vehicle Control," *Proc. IEEE*, vol. 100, 2012, pp. 660–671.

**Mary (Missy) Cummings** is the director of the Humans and Autonomy Laboratory at Duke University. She is also a principle investigator in the MIT Computer Science and Artificial Intelligence Laboratory. Her research interests include human-autonomous system collaboration, human-systems engineering, and the ethical and social impact of technology. Cummings has a PhD in systems engineering from the University of Virginia. She's an IEEE Senior Member. Contact her at m.cummings@duke.edu.