Humans in Autonomy: Verifiable, Multi-Scale, Co-Design

Ella Atkins, Randal Burns, and Mark Campbell

March 18, 2014

Motivation and Key Research Questions

The evolution of autonomy has revealed that the humans need to be inextricably involved in all dimensions of autonomous systems, not just managing those systems, but seamlessly integrating human capabilities, human information gathering, and human decision making.

> *“. . . the true value of unmanned systems is not to provide a direct human replacement, but rather to extend and complement human capability.”*
>
> DSB Task Force Report, The Role of Autonomy in DoD Systems, 2012.

The DSB asserts that the benefits of Autonomy lie in unlimited persistence, reducing risk to humans, and reducing cognitive load. Autonomy should in no way eliminate or reduce human capabilities. This shifts thinking so that autonomy becomes a capability to be integrated into a complex human-machine system, rather than a functional replacement for a human.

With this conceptual shift, how can autonomy integrate into human-machine systems so that they maximize the performance of the system as a whole? Insight into system performance requires measures and models of human performance both standalone and in the presence of autonomy. Success will provide the capability to extend performance to systems that are multi-scale networks of humans and autonomy.

Beyond performance, it is essential to have guarantees that autonomous systems operate correctly within bounds set by human oversight, not specifications or requirements.

> *“Autonomous and semi-autonomous weapon systems shall be designed to allow commanders and operators to exercise appropriate levels of human judgment over the use of force.”* A. B. Carter, DoD Directive 3000.09, November 21, 2012

This becomes increasingly difficult as autonomy grows in complexity, scope, and scale. Advances are needed in the design of human-autonomy interaction, to ensure that autonomy reflects operator and command intent. Advances are also needed in proving the correctness of autonomous systems. This must include the correctness of humans, as their capabilities and tasks are networked into autonomy and will often serve as inputs to autonomous capabilities.

Because autonomy is evolving toward human collaboration, capability enhancement, and complex systems, new definitions of autonomy are needed to guide innovation and set goals. The DoD Priority Steering council recently defined autonomy as the “capability and freedom to self-direct to achieve mission objectives”. They also put forth that:

> *“Military Power in the 21st Century will be defined by our ability to adapt—this is THE hallmark of autonomy”*
>
> DoD Priority Steering Council, Nov. 2012

The role of science and technology development in autonomy is to meet this definition of autonomy while integrating humans, resolving conflicts between self-direction and correctness and ensuring that adaptation reflects human decisions.

The authors note that this thinkpiece focuses on how science and technology can contribute to achieving autonomy goals related to human collaboration in autonomous systems. As such, it does not focus on acquisition, security, policy, or law.

Science and Technology Research Questions:

An analysis of the requirements and risks reveals technical challenges to overcome and capability gaps to fill in order to fully integrate humans into collaborative autonomy.

- *How do we model, verify, and utilize humans as collaborative decision makers and actors in correct-by-construct autonomous systems?*

Many automated systems today are designed primarily with automation in mind, not necessarily human interaction, leading to sub-optimal or incorrect integrated performance. While the typical stated goal of autonomy is to remove the human element from particular tasks, it is critical to realize that the addition of autonomy simply moves humans to a different point/level of interaction with the autonomy; the human is typically not fully removed. As such, one must consider the autonomy+human as an integrated âĂŸsystemâĂŹ when evaluating performance, robustness, and effectiveness. The ultimate success of many autonomous systems is intimately tied to interactions with a human.

Modeling even a portion of human capabilities will enable the ability to plan, optimize, and analyze an integrated human-autonomy system. Probabilistic modeling of human capabitilies is reaching a maturity level that could enable human-machine interactions to be more prospective, rather than reactive, which has typically been the characteristic of previous research on humans and automation. Importantly, the ability to probabilistically model human capabilities would enable the concept of 'correctness' to be considered more deeply, such as with formal Verification and Validation methods.

- *How do we create multi-scale, model-based autonomy (controllers, communication, and computation) that scale to the battlespace?*

Emergent autonomous systems cannot be thought of simply as controllers that automate functions. Rather, they are networks of humans and machines that perform heterogeneous complex tasks, such as data integration, navigation, target detection, and distributed sensing. The "controllers" for autonomous capabilities must include computation and communication, making the human-machine network a distributed computing system with a network protocol. This must include human computation for tasks in which humans exceed machines, such as language translation and object classification in images. It must also include protocols for humans to communicate with autonomy.

For complex, human-machine systems, model-based autonomy must be multi-scale, i.e. define properties and invariants at multiple spatial and temporal scale and integrate or link the scales into the battlespace view. Autonomy must be decomposed into multiple models. First, the state space of a human-machine network is so large as to be unsolvable in practice. Also, models must be concise and represent related concepts to produce meaningful correctness guarantees and result in controllers. Autonomy will consist of a hierarchical network of models that are solved in detail individually and summarized to be integrated into higher-level models.

- *How do we co-design complex mission systems to optimally and acceptably incorporate human and autonomy decision-makers and actors?*

As systems more tightly integrate human and machine elements, the systems engineering process must evolve to provide solutions that optimally or acceptably *co-design* the system over both humans and autonomy. The traditional systems engineering process translates requirements to designs to implementations. A fundamental assumption of this process is that we can create and alter product design and implementation as needed to meet the requirements. This assumption must be relaxed to account properly for humans whose capabilities are extremely versatile and adaptive but who cannot be "redesigned" to meet requirements except through careful training. The diversity inherently present in co-designed human-autonomy systems will requires new models to better accommodate human system elements and a more model-based approach to systems engineering than is used today.

The co-designed human-autonomy system will be comprised of autonomy elements with hardware and software customized to perform particular tasks, e.g., persistent surveillance, and humans expected to have received customized training to perform particular tasks, e.g., adapt mission goals based on incoming information. Mission success is contingent on acceptable performance by both. This description is sufficiently general to fit into today's paradigms; the difference is that today systems engineering focuses strictly on design and implementation of the machine systems (the autonomy), leaving consideration of how these machines interact with humans (apart from graphical user interfaces) to be determined once the system is implemented.

A co-designed system will infuse models of humans and autonomy throughout all phases of system engineering. The products of the co-design system engineering process will then include both designs and implementations of autonomy elements that have been verified to exactly match their corresponding requirements and translations of human element requirements (roles and responsibilities) to training protocols along with metrics used to verify that human performance has reached level(s) required by the overall system. Validation of the co-designed system will then involve tests with fully-verified autonomy and fully-trained human actors engaged in realistic deployment scenarios.

Background

The implications and conclusions of this think piece are built upon five key technical areas: Verification & Validation, which is common in computer software and autonomy; correct by construction controllers which are generated with guarantees; probabilistic modeling of human decision making; collaborative autonomy, and model based system engineering. Each of these are briefly described here, with a few example references.

Verification & Validation (software v. autonomy)

A recent study by the Office of the US Air Force Chief Scientist [1] cited Verification and Validation (V&V) is a key limitation in the ability to achieve the high impact gains that can be realized from autonomy.

> *"Increased use of autonomy. . . will depend on development of entirely new methods for enabling 'trust in autonomy' through verification and validation (V&V) of the near-infinite state systems that result from high levels of adaptability and autonomy."*
>
> A Vision for Air Force Science and Technology 2010-30, USAF Chief Scientist, 2011.

It is important to first understand the definitions of Verification and Validation:

Verification: Requirements evaluation during *development*

Validation: Requirements evaluation after *integration*

Thus, the process of verification is to incrementally and systematically evaluate whether a system or subsystem is meeting requirements during the design process. Validation, on the other hand, evaluates

System	Lines of Code (loc)
F-4A	1,000
F-15A	50,000
F-16C	300,000
F-22	2,500,000
F-35	18,000,000

Table 1: Critical lines of code for aircraft; $\sim$4-6 errors per 1000 LOC, $\sim$0.1-1.0 errors per 1000 critical LOC

whether the completed system meets the end customer requirements in a practical setting. If done well, Verification can speed up of the Validation process because portions of the underlying system have already been verified.

The process of V&V has been known for a long time, and is a formal part of nearly any systems engineering process [2]. The importance of systematic processes has increased as the complexity/safety/security of the system has increased, such as in cars, airplanes, and spacecraft. V&V for these systems has typically taken the form of empirical testing because it is closer to the end operational state, and people typically accept empirical testing easier than other options such as simulations.

Most of these complex systems have grown in complexity more because of software growth than other components. Autonomous systems are particular challenging, as the physical system has not changed as much as the internal 'intelligence' of the software. With the software growth, however, comes the typical challenges in verification of the system (now, a physical+software system). Dr. Werner Dahm, former Chief Scientist of the Air Force, has given several talks on the Air Force study in Ref. [1], and cited the growth and errors in software. A small summary is given below:

Current V&V methods are not sustainable as systems increase in complexity, given this growth. Dr. Dahm argues that current systems with autonomous elements are tested to exhaustion of the budget, rather than to a formal level of V&V.

V&V in software has also increased in importance over the years with the growth of programs and applications. Risk management for large software systems has lead to the "spiral model" of development [3] that builds out systems based on successive cycles each that incorporate V&V and links V&V to design and the evaluation of risk. The process commonly employs multiple groups for V&V that are independent from developers, including a test team that are part of the software development process and users and customers that participate in user acceptance *beta* testing.

With all applications, the time required for the processes for V&V increases, creating pressures on safety/security/costs, etc. As such, recent research in the software engineering community has focused on developing automated, and formal, verification tools. Formal methods is a well established area of research concerned with formally specifying systems and properties, developing techniques to prove/disprove that a system satisfies a property (verification), and in some cases, developing methods to generate a system from required properties (synthesis). Model checking [4] is a verification technique that exhaustively searches the state space of a system in order to either verify that a property holds over all of the system's executions or to find a counter example, e.g. using temporal logic of actions [5]. Formal V&V tools have matured so that they can provide formal proofs for large concurrent systems; the Alloy Analyzer has been used to correct the Chord peer-to-peer protocol [6] and verify the Internet's Border Gateway Protocol [7]. Other advances adds V&V capabilities to existing languages and development tools, such as the verifiable C compiler [8].

The robotics community has borrowed/expanded these concepts for the verification of autonomy. In particular, research has been conducted in developing approaches for formal verification of software for autonomous systems. Spin [9] and NuSVM [10] are two examples of powerful model checkers that have been used to verify autonomy software generated from higher level specifications. SAT solvers (e.g. [11, 12]) are powerful tools that check whether a propositional logic formula has a satisfying assignment, a technique that exhaustively searches all executions of a system up to length k. All of these tools typically check logical consistency of a specification and reports on deadlocks, race conditions, incompleteness, and assumptions.

Most recently, the community has developed 'probabilistic model checkers' such as PRISM [13, 14] which are designed to verify software to a particular level of probability. These tools typically verify a system through symbolic data structures and algorithms as well as exhaustive search. Importantly, state of the art tools are being used to verify software and autonomy, even probabilistically.

Correct by construction controllers

In the application of robotics, the formal verification methods have been used to generate 'correct by construction' controllers from high-level task specifications in a manner that provides guarantees about the behavior of the autonomy [15–18]. This is the process whereby a system is modeled and a list of specifications in generated. The specifications are typically at the higher level, such as 'visit all rooms until you find my keys' or 'do not go into room X.' In this case, hybrid controllers (discrete and continuous elements) are automatically generated and guaranteed to satisfy the original specifications listed. If the specifications cannot be met, then the controller is not generated. The flow chart of the process is given below.

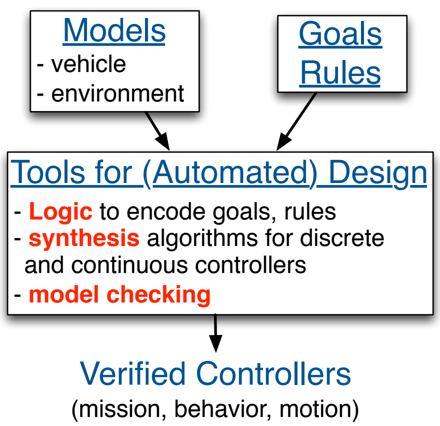
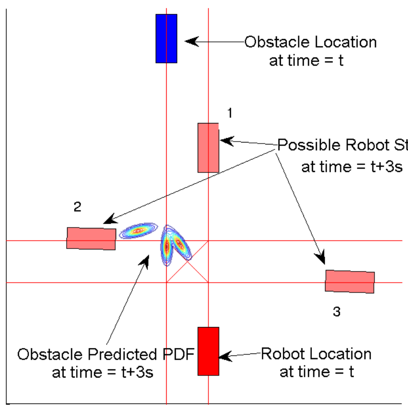


Figure 1: correct by construction example

These concepts have been extended to using probabilistic model checkers, and this work is most applicable to the concepts proposed in this think piece. Typically, an off-the-shelf model checking software such as PRISM [13, 14] is used to find the probabilities of satisfying the desired set of logic based specifications. Given a full characterization of the system (known probabilities for environment and sensor models), model checking techniques can be used to find upper and lower bounds on the probability that the autonomy will satisfy the set of specifications.

Figure 2 shows a recent example of a taxi driver with a specification on collision probability [19]. In this case, the authors have defined collision probability based on the current location density of an object (e.g. another car) in an environment, a map of the road structure, and a temporal/probabilistic prediction of the location density into the future, based on typical driving standards (Figure 2(left)). A collision bound is then calculated and used as a design specification for the controller. A controller can then be generated by selecting a 'desired probability of collision', and generating and checking a controller for the taxi driving in an environment with other cars. Figure 2(right) shows the case when the other cars in the environment are modeled as not obeying any rules of the road.By selecting a 'desired probability of collision' and generating a controller, different autonomous driving behaviors can be realized, such as conservative driving or agressive driving.



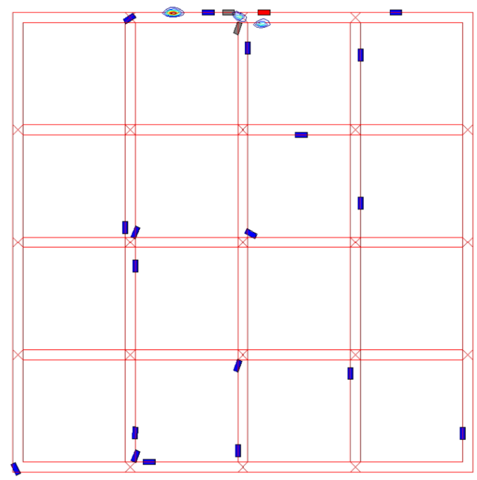


Figure 2: collision probability taxi driver example

Given that there are now approaches to generating software for autonomous systems that are probabilistically correct, can we now begin to include human interaction (and models of humans) such that the controllers are designed with humans in mind?

Probabilistic modeling and humans

Modeling even a portion of human capabilities will enable the ability to plan and optimize an integrated human-autonomy system. Modeling human-machine interactions in large-scale networked systems can be useful in making research more prospective, rather than reactive, which has typically been the characteristic of previous research on humans and automation. Modeling can also inform the design of future interfaces to support operators of multiple UV systems in the presence of concurrent operational and cognitive uncertainties.

Research in modeling of human capabilities has been on-going to many years, from the early work on simple cognitive functions and interaction with autonomy [20], to more recent work attempting to model extensive cognitive capabilities [21] and even the brain [22]. Integrated databases have been developed for modeling/prediction of perception and motor skills [23–26]. Currently, these databases are focused on low level human skills, and do not integrate with the environment (such as the use of autonomy models). ACT-R [21] is a cognitive architecture attempting to model a full range of human cognitive tasks, including the way we perceive, think about, and act on the world. The ACT-R architecture has developed modules representing perceptual attention, motor programming, long-term declarative memory, goal processing, mental imagery and procedural competence. Applications have included air traffic control [27,28] and multiple agents in military environments [29].

The human factors community, on the other hand, has focused on empirically driven tests in order to gain insightful observation of trends, but not formal models; one fruitful area has been in supervisory control of UAVs where several human operators are typically required to control current unmanned aerial vehicle (UAV) platforms [30, 31]. Given the goal of one operator to many UAVs, automation support, even if imperfect, is mandated [32–34]. However, the extra task load generated by handling imperfect automation may interfere with adequately supervising a larger number of UAVs. Recent estimates of an operator's capacity to control multiple UAVs range from 1 to 16 [35], but more precise estimates may be calculated by considering the impact of UAV coordination demands, UAV interaction and neglect times, automation reliability, mission type and operator tasks and the task-to-robot ratio [35–39]. Research on operators in Air Traffic Control [40, 41] has provided valuable insight into how users make decisions as a function of parameters such as stress, interface type, and time. These works typically derive key performance metrics from trends in the data, but do not formally model them.

Many generic 'non-cognitive' probabilistic models have been proposed as alternatives to well-known detailed cognitive computational models for predicting human-in-the-loop performance in networked unmanned vehicle applications. In [42] and [43], for instance, human operators are modeled dynamically via probabilistic Markov models in order to capture random transitions between abstract discrete states that influence decision-making and task performance metrics. In [44], discrete-event task simulations with probability distributions on operator servicing times are used to explicitly model the performance effects of changing workload and vehicle utilization in a multi-UAV supervisory task. Both cognitive and non-cognitive dynamic probabilistic human-operator models can be used to generate sample-based performance prediction statistics via repeated random simulations of closed-loop task execution, and as such can provide useful insight into specific scenarios that lead to good/bad operator performance. However, such dynamic probabilistic models require a high level of detail and much training data to explicitly account for the effects of various task/network-related factors (e.g. number of agents, task load). These models also do not explicitly account for individual factors, e.g. differences in working memory capacity. Furthermore, many simulations must be run with dynamic models in order to make performance predictions for a single set of operating conditions, which can be cumbersome for exploring many different network/task conditions.

A new class of probabilistic models has recently been developed that are potentially useful for prediction and verification of human operator performance in human-autonomysystems, either in the sense of performing detailed analyses related to dynamical process simulations or performing gross 'high-level' analyses of human-machine system performance that abstract away certain dynamical details. Two of these models, *Gaussian Process (GP) regression* and *Bayesian networks (BN)*, can enable direct 'function-like' performance predictions without requiring simulations or an explicit model of the operator's decision-making processes [45]. Any expected variability arising from differences in these and other unmodeled factors related to task dynamics are described by the estimated probabilities associated with each prediction. A third type of model, *probabilistic discriminative classification models*, can be used to capture stochastic *non-Markovian state-dependent switching behaviors* for discrete supervisory decision making by human operators in detailed process models [46], [47], [48], which is currently not realizable with the probabilistic Markov or discrete event models mentioned above.

As an example, consider the case of probabilistically modeling human observations at a macro level (human observations and/or tasks), in an effort to more formally exchange information with an autonomous system. Both discrete [49] and continuous human inputs [50] have been modeled, including a rich set of structured inputs such as 'The target is near the tree and heading quickly toward you' or 'There is nothing behind the wall.' The key challenge is probabilistically modeling terms such as 'near' or ‘behind’.

Leveraging the fact that discrete random variables nicely represent soft (human) categorical information, the human information has been shown to be well-modeled via probabilistic classifiers [51–54]. These classifiers are typically learned from human observation data, such as computer point and click; chat inputs; and natural language processing. Figure 3 shows a simple example of learning soft human categorical information from data.

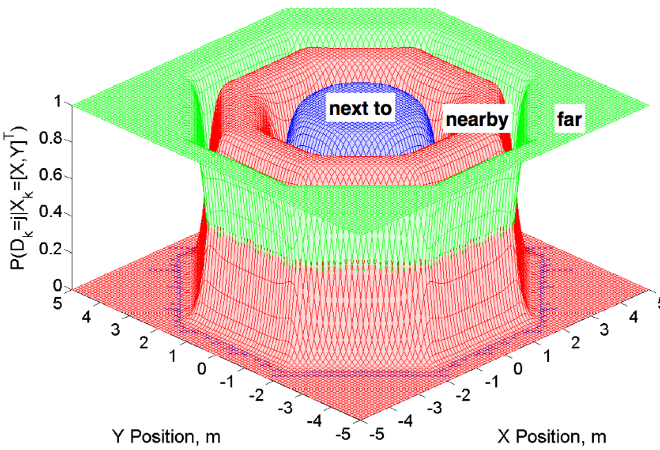


Figure 3: Learned likelihood model for relations ‘next to’, ‘nearby’, and ‘far.’

These likelihood functions then enable a host of subsequent functions, from cooperative planning to information fusion and inference. Ref. [54] empirically investigates how human information can be used by an autonomous robot in a search mission. Even though human subjects could not directly command robots, it was found that fusing human and robot observations greatly improved object search performance (i.e. number of targets found, time to find objects, object localization error) compared with baseline searches using robot observations alone. Human observations were particularly useful for correcting missed object detections and reducing the distances traveled by the robot, whereas the use of *negative information* (e.g., ‘Nothing is near the bridge’) was shown to be particularly important in the integrated human+robot team performance.

The use of Gaussian Process models has been shown to be particularly insightful into human tasking, including variations over users. In a collaboration between researchers in the human factors community and the autonomy community, collected operator data was modeled using different statistical modeling methods to study the ability to predict human operator performance in an air defense simulation scenario [55]; performance metrics were modeled as a function of task load, message quality, and operator working memory capacity. It was found that state-of-the-art Gaussian Process (GP) regression models can make predictions with uncertainty bounds that are more informative than traditional linear regression and discrete Bayesian network (BN) prediction models. More specifically, off-line tests of human operator metrics (such as working memory) are predictive in eventual performance of the operators in the UAV tasking environment (such as red zone performance). The GP models also nicely capture uncertainty in area of the model with little date, and trends/anomalies with user capabilities. Figure 4 shows an example of one such predictive model, where task performance is captured as a function of working memory, which can be evaluated off-line in a priori tests. Several studies have shown that individual differences in working memory capacity play a major role in determining how well a person can focus attention on visual tasks and cope with distractors such as irrelevant messages [56].

More generally, working memory is thought to be a key component of executive control processes that underlie effective multi-tasking and decision-making in time-critical tasks [57, 58]. Therefore, individual differences in working memory capacity could be modeled and used in developing verifiable specifications.

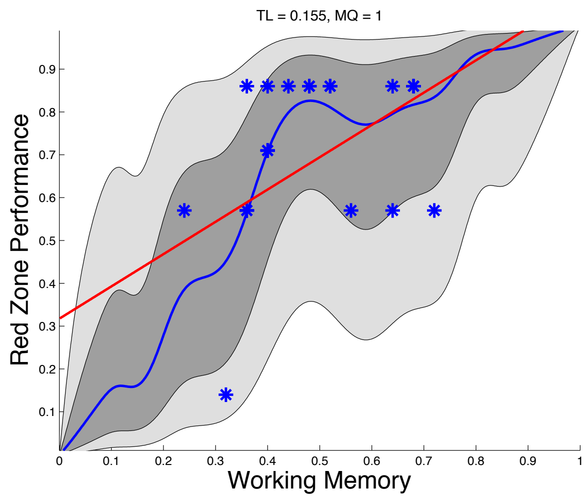


Figure 4: Example of probabilistic modeling of human capabilities, correlating performance with working memory.

In summary, there are many classes of models of human tasking, behaviors, and decision making. Importantly, there is a growing maturity in probabilistic models of human decision making can could enable fundamental studies in formal verification of human-autonomy systems.

Collaborative autonomy

Collaboration between autonomy and human actors can involve computation, communication, navigation, sensing, and other decision-making elements. A complex mission will require interaction between a distributed collection of decision-makers and physically-embodied vehicles. Decision-makers, vehicles, and soldiers may operate in shared or distinct environments. To-date, most work in collaborative autonomy has focused on optimization and planning for collections of homogeneous vehicles (e.g., unmanned aircraft) or computational nodes (e.g., cloud computing elements). Emphasis has been placed on scalability to large sets of tasks involving large quantities of data to collect and process.

There are two basic strategies for planning collaborative or multi-agent missions: centralized and decentralized. Centralized planners require comprehensive knowledge of the tasks and actors to accomplish them. A centralized planner can devise a globally-optimal solution but scalability is a major drawback. Truly decentralized planners enable individual actors or local actor groups to coordinate their activities through protocols that are generally intuitive and emphasize scalability.

Substantial research has focused on organizing the motions of physically-embodied vehicles asked to collaboratively accomplish a particular mission. Cooperative path planning algorithms have been developed to efficiently and autonomously allocate and plan vehicle motions. [59, 60] Scalable cooperative control or “swarming” algorithms have been devised to organize vehicle motions through information consensus [61], including cooperative timing and other coordination functions. [62]

For missions specified by a set of computational and communication tasks, decentralized task allocation and selection algorithms have been developed. For example, market-based (auction) protocols [63, 64] use agent bidding to determine an allocation and scheduling policy; research has shown such systems will reach equilibrium, and decentralized algorithms scale well to large collections of agents. Consensus-based algorithms have also been applied to distributed task allocation problems. [65]

Humans also play a central role in collaborative systems, even when these systems include state-of-the-art autonomy elements. Early human-robot interaction placed humans in supervisory and teleoperation roles; with this structure the only "autonomy" was realized in converting a task-level instruction into a motion sequence. Automation for vehicles, e.g., aircraft flight management systems, has evolved such that onboard or remote pilots supervise rather than directly controlling the vehicle. Human-robot interaction researchers have also studied protocols for effectively interacting, establishing metrics that enable a robot to determine when and how to ask for help. [66] Recognition of natural motions or gestures also shows promise as a means to reduce workload for humans sharing an environment with an autonomous robot or vehicle. [67]

Parasuraman [68] describes the notion of adaptive automation to optimize or balance workload and maintain human situational awareness. The adaptation or switching process itself may be user-controlled or system-driven. While it is generally agreed that humans are most versatile in understanding complex and novel data and translating this data to decisions, automation will be capable of monitoring more data with lower latency given sufficient computational resources. Any new or more extensive autonomy must not only demonstrate acceptable performance but also must gain trust through use over time.

For physically-proximal collaboration, researchers have begun investigating methods that enable a robot to interact with a human companion without requiring the human to explicitly communicate his/her perceptions and goals. Example studies include prediction of human intent and perspective-taking. In perspective-taking, an actor projects him/itself into the perceptual and cognitive state of a companion. In doing so, the actor can better predict how to act or react so that the collaborative goal can be achieved. Such behaviors are difficult to program exhaustively but early work has shown promise with robotic systems learning through a series of tasks with analogue to human child learning. [69]

Human intent prediction, accomplished to-date using uncertain modeling methods such as the computationally-intensive partially-observable Markov Decison Processes, [70] would enable a robot and human to work together without the requirement for the human to verbalize or gesture each task or action he/she intends to accomplish. Such a capability can enable more closely-coupled activities with reduced chance of physical or mental conflict. Intent prediction has also been used from the perspective of assisting data analysts, [71] although to-date analysts still prefer to use automation as assistive tools rather than trusting it over the full data-to-decision cycle.

Model-based system engineering

A variety of systems and software engineering processes have been adopted by industry. In software engineering, the Royce Waterfall and Boehm Spiral models [72] were devised to form logical processes by which potentially complex systems could be specified, designed, implemented, and tested over their product lifecycles. Such models were found to be challenging to use in practice, particularly when baseline requirements changed. The V or "Vee" model for systems engineering [73] provides an improved scheme for functional systems engineering in that it accounts for an evolving baseline. Shown in Figure 5, the baseline product/system requirements evolve to a design with progressively increasing detail. The development cycle creates the left leg of the V, while the testing process forms the right leg. The steps in development include decomposition, definition, and verification, while the right leg represents validation through testing. Correct-by-construction methods can support verification prior to testing (on the left leg) which can in turn improve the efficiency and completeness of the testing (validation) phases of the life cycle.

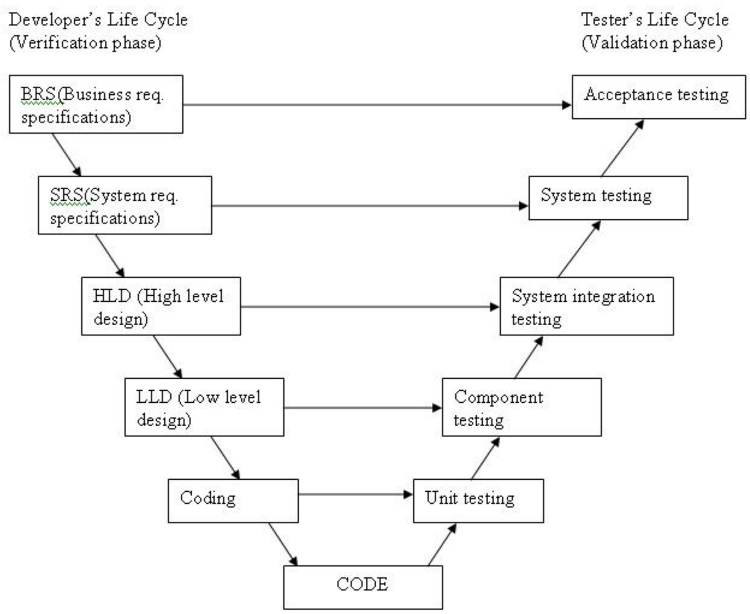


Figure 5: The V-Model Systems Engineering Process (http://istqbexamcertification.com/wp-content/uploads/2012/01/V-model.jpg)

Functional system design performed using any of the traditional systems engineering methods can result in a highly-complex and coupled overall design, even if a hierarchical functional structure such as that shown in Figure 6 is used to simplify the specification of each component or subsystem. Since compehensive (exhaustive) testing is not feasible, it is difficult to identify cases where a specific and unusual combination of parameter values causes a problem. System developers therefore have tried to build tests that cover the space as well as possible.

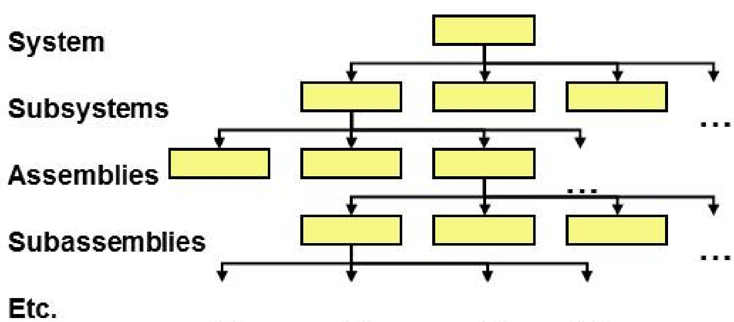


Figure 6: Functional Decomposition

Model-based systems engineering [72, 74] provides a more object-oriented methodology. Shown in Figure 7, all lifecycle processes are devised around a suite of subsystem or component models.

Substantial emphasis is placed on generating correct state-based models for each "actor" in the system, then these models are re-used throughout all phases of system engineering. The modularity and reusability afforded by the model-based systems engineering process improves scalability and consistency over functional design schemes. Tools for state analysis offer a formal method to verify behaviors to augment designer scrutiny of tabulated requirements their satisfaction in product design. Model-based engineering has been successfully applied to a number of domains, including automotive [75, 76] and spacecraft mission design [77].

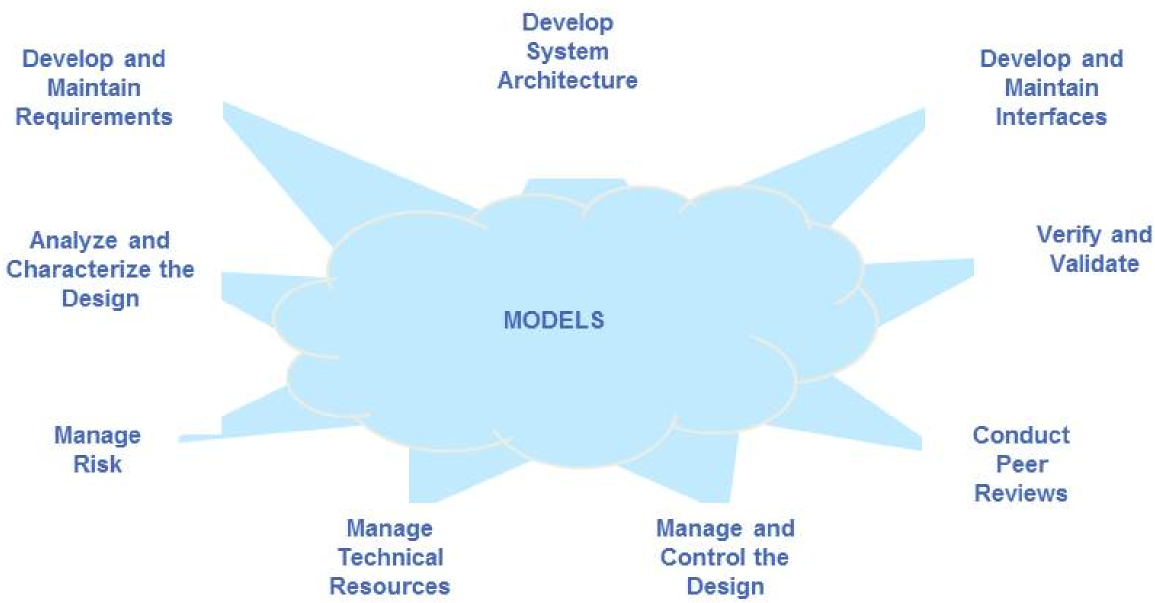


Figure 7: Model-based System Engineering [77]

Most products designed with the above systems engineering principles have resided on a single physically-embodied vehicle or system plus any network-based support, e.g., spacecraft mission control, a ground control station for an unmanned aircraft system (UAS), connection to the "cloud" for database or computational resource access. Researchers have begun working toward application of model-based systems engineering to multi-agent domains [78]. The term "multi-agent" can be used to represent agent-based models applied to a system with a single or multiple physical embodiments. The scalability and re-use offered by model-based systems engineering is also captured by agent-based modeling paradigms.

Reference Mission

To ground the discussion of collaborative human-autonomy systems a reference mission is proposed that involves human-autonomy teams performing a diverse set of complementary tasks. The overall mission objective is to identify thermal signatures representing persons of interest, and to detect and avoid or disable improvised explosive devices (IEDs) as they are encountered.

This mission, depicted in Figure 8, requires a pervasive collection of human and autonomy teams that acquire and share data, identify targets, and react appropriately. In the field, teams of warfighters may be supported by ground and over-the-horizon unmanned aircraft as well as personnel carriers. The robotic systems can serve as scouts, providing both over-the-horizon and over-the-shoulder situational awareness. Increased autonomy can enable the human team members to maintain better awareness of their own environment, increasing their effectiveness and reducing risk to themselves. This risk is particularly high when a teleoperator projects his senses into the robot; if the robot could instead receive high-level commands the soldier could remain more aware and mobile during the operation. To complement local actors, high-altitude UAVs can provide a bird's-eye view of the operating area, collecting substantial data to share with remote operators and analysts.

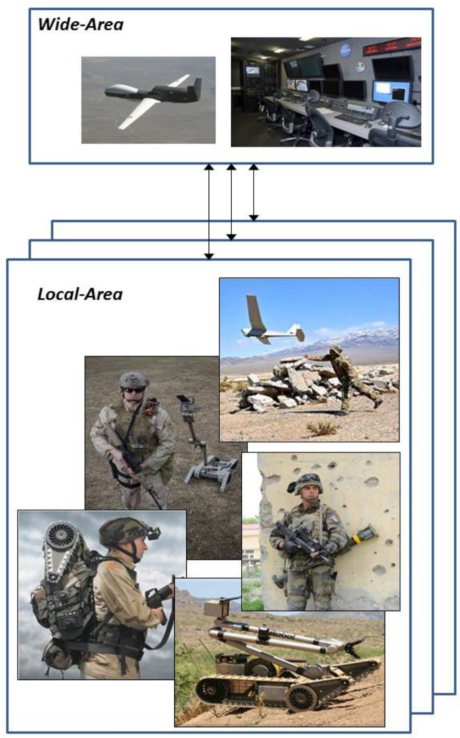


Figure 8: A Reference Mission requiring Cooperative Human-Autonomy Teams

In this scenario, ground robots might be responsible for IED detection and disarming to reduce risk to their human companions. The human actors that deploy and operate locally would be responsible for directly observing the environment, interacting with entities in the environment including the local population, and making real-time decisions based on the mission and their observations. Hand-carried devices such as [secure] cell phones can provide the soldier handheld communication and data display capability necessary to interact with local actors and remote operators and analysts in data centers. Manned vehicles would offer a suite of larger onboard sensors, computers, and communication devices to support the team. Today, roles and responsibilities are manually determined. In the future, progress in human-autonomy research can enable a more seamless end-to-end operational paradigm that improves mission effectiveness and efficiency while also reducing risk to soldiers in the field.

Key Research Directions

Given the motivation from the DSB and other DoD studies, and the background in these five areas, our goal was to identify capability gaps and key research objectives that integrate autonomy and humans (as command, analysts, operators, and soldiers) that are:

- Model-based, probabilistic, and verifiable (correct by construction)
- Scalable: self-organizing networks of heterogeneous actors that perform navigation, sensing, communication, and computation
- Agile: can be rapidly validated and deployed for specific exploitation missions

Research directions are described below that will enable these goals.

Probabilistic Models of Human Capabilities in Correct by Construction Frameworks

A key research direction is to leverage probabilistic models of human capabilities, and integrate these models into formal frameworks that enable the generation of correct by construction controllers for autonomous systems. Importantly, not all human capabilities have to be modeled; only *some* capabilities. And, these models can be probabilistic because the current state of the art in formal methods includes formal tools that consider probability.

These models can be derived from underlying knowledge of human capabilities, but will mostly be derived from data collected from human tests. Models could be general (a group of humans), or custom (a specific person). A key challenge is minimizing the data collection required to develop such models. As models of human capabilities mature, they can be integrated into the same framework, and new controllers will be automatically generated that are richer in their ability to work together with humans.

If a framework exists whereby probabilistic models of human capabilities are integrated into a verifiable framework, then we could:

- Automatically verify (and auto-generate) correct by construction controllers for autonomy. These controllers would be designed with the *integrated* human+autonomous system in mind because the specifications are on the task itself, and the models include both the humans and autonomy.
- Generate autonomy software that is 'matched' to human capabilities. Given that a different probabilistic model of human capabilities could be developed for different people, it is envisioned that the associated controllers that are automatically generated will consider each person individually.
- Speed validation. In any V&V framework, any improvement in verification (such as increasing speed with automatic verification of formal methods), will in turn speed the end validation process.

As an example, consider the case of a network of humans and autonomous robots working together on a task, such as surveillance of an area, rescue mission in a forest or building, or other. Figure 9 shows an abstraction of such a network, with N autonomous vehicles. For this example, we will assume that the human analyzes data feeds for objects and false detections. During a prior testing, models of human capabilities of three particular operators showed that they each have important characteristics:

- Operator A has faster eye-hand coordination
- Operator B has a photographic memory
- Operator C's performance drops rapidly with task load

Given models of these capabilities, the concept of *'personalized autonomy'* can be realized whereby the correct-by-construction controller for the autonomous takes these abilities into account. Importantly, the controller is designed with *collaborative performance* for the task in mind, not separately. This capability would be useful not only for the original design effort, but also operational deployment with the concept of adaptive tasking [34].

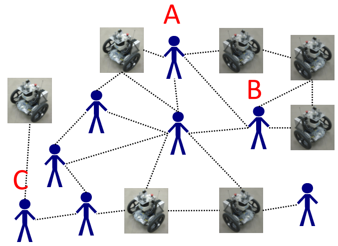


Figure 9: A network of humans and autonomous systems working together on a task.

Rapid Validation

A key challenge in future systems is the speed and ability to validate systems prior to deployment. Future autonomy (adaptive, agile) are near infinite state systems, and traditional empirical validation (Test & Evaluate, T&E) will not scale [1].

An example from the DARPA Urban Challenge (DUC) in 2007 crystallized these challenges [79,80]. The Cornell team developed an autonomous driving car for the competition. In the 12-18 months prior to the competition, the Cornell team tested and evaluated sub-components and the integrated system, with much of the final six months devoted to systems level testing and evaluation (i.e. validation). During the semi-finals of the DUC, Cornell's car, Skynet, could not navigate through a course with many cars parked in either side of the road; this was because the road was narrow, and Skynet is large (a Chevrolet Tahoe). The only way to navigate the course was enable Skynet to slightly cross over the double yellow line in the middle of the road - a capability that the Cornell team took great lengths to avoid.

In order to make it into the DUC finals, the Cornell team had to make a small software change (a few lines of code) in order to enable Skynet to navigate closer to side cars and over the double yellow line. However, the Cornell team kept asking the same questions: *Will this change create other problems? Will this change invalidate the months of validation tests that the team had completed*. In the end, Skynet completed both the semi-finals and finals, completing the competition as one of six finishers. However, not without a lot of stress about not knowing the effects of such a small change.

One 'promise' of a verifiable, probabilistic, correct by construction framework is a shorter validation time. We envision that such tools will enable this improvement. Consider the case of a single operator and UAV, working together on a surveillance task (Figure 10). The specific task is to find and identify certain people in crowded spaces. The operator and UAV must work together on such a task, as the UAV must position itself appropriately (height, location); search and maintain camera field of view on important people; continue to fly why completing its sensing task. The human tasks the UAV to a general area, watches the video, and picks the object of interest that the UAV must track. Consider that the UAV, software, and operator interface were developed, and validation tests occurred over many months that demonstrated reliability to a high level of probability.

Here is an important question: *After one year in operation, the DoD wants to add a new capability, that of automatic entity detection in the UAV software, and use this detection to re-task the UAV.* Will the past T&E validation be invalidated? i.e. not useful any more? Or can some be used? How much additional T&E is required to deploy the system again?

Further, acceptance should be based on the completed task (search, ID, track objects on the ground), and this task is dependent on *both* the human and UAV. Thus, if the UAV performs better at its tasks, but the human does not understand/trust the new software, the *integrated* human+UAV system could perform worse than it did previously.

Figure 10: An operator tasking a UAV.

While the proposed framework may not solve all problems, it is conceivable that if the software for the UAV was auto-generated to create probabilistically correct by construction (verified) software, then it is conceivable that the addition of UAV capabilities (and perhaps some human tests/models) would enable a faster verification process.

Multi-scale architectures for human autonomy

The model-checking needed by correct-by-construction controllers could limit the scalability of autonomy. Model-checking generates controllers, validates correctness, and ensures performance based on an exploration of the state-space of a finite model system. As autonomy scales, so does the state space, making it impossible to evaluate models that describe the entire system.

To avoid the explosion of state, systems must be decomposed into networks and hierarchies of interacting components that represent systems at multiple scales. Decomposition of a system into parts can be done by data, task, actor, or localization.

This approach to scalability inherits from service-oriented architectures used to build Internet and Cloud software in which complex systems are assembled from multiple simple services with well-defined interfaces. Software systems built in this manner are agile; they may be updated, modified or patched incrementally. Modifications are deployed directly into the running system after validation against the interface specification. Often they are deployed to small fractions of the total system for testing on the operational system. Modifications are rolled back if they disrupt function. This is in contrast to monolithic software engineering processes in which entire systems are built, verified, validated, and deployed. The time to deploy any new function in monolithic software grows with system complexity and systems quickly become unmanageable.

We provide an abstract example that demonstrates how a system-wide goal can be achieved hierarchically by a network of independently autonomous actors coordinating locally with other system elements. The example is one of autonomously navigating UAVs of sensing an infra-red spatial field to specified accuracy, i.e. to a maximum known error across the sensing space. The example leverages a Gaussian process estimation technique, known as kriging, to localize information and allow the sensors to act independently or in small groups. Kriging has the property that it estimates a field based on a finite number of sensors and, more importantly, provides known error on the field estimate. Figure 11 shows this process, demonstrating the original field and sensor locations (left), the estimated field (center), and the known error in the estimate (right). Because the estimate for the field has local support, sensors communicating locally with a small set of other sensors can determine an overestimate of the local error. They use this information to autonomously navigate to regions of high error, which

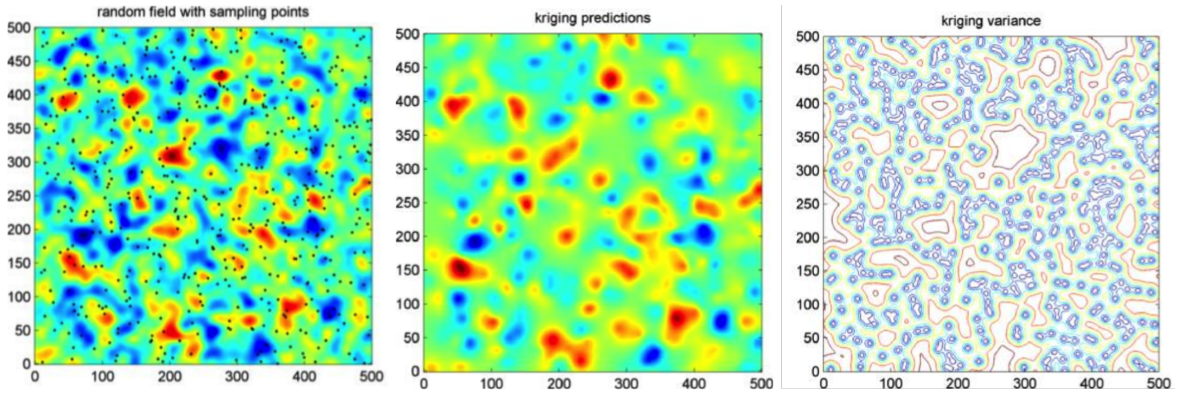


Figure 11: A heatmap visualization of a random field to be sensed and the sampling points, i.e. the present locations of the UAVs (left). The approximation of the field based on kriging (Gaussian process estimation) of the current sensor locations (center). The error in the sensed field (right). Images from `http://www.mathworks.com/matlabcentral/fx_files/29025/1/kriging.jpg`.

will drive down global error. Figure 12 shows one possible navigation strategy used by a small group collaborating based on local information. They use a greedy approach in which they navigate to closest areas of high error. Navigation will continue, monotonically decreasing error in the field, until the goals have been met.

This example demonstrates desirable properties of autonomous systems that will achieve scale: it is *goal-based* and *declarative*. With goal-based autonomy, one expresses the objective of the system in a simple statement, in this case, sense the field with known error. Goal-based autonomy allows us to assemble multiple autonomous systems in hierarchies and networks. The simplicity of the goal means that this autonomous subsystem has a well known interface for other systems that want to consume its output. Describing the outcome of a complex system in a simple goal will allow multiple such systems to interact and will be necessary to scale autonomy to the battlespace. Declarative indicates that the subsystem is tasked by a statement of *what to do* rather than *how to do it*. This allows the subsystem to self-optimize, choosing from among many possible execution strategies based on local knowledge. Declarative strategies compartmentalize complexity, allowing each system to use the high-resolution data needed to accomplish its goal, but hide that complexity from the overall system.

Human-Automation System Co-Design

Model-based systems engineering holds promise to improve scalability and reusability in complex engineered systems but to-date has only been used to design automation and autonomy system elements. In traditional systems engineering processes, humans are primarily considered through interfaces specified in the design, not as integral elements of the system being designed. There is good reason for this trend: while a hardware and software system element can be fully customized, the human cannot and should not be “customized” in the sense of being genetically altered or reconstructed. However, human capabilities are versatile and can be observed/modeled. A human actor can then be trained to effectively assume a particular role in a collaborative human-automation system, and this human training can itself be optimized as part of the overall system design.

We propose the study of human-automation system *co-design* in which the systems engineering

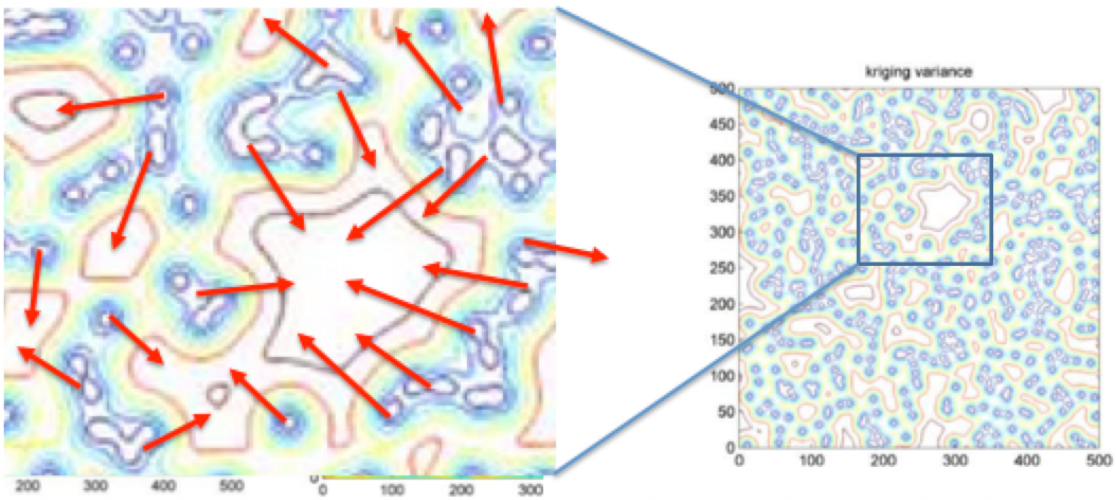


Figure 12: UAVs navigating to regions of high error based on local information when collaboratively sensing a field. Original images from `http://www.mathworks.com/matlabcentral/fx_files/29025/1/kriging.jpg`. The zoom effect and the motion of sensors indicated by arrows have been added.

effort models and optimizes the holistic system over human-machine system trade spaces. [1] For a truly integrated co-design process, appropriate models, metrics, design space variables, and constraints must be defined across both human and autonomy elements of the system. Designs should be free to assign each role to human actors, autonomous actors, or a team of both based on defined models and metrics.

A number of challenging research issues must be addressed before co-design can be effective. Open questions include:

- How do we model the roles and capabilities of the human?
- How do we allocate human and autonomy elements during the design process?
- How do we define common metrics for the human and autonomy elements?

The outcome of the co-design systems engineering process will be a product that requires human and autonomy actors to prepare and deploy effectively over a long term. The system lifecycle will involve designing then executing missions, learning from results, and adapting the system and actors to improve future mission effectiveness. The implementation of a co-designed subsystem will look quite different depending on whether the role assigned to that subsystem is accomplished by a human or autonomy actor. The basic question to be addressed for each is:

- *Human:* What training and profiling and profiling will best enable accurate model development and mission readiness?

[1]This process of making design choices over a complex systems trade space including humans and automation is cited as one of the three primary views for making decisions in "The Role of Autonomy in DoD Systems" (Defense Science Board, July 2012).

- *Autonomy:* What software and hardware development, database preparation, and verification activities will optimize capabilities?
- *Both:* How will human-autonomy actors (subsystems) interact, and over what time scales can mission design and adaptation be performed?

Consider the person of interest and IED detection reference mission proposed above and shown previously in Figure 8. This mission contains some capabilities that are unique to humans and autonomy actors and others that are candidates for co-design. Humans are uniquely able to directly interact with the local population, and humans will be assigned responsibility for weapon handling and discharge. Airborne sensors will almost certainly be carried by unmanned aircraft, and autonomy will be responsible for high-speed data manipulation and communication. On the other hand, data-to-decisions activities may be assumed by humans, autonomy, or a collaboration of both. Human and autonomy actors are both capable of collecting and disseminating data as well as mobilizing assets.

Human-autonomy interactions are required at the mission level (wide-area) and in the immediate area of the operation. Wide-area collaborations will center around big data analysis and interpretation, while local-area collaborations will strive to achieve shared and sufficiently comprehensive situational awareness. Local-area collaborations will also need to offer effective support for soldiers particularly in life-threatening, fast-paced situations. Collaboration between wide-area and local-area system elements is crucial and will involve direct communication (voice/text) from human actors in combination with data streams from autonomy and collaborative system elements. While the co-designed collaborative human-autonomy system will be unquestionably “complex”, achieving an effective co-design process holds substantial promise for improving future mission effectiveness.

Implications: Potential Impact

This think piece recommends the pursuit of fundamental research that contributes to bridging the human-autonomy gap through a combination of probabilistic modeling of human capabilities, formal methods in verification, multi-scale approaches, and human-automation co-design. We envision that our research will impact many, if not most, autonomous systems that inherently require interaction between humans and autonomy. Examples include service robotics; personal robotics; small businesses such as in design/manufacturing/assembly/packaging/shipping of products; planetary exploration; the national air space; and defense and intelligence applications.

Successful research and development in this area will lead to the following important contributions:

- Better (seamless, natural, efficient) collaboration between humans and autonomy, addressing the DSB human-autonomy goals
- More effective use of humans with autonomy
- Faster and more reliable verification and validation of autonomy (and humans+autonomy)
- Robust deployment of human-autonomy systems at the mission scale (not the subsystem scale)
- Human and autonomy elements, optimized via system co-design for specific taskings

Our think piece also brought up many additional questions, both because of the challenging area we were studying, and the limited scope of the think piece. These include:

- How can acceptance be addressed? Our initial goal was to make inroads into acceptance. However, this topic was challenging and we were not able to make strong statements in this area. Why? Are there technical approaches that can more easily address acceptance?

- How far can modeling of human capabilities go? The use of models of human capabilities received mixed reaction from some, primarily because of the challenge of the task (e.g. accuracy) and how much has to be modeled. It is hypothesized here that even *some* modeling of human capabilities would be helpful in the proposed verification framework. What does some mean? What capabilities achieve the best collaborative results? How accurate do these models have to be? What if the model is incomplete? What about difficult to model situations, such as emergencies?
- Can we formalize trust? Clearly, the adoption of automation is intimately tied to the ability of humans to trust that it will work for them. Can trust be modeled? Can trust be incorporated into the design/validation approach in order to speed acceptance?
- How do we model the psychology of human beings?
- How do we define a probabilistic specification for human+autonomy systems? Is it a worst case (operator falls asleep), or an average/variation? What if a system falls outside of the specified bounds, either in design or in operation?
- A formal task specification is probabilistic, with a particular level of success probability. What happens when the situation is outside the bounds? How will the system degrade? Current formal V&V methods do not address this, but could they? Would a mediation, or a correction mechanism work?
- How can formal methods enable verifiable interactions between humans and automation, which may include feedback loops (verifying a command can be met) and adaptability (changing or switching tasks to provide a higher probability of success).
- How can the fact that humans learn over time be leveraged? For example, a man or woman in combat for six months will do a lot better than a starting person. Also, additional information on the human can be collected, such as how they handle emergency situations.
- Can acceptance be accomplished without full system validation? Currently, the state of the art in acceptance is full testing and evaluation, with the scale of the approach proportional to the complexity of the technology. With the promise of formal methods to generate validated controllers, it is clearly a key question as to how to leverage this fundamental characteristic in an effort to reduce acceptance times, which are scaling out of control. Will modular approaches with formal methods improve acceptance?
- How will real and perceived risk change? Human perception of outcomes can be different when comparing autonomy and humans. For example, compare the reaction of an accident or munition if the cause was by a human or by an autonomous system. At times, it appears that autonomy could be held to a higher standard. Are there ways that technology can be developed that influence the real/perceived risk?
- What are the security risks with formal verification approaches?

These questions are important to the overall implications and impact of the research directions, and therefore must be addressed in concert with the technological developments of the proposed research directions.

Acknowledgements

This think piece, and our experiences with the DSSG program in general, were enriched by many, and we would like to acknowledge them here. In particular, visits to LANL (including the Data Science at Scale Team), NRL, Army/Marine facilities, and DARPA led us to consider this as a possible topic and helped us frame the key issues. The use and adoption of technology, as well as the openness of our researchers and troops to their use were clear at many of these stops. Rebecca Grier, Frank Moses, Shelley Cazares at IDA contributed suggestions, references, and thoughtful dialogue on our topic, making our think piece much stronger. A visit to Palantir near the end of our DSSG sessions provided a nice glimpse at how a company has worked with the DoD to enable the adoption of new software and technology to support analysts. We would like to thank Raja Parasuraman (GWU), Scott Galster (AFLR/HE), and Hadas Kress-Gazit (Cornell), each of whom conduct research in related fields (human factors, formal methods), for their patience in answering questions, support with references and ideas, and interesting dialogue on our topic.

Importantly, we would like to thank the mentors and our DSSG colleagues for making the DSSG experience a truly remarkable one for us at this point in our careers. The discussions about DSSG, DoD and life in general as we made our way around the country were amazing. The mentors have shown us that we can be in awe of them not only for their past service, but their continued selflessness in giving back to this type of program. While we will miss our DSSG colleagues, we know that we have dinner plans any time we travel to your university. We look forward to keeping in touch.

Finally, we would like to thank Bob Robert and Katie Gliwa. Bob for starting the DSSG program, and coming back to run it now with the enthusiasm of a new DSSG member. And Katie for her remarkable professionalism, organization and wit.

References

[1] Office of the US Air Force Chief Scientist, "Technology horizons: A vision for air force science and technology 2010-30," September 2011.

[2] B. S. Blanchard and W. J. Fabrycky, *Systems Engineering and Analysis*. Prentice Hall, 2010.

[3] B. W. Boehm, "A spiral model of software development and enhancement," *Computer*, vol. 21, no. 5, pp. 61–72, 1988.

[4] E. M. Clarke, O. Grumberg, and D. A. Peled, *Model Checking*. Cambridge, Massachusetts: MIT Press, 1999.

[5] L. Lamport, *Specifying Systems: The TLA+ Language and Tools for Hardware and Software Engineers*. Addison-Wesley, 2002.

[6] P. Zave, "Using lightweight modeling to understand Chord," in *Computer Communication Review*, 2012.

[7] M. Arye, R. Harrison, R. Wang, P. Zave, and J. Rexford, "Toward a lightweight model of BGP safety," in *Workshop on Rigorous Protocol Engineering*, 2011.

[8] E. Cohen, M. Dahlweid, M. A. Hillebrand, D. Leinenbach, M. Moskal, T. Santen, W. Schulte, and S. Tobies, "VCC: A practical system for verifying Concurrent C," in *Theorem Proving in Higher Order Logics*, 2009.

[9] "Spin: Model checker." http://spinroot.com.

[10] "Nusvm: Model checker." http://nusmv.fbk.eu.

[11] "An extensible sat-solver," in *SAT* (E. Giunchiglia and A. Tacchella, eds.), vol. 2919 of *Lecture Notes in Computer Science*, (Santa Margherita Ligure, Italy), pp. 502–518, Springer, 2003.

[12] M. Herbstritt, "zchaff: Modifications and extensions," 2001.

[13] M. Kwiatkowska, G. Norman, and D. Parker, "PRISM: Probabilistic symbolic model checker," in *Proc. Tools Session of Aachen 2001 International Multiconference on Measurement, Modelling and Evaluation of Computer-Communication Systems* (P. Kemper, ed.), pp. 7–12, September 2001.

[14] "Prism: Probabilistic model checker." http://www.prismmodelchecker.org/.

[15] M. Antoniotti and B. Mishra, "Discrete event models + temporal logic = supervisory controller: Automatic synthesis of locomotion controllers," *IEEE International Conference on Robotics and Automation*, 1995.

[16] G. E. Fainekos, H. Kress-Gazit, and G. J. Pappas, "Hybrid controllers for path planning: A temporal logic approach," in *Proceedings of the 44th IEEE Conference on Decision and Control*, pp. 4885 – 4890, December 2005.

[17] H. Kress-Gazit, G. E. Fainekos, and G. J. Pappas, "Temporal-logic-based reactive mission and motion planning," *IEEE Transactions on Robotics*, vol. 25, no. 6, pp. 1370–1381, 2009.

[18] T. Wongpiromsarn, U. Topcu, N. Ozay, H. Xu, and R. M. Murray, "TuLiP: A software toolbox for receding horizon temporal logic planning," in *International Conference on Hybrid Systems: Computation and Control*, 2011. submitted, software available at `http://www.cds.caltech.edu/tulip`.

[19] B. Johnson, F. Havlak, M. Campbell, and H. Kress-Gazit, "Execution and analysis of high-level tasks with dynamic obstacle anticipation," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pp. 330–337, IEEE, 2012.

[20] T. Sheridan, *Telerobotics, Automation, and Human Supervisory Control*. Cambridge, MA: The MIT Press, 1992.

[21] J. R. Anderson, M. Matessa, and C. Lebiere, "Act-r: A theory of higher level cognition and its relation to visual attention," *Human-Computer Interaction*, vol. 12, no. 4, pp. 439–462, 1997.

[22] N. I. of Health, "The brain initiative." http://www.nih.gov/science/brain/, 2013.

[23] D. E. Kieras and D. Meyer, "Epic: A cognitive architecture for computational modeling of human performance." http://www.eecs.umich.edu/ kieras/epic.html.

[24] D. E. Kieras, D. E. Meyer, J. A. Ballas, and E. J. Lauber, "Modern computational perspectives on executive mental processes and cognitive control. where to from here?," Tech. Rep. TR-98/ONR-EPIC-12, University of Michigan, 1999.

[25] M. D. Byrne, "The human-computer interaction handbook: Fundamentals, evolving technologies and emerging applications," in *Cognitive architecture* (J. Jacko and A. Sears, eds.), Mahwah, NJ: Lawrence Erlbaum, 2003.

[26] M. Byrne, "Act-r/pm: Act-r perceptual-motor." http://chil.rice.edu/projects/RPM/index.html.

[27] C. Lebiere, J. R. Anderson, and D. Bothell, "Multi-tasking and cognitive workload in an act-r model of a simplified air traffic control task," in *Proceedings of the 10th Conference on Computer Generated Forces and Behavior Representation*, 2001.

[28] N. A. Taatgen, C. Lebiere, and J. R. Anderson, "Modeling paradigms in act-r," *Cognition and multi-agent interaction: From cognitive modeling to social simulation*, pp. 29–52, 2006.

[29] B. J. Best and C. Lebiere, "Cognitive agents interacting in real and virtual worlds," *Cognition and multi-agent interaction: From cognitive modeling to social simulation*, pp. 186–218, 2006.

[30] N. Cooke, H. Pringle, H. Pedersen, and O. Connor, "Human factors of remotely operated vehicles. advances in human performance and cognitive engineering (vol. 7)," 2006.

[31] M. L. Cummings, A. S. Brzezinski, and J. D. Lee, "Operator performance and intelligent aiding in unmanned aerial vehicle scheduling," *Intelligent Systems, IEEE*, vol. 22, no. 2, pp. 52–59, 2007.

[32] M. J. Barnes and F. Jentsch, *Human-robot interactions in future military operations*. Ashgate Publishing, Ltd., 2010.

[33] R. Parasuraman, S. Galster, P. Squire, H. Furukawa, and C. Miller, "A flexible delegation-type interface enhances system performance in human supervision of multiple robots: Empirical studies with roboflag," *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, vol. 35, no. 4, pp. 481–493, 2005.

[34] R. Parasuraman, K. A. Cosenzo, and E. De Visser, “Adaptive automation for human supervision of multiple uninhabited vehicles: Effects on change detection, situation awareness, and mental workload.,” *Military Psychology*, vol. 21, no. 2, p. 270, 2009.

[35] C. Wickens, S. Dixon, and M. Ambinder, “Workload and automation reliability in uavs.,” in *Human Factors of Remotely Operated Vehicles* (N. Cooke, H. Pringle, H. Pedersen, and O. Connor, eds.), 2006.

[36] M. L. Cummings and P. J. Mitchell, “Predicting controller capacity in supervisory control of multiple uavs,” *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, vol. 38, no. 2, pp. 451–460, 2008.

[37] E. de Visser and R. Parasuraman, “Adaptive aiding of human-robot teaming: Effects of imperfect automation on performance, trust, and workload,” *Journal of Cognitive Engineering and Decision Making*, 2011 (in press).

[38] S. Galster, B. Knott, and R. Brown, “Managing multiple uavs: Are we asking the right questions?,” in *Proceedings of the Human Factors and Ergonomics Society 50th Annual Meeting*, (San Francisco, CA), pp. 2639–2643, Human Factors and Ergonomics Society, 2006.

[39] R. Parasuraman and V. Riley, “Humans and automation: Use, misuse, disuse, abuse,” *Human Factors*, vol. 39, no. 2, pp. 230–253, 1997.

[40] S. M. Galster, J. A. Duley, A. J. Masalonis, and R. Parasuraman, “Air traffic controller performance and workload under mature free flight: Conflict detection and resolution of aircraft selfseparation,” *International Journal of Aviation Psychology*, vol. 11, pp. 71–93, 2001.

[41] E. Rantanen, “Development and validation of objective performance and workload measures in air traffic control,” Tech. Rep. Technical Report AHFD-04-19/FAA-04-7, University of Illinois, Urbana Champaign, Sept 2004.

[42] X. Fan, J. Yen, and P.-C. Chen, “Learning HMM-based cognitive load models for supporting human-agent teamwork,” *Cognitive Systems Research*, vol. 11, pp. 108–119, 2010.

[43] F. Heger and S. Singh, “Sliding autonomy fir complex coordinated multi-robot tasks: Analysis and experiments,” in *Robotics Science and Systems*, 2006.

[44] B. Donmez, M. Cummings, and C. Nehme, “Modeling Workload Impact in Multiple Unmanned Vehicle Supervisory Control,” *IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans*, vol. 40, no. 6, 2010.

[45] N. Ahmed, E. deVisser, T. Shaw, A. Mohamed-Ameen, M. Campbell, and R. Parasuraman, “Statistical modeling of networked human-automation performance using working memory capacity,” *Ergonomics*, vol. in press, 2013.

[46] F. Bourgault, N. Ahmed, D. Shah, and M. Campbell, “Probabilistic operator-multiple robot modeling using bayesian network representation,” *Proc. GNC, Hilton Head, SC*, 2007.

[47] N. Ahmed and M. Campbell, “Multimodal operator decision models,” in *ACC 2008*, (Seattle), 2008.

[48] N. Ahmed and M. Campbell, “Variational Bayesian learning of probabilsitic discriminative models with latent softmax variables,” *IEEE Trans. on Sig. Proc.*, vol. 59, no. 7, pp. 3143–3154, 2011.

[49] N. R. Ahmed and M. E. Campbell, “On estimating simple probabilistic discriminative models with subclasses,” *Expert Systems with Applications*, vol. 39, pp. 6659–6664, June 2012.

[50] F. Bourgault, A. Chokshi, J. Wang, D. C. Shah, J. R. Schoenberg, R. Iyer, F. Cedano, and M. E. Campbell, “Scalable Bayesian human-robot cooperation in mobile sensor networks,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2342–2349, IEEE, Sept. 2008.

[51] N. Ahmed, E. Sample, and M. Campbell, “Bayesian multi-categorical soft data fusion for human-robot collaboration,” *IEEE Transactions on Robotics*, vol. 29, pp. 189–206, February 2013.

[52] N. Ahmed and M. Campbell, “Variational Bayesian data fusion of multi-class discrete observations with applications to cooperative human-robot estimation,” in *Proc. International Conf. On Robotics and Automation (ICRA 2010)*, (Anchorage, AK), pp. 186–191, 2010.

[53] N. Ahmed, E. Sample, K. Ho, T. Hoossainy, and M. Campbell, “Categorical soft data fusion via variational bayesian importance sampling with applications to cooperative search,” in *American Control Conference (ACC)*, pp. 1268 –1273, 2011.

[54] N. Ahmed and M. Campbell, “Variational Bayesian learning of probabilsitic discriminative models with latent softmax variables,” *IEEE Transactions on Signal Processing*, vol. 59, no. 7, pp. 3143–3154, 2011.

[55] N. Ahmed, E. de Visser, T. Shaw, A. Mohamed-Ameen, M. Campbell, and R. Parasuraman, “Statistical modelling of networked human-automation performance using working memory capacity.,” *Ergonomics*, Dec. 2013.

[56] R. Engle, “Working memory as executive attention,” *Current Directions in Psychological Science*, vol. 11, pp. 19–23, 2002.

[57] R. Parasuraman, “Neurogenetics of working memory and decision making under time pressure,” in *Applied Human Factors and Ergonomics* (W. Karwowski and G. Salvendy, eds.), Boca-Raton, FL: Taylor and Francis, 2010.

[58] M. Endsley, “Toward a theory of situation awareness in dynamic systems,” *Human Factors: The Journal of the Human Factors and Ergonomics Society*, vol. 37, no. 1, pp. 32–64, 1995.

[59] A. Tsourdos, B. White, and M. Shanmugavel, *Cooperative path planning of unmanned aerial vehicles*, vol. 32. Wiley, 2010.

[60] J. Bellingham, M. Tillerson, A. Richards, and J. P. How, “Multi-task allocation and path planning for cooperating uavs,” in *Cooperative Control: Models, Applications and Algorithms*, pp. 23–41, Springer, 2003.

[61] W. Ren, R. W. Beard, and E. M. Atkins, “Information consensus in multivehicle cooperative control,” *Control Systems, IEEE*, vol. 27, no. 2, pp. 71–82, 2007.

[62] T. W. McLain and R. W. Beard, “Coordination variables, coordination functions, and cooperative timing missions,” *Journal of Guidance, Control, and Dynamics*, vol. 28, no. 1, pp. 150–161, 2005.

[63] M. P. Wellman, W. E. Walsh, P. R. Wurman, and J. K. MacKie-Mason, "Auction protocols for decentralized scheduling," *Games and Economic Behavior*, vol. 35, no. 1, pp. 271–303, 2001.

[64] W. E. Walsh and M. P. Wellman, "Decentralized supply chain formation: A market protocol and competitive equilibrium analysis," *J. Artif. Intell. Res.(JAIR)*, vol. 19, pp. 513–567, 2003.

[65] H.-L. Choi, L. Brunet, and J. P. How, "Consensus-based decentralized auctions for robust task allocation," *Robotics, IEEE Transactions on*, vol. 25, no. 4, pp. 912–926, 2009.

[66] T. Fong, C. Thorpe, and C. Baur, "Collaboration, dialogue, human-robot interaction," in *Robotics Research*, pp. 255–266, Springer, 2003.

[67] C. L. Nehaniv, K. Dautenhahn, J. Kubacki, M. Haegele, C. Parlitz, and R. Alami, "A methodological approach relating the classification of gesture to identification of human intent in the context of human-robot interaction," in *Robot and Human Interactive Communication, 2005. ROMAN 2005. IEEE International Workshop on*, pp. 371–377, IEEE, 2005.

[68] R. Parasuraman and C. D. Wickens, "Humans: Still vital after all these years of automation," *Human Factors: The Journal of the Human Factors and Ergonomics Society*, vol. 50, no. 3, pp. 511–520, 2008.

[69] J. G. Trafton, N. L. Cassimatis, M. D. Bugajska, D. P. Brock, F. E. Mintz, and A. C. Schultz, "Enabling effective human-robot interaction using perspective-taking in robots," *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, vol. 35, no. 4, pp. 460–470, 2005.

[70] A.-B. Karami, L. Jeanpierre, and A.-I. Mouaddib, "Human-robot collaboration for a shared mission," in *Human-Robot Interaction (HRI), 2010 5th ACM/IEEE International Conference on*, pp. 155–156, IEEE, 2010.

[71] E. Santos, H. Nguyen, Q. Zhao, and H. Wang, "User modelling for intent prediction in information analysis," in *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, vol. 47, pp. 1034–1038, SAGE Publications, 2003.

[72] J. A. Estefan *et al.*, "Survey of model-based systems engineering (mbse) methodologies," *Incose MBSE Focus Group*, vol. 25, 2007.

[73] K. Forsberg and H. Mooz, "System engineering for faster, cheaper, better," in *1999 Ninth annual international symposium (INCOSE), Brighton, England*, Citeseer, 1999.

[74] L. Balmelli, D. Brown, M. Cantor, and M. Mott, "Model-driven systems development," *IBM Systems Journal*, vol. 45, no. 3, pp. 569–585, 2006.

[75] A. Pretschner, M. Broy, I. H. Kruger, and T. Stauner, "Software engineering for automotive systems: A roadmap," in *2007 Future of Software Engineering*, pp. 55–71, IEEE Computer Society, 2007.

[76] E. Bringmann and A. Kramer, "Model-based testing of automotive systems," in *Software Testing, Verification, and Validation, 2008 1st International Conference on*, pp. 485–493, IEEE, 2008.

[77] D. Dvorak, R. Rasmussen, G. Reeves, and A. Sacks, "Software architecture themes in jpl's mission data system," in *Aerospace Conference Proceedings, 2000 IEEE*, vol. 7, pp. 259–268, IEEE, 2000.

[78] J. M. Gascueña, E. Navarro, and A. Fernández-Caballero, "Model-driven engineering techniques for the development of multi-agent systems," *Engineering Applications of Artificial Intelligence*, vol. 25, no. 1, pp. 159–173, 2012.

[79] M. E. Campbell, M. Egerstedt, J. P. How, and R. M. Murray, "Autonomous driving in urban environments: approaches, lessons and challenges.," *Philosophical transactions. Series A, Mathematical, physical, and engineering sciences*, vol. 368, pp. 4649–72, Oct. 2010.

[80] I. Miller and M. E. Campbell, "Team Cornell's Skynet: Robust perception and planning in an urban environment," *Journal of Field Robotics*, vol. 25, no. 8, pp. 493–527, 2008.