Holistic Modeling for Human-Autonomous System Interaction

M.L. Cummings

Duke University

Box 90300 144 Hudson Hall

Durham, NC 27708

919.660.5306

m.cummings@duke.edu

A. Clare

Massachusetts Institute of Technology

77 Massachusetts Ave

Cambridge, MA 02139

617.252.1512

aclare@mit.edu

Acknowledgements

This research was funded in part by the Office of Naval Research, the Air Force Office of Scientific Research, Charles River Analytics, and Aurora Flight Sciences.

Abstract

For complex systems that embed automation, but also rely on human interaction for guidance and contingency management, holistic models are needed that provide for an understanding of the individual human and computer elements, and address the critical interactions of such complex systems. Discrete event simulation (DES) models and system dynamics (SD) models are two different approaches that can be used to address these requirements. Both modeling approaches can support the designers of future autonomous vehicle (AV) systems by simulating the impact of alternate designs on vehicle, operator, and system performance. However, the DES modeling approach is likely best suited for using probabilistic distributions to accurately model an operator who is a serial processor of discrete tasks, as well as an environment with randomly occurring events. The SD modeling approach is better suited for modeling continuous performance feedback that is temporally dependent and is affected by qualitative variables such as trust.

Keywords: human performance modeling, discrete event simulation, system dynamics, autonomous vehicles

Relevance of findings: This article illustrates how discrete even simulation and system dynamics models can be used to explore and diagnose human-automation performance parameters**.** Such approaches allow designers the ability for cost and time effective evaluations without conducting extensive experimentation, which is critical for the rapidly evolving world of autonomous systems.

Introduction

A variety of empirical and analytical methods have been developed to inform design decisions for new systems that leverage increasing degrees of automation. Such methods include the use of human-in-the-loop studies with prototypes as test environments, cognitive walkthroughs, cognitive task analyses, heuristic analyses and other structured techniques for the development of predictive decision making (Crandall, et al., 2006; Dekker and Woods, 1999; Hollnagel, 2003; Klein, 2000; Klein, et al., 1989; Lewis and Wharton, 1997; Nielson and Mack, 1994; Novick and Hollingsed, 2007; Preece, et al., 1994; Rubin, et al., 1988; Smith, et al., 1998; Smith et al., 2006).

Various types of computational models and simulations have also been implemented to help designers envision the impact of their design on human and system performance. Such models range from extensions of the GOMS model (Card, et al., 1983; John, 1995; Kieras, 1991, 2004; St. Amant, et al., 2005; Williams, 2005) to the use of cognitive architectures that can be used to model performance in a given task/system context and generate performance predictions (Byrne and Kirlik, 2005). Examples of such cognitive architectures are ACT-R (Anderson, 1993; Anderson, et al., 2004), SOAR (Laird et al., 1987; Nason, et al., 2005) and EPIC (Kieras and Meyer, 1997; St. Amant, et al., 2005). More recently, other types of simulation models have also been developed to evaluate human-automation systems and to provide more holistic evaluations of human system performance (Bolton, et al., 2013; Pritchett and Christmann, 2011; Nehme, et al., 2008; Donmez, et al., 2010).

This effort focuses on two classes of models that are routinely used in other domains like logistics and industrial process modeling, but that are under-utilized in human-system integration settings. These are Discrete event simulation (DES) models and system dynamics (SD) models, which are simulation models that allow for cost and time effective evaluation of different design parameters for futuristic systems without conducting extensive human-in-the-loop experimentation, which is particularly critical in early conceptual design phases. This is especially true for human supervisory control systems that require human interaction with and supervision of a complex system with embedded automation. Holistic models such as discrete event simulations or system dynamics models can be used to not only understand the individual elements of complex automated systems that incorporate humans, but also understand the critical interactions that take place between the two.

The discrete event simulation (DES) approach is based on queuing-based constructs including events, arrival processes, service processes, and queuing policies to model the human operator as a serial processor of tasks (Law & Kelton, 2000). The input variables are primarily the distributions of the arrival rate of various operator tasks and the distributions of service times for these tasks. These distributions either must be estimated or drawn from previous experimental data. While their use in human performance modeling has been limited, DES models have been successfully applied to supervisory control domains such as air traffic control (Schmidt, 1978), as well as multiple unmanned vehicle control by a single operator (Donmez, Nehme, & Cummings, 2010; Nehme, Mekdeci, Crandall, & Cummings, 2008; Nehme, 2009).

System Dynamics (SD) draws inspiration from basic feedback control principles to create simulation models (Sterman, 2000). SD constructs (stocks, flows, causal loops, time delays, feedback interactions) enable investigators to describe and potentially predict complex system performance, which would otherwise be impossible through analytical methods (Forrester, 1961). SD models have been used in a number of large and small scale systems with social elements including management, economics, logistics, education, and disease spread (Sterman, 2000). More relevant to real-time human-automation collaborative scheduling and human decision making, a few SD models have been developed to represent human supervisors monitoring automated systems (White, 2003), a number of command and control applications (Coyle, Exelby, & Holt, 1999), and human decision-making in high-stress situation with interruptions (Rudolph & Repenning, 2002).

In order to illustrate how such models can be used to explore human-autonomous system design trade spaces, DES and SD models will be presented attempting to capture the complex interactions of a single operator controlling multiple autonomous vehicles. Each modeling approach will be explained, and then compared in a discussion of which method is the most appropriate for various applications that involve human decision in a supervisory control task.

Discrete Event Simulation

Discrete event simulation (DES) is a technique which models a system as a chronological sequence of events representing changes in system states (Law & Kelton, 2000). DES is particularly suited to modeling human-autonomous systems due to their event-driven nature, i.e., a human tells a ground robot where to search next once one goal has been achieved. Based on queuing theory, DES models represent the human as a single server serially attending the arrival of complex events (Carbonell, Ward, & Senders, 1968; Senders, 1964; Sheridan, 1969). Queues for the operator's attention build as events arrive that require on operator's attention, and if not attended to immediately, are either delayed or forgotten, with perhaps serious consequences for the autonomous system. DES models can represent operator parallel processing through the introduction of multiple servers (Liu, 1997; Liu, Feyen, & Tshimhoni, 2006), if appropriate.

In order to illustrate just how a DES model can be used to predict possible system performance, a DES model designed for single operator control of multiple unmanned vehicles will be discussed (Nehme, 2009). The test bed used for this DES model is called the Research Environment for Supervisory Control of Heterogeneous Unmanned Vehicles (RESCHU) simulator, discussed in detail in the next section.

The RESCHU Test Bed

RESCHU allows operators the ability to control a team of Autonomous Vehicles (AVs) composed of unmanned air and underwater vehicles (UAVs and UUVs). The primary mission for all vehicles is surveillance with the ultimate goal of locating specific objects of interest in urban coastal and inland settings. There exists a single UUV type, and two UAV types, with one providing high level sensor coverage (a High Altitude Long Endurance UAV), while the other provides more low-level target surveillance and video gathering (a Medium Altitude Long Endurance UAV). The primary assumptions in RESCHU are that the AVs are capable of maintaining stable flight (UAV) or sustained forward motion (UUV), and executing all navigation and flight control actions with goal-based discrete input by the operator

The RESCHU interface consists of five major sections (Figure 1). The map (Section A) displays the locations of vehicles, threat areas, and areas of interests (AOIs). Vehicle control is carried out on the map, where operators can change vehicle paths or assign a new AOI to an AV. Significant events in the mission (i.e., vehicles arriving to goals, or automatic assignment to new targets) are displayed in the message box, along with a timestamp (Section C). When the vehicles reach an AOI, a simulated video

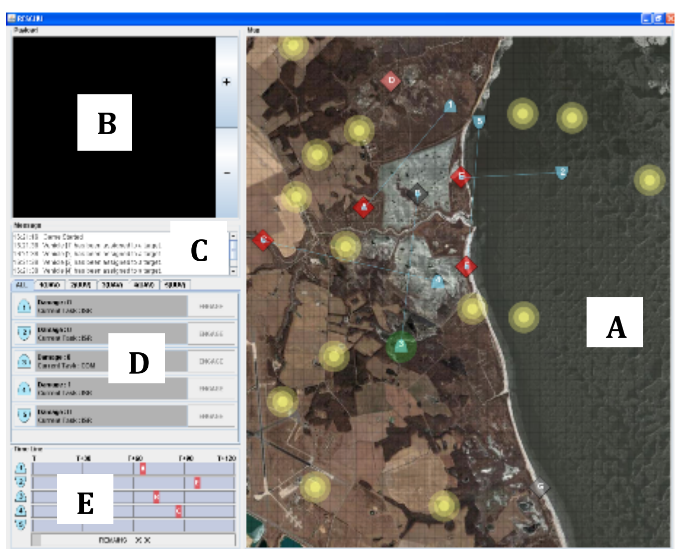

Figure 1. RESCHU interface (A: map, B: camera window, C: message box, D: control panel, E: timeline)

feed is displayed in the camera window, where participants visually identify a target. Example targets and objects of interest included cars, swimming pools, helipads, etc. The control panel (Section D) provides AV health information, as well as information on the AV's mission. The timeline (Section E) displays the estimated time of arrival to waypoints and AOIs.

When the AVs complete their assigned tasking, an automated path planner automatically assigns the HALE UAV to an AOI that needs intelligence, and the MALE UAVs and UUVs to AOIs with pre-determined targets. The automatic assignments AOIs are made with a greedy algorithm and thus are not necessarily the optimal choice. The operator can change the assigned AOI, and must reroute the AVs to avoid threat areas by changing an AV's goal or adding a waypoint to the path of the AV in order to go around the threat area.

When an AV arrives at an AOI, a visual flashing alert indicates that the operator can engage the camera. The operator then completes a search task by panning and zooming the camera until the specified target is located. Once complete, the AV is automatically re-assigned to a new AOI. Participants maximize their score by 1) avoiding threat areas that dynamically change, 2) completing search tasks correctly, 3) taking advantage of re-planning when possible to minimize vehicle travel times between AOIs, and 4) ensuring an AV is always assigned to an AOI whenever possible.

The DES Model of the RESCHU Environment

The DES model was specifically designed to evaluate the impact of changing vehicle team structure, (such as numbers and types of vehicles), as well as the impact of operator attention allocation strategies on overall system performance. This model can output a prediction of how many AVs of different types a single operator can control, but it can also predict how much autonomy a team of vehicles will need if a certain level of operator workload is desired, as well as the impact of different operator control strategies and vehicle collaboration strategies.

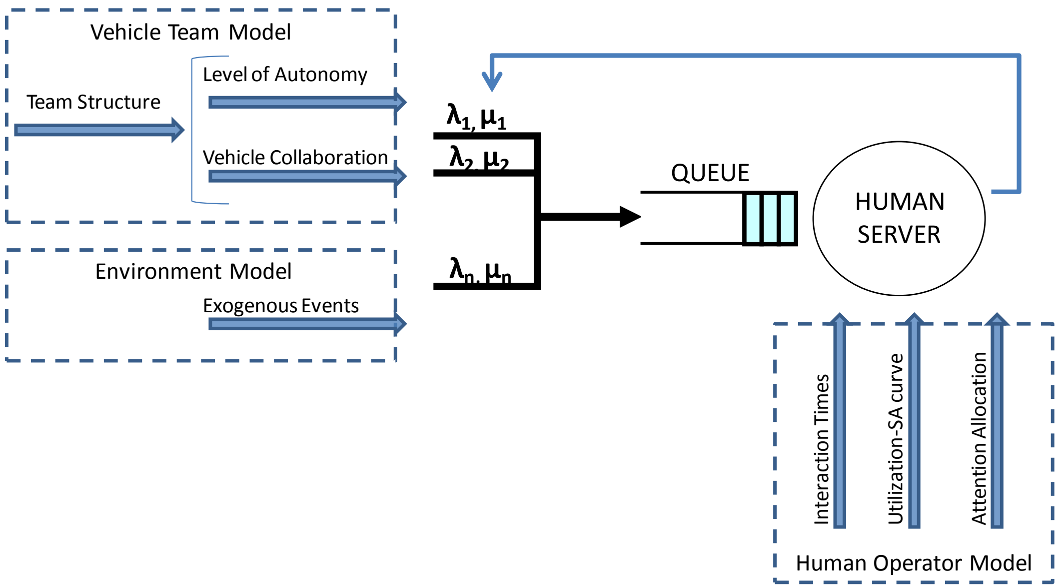

Figure 2. A DES model of a single operator controlling multiple unmanned vehicles

There are three components of the DES model in Figure 2 that can be generalized to human-autonomous system interaction: 1) the vehicle-team model, 2) the environment model, and 3) the human-operator model. For the vehicle-team model in Figure 2, team structure means the number and types of AVs under control (which could be dissimilar, i.e., meaning that the system could include a UAV and a UUV). The level of vehicle autonomy is captured through a concept known as Neglect Time (NT) (Olsen & Goodrich, 2003), since it represents the time a vehicle can operate without human intervention. Thus the more autonomous a vehicle, the longer it can be neglected. For example, Mars Rovers are much more autonomous than present-day AVs since the Rovers can operate for days without human intervention, while operational UGVs need human input for constant operation. Lastly, for the vehicle-team model, the degree of autonomy between vehicles is represented by the vehicle collaboration input. This represents how much vehicles coordinate among themselves for task completion, which increases as teams of AVs become more decentralized (Cummings, How, Whitten, & Toupet 2012).

When discrete events occur, they generate tasks that can cause a queue for the operator's attention. There are generally two types of tasks that are generated in this system. The first type is endogenous events, which are created internal to the system. Endogenous events can be either vehicle-generated, (e.g., a vehicle requires a new goal upon arriving at the previously-directed goal), or operator induced (e.g., an operator may think a new path is better than the one planned by the automation and thus changes it). The second type is exogenous events, which are environmentally driven (and thus labeled as the environment model in Figure 2). They may be unexpected, such as an emergent threat area which requires the operator to re-planning AV paths. The interarrival times for these events are represented by $\lambda_{i's}$.

The last major component of the model in Figure 2 is the Human Operator Model. Based on a single server queue with multiple input streams, the operator can attend to only one complex event at a time, generating the service time μ_i, which could result in a queue of tasks. The length of time an operator interacts with the system is captured through a probability distribution of service times, $\mu_{i's}$, thus capturing variability of performance both between and within different operators.

Interaction Time (IT) in Figure 2 represents how an operator interacts with a single event/AV. How and when operators elect to attend to the AVs as a group is termed attention allocation (Crandall & Cummings, 2007). In the model in Figure 2, attention allocation is represented by two variables. This first is management strategy, which can vary between micro (operators feel they must constantly direct and correct the automation), and macro (operators are comfortable letting the automation do most of the work and only intervene when the automation requests intervention.) This variable can act as a trust proxy.

The second attention allocation variable is the order in which the different vehicles are serviced. When multiple vehicles require operator attention simultaneously, the operator must select the next vehicle to be serviced. Typical queuing strategies such as First In First Out (FIFO) or preemptive priority queuing (highest priority always gets moved to the head of the queue) can be used to represent operator selection strategies (Pinedo, 2002).

While the service time and queuing strategy representations are not new in DES and queuing theory, the novel contribution of the Human Operator Model in Figure 2 is the relationship between utilization and situation awareness. Situational awareness (SA) is defined as the combination of perception of elements in the environment, the comprehension of their meaning, and the projection of their status in the future as relevant to the management process (Endsley, 1995). Utilization is defined as percent busy time, or the time an operator is actively engaged in some kind of task. When operators are very busy with high levels of utilization, they could be too busy to accumulate the information required to build SA, which has been shown to negatively affect performance (Cummings, Clare, & Hart, 2010). Conversely, when under-utilized, operators can become less alert or attentive, resulting in low SA and degraded performance (Cummings, Mastracchio, Thornburg, & Mkrtchyan, 2013).

In the model in Figure 2, the effect of both high and low utilization and reduced situational awareness is the creation of additional wait times, which increases the time it takes the operator to notice the needs of the system. This penalty, which is added to the service time, is expressed through a concave upwards parabolic function inspired by the Yerkes Dodson Law (Yerkes & Dodson, 1908). Previous research has shown that using such a relationship can substantially improve model performance as compared to a more traditional DES model that does not explicitly account for the impact of operator workload on performance (Donmez et al., 2010).

DES as a Design Tool

As previously discussed, a potential strength of a DES model is in the conceptual design stage of an autonomous vehicle network, specifically in terms of the generation of requirements and initial design stages. Designers and engineers of unmanned vehicle teams in the future will face a myriad of choices including various vehicle types and capabilities, or even the same vehicle but with many different capabilities due to modular sensor design. These different design choices should also consider the impact on the ability of operators to supervise the teams effectively.

Therefore, in building effective AV systems, the collective impact of different design choices must be taken into consideration to ensure effectiveness of operator supervision.

Given the limitations of human-in-the-loop experimentation for guiding design choices in futuristic heterogeneous unmanned and autonomous systems, a DES model like the one depicted in Figure 2 can be used to provide rapid prototyping evaluation capabilities. The model is able to describe the behavior of the system in question, given a specific set of input conditions, as well as predict the effects of changes in one or more system design variables on the behavior of the system.

Figure 3 is an example of a trade space exploration that was generated by the model in Figure 2 using data to form the distributions from 80 participants who conducted missions using RESCHU (Figure 1) . Consider the case of an engineer who wishes to design a system where a single operator must analyze imagery from multiple UAVs under his/her supervision in a search mission where possible targets of interest could be hostile. The number of UAVS, service times for image search tasks, and reactions times to emergent events to represent an operator's workload/situation awareness can be used as inputs to a simulation model like that previously described, with the model in Figure 2 generating predicted average search task wait times with the number of objects correctly identified (the key performance parameter). In order to generate a Pareto front, the four curves in Figure 3 represent four different design cases with the number of UAVS under control varied from 1-6. For the base curve, the switching strategy was set to operators prioritizing vehicle path planning over an imagery search task, with set times for the image search times (labeled Service Times in Figure 3) and the time it takes for an operator to recognize an event needs servicing (labeled Reaction Times in Figure 3.)

Each of the other three curves in Figure 3 varies just one of the base-case variables. So the Switching Strategy curve represents a shift in operator priorities such that image search was given priority over vehicle replanning, and the other two variables were held constant. Similarly, for the Service Times curve, all service times were improved by 5s and for the Reaction Times curve, all event recognition times improved by 5s (and all other variables held the same as in the base case.)

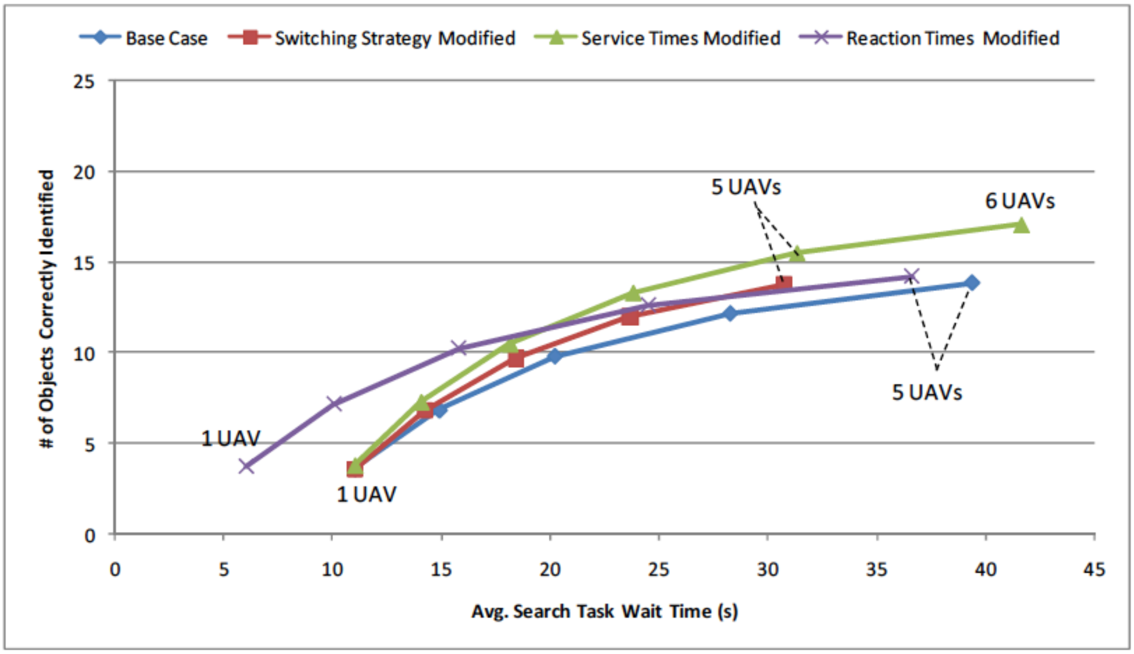

Figure 3: DES-Generated Trade Space for Multiple UAV Network Design

Figure 3 illustrates that changing the switching strategy, reducing service times, and reducing reaction times all lead to a shift in the tradeoff curve, such that the curves all improve upon the base case. The best case depends on the desired outcomes of the designer, and is potentially a subjective choice. If a designer wants the most objects identified, and there is not a significant penalty in terms of time pressure, cost of operation, or safety then 6 UAVs with a decision support system that can speed up target detection would be the best design choice. However, if only 3 vehicles were available and wait times should be minimal, then a system with improved reactions times in terms of emergent events would be the best case.

So as seen in Figure 3, the use of a DES can be very helpful in early design stages but as with all modeling efforts, there are limitations. This DES modeling approach is not suitable for modeling AV systems at a lower cognitive level where parallel processing is of interest. This is because a primary assumption is that the operator is serially attending to complex tasks, in keeping with queuing theory. This kind of DES modeling is also not appropriate for modeling systems that entail a low task loading, because the model in Figure 2 assumes at least 40% utilization (Donmez et al., 2010), so systems with long latencies between human interactions would likely not be well represented. This also relates to another major limitation that these models assume some level of observable human-computer interaction. It remains an open questions how to use a DES approach to model systems that require extensive monitoring with little action, such as multiple screen image monitoring.

System Dynamic Modeling

There are a number of reasons that SD models are particularly appealing for modeling AV systems. The first is the ability of SD models to capture non-linear processes (Sweetser, 1999). Since human performance does not generally adhere to linear models (Gao & Lee, 2006), using non-linear behavioral and performance representations will be critical for the external validity of the model. Second, SD models can include both qualitative and quantitative data (Sterman, 2000), and often qualitative data is essential to human performance modeling (Hancock & Szalma, 2004). Third, SD models are effective at capturing the impact of latencies and feedback interactions on the system, which is essential for modeling a human operator and the impact of delays in perception of system performance on operator behavior and trust.

The SD modeling process can be broken down into five major phases (Sterman, 2000). First, in the problem articulation stage, the overall problem that the model is attempting to represent is identified, along with key variables to be captured within the boundary of the model. In the second stage, a "dynamic hypothesis" is developed. A dynamic hypothesis is defined as a theory that explains the behavior of the system as an endogenous consequence of the feedback structure of the holistic system (Sterman, 2000). It is a working hypothesis that guides the modeling effort and is continuously tested and refined throughout the model building and testing process.

In the third stage, the dynamic hypothesis is mapped into causal loops and stocks and flows in order to formulate the simulation model and estimate exogenous parameters. The SD community defines endogenous variables simply as those variables which are calculated within the model, while exogenous variables are assumed parameters which lie outside of the model boundary (Sterman, 2000). The fourth stage, testing the model, includes comparison of model outputs to experimental data sets, robustness under extreme conditions, and sensitivity analyses. The fifth stage, policy design and evaluation, includes evaluating the ability of the model to predict performance under new circumstances, which is why this method is useful in the conceptual design phases of the systems engineering process.

The CHAS SD Model

Using the SD modeling phases described above, for the case of a single operator supervising multiple autonomous vehicles, a Collaborative Human-Automation Scheduling (CHAS) model was developed to capture non-linear human behavior and performance patterns, latencies and feedback interactions in an autonomous system, as well as qualitative variables such as human trust in automation (Clare, 2013). The CHAS model was developed to aid a designer of future AV systems by simulating the impact of changes in system design and operator training on human and system performance. This can reduce the need for time-consuming human-in-the-loop testing that is typically required to evaluate such changes.

A collaborative, multiple AV simulation environment was used to develop the CHAS model, called Onboard Planning System for UVs Supporting Expeditionary Reconnaissance and Surveillance (OPS-USERS). This simulation environment leverages decentralized algorithms for vehicle routing and task allocation (Figure 4). Operators controlled multiple, heterogeneous AVs for the purpose of searching an area of interest for new targets, tracking these targets, and approving weapons launch if a target was deemed hostile. The operators were able to assign tasks to the vehicles as a group, and schedules were presented to the operator for approval at various intervals (called the replanning interval). More details about this test bed can be found elsewhere (Clare, 2010), but the primary difference between this simulation and RESCHU (Figure 1) is the degree of autonomy between the AVs. In RESCHU, AVs shared very little information but in the OPS-USERS paradigm, AVs were in constant communication, sharing significant packets of information. As a result, the operator in OPS-USERS acts as a coach, calling plays that the vehicles independently determine how to execute, but in RESCHU, the operator must command each and every vehicle.

A simplified CHAS model of the OPS-USERS environment is shown in Figure 5, which depicts the three major feedback loops: the Trust in Automation loop, the Expectations Adjustment loop, and the Cognitive Overload loop. These three loops consist of separate causal pathways. The full diagram of the CHAS model is presented in Figure 6, showing the three main feedback loops (Clare, 2013).

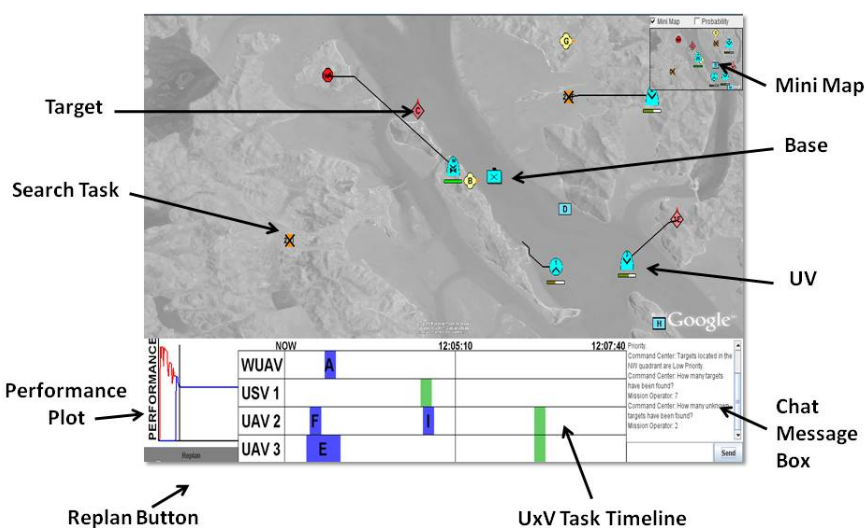

Figure 4. The OPS-USERS Test Bed

The Trust in Automation loop, shown in the red dashed line box in Figure 5 draws from the "perception, cognition, action" loop in the human information processing model developed by Wickens and Hollands (2000). The loop represents how the operator's perception of the performance of the system impacts his or her trust in the automation and thus influences the operator's decisions to intervene in the operations of the semi-autonomous vehicles. As shown in Figure 5, the operator has a time-delayed perception of how the system is performing. The operator's trust during the mission begins at an initial level and then adjusts based on the difference between the operator's expectations of how well the system should be performing and the operator's perception of system performance.

It is likely that the operator's trust has some inertia (Lee & Moray, 1994) and thus adjusts with a time delay. As the operator loses (or gains) trust in the AS, the operator will choose to intervene more (or less) frequently, for example by creating new tasks for the AVs or requesting a new schedule for the AVs. Similar results have been found in other work (Bailey & Scerbo, 2007). This decision to intervene has an impact on the operations of the team of semi-autonomous AVs,

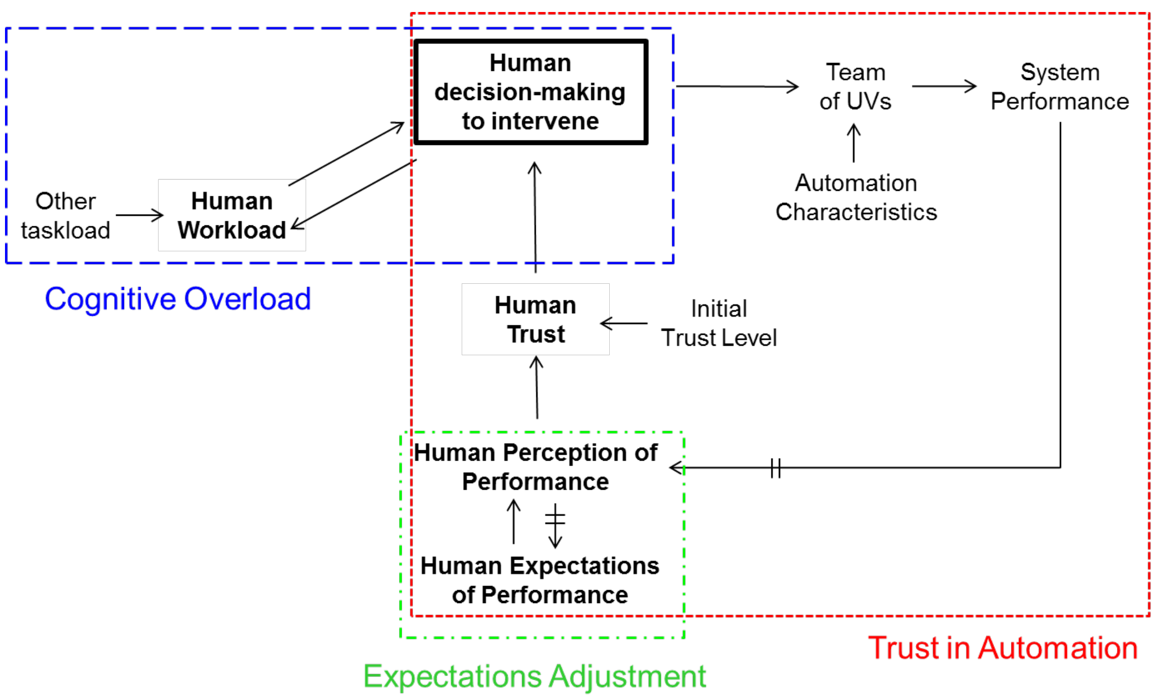

Figure 5: Simplified Collaborative Human-Automation Scheduling System Dynamics Model

which influences the performance of the system, completing the feedback loop.

Second, the Expectations Adjustment loop, shown in the green dashed line box, represents how the operator's expectations of performance can change throughout the mission. The operator's initial expectations of performance are likely set by training, previous experience, or by instructions from a supervisor. However, as the operator perceives how the system is actually performing, there is likely a time-delayed adjustment of the operator's expectations to conform to his or her perceived reality of how well the system is doing.

Third, the Cognitive Overload loop, shown in the blue dashed line box, represents the impact that excessive cognitive workload can have on system performance. The System Dynamics modeling community typically separates the positive and negative effects of a variable into distinct loops (Sterman, 2000). The Trust in Automation loop, as previously described, captures the positive effects of increasing workload, assuming that an increasing rate of interventions leads to higher performance. It should be noted that this assumption does not hold for all systems, as previous studies have shown that frequent human intervention can potentially have a negative impact on automation (Beck, Dzindolet, & Pierce, 2005; Parasuraman & Riley, 1997), as some decentralized algorithms may need time to stabilize (Walker et al., 2012). The automation in the multiple UAV control test bed used to validate CHAS has been found to be provably good, but suboptimal (Choi, Brunet, & How, 2009), and previous experiments have shown that a moderate rate of intervention results in higher performance than a low frequency of intervention (Clare, Maere, & Cummings, 2012; Cummings et al., 2010).

The CHAS model captures the fact that the frequency with which the operator decides to intervene in the system also has an impact on human cognitive workload. The Cognitive Overload loop only captures the negative effects of *high* workload, and thus is dormant when the operator has low or moderate workload, having little effect on the model. Human workload is also driven by task load, i.e. the level of tasking that an operator is asked to perform by the system (Clare & Cummings, 2011). The feedback loop is completed by modeling the potential for cognitive overload, where high levels of human workload can decrease the effectiveness of the operator's interventions in the system, thus decreasing system performance (Cummings & Guerlain, 2007; Rouse, 1983;

Schmidt, 1978). This model is specifically designed to simulate moderate to high task load missions but one limitation, as in the DES model, is the lack of investigation in low task load missions.

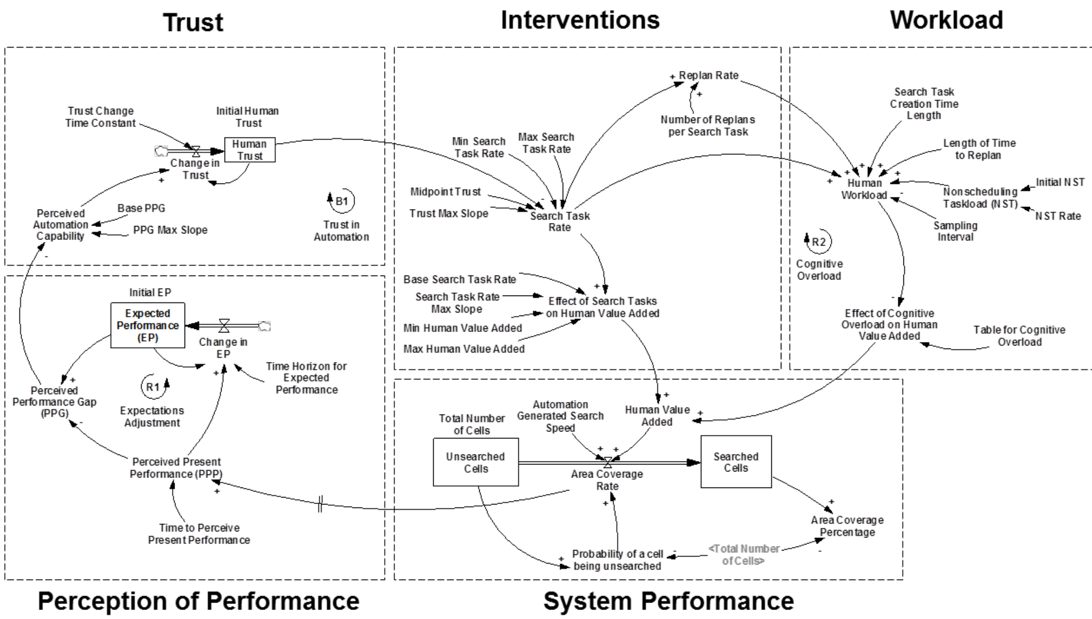

Figure 6: Fully Developed CHAS Model with Major Core Components Labeled.

The CHAS model in Figures 5 and 6 has other limiting assumptions. First, the CHAS model is a computational model of human behavior and decision-making and thus makes a number of simplifications and assumptions about the complexities of human perception, cognition, emotions, and decision-making. Second, the CHAS model requires sufficient data to validate and tune the causal relationships throughout the model. Thus, the model is limited to assisting designers who aim to make evolutionary changes to existing systems. Third, the CHAS model in its current form is limited to simulating single operator, moderate to high task load, multiple UAV control missions. The assumption is that the vehicles are semi-autonomous and the human operator only guides the high-level goals of the vehicles, as opposed to guiding each individual vehicle. In addition, the CHAS model assumes that the operator is physically removed from AV environment, thus there is no direct human-robot interaction. Finally, the CHAS model does not consider vehicle failures, communications delays, and safety policies.

SD Modeling as a Design Tool

Like the DES approach in Figure 2, the CHAS model enables tradespace exploration within a much larger design space than would be possible using real systems. Variables that can either be held constant or manipulated include the human operator's abilities, the number of AVs or the level of automation of the AVs. For example, a system designer may want to explore different skills sets for human operators. As an illustration, recently the U.S. Air Force started a new program that trains UAV operators with no piloting experience (Clark, 2012). It is not clear how new populations such as these will be affected in terms of workload, and whether new tools or advanced automation is needed to offset potential high workload conditions. It is possible that some operators can sustain performance up to 80%, 90%, or even close to 100% utilization, while others begin to experience cognitive overload as low as 50% utilization. For the purposes of this discussion, the utilization level where cognitive overload begins will be called the cognitive overload onset point.

Thus, the CHAS model can aid a system designer in choosing the most appropriate level of intervention given the different cognitive overload onset points of various operators. To

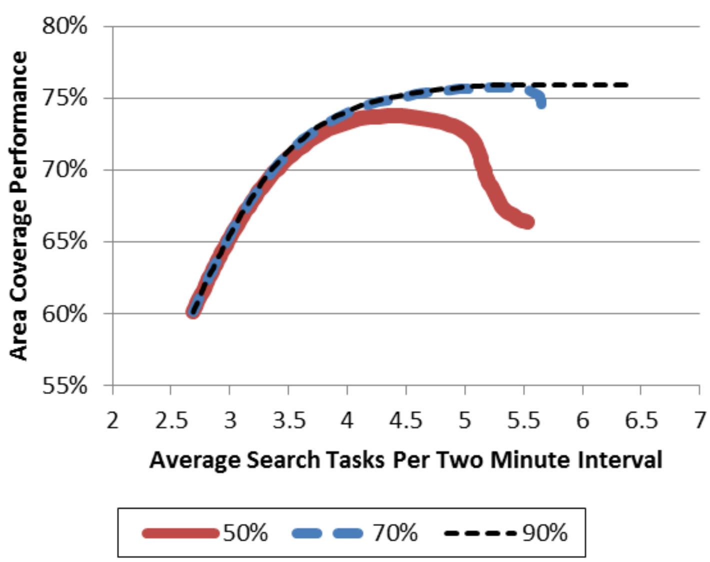

Figure 7: Comparing Increasing Cognitive Loads on Overall Performance

demonstrate this, for a single operator-multiple autonomous vehicle (AV) control and supervision task similar to that discussed in the previous section, the CHAS model was run with three different levels of anticipated workload as defined as utilization or percent busy time, 50%, 70% and 90%. As shown in Figure 7 where the x axis represents increasing task load as dictated by a group of searching AVs generating search tasks every two minutes, and the performance metric as the percentage of overall area searched, operators who can sustain performance at utilization levels up to 90% can perhaps be prompted to intervene up to 6 times per two minute interval to maximize system performance. In contrast, operators who begin to experience cognitive overload onset at 50% utilization should only be prompted to intervene 4.5 times per two minute interval.

It is also notable that there is not a large performance improvement when moving from 4.5 to 6 search tasks per two minute interval. This indicates that the system may be robust to varying skill levels and cognitive overload onset points, a key advantage of a goal-based architecture for real-time human-automation collaborative scheduling. Thus the use of CHAS can help designers determine such points of robustness so they can determine where the best use of resources likely is. It is important to note that because of the complexity and interactions in these systems, there is no known closed-form solution approach that could provide such boundary estimations so the use of simulation to do so is a critical tool for system design exploration.

Model Comparison

Two different simulation approaches, discrete event simulation and system dynamics modeling, were presented in the context of similar multiple AV control environments. Both modeling approaches were designed to support the designers of future AV systems by simulating the impact of alternate designs on vehicle, operator, and system performance. However, there is an ongoing argument in the modeling community about the scenarios for which SD is more appropriate than DES (i.e. Özgün & Barlas, 2009; Sweetser, 1999; Tako & Robinson, 2008). Given these two different modeling approaches, what can be said about which is better or more appropriate in attempting to model human-complex system interaction in various situations?

In order to compare these two models, the DES model was adapted to those parameters of the OPS-USERS test bed (Figure 4) using the data from an actual experiment with the OPS-USERS test bed. In this experiment, 31 operators performed two 10-minute long simulated missions (A.S. Clare & Cummings, 2011). Operators were prompted to view automation-generated schedules at prescribed intervals of either 30 or 45 seconds. The two different frequencies of schedule presentation served to modulate the task load of the operator, such that 30s replan intervals should induce higher workload than the 45s intervals. All operators experienced both replan intervals in a counterbalanced and randomized order.

The average observed utilization of operators in the 30s and 45s replan prompting interval conditions is shown in Figure 8. The CHAS and DES models were then applied to simulate this experiment, and the utilization results from these simulations are compared to the experimental results, also in Figure 8. The DES model used to replicate the results of this experiment (called the Multiple Unmanned Vehicle DES, or MUV-DES) was more accurate for the 30s replan prompting interval condition as compared to the CHAS model, which had an average utilization prediction that was higher than the experimental data. For the 45s replan prompting interval data, both models were slightly off, although both predictions fell within the 95% confidence interval of the data. Both models captured the decrease in utilization from the 30 to 45 second replan prompting interval conditions.

While both models replicate average utilization values for the experimental data, a qualitative comparison of the features of MUV-DES and the CHAS models is more revealing. The DES model does not consider the interaction between human-AV system variables (i.e., the variable inputs to the model), which can result in lower predictive accuracy. In addition, the DES model does not explicitly consider the impact of trust on operator or system performance (Nehme, 2009).

However, SD models do not represent variability in the data using the same probabilistic distributions that DES models do so if the boundaries of system performance are of concern, SD models fall short.

The CHAS model was built upon the DES model to capture the feedback interactions among perception, workload, trust, decisions to intervene, and performance. It also explicitly represents qualitative variables such as human trust and its impact on the rate at which humans intervene into the operations of the team of AVs, and thus on system performance. The dynamics of trust are captured by enabling trust to adjust over time throughout the mission with some inertia. This is simply not possible in a DES formulation. One important caveat to the SD approach is that each model is fine-tuned to represent a specific system under specific conditions, so the level of abstraction is low with difficulty in generalizing the results beyond the specific system. Moreover, it is difficult in terms of SD model building and validation when even small system changes are made.

Another important difference is that CHAS model can provide predictions of continuous measures, a key attribute of all SD models. While the DES model provided predictions of system performance based on the occurrence of events (such as the arrival of targets), this form of

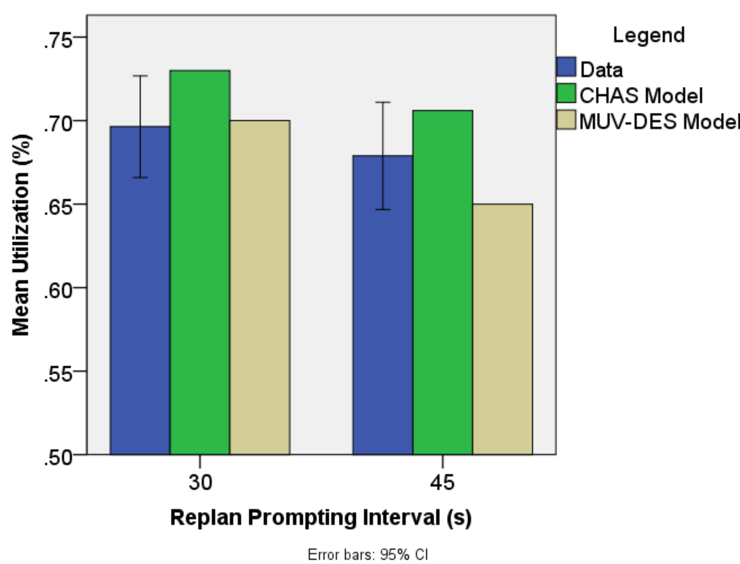

Figure 8: Average observed utilization for single operator-multiple UAV high task load experiment, with CHAS and DES predictions and 95% confidence intervals.

performance prediction is not as useful for a continuous performance metric such as area coverage.

Both the DES and the CHAS models have specific domains for which they are most appropriate. The DES model is likely best suited for using probabilistic distributions to accurately model an operator who is a serial processor of discrete tasks, such as visually identifying targets. The CHAS model is better suited for modeling continuous performance feedback that is temporally dependent and is affected by qualitative variables such as trust. These comparisons are summarized in Table 1. Ultimately the best model is the one that enables diagnosticity, in that it allows a system designer to more precisely characterize the reasons behind behavior and performance patterns, but in a manner that reduces the reliance on assumptions and estimations that may not be well characterized.

Table 1: A Comparison of Discrete Event Simulation and System Dynamics Models

	Discrete Event Simulation Model	**System Dynamics Model**
Model classification	Stochastic	Deterministic
Includes qualitative variables	No	Yes
Second order and higher interactions represented	No	Yes
Models can be easily changed to represent new system parameters	Yes	No
Level of abstraction	High	Low
Model outputs	Better for event-related measures	Better for continuous measures

Conclusion

With the move towards increasingly automated systems, the development of models that can both replicate observed human-system behavior and predict likely outcomes given various changes in automation architectures and human tasking will also become more critical. To this end, this paper has discussed the application of two modeling techniques, discrete event simulation and system dynamics models, used predominantly in other settings, that can be extended to represent various facets of supervisory control of autonomous systems. Ultimately such models enable cost and time effective evaluations without conducting extensive experimentation, critical for the rapidly evolving world of autonomous systems.

One area of research that is lacking is if and how these approaches could be combined for autonomous systems modeling. SD models have been used to inform inputs for DES models and vice versa (Chahal & Eldabi, 2008; Venkateswaran & Son, 2005) , but the only known integration of the two models into a single model occurred in the domain of breast cancer screening where the attempt was to determine the impact of various screening policies (Tejada, 2012). How to combine these two modeling techniques for autonomous system supervisory control where the system has potentially emergent behavior that could change significantly in response to system uncertainties is a research area left unexplored.

A clear strength of the system dynamics approach is the ability to use qualitative variables, while the discrete event simulation approach provides flexibility in exploring various system architectures with inherent uncertainty not afforded by the systems dynamics approach. Given the predominant role that human trust (a qualitative variable) plays in autonomous system operation, such a hybrid model would be a very powerful design trade space exploration tool, especially since autonomous systems embody significant stochastic reasoning.

References

Anderson, J.R. (1993). Rules of the Mind. Hillsdale, NJ: Lawrence Erlbaum.

Anderson, J. R., Bothell, D., Byrne, N., Douglass, S., Lebiere, C. and Qin,Y. (2004). An integrated theory of the mind. *Psychological Review*, 111(4), 1036-1060.

Bailey, N. R., & Scerbo, M. W. (2007). Automation-induced complacency for monitoring highly reliable systems: the role of task complexity, system experience, and operator trust. *Theoretical Issues in Ergonomics Science, 8*(4), 321-348.

Beck, H. P., Dzindolet, M. T., & Pierce, L. G. (2005). *Take the Advice of a Decision Aid: I'd Rather be Wrong!* Paper presented at the Human Factors and Ergonomics Society Annual Meeting, Orlando, FL. .

Carbonell, J. R., Ward, J. L., & Senders, J. W. (1968). A queuing model of visual sampling experimental validation. *IEEE Transactions on Man Machine Systems, 9*(3), 82-87.

Chahal, K., & Eldabi, T. (2008). *Applicability of Hybrid Simulation to Different Modes of Governance in UH Healthcare.* Paper presented at the 2008 Winter Simulation Conference.

Choi, H. L., Brunet, L., & How, J. P. (2009). Consensus-Based Decentralized Auctions for Robust Task Allocation *IEEE Transactions on Robotics, 25*(4), 912 - 926.

Clare, A. S. (2010). *Dynamic Human-Computer Collaboration in Real-time Unmanned Vehicle Scheduling.* Cambridge, MA.

Clare, A. S. (2013). *Modeling Real-Time Human-Automation Collaborative Scheduling of Unmanned Vehicles.* (PhD), Massachusetts Institute of Technology, Cambridge, MA.

Clare, A. S., & Cummings, M. L. (2011). *Task-Based Interfaces for Decentralized Multiple Unmanned Vehicle Control.* Paper presented at the AUVSI Unmanned Systems North America, Washington DC.

Clare, A. S., Maere, P. C. P., & Cummings, M. L. (2012). Assessing Operator Strategies for Real-time Replanning of Multiple Unmanned Vehicles. *Intelligent Decision Technologies, 6*(3), 221-231.

Clark, L. (2012). 18X Pilots Learn RPAs First. Retrieved April 7, 2013, from http://www.holloman.af.mil/news/story.asp?id=123289389

Coyle, J. M., Exelby, D., & Holt, J. (1999). System Dynamics in Defence Analysis: Some Case Studies. *Journal of the Operational Research Society, 50*, 372-382.

Crandall, J. W., & Cummings, M. L. (2007). Identifying predictive metrics for supervisory control of multiple robots. *IEEE Transactions on Robotics - Special Issue on Human-Robot Interaction, 23*(5), 942-951.

Cummings, M. L., Clare, A., & Hart, C. (2010). The Role of Human-Automation Consensus in Multiple Unmanned Vehicle Scheduling. *Human Factors, 52*(1), 17-27.

Cummings, M. L., & Guerlain, S. (2007). Developing Operator Capacity Estimates for Supervisory Control of Autonomous Vehicles. *Human Factors, 49*(1), 1-15.

Cummings, M. L., How, J., Whitten, A., & Toupet , O. (2012). The Impact of Human-Automation Collaboration in Decentralized Multiple Unmanned Vehicle Control. *Proceedings of the IEEE, 100*(3), 660-671.

Cummings, M. L., Mastracchio, C., Thornburg, K. M., & Mkrtchyan, A. (2013). Boredom and Distraction in Multiple Unmanned Vehicle Supervisory Control. *Interacting with Computers, 25*(1), 34-47.

Donmez, B., Nehme, C., & Cummings, M. L. (2010). Modeling Workload Impact in Multiple Unmanned Vehicle Supervisory Control *IEEE Systems, Man, and Cybernetics, Part A Systems and Humans, 99*(1-11).

Endsley, M. R. (1995). Toward a theory of situation awareness in dynamic systems. *Human Factors, 37*(1), 32-64.

Forrester, J. W. (1961). *Industrial Dynamics*. Cambridge, MA: MIT Press.

Gao, J., & Lee, J. D. (2006). Extending the Decision Field Theory to Model Operators' Reliance on Automation in Supervisory Control Situations *IEEE Transactions on Systems, Man and Cybernetics - Part A: Systems and Humans, 36*(5), 943-959.

Hancock, P. A., & Szalma, J. L. (2004). On the relevance of qualitative methods for ergonomics. *Theoretical Issues in Ergonomics Science, 5*(6), 499-506.

Law, A. M., & Kelton, W. D. (2000). *Simulation modeling and analysis* (3rd ed.). Boston: McGraw-Hill International Series.

Lee, J. D., & Moray, N. (1994). Trust, Self-Confidence, and Operators' Adaptation to Automation. *International Journal of Human-Computer Studies, 40*(1), 153-184.

Liu, Y. (1997). Queuing network modeling of human performance of concurrent spatial and verbal tasks. *IEEE Transactions on Systems, Man, and Cybernetics, 27*(2), 195-207.

Liu, Y., Feyen, R., & Tshimhoni, O. (2006). Queuing network-model human processor (QN-MHP): a computational architecture for multitask performance in human-machine systems. *ACM Transactions on Computer-Human Interaction, 13*(1), 37-70.

Nehme, C., Mekdeci, B., Crandall, J. W., & Cummings, M. L. (2008). The Impact of Heterogeneity on Operator Performance in Future Unmanned Vehicle Systems. *The International C2 Journal, 2*(2).

Nehme, C. E. (2009). *Modeling human supervisory control in heterogeneous unmanned vehicle systems.* (Doctor of Philosophy), Massachusetts Institute of Technology, Cambridge, MA.

Olsen, D. R., & Goodrich, M. A. (2003). *Metrics for Evaluating Human-Robot Interactions*. Paper presented at the Performance Metrics for Intelligent Systems, Gaithersburg, MD.

Özgün, O., & Barlas, Y. (2009). *Discrete vs. Continuous Simulation: When Does It Matter?* Paper presented at the 27th International Conference of The System Dynamics Society, Albuquerque, NM.

Parasuraman, R., & Riley, V. (1997). Humans and Automation: Use, Misuse, Disuse, Abuse. *Human Factors, 39*(2), 230-253.

Pinedo, M. (2002). *Scheduling: Theory, Algorithms, and Systems* (2nd ed.). Englewood Cliffs, NJ: Prentice Hall.

Rouse, W. B. (1983). *Systems Engineering Models of Human-Machine Interaction*. New York: North Holland.

Rudolph, J. W., & Repenning, N. P. (2002). Disaster Dynamics: Understanding the Role of Quantity in Organizational Collapse. *Administrative Science Quarterly, 47*(1), 1-30. doi: 10.2307/3094889

Schmidt, D. K. (1978). A queuing analysis of the air traffic controller's workload. *IEEE Transactions on Systems, Man, and Cybernetics, 8*(6), 492-498.

Senders, J. W. (1964). The human operator as a monitor and controller of multidegree of freedom systems. *IEEE Transactions on Human Factors in Electronics, 5*(1), 2-5.
Sheridan, T. B. (1969). On how often the supervisor should sample. *IEEE Transactions on Systems, Science, and Cybernetics, 6*(2), 140-145.
Sterman, J. D. (2000). *Business Dynamics: Systems Thinking and Modeling for a Complex World*. New York: Irwin/McGraw-Hill.
Sweetser, A. (1999). *A Comparison of System Dynamics (SD) and Discrete Event Simulation (DES)*. Paper presented at the 17th International Conference of the System Dynamics Society and 5th Australian & New Zealand Systems Conference, Wellington, New Zealand.
Tako, A. A., & Robinson, S. (2008). Comparing Discrete-Event Simulation and System Dynamics: Users Perceptions. *Journal of the Operational Research Society*, 1-17.
Tejada, J. J. (2012). *An Integrated Discrete-Event/System Dynamics Simulation Model of Breast Cancer Screening for Older US Women.* North Carolina State University, Raleigh.
Venkateswaran, J., & Son, Y. J. (2005). Hybrid System Dynamic—Discrete Event Simulation-Based Architecture for Hierarchical Production Planning. *International Journal of Production Research,, 43*(20), 4397-4429.
Walker, P., Nunnally, S., Lewis, M., Kolling, A., Chakraborty, N., & Sycara, K. (2012). *Neglect Benevolence in Human Control of Swarms in the Presence of Latency*. Paper presented at the IEEE International Conference on Systems, Man, and Cybernetics (SMC), Seoul, Korea.
White, A. S. (2003). *A Qualitative Systems Dynamic Model of Human Monitoring of Automated Systems.* Paper presented at the 21st International Conference of the Systems Dynamics Society, New York, NY.
Wickens, C. D., & Hollands, J. G. (2000). *Engineering Psychology and Human Performance* (3rd ed.). Upper Saddle River, N.J.: Prentice Hall.
Yerkes, R. M., & Dodson, J. D. (1908). The relation of strength of stimulus to rapidity of habit-formation. *Journal of Comparative Neurology and Psychology, 18*, 459-482.