Framework to Support Scenario Development for Human-Centered Alerting System Evaluation

Matthew L. Bolton, *Member, IEEE*, Sinan Göknur, and Ellen J. Bass, *Senior Member, IEEE*

***Abstract*—The purpose of the framework introduced here is to support the development of evaluation scenarios that are capable of assessing system level performance while considering the system, the humans that interact with it, and the environment. The following five step framework is presented and applied to a pilot self separation task: 1) identify entities critical to system design, development, and operation and define their goals and properties as they relate to the system being studied; 2) define a subset of functionality for evaluation (define an execution sequence); 3) map entity properties to the execution sequence to identify independent variables; 4) translate entity goals into a set of system goals that can be used to identify dependent measures; and 5) iterate through each step to ensure the models produced are internally consistent.**

***Index Terms*—Air traffic control, alarm systems, alerting systems, decision support systems, man machine systems.**

I. Introduction

THE performance evaluation of human–machine systems should consider the technological and ecological components of the system [41] as well as the human element (including variance in response behavior and limitations of human performance; see [19]). Because of the complexity involved and the many conditions under which an evaluation must take place, cohesive frameworks to conduct such systems evaluations are necessary [22], [34]. This study presents a framework to develop scenarios to evaluate alerting systems and applies it to the evaluation of a prototype cockpit alerting system for air traffic management.

A. Alerting Systems

The nature of the information conveyed by an alert can vary widely. They can display information for a single sensor, integrate information from multiple sources, or be used to analyze trends, and assess hazards using computationally intensive algorithms and large databases [4], [26]. Some alerting systems may also recommend particular user behavior (see, for example, [14]).

1) Alerting System Performance: There are a variety of system and ecological challenges associated with alerting systems including determining the appropriate sensor accuracy, selecting alerting threshold, defining hazard metrics, identifying alerting criteria, and building efficient algorithms [3]–[5], [20], [26]. There are also a variety of human factors issues that are associated with alerting systems given that system performance and integrity is often dependent on a human's ability to recognize and respond to alerts [8], [20], [23], [35]. The presence of an alerting system requires that the human to adapt his or her task to account for the information the alerting system provides [23]. This may, in turn, impact how the human operator seeks out information and utilizes cues [25], [37].

The alerting system display design also plays a key role. For example, Sarter and Woods [30], [31] found that displays that communicated the strategy being employed by the automation improved operator conformance to automated alerts. Skjerve and Skraaning [33] founded that the time that human operators took to detect critical events as well as subjective measure of human–automation cooperation were improved in nuclear power plant monitoring tasks when interfaces designed to increase the observability of the automation's activity were used.

Beyond the alerting system's end user, there are a variety of other stakeholders who also influence the system performance. The alerting system's design team may develop a design that provides resolutions based on assumptions about operator conformance (whether the operator reacts to the alerting system in the way the designer intended). Further, the type of alerts offered and the nature of operator conformance expected by the designer are influenced (directly or indirectly) by the goals and regulations of administrative, regulatory agencies, and/or indirect users of the system (those benefiting from its operation but not interacting with it directly).

Thus, the performance of an alerting system depends on the interaction of all of these stakeholders, the system itself, and the operational environment in which it is embedded. As such, a system evaluation must take these complex interactions into account when evaluating the overall system performance.

2) Operator Conformance: Operator conformance to alerting systems can be affected by a variety of factors that are related to the system's design such as how complex, reliable, and strategically similar the alerting algorithm is to the operator's operational strategy [26], [29]; what information is available to the operator and how that information is presented [25], [28]; and the false alarm rate of the alerting system [26], [39]. Properties of the operator can also affect performance due to varying

Manuscript received May 1, 2013; accepted September 23, 2013. Date of publication October 17, 2013; date of current version November 26, 2013.

M. L. Bolton is with the Department of Mechanical and Industrial Engineering, University of Illinois at Chicago, Chicago, IL 60607 USA (e-mail: mbolton@uic.edu; matthewbolton@gmail.com).

S. Göknur is with the Department of Visual & Media Arts, Duke University, Durham, NC 27705 USA (e-mail: sgoknur@gmail.com).

E. J. Bass is with the College of Computing and Informatics & College of Nursing and Health Professions, Drexel University, Philadelphia, PA 19104 USA (e-mail: Ellen.J.Bass@Drexel.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/THMS.2013.2283399

degrees of trust in the automated alerting system [9], [26], the mental workload of the operator when an alert is issued [26], and over and under reliance of the operator on the automation [3], [5], [9], [29].

As a result of these factors, it may be inappropriate to assume that operators will conform to the alerting system [28], [29]. Immediate alerting systems (systems that only issue results for immediately inclement conditions) often provide tactical resolutions—A reactionary means of responding to the alert [32]. A variety of metrics exist to measure the operator conformance to immediate alerts. These include how long it takes operators to respond to alerts (response time) [9], procedural compliance for response actions (performing the proper tactical resolution) [43], and signal detection theory (SDT) metrics [35]. However, immediate alerting systems can be dangerous as they can result in operators responding to erroneous alerts (noise in an SDT context) or executing an inappropriate resolution [9].

More sophisticated systems circumvent these issues by using predictive alerts, alerts that warn of potential future conditions and may allow for a more reasoned operator response [9]. Such systems often offer strategic solutions—A set of advisory actions that are based on the long-term goals of the system [21]. Some predictive alerting systems provide multilevel alerts where alert levels will increase as a detected phenomenon becomes more imminent [4]. This may entail a switch from a strategic solution to a tactical one.

Because of the predictive nature of these systems and the increased operator reaction possibilities they afford, it is more difficult to assess conformance and its impact on total system performance. For example, it may sometimes be more beneficial for the operator to delay response to an alert because system strategic solutions may improve over time due to the availability of new information or alert conditions may subside before action is necessary. However, there may also be negative implications for a delayed reaction: escalating hazards and alerts as well as increased stress and workload (both due to the imminence of hazards and the potential availability of multiple alerts and resolutions).

There are a variety of measures used to evaluate conformance in such systems. These include: time in alert [27], number of alerts received, number of hazards that have occurred [2], [38], types of avoidance maneuvers that are provided by the alerting system, and types of avoidance maneuvers that are used by the operator [2]. While all of these provide interesting insight into conformance, none provide a complete picture of system performance, even when used concurrently.

B. Objectives

Clearly, there is a need to identify metrics that are capable of evaluating the effects of conformance on the performance of systems that utilize predictive alerts. Further, given the complex nature of the interactions that are involved in alerting systems, there is a need for a means of determining what factors, aside from conformance, may impact system performance. In the following section, we introduce a framework to identify these factors. We then use it to construct an evaluation of a predictive alerting system.

II. Framework for Scenario Development for Human-Centered Alerting System Evaluation

The purpose of the framework introduced here is to support the development of evaluation scenarios that are capable of assessing system level performance while considering the system, the humans that interact with it, and the environment. The framework has the following steps:

1) identify entities critical to system design, development, and operation and define their goals and properties as they relate to the system being studied;
2) define a subset of functionality for evaluation (define an execution sequence);
3) map entity properties to the execution sequence to identify independent variables;
4) translate entity goals into a set of system goals that can be used to identify dependent measures; and
5) iterate through each step to ensure the models produced are internally consistent.

In the following, we describe each step. However, we first introduce a predictive alerting system example to illustrate the process for each.

A. Before Starting: Development of Domain Knowledge

1) A Cockpit Alerting System for Air Traffic Management: The framework is not meant to help analysts build knowledge about the domain but rather to help facilitate the identification of constants, independent variables, and dependent variables that are necessary to perform a system evaluation. Thus, before applying this evaluation framework, analysts should be familiar with the domain in which the system they are evaluating operates; the relevant individuals, organizations, and subsystems for which the system is being evaluated; what phenomena they want to evaluate in a systems context; and what resources (including facilities, apparatus, personnel, funds) are available to perform the evaluation.

Thus, before we can demonstrate how this step is performed, the relevant background information for our illustrative air traffic alerting system example is presented.

NASA continues to develop a set of operational concepts, procedures, and decision support tools to improve operations of the national airspace system. In these, pilots are expected to have additional responsibilities that are associated with maintaining aircraft separation. One operational concept involves having flight crews, with properly equipped aircraft, manage separation from other aircraft during the en route and terminal-transition domains [2], [38].

A supporting technology includes a predictive, multilevel, cockpit alerting system to call a pilot's attention to potential airspace conflicts and make recommendations for a course of action. One early prototype, the autonomous operations planner (AOP), was designed to alert pilots to conflicts in the airspace, where conflicts represent loss of separation (LOS)(when aircraft get within 5 nmi laterally and 1000 ft vertically of each other) or collision between aircraft (when aircraft get within 0.15 nmi laterally and 300 ft vertically of each other) [1]. AOP supports a four-level alerting scheme. A level 0, alert indicates that LOS is a possible, but not current, threat to the ownship. A level 1,

alert is issued when a conflict is predicted to occur within 8 min. Level 1 alerts are accompanied by the availability of strategic solutions, where the flight management system's flight plan is modified to avoid the conflict [2]. If the conflict is predicted to occur within 5 min, AOP generates a level 2 alert, known as a conflict detection zone alert. An alert level 2 is accompanied by a tactical resolution that recommends a heading and/or vertical speed change. Strategic resolutions may also be available but are not guaranteed. A level 3 alert, known as a collision avoidance system (CAS) alert, indicates that a collision may be impending within 1 min [2], [38]. AOP provides only tactical resolutions for CAS alerts.

To facilitate the study of pilot interaction with this alerting system, NASA Langley has developed simulation capabilities in which up to eight human pilots can fly simulated aircraft that are interacting as part of a larger traffic management simulation [2]. Because this simulation models the entire airspace as well as the cockpit alerting system, it can be used in experiments that are designed to evaluate how operator conformance with the alerting system impacts the system performance. An automated pilot agent supports simulation experiments without human subjects [15].

As AOP is a predictive alerting system and the impact of operator conformance on the total system performance is not understood, the effect of operator conformance to AOP on the system performance provides an example application of the evaluation framework.

B. Step 1: Identify System Entities

The objectives of a system are determined by its stakeholders [42] while how a system meets these objectives is determined by the interactions between the technological, ecological, and shareholding entities (sometimes humans) that compose the system. Thus, the first step in constructing a systems evaluation is to identify these entities and to describe the relationships between them.

For the purpose of classification, we consider two types of entities: stakeholding entities (SHEs), and technological and/or ecological entities (TEEs). SHEs constitute any entity vested in the performance of a system including human operators who directly interact with the system, customers that have purchased services facilitated by the system, individuals or groups that have capital investment in the system, entities that regulate the system, and the system designers. TEEs constitute any other type of entity that may influence the system performance, but do not hold stake in the system's performance. These may include the particular subsystem or technological product that is the focus of the evaluation, other technological subsystems that interact with it, and the environment or environmental subsystems.

SHEs are defined based on their tasks, goals, and properties. An entity's tasks define what the entity is doing in the system, and thus, help define its relationship to the other system entities. Goals define what the entity is attempting to accomplish with these tasks. Properties identify internal properties of the entities that may affect their ability to perform tasks or achieve goals. Because they are not actively performing tasks within the system, TEEs are only defined by their properties. In this case, properties constitute sources of variance within the entity that may affect how other entities interact with them. To define the relationships between the entities at this stage in the process, one can identify the existence of interaction and the direction in which the interaction takes place (what entity is exerting its influence).

To illustrate these concepts, we now identify the entities for the air-traffic cockpit alerting system example and the relationships between them. Fig. 1, an entity interaction chart, is a visual model used for this purpose. This figure depicts two TEEs, the cockpit alerting system itself and the environment. In this case, the environment represents the airspace with properties (weather, air traffic geometry, and no fly zone locations) that are capable of describing the airspace. The cockpit alerting system's properties represent variables that are capable of representing the state of the alerting system: computational resources, the functional state of the equipment, and the availability of conflict resolutions. Because the alerting system provides resolutions to anticipated air traffic events that will change the state of the airspace and the nature of the airspace environment will determine what resolutions are available, properties of each influence the properties of the other.

Fig. 1 also contains six SHEs and their tasks, goals, and interactions: airline passengers, aerospace regulatory agencies, air traffic control, the pilots, airline policy makers, and the alerting system designers. Direct interactions are modeled using arrows. Indirect relationships can be inferred by tracing interactions across entities. For example, passengers will influence airline policy makers by providing customer feedback, airline policy makers will make policy changes for pilots to follow, and pilot behavior will then influence customer flying experience.

C. Step 2: Define a Subset of Functionality for Evaluation

The next step in the evaluation framework is to identify a subset of system functionality for evaluation and to construct the sequence of events associated with it. For the air traffic alerting example, the goal is to evaluate how pilot conformance to alerts will affect the total system performance. We next define the sequence of events associated with an alert and the resulting pilot response [see Fig. 2(a)]. In this, the following occurs: 1) a traffic event occurs, 2) the alerting system generates an alert in response to the traffic event, 3) resolutions are offered to the pilot, 4) there is a delay in pilot response, and 5) the pilot responds to the alert.

Each event is attributed to exactly one entity. If the event is not associated with an existing entity, this may suggest that an additional entity should be added to the entity interaction model. If more than one entity is associated with a given event, one should consider decomposing the event up so that each event is associated with one entity.

D. Step 3: Map Entity Properties to Execution Sequences Events

To identify variables (independent variables and constants), one must identify factors that may affect performance for the execution sequence of interest by mapping entity properties to the sequence of events identified in step 2. Since each event

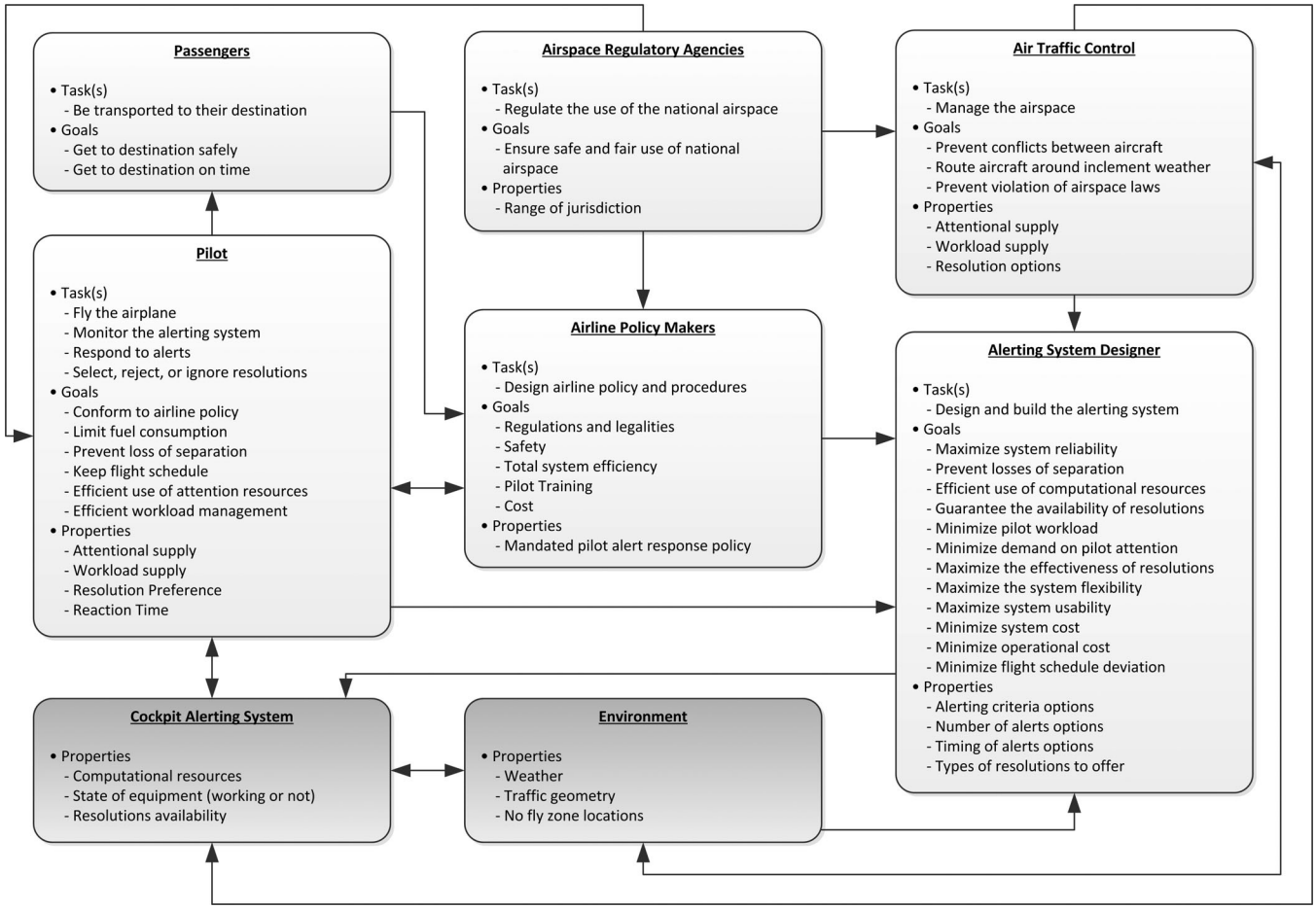

Fig. 1. Entity interaction chart for the cockpit air traffic alerting system. TEEs have a darker background. Lines indicate interaction where the entity on the arrowed end of the line is influenced by the entity on the nonarrowed end.

in the execution sequence is associated with exactly one entity, only properties from the associated entity and those that directly interact with it (as identified in step 1) should be capable of influencing that event.

This process is represented visually for the air traffic alerting system example in Fig. 2(b) (only properties deemed relevant to the execution sequence are included). When performing this step, analysts should ensure that they consider the properties of all relevant entities.

E. Step 4: Map StakeHolder Goals to System Goals

The goals of the SHEs are used to define a set of system goals that later help to define dependent measures. Overlap between the goals of individual SHEs should be identified in order to create a single set of system goals. This process is illustrated for the air traffic alerting system example in Fig. 3. Here, the goals for each SHE are listed vertically and connected to corresponding goals from other SHEs using horizontal lines.

F. Step 5: Iteration and Finalization

Iteration and refinement are encouraged. Any changes incorporated at a given step should be propagated to the other steps.

III. Case Study

Here, the framework is executed as a case study proof of concept using simulation. Multilevel alerting systems that act as predictive aids are becoming more prevalent. The total system performance of a system including a human operator and an alerting system relies on the human operator's response to the alerting system. The combination of response time and resolution strategy may lead to different quality of the total system performance. The simulation was designed to investigate the effects of operator response delay time with a multilevel alerting system on the total system performance. A simulated pilot responds to the output of a traffic conflict alerting system that provides traffic alerts and resolutions. In addition to the pilot's response delay time, scenario length (distance to point of closest approach), the pilot model's sensitivity to alert level upgrades, resolution preference, and whether the traffic encounter would lead to a collision were considered. When the simulated pilot is sensitive to alert level upgrades, it interrupts the original delay count and modifies it to respond to the upgrade earlier. Therefore, pilot's model sensitivity to alert level upgrades introduces additional response times.

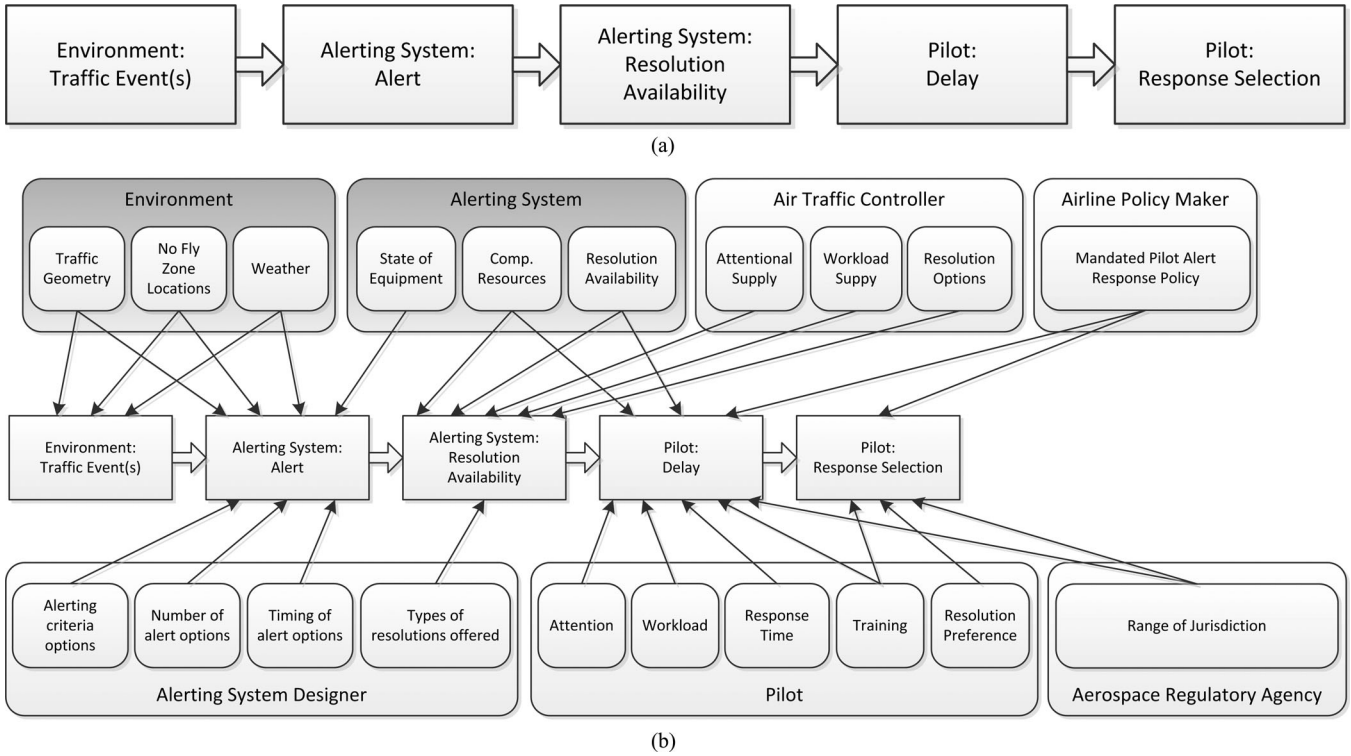

Fig. 2. (a) Execution sequences associated with an AOP alert and the pilot response. (b) Mapping of entity properties to the events in the execution sequence.

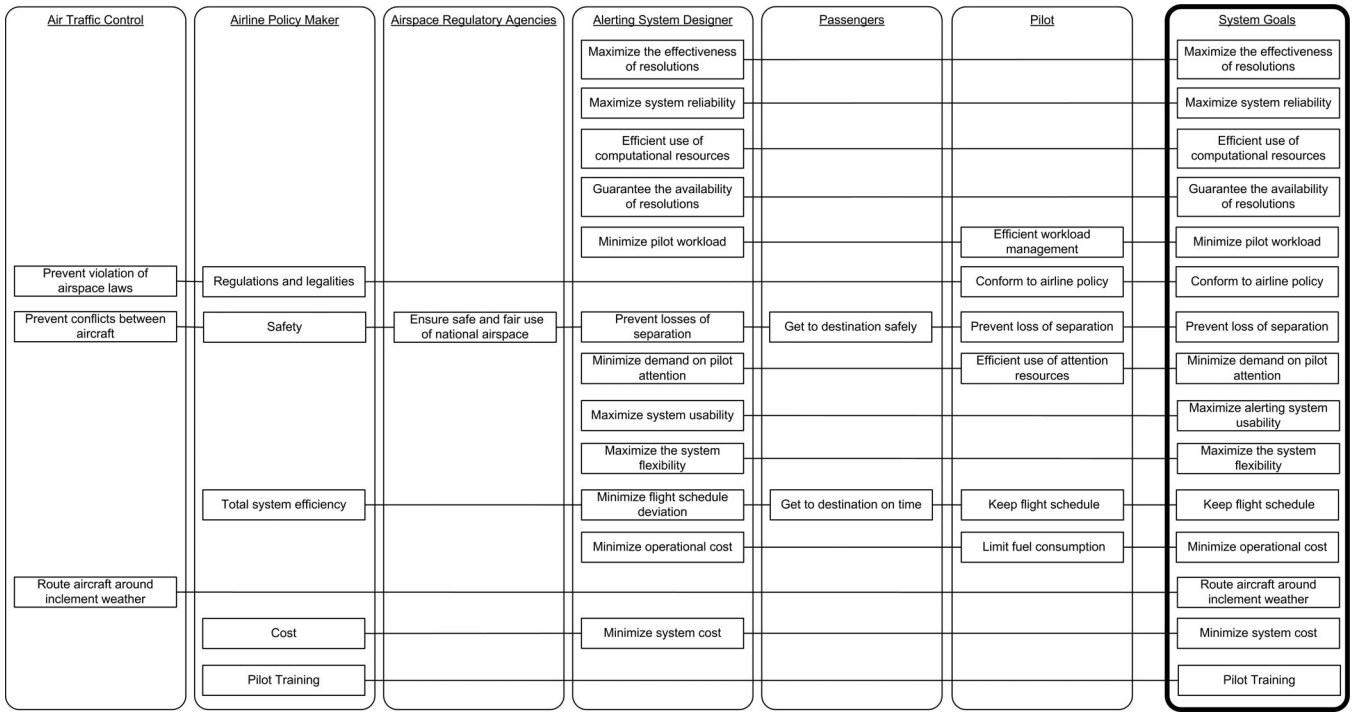

Fig. 3. Mapping of SHE goals to the set of system goals.

A. Methods

1) Apparatus: The air-traffic simulation (TMX) was designed as a desktop air traffic simulation with scripting capabilities [36]. ASTOR was designed as a configurable part-task flight deck simulator that can be used to control TMX aircraft. The configuration used in this study included the AOP alerting system, the control display unit (CDU), the glareshield control panel (GCP), and the navigation display (ND) (see Fig. 4).

AOP alerts are presented on the ND using both an intruder aircraft (a chevron shape) and a line (a band) along the

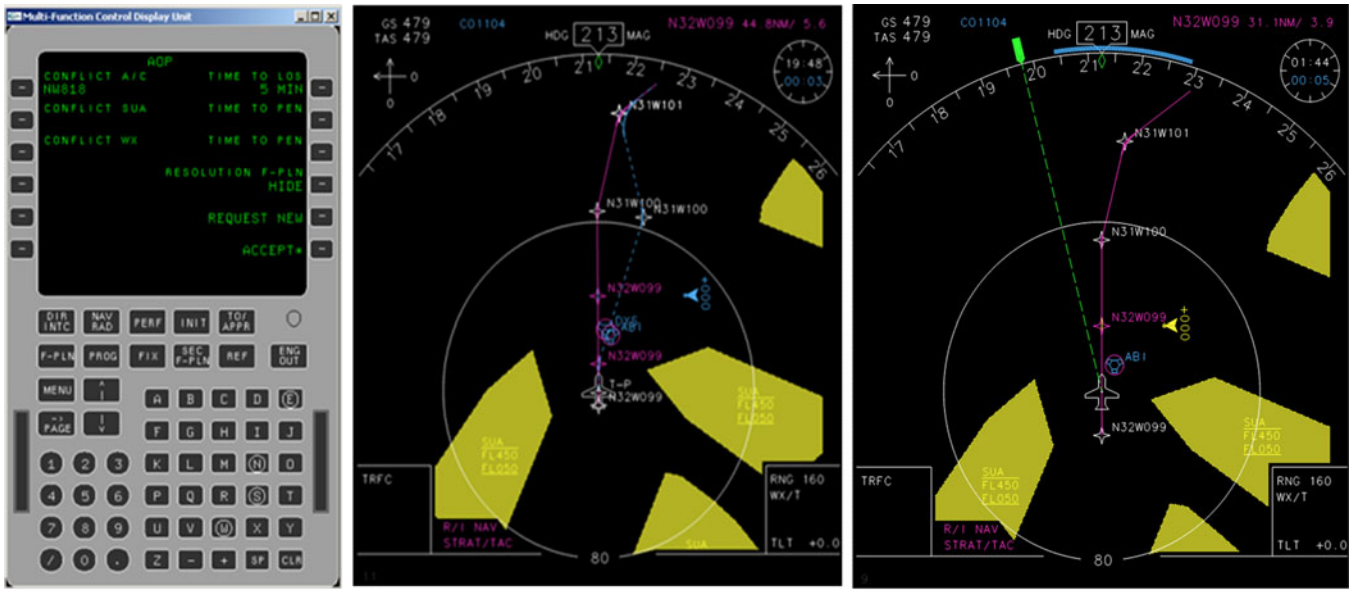

Fig. 4. ASTOR cockpit simulation (a) CDU interface to AOP. (b) ND with an intruding aircraft at a level 1 alert and a strategic resolution. (c) ND with an intruding aircraft at a level 2 alert and a tactical resolution.

outside of the heading display. The intruder aircraft indicates the location of the aircraft that caused the predicted conflict. The band indicates headings for which a conflict will occur/persist. Alerts are colored based on their level. A level 0 alert displays a black intruder aircraft with a blue outline (no band is associated with level 0 alerts). Level 1, 2, and 3 alerts displays solid blue, yellow, and red intruder aircraft and bands, respectively.

Strategic solutions and tactical solutions are displayed on the ND. Magenta lines on the ND represent the original flight plan; the dashed blue line depicts AOP's strategic resolution. The pilot may accept the resolution, hide it (if it is associated with a level 1 alert), or request a new resolution by using the options displayed on the CDU. Heading-based tactical resolutions are displayed by using green dashed lines pointing to the suggested heading. The pilot can conform to the tactical resolution by adjusting the heading of the aircraft through the GCP.

A custom built simulated pilot agent was used to respond to horizontal conflicts related to level 1, 2, and 3 AOP alerts [15]. The pilot agent was capable of identifying intruder aircraft on the ND; requesting, evaluating, and accepting/rejecting strategic resolution through interaction with AOP via the CDU; evaluating tactical resolutions presented on the ND; and executing tactical resolutions by changing ownship's heading via the GCP. The actions taken by the pilot agent, and the order they were executed, was dependent on the nature of the alert. Fig. 5 contains the enhanced operator function model [6] representation for the pilot response to an AOP alert level 2.

2) Independent Variables: In step 3, entity properties were identified that could potentially impact the system performance when a pilot is responding to an AOP alert (see Fig. 2). Only six entity properties are supported by the simulation environment: the environment's traffic geometry; the alerting system's equipment state; and attention, workload, response time, and resolution preference for the pilot.

With respect to traffic geometries, the simulation trials used a variant of the traffic configuration used in the *overconstrained conflict* traffic problem used in [38].Within this configuration, two traffic geometry conditions were created: one where ignoring alerts and resolutions would result in a collision and one where it will only result in a loss of separation. Thus, the independent variable of interest was called the collision course indicator. Nominally, AOP will detect a conflict 8 min before it happens. However, in the event of alerting logic failure, AOP may not issue an alert as punctually (for example, an error in the ADS-B communication network could make traffic positions temporarily unavailable). To replicate such a system state, variants of the traffic geometry configuration were set up so that the two aircraft would start closer to the point of closest approach, giving the automation less time to predict the conflict. This created six different scenario lengths (in minutes): eight (the nominal case), seven, five, three, and two (see Fig. 6).

Since the goal of this evaluation was to assess the impact of pilot conformance to the automation on total system performance, pilot response time is very important. To address response time, six response delays were established that determined how long the pilot agent would wait before responding to an alert.

1) Proactive: The pilot delays response until 30 s after the first alert is issued.
2) Late: The pilot delays response until 10 s before the alert upgrades to a higher level.
3) After upgrade: The pilot responds the alert within 50 s after the alert upgrades to a higher level.
4) Late after upgrade: The pilot delays responding to the alert until 30 s before the alert upgrades for the second time.
5) After second upgrade: The pilot responds to the alert within 50 s after the second alert upgrade.
6) Extremely late: The pilot responds 20 s before a collision occurs.

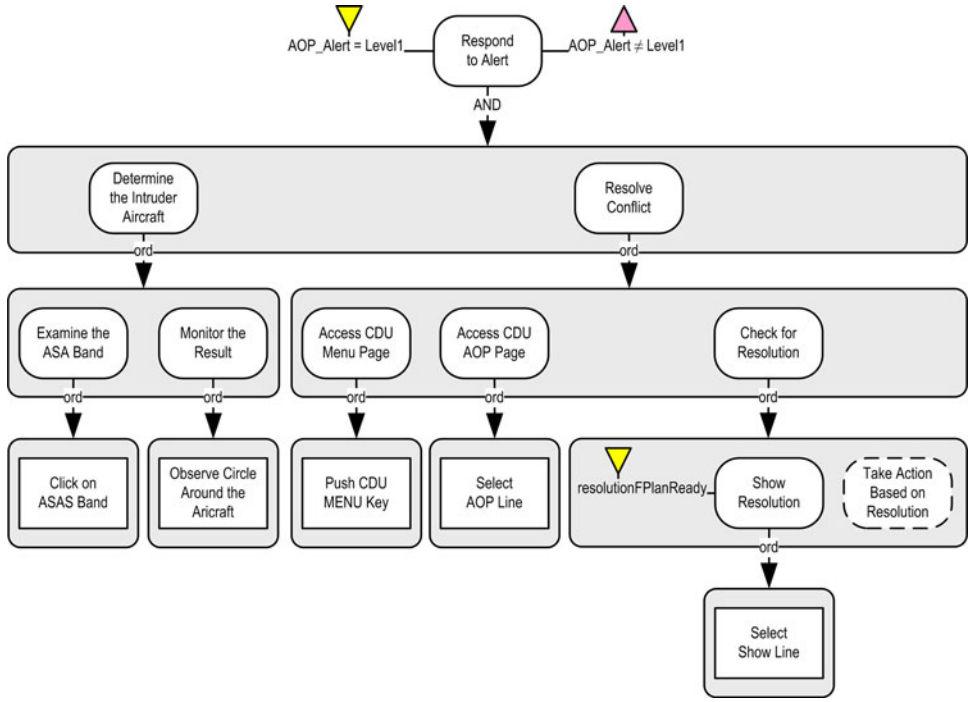

Fig. 5. Enhanced operator function model for the pilot response to an AOP alert level 2.

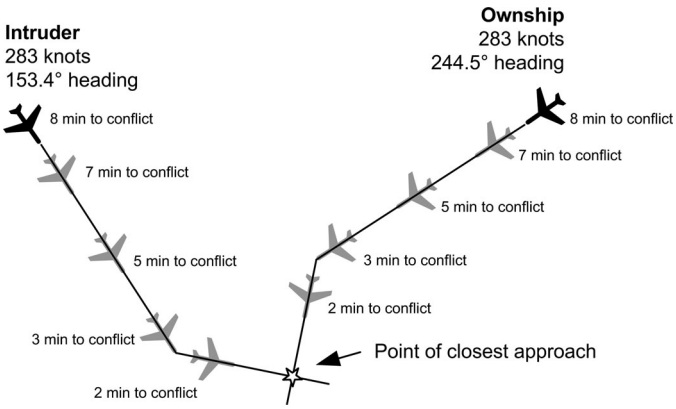

Fig. 6. Traffic geometry for the five scenario lengths.

Given that pilot workload and situation awareness will affect pilot response time, pilot response to delay is used as a proxy for these in this experiment.

Pilot resolution preference is accounted for by having the pilot agent prefer either a strategic or tactical solution. Thus, when both strategic and tactical solutions are available, the pilot agent will select the one it prefers. Otherwise, it will select the only available solution.

3) Dependent Variables: Using the list of system goals from step 4, one can then identify measurable quantities or observable qualities that are capable of indicating if and/or how well these goals are being met. Fig. 7 highlights the measures for the air traffic alerting system example.

Epsilon (ε) is a measure of conflict separation that converts the 3-D separation measurements between aircraft into a single dimension measure using the equation

$$\varepsilon_{ij}(t) = \frac{\Delta X_{ij}^2(t)}{a^2} + \frac{\Delta Y_{ij}^2(t)}{b^2} + \frac{\Delta Z_{ij}^2(t)}{c^2}$$

where $\Delta X_{ij}^2(t), \Delta Y_{ij}^2(t)$, and $\Delta Z_{ij}^2(t)$ are the relative lateral and vertical distances between two aircraft $(i$ and $j)$ using a Cartesian axis system where X points East, Y points to North, and Z points upwards [10]. a, b, and c are constants that define an ellipsoid representing the separation space around each aircraft $(a = b = 5$ nmi and $c = 1000$ ft). Thus, a loss of separation occurs when the separation between the aircraft puts them within this ellipsoid $(\varepsilon_{ij}(t) < 1)$. Minimum epsilon represents the minimum separation between two aircraft for a given time period. Because of this, it can be used as a measure of system effectiveness and reliability (how well the system maintains separation between aircraft). Further, because ε provides an assessment of when loss of separation occurs, it can also be used to determine if the system is capable of preventing loss of separation via the count of separation violations [2], [38].

The count of strategic and tactical resolutions [2] was used to provide insight into system reliability and its ability to guarantee the availability of resolutions (a reliable system would be able to provide resolutions to every alert), how efficiently the system was using computation resources (the more resolutions the system generates, the more computational resource being used to generate resolutions), and pilot workload (the more resolutions that are offered, the more workload placed on the pilot).

Pilot workload was also associated with the time until the predicted conflict when the pilot responds (TUC) since pilot workload will increase as an impending conflict nears.

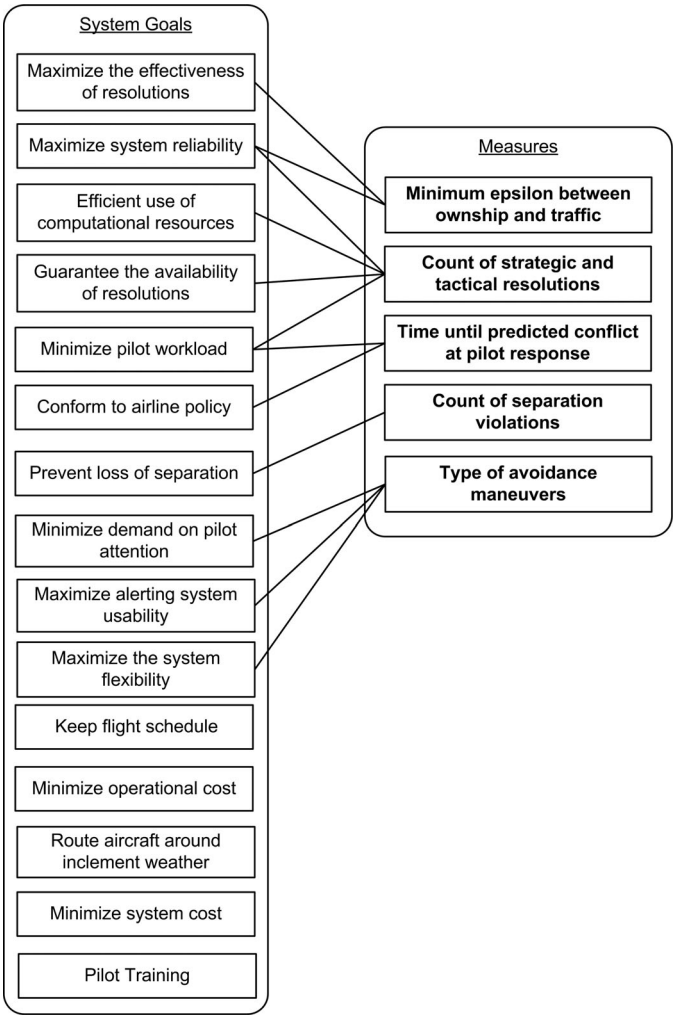

Fig. 7. Mapping of system goals to measurable quantities.

The type of resolutions offered by the automation was used as a measure of usability and the flexibility goals (with both strategic versus tactical resolutions, the system potentially provides more usable functionality). An increase in the types of resolutions available will require pilots to use more attentional resources to choose between resolutions; this can indicate higher demand on attentional resources.

4) Experimental Design: A single replicate factorial design was used in this experiment. However, not all combinations of the independent variables result in valid trials (see Fig. 8). Of the $6 \times 6 \times 2 \times 2 \times 2 = 288$ theoretically possible trials only 62 were valid. In 7- and 8-min long scenario, the simulation can proceed through all three alert levels, and thus, all six delay times are valid. However, for the shorter scenarios, only alert levels 2 and 3 are experienced. Thus, the late after upgrade and after second upgrade delays are not valid for these scenarios. Additionally, when the intruder aircraft and the ownship are not on a collision course, an AOP alert level 3 will never be generated. In these situations, the late after upgrade and after second upgrade responses are irrelevant. The resolution preference is only applicable to cases where the simulated pilot responds during an alert level 2 when both tactical and strategic resolutions are ready. For all other cases, the simulated pilot responds based on the only resolution available. Finally, when the pilot responds sensitively to alert upgrades, two of the scenarios become the same when the pilot responds 15 or 5 s after the alert depending on the alert level. In these cases, the replicates were eliminated from the experimental run.

IV. Results

General linear model analyses of variance were used to assess the effects of the four independent variables on minimum epsilon and time until conflict. Variables found to be significant were evaluated using a Bonferroni *post hoc* analysis. Results of these analyses are reported as significant with $\alpha = 0.05$. Results for the count of LOS and the counts of strategic and tactical resolutions are presented based on the rates with which they occurred.

A. Minimum Epsilon

An ANOVA analysis indicated that there were significant differences between scenario lengths ($F(4, 50) = 15.67, p < 0.01$), response delays ($F(5, 50) = 26.46, p < 0.01$), and collision course indicators ($F(1, 50) = 4.15, p = 0.05$). A *post hoc* analysis revealed that there were significantly higher minimum epsilons for trials with 7- and 8-min lengths than there were for trials with 2-, 3-, and 5-min lengths [(see Fig. 9(a)]. A *post hoc* also showed that there were significantly higher minimum epsilons for proactive actions than for pilots that responded after upgrade, and for pilots that responded proactively, late, or after upgrade compared to all of the more delayed responses [see Fig. 9(b)]. There were significantly lower minimum epsilons for scenarios without the collision condition [see Fig. 9(c)].

B. Time Until Contact

For TUC, ANOVA results indicated that there were significant differences between scenario lengths ($F(4, 50) = 22.32$, $p < 0.01$) and response delay ($F(7, 50) = 34.21, p < 0.01$). For scenario length, a *post hoc* analysis revealed that there was significantly more time until conflict for the 7- and 8-min scenarios than for the shorter scenarios. For delay time, the *post hoc* analysis revealed that proactive pilots had significantly higher TUC than for all the other delay schemes [see Fig. 10(a)]. Further, late, after upgrade (sensitive), and after upgrade delays produced higher average TUC than the remaining, longer delay schemes [see Fig. 10(b)].

C. LOS Count

Given the nature of the experimental scenarios, there was only one loss of separation per scenario. Thus, the LOS count for a given scenario could have one of two values: 0 or 1. An examination of the data revealed that LOS occurred as a function of the delay type and scenario length, with 100% of the trials with specific delay type and scenario length combinations producing an LOS and no LOS occurring under any other conditions. The delay type/scenario length combinations that produce LOS are shown in Table I. This reveals that for late after upgrade and long scenarios, an LOS will always occur. For late and after

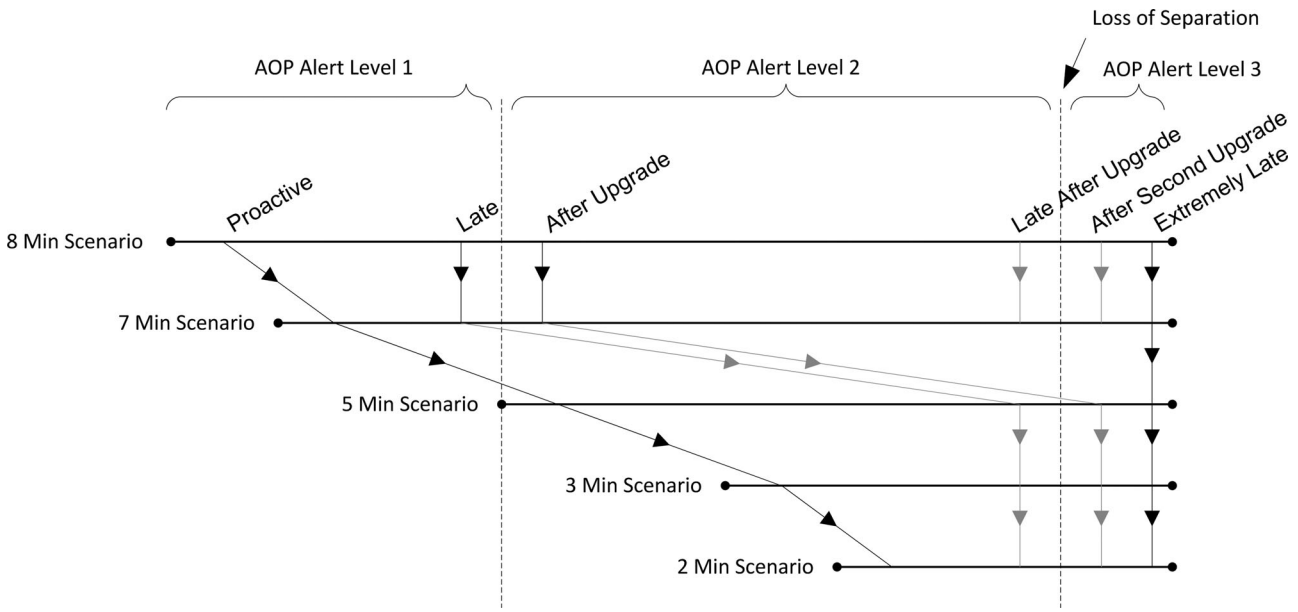

Fig. 8. Interaction between the six types of response delay and the five scenario lengths. All six delays occur for the 8- and 7-min scenarios. Only the proactive, late, after upgrade, and extremely late delays are valid for 5-, 3-, and 2-min scenarios. Because a level 3 alert will only occur in scenarios with collisions, response delays based on the transition between a level 2 and level 3 alert are only valid for scenarios with a collision (shown with gray arrows above).

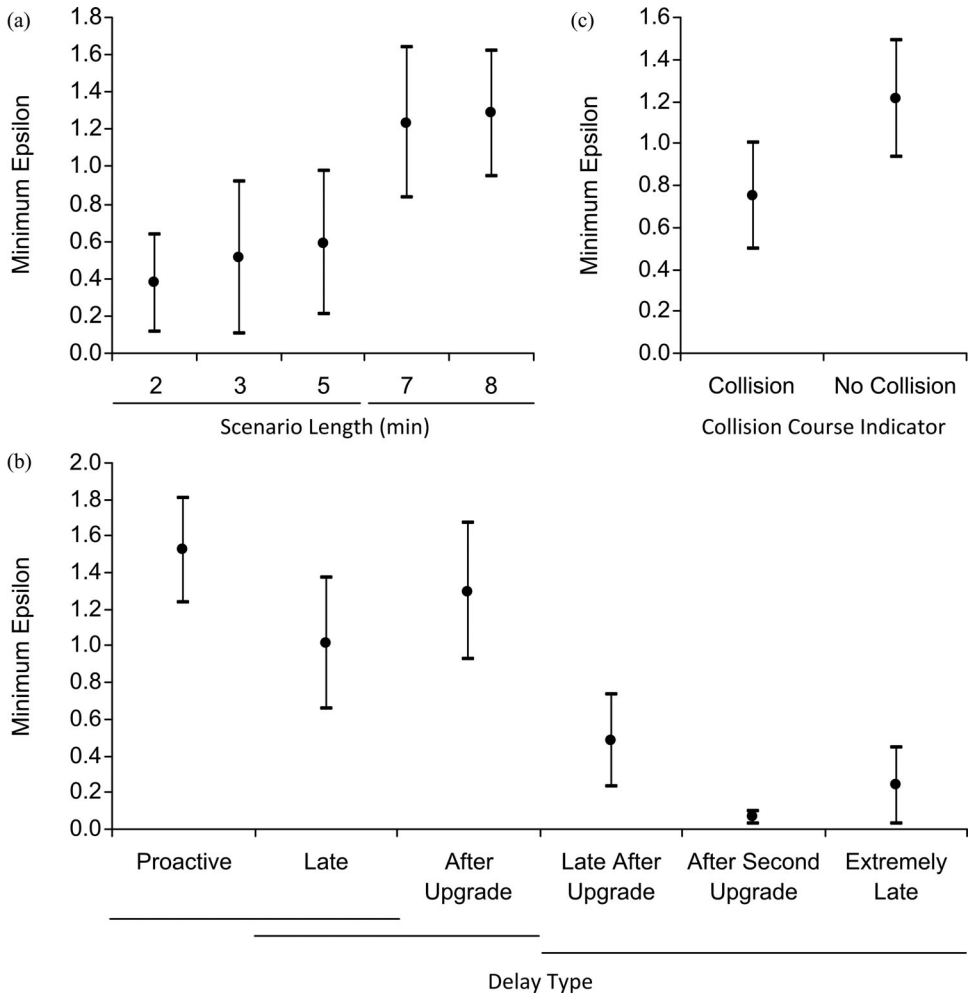

Fig. 9. 95% confidence interval plots for minimum epsilon where lines under charts indicate homogenous subsets according to a Bonferroni *post hoc* analysis. (a) Scenario length, (b) collision course, and (c) delay type.

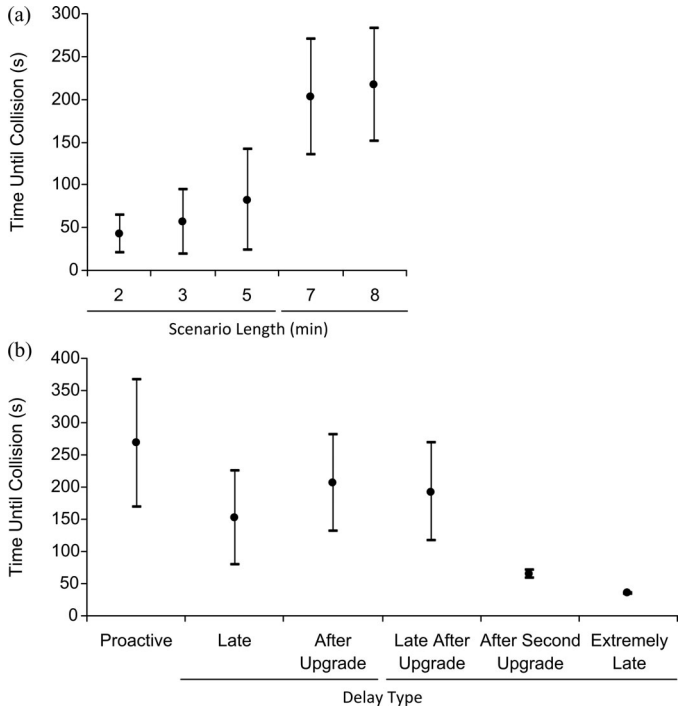

Fig. 10. 95% confidence interval plots for TUC where lines under charts indicate homogenous subsets according to a Bonferroni *post hoc* analysis. (a) Scenario length and (b) Delay type.

TABLE I
INCIDENTS OF LOS AS A FUNCTION OF SCENARIO LENGTH AND DELAY TYPE

	Scenario Length (min)				
Delay Type	2	3	5	7	8
Proactive	X				
Late	X	X	X		
After Upgrade	X	X	X		
Late After Upgrade	na	na	na	X	X
After Second Upgrade	na	na	na	X	X
Extremely Late	X	X	X	X	X

Note. An X indicates that the specified scenario length and delay type combination produces a LOS in all scenarios. An "na" indicates that the particular scenario length and delay type combination is not applicable.

upgrade delays, LOS occurs for all but 7- and 8-min. scenarios. For proactive pilots, LOS only occurs for 2-min scenarios.

D. Count of Tactical and Strategic Resolutions

The count of strategic resolutions also proved to be a function of scenario length and delay type (see Table II), with the number of strategic resolutions increasing with an increase in delay for all but 2- and 3-min scenarios. The count of tactical resolutions was a function of scenario length, delay type, and collision indicator (see Table III). In all cases, the number of tactical resolutions increased with delay time. However, in the no collision scenarios, the count of tactical resolutions decreased for the extremely late delay.

V. DISCUSSION OF THE CASE STUDY

With the results reported, the framework can now be used to help interpret them. This is accomplished by assessing the impact of the independent variables on system goals using the mapping of goals to measurable quantities.

A. Pilot Resolution Preference

Pilot resolution preference had no effect on any of the tested dependent measures.

B. Collision Course Indicator

The fact that there were smaller minimum epsilons for scenarios with a collision indicate that the effectiveness of resolutions provided by the system and system reliability were best supported when aircraft were not on a collision course. The fact that the collision condition resulted in more tactical resolutions in scenarios with extremely late pilot response delay (see Table III) indicates that system reliability increased to compensate for the more dangerous conditions present in these scenarios. Further, the alerting system's ability to provide tactical resolutions under both conditions supports the system goal of guaranteeing the availability of resolutions. However, because an increase in the number resolutions may indicate an increase in pilot workload and the computational resources necessary to provide them, these advantages may come at the expense of pilot workload and the availability of computational resources.

C. Scenario Length

Higher minimum epsilons and TUC values were observed more with 7- and 8-min scenarios than for shorter scenarios. These results indicate that resolutions were more effective, the system was more reliable, pilot workload was reduced, and airline policy better adhered to when the automation detected the conflict earlier. The goal of preventing LOS was also best facilitated by the 7- and 8-min scenarios, which prevented LOS in all but the three longest delay schemes (see Table I). LOS was only averted in the shorter 2- and 5-min scenarios for proactive pilot delay types. LOS occurred for all 2-min scenarios.

Given that the varying scenario length was meant to serve as a proxy for the functional state of the alerting system (such as communication delays, and alert detection problems), these results indicate how critical it is that the system provide timely resolutions. Doing so helps ensure that an LOS does not occur (which is critical to aircraft safety) and significantly improves the efficiency of resolutions.

Additionally, the number of available strategic resolutions generally increased with scenario length and the number of available tactical resolutions decreased with scenario length (see Tables II and III). With this distribution, a resolution was always available, supporting the goals of guaranteeing the availability of resolutions and maximizing system reliability. However, this distribution also ensures that more resolutions are available in median scenario length, increasing pilot workload and computational resource utilization.

D. Pilot Response Delay

The shorter pilot delay types (proactive, late, and after upgrade) resulted in significantly larger minimum epsilons and

TABLE III
COUNT OF TACTICAL RESOLUTIONS AS A FUNCTION OF SCENARIO LENGTH, DELAY TYPE, AND COLLISION INDICATOR

Scenario Length (Min)	Delay Type						
	Proactive	Late	After Upgrade	Late After Upgrade	After 2nd Upgrade	Extremely Late	
						Collision	No Collision
8	0	0	1	1	2	2	1
7	0	0	1	1	2	2	1
5	1	1	2	na	na	2	1
3	1	1	2	na	na	2	1
2	1	1	2	na	na	2	1

Note. "na" indicates that the particular combination of variable never occurred and was therefore not applicable.

TABLE II
COUNT OF STRATEGIC RESOLUTIONS AS A FUNCTION OF SCENARIO LENGTH AND DELAY TYPE

Scenario Length (Min)	Delay Type					
	Proactive	Late	After Upgrade	Late After Upgrade	After 2nd Upgrade	Extremely Late
8	1	1	2	2	2	2
7	1	1	2	2	2	2
5	0	1	2	na	na	1
3	0	0	0	na	na	0
2	0	0	0	na	na	0

Note. "na" indicates that the particular combination of variable never occurred and was therefore not applicable.

TUC than the longest delay times (late after upgrade, after second upgrade, and extremely late). These results indicate that resolutions were more effective, the system was more reliable, pilot workload was reduced, and airline policy better adhered to when the pilots responded to conflicts earlier. Further advantages to early response (late delay) is seen in the LOS data (see Table I) where late after upgrade and longer delay schemes always resulted in an LOS, late and after upgrade delays resulted in LOS for all but the 7- and 8-min scenarios, but LOS was avoided for all but 2-min scenarios for proactive delays. Thus, as with scenario length, how a pilot delays response to an alert has a serious impact on aircraft safety (LOS), system reliability, and system efficiency.

An increase in pilot delay was associated with an increase in the number of available strategic and tactical resolutions, indicating increases in system reliability, resolution availability, pilot workload, and computational resource utilization.

This case study illustrated the benefit of an in-cockpit traffic alerting system to detect errors early and for pilots to respond to them proactively. Additionally, while early conflict detection and pilot response help contribute to safety, there are tradeoffs associated with the reduced availability of resolutions.

Given the limitations of the simulation and the assumptions used to constrain the design space, there are still many different goals and domain factors yet to be explored. A limitation of this empirical study is that it is conducted via simulation and actual pilots could behave differently. For example, Ellerbroek *et al.* [11], [12] founded that the type of display representation can impact planning behaviors, an aspect not modeled herein. In addition, because of the number potential options for traffic geometries, weather, and no fly zone locations, this experiment used a very constrained set of traffic geometry scenarios. However, because these factors were identified as being important in Fig. 2, further experiments could explore how these environmental properties impact system goals.

Not all of the goals identified in step 4 have been mapped to measurable quantities and some measures are indirect indicators of quantities that would be directly relevant to goals. This is due to the limitations of the simulation infrastructure being used. For example, if the evaluation were being run with human subjects instead of automated pilot agents, workload and situation awareness measures like NASA TLX [17] and SAGAT [13] could be used to assess these quantities rather than the indirect measures listed here. Further, if the simulation provided fuel consumption or path-delay data, these could be used as dependent measures for the respective system goals. However, by mapping SHE goals to measurable quantities, and identify goals that are incapable of being evaluated for a given evaluation procedure, this framework will help analysts target future evaluation efforts to cover goals not previously considered.

As such, a variety of system goals were not addressed in this experiment. Thus, further experiments could be conducted to assess how the parameters of this experiment (or parameters not assessed in this experiment) impact dependent measure that would map to the unconsidered goals. For example, the simulation could be modified to provide fuel consumption and schedule delay data for each resolution. These could then be used to assess how well a given scenario met the flight schedule and operational cost minimization goals.

VI. GENERAL DISCUSSION

As has been demonstrated, the evaluation framework discussed in this paper is useful to develop evaluations for alerting

systems. By identifying SHEs and TEEs; describing how each entity interacts; recognizing which entity properties impact different stages in an execution sequence; and deriving system goals from stockholder goals; this framework allows researchers to identify independent variables and dependent measures that can be used to evaluate how well the system goals are being met under different operation conditions. Further, because it is unlikely that any complex system will be capable of being evaluated in a single study, the framework allows one to see which entity properties and system goals were not previously considered so they can be used to plan future evaluations.

The framework as used here employs a nonstandard modeling notation. This was done to account for the very specific content that needs to be represented. Thus, any analyst who is evaluating a system without any existing specification or system model could use the modeling infrastructure provided here to perform the evaluation. However, because systems evaluations may occur with varying degrees of preexisting system specifications, it may be worth investigating the integration of the evaluation framework into preexisting modeling technology.

A survey of the systems engineering literature reveals a number of standard modeling infrastructures that could be used in the evaluation framework. For the entity interaction chart, one possibility is the entity interaction diagram [40]. In such diagrams, entities are represented as rectangular blocks (nodes) and their interactions (called relationships) are modeled as labeled directed and undirected graph edges. In the context of the evaluation framework, SHEs and TEEs would be represented as the nodes, and entity tasks could be used to define the edges between the nodes. Higraphs [18] could also be used in a similar manner.

There is also a number of behavior modeling techniques that could be used to represent the execution sequence (and the influence of entity properties on it) used in the evaluation framework. These include function flow diagrams, behavior diagrams, finite-state machines, state charts, control flow diagrams, and Petri nets [42].

Finally, the use of general-purpose modeling languages used in the systems and software engineering community would be beneficial. Unified Modeling Language (UML), offers a variety of modeling technologies [7] as does SYSML [16]. If the evaluation framework could be adapted to work with languages like UML or SYSML, it could be used in many different evaluations of software systems.

REFERENCES

[1] R. Barhydt, T. M. Eischeid, M. T. Palmer, and D. J. Wing, "Regaining lost separation in a piloted simulation of autonomous aircraft operations," presented at the 5th USA/Eur. Air Traffic Management R&D Semin., Budapest, Hungary, Jun. 2003.

[2] B. Barmore, E. Johnson, D. J. Wing, and R. Barhydt, "Airborne conflict management within confined airspace in a piloted simulation of DAG-TM autonomous aircraft operations," presented at the 5th USA/Eur. Air Traffic Manag. R&D Semin., Budapest, Hungary, Jun. 2003.

[3] E. J. Bass and A. R. Pritchett, "Human-Automated Judge Learning: A methodology for examining human interaction with information analysis automation," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 38, no. 4, pp. 759–776, Jul. 2008.

[4] E. J. Bass, S. T. Ernst-Fortin, R. L. Small, and J. T. Hogans Jr., "Architecture and development environment of a knowledge-based monitor that facilitate incremental knowledge-base development," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 34, no. 4, pp. 441–449, Jul. 2004.

[5] E. J. Bass, L. A. Baumgart, and K. K. Shepley, "The effect of information analysis automation display content on human judgment performance in noisy environments," *J. Cognit. Eng. Decision Making*, vol. 7, no. 1, pp. 49–65, 2013.

[6] M. L. Bolton, R. I. Siminiceanu, and E. J. Bass, "A systematic approach to model checking human-automation interaction using task-analytic models," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 41, no. 5, pp. 961–976, Sep. 2011.

[7] G. Booch, *The Unified Modeling Language User Guide, 2/E.* Delhi, India: Pearson Education, 2005.

[8] S. S. Choi, J. K. Park, J. H. Hong, H. G. Kim, S. H. Chang, and K. S. Kang, "Development strategies of an intelligent human-machine interface for next generation nuclear power plants," *IEEE Trans. Nucl. Sci.*, vol. 43, no. 3, pp. 2096–2114, Jun. 1996.

[9] S. D. Davis and A. R. Pritchett, "Alerting system assertiveness, knowledge, and over-reliance," *J. Inf. Technol. Impact*, vol. 1, no. 3, pp. 119–143, 1999.

[10] W. den Braven, "Analysis of aircraft/air traffic control performance," in *Proc AIAA Guid., Navigat., Control Conf.*, Baltimore, MD, USA, Aug. 7–10, 1995, pp. 1721–1738.

[11] J. Ellerbroek, K. C. R. Brantegem, M. M. van Paassen, N. de Gelder, and M. Mulder, "Experimental evaluation of a coplanar airborne separation display," *IEEE Trans. Human-Mach. Syst.*, vol. 43, no. 3, pp. 290–301, May 2013.

[12] J. Ellerbroek, K. C. R. Brantegem, M. M. van Paassen, and M. Mulder, "Design of a Coplanar Airborne Separation Display," *IEEE Trans. Human-Mach. Syst.*, vol. 43, no. 3, pp. 277–289, May 2013.

[13] M. R. Endsley, "Situation awareness global assessment technique (SAGAT)," in *Proc. Nat. Aerosp. Electron. Conf.*, 1998, pp. 789–795.

[14] FAA. (2011). FAA AC 120-55C - Air Carrier Operational Approval and use of TCAS II. [Online]. Available: http://www.faa.gov/documentLibrary/media/Advisory_Circular/AC%20120-55C.pdf

[15] S. Göknur, M. L. Bolton, and E. J. Bass, "Adding a motor control component to the operator function model expert system to investigate air traffic management concepts using simulation," presented at the IEEE Int. Conf. Syst., Man, Cybernet., Hague, The Netherlands, Oct. 2004.

[16] T. Weilkiens, *Systems Engineering with SysML/UML: Modeling, Analysis, Design.* Morgan Kaufmann, 2008.

[17] S. G. Hart and L. E. Staveland, "Development of NASA-TLX (Task Load Index): Results of empirical and theoretical research," in *Human Mental Workload*, P. A. Hancock and N. Meshkati, Eds. New York, NY, USA: Elsevier Science Publishers, 1988, pp. 77–106.

[18] D. Harel, "On visual formalisms," *ACM Commun.*, vol. 31, no. 5, pp. 514–530, 1988.

[19] J. G. Hollands and C. D. Wickens, *Engineering Psychology and Human Performance*. Englewood Cliffs, NJ, USA: Prentice Hall, 1999.

[20] J. K. Kuchar, "Methodology for alerting-system performance evaluation," *J. Guid., Control, Dyn.*, vol. 19, no. 2, pp. 438–444, 1996.

[21] K. Latorella and J. Chamberlain, "Tactical vs. strategic behavior: General aviation piloting in convective weather scenarios," presented at the Human Factors Ergon. Annu. Meet., Baltimore, MD, USA, 2002.

[22] D. Meister and G. F. Rabideau, *Human Factors Evaluation in System Development*. New York, NY, USA: John Wiley, 1965.

[23] J. Meyer and Y. Bitan, "Why better operators receive worse warnings?" *Human Factors*, vol. 44, no. 3, pp. 343–353, 2002.

[24] S. Mondoloni, M. T. Palmer, and D. J. Wing, "Development of a prototype airborne conflict detection and resolution simulation capability," presented at the AIAA Guidance, Navigation Control Conf., Monterey, CA, USA, 2002, paper AIAA-2002-4446.

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

[26] A. R. Pritchett, "Reviewing the safety contribution of cockpit alerting systems," *Human Factors Aerosp. Safety*, vol. 1, no. 1, 2001.

[27] A.R. Pritchett and J. R. Hansman, "Pilot non-conformance to alerting system commands," presented at the 9th Int. Symp. Aviation Psychol., Columbus, OH, USA, 1997.

[28] A. R. Pritchett and B. Vándor, "Designing situation displays to promote conformance to automatic alerts," in *Proc. 45th Annu. Meet. Human Factors, Ergonomics Soc.*, 2001, pp. 311–355.

[29] A. R. Pritchett, E. S. Fleming, J. J. Zoetrum, W. P. Cleveland, V. M. Popescu, and D. A. Thakkar, "Examining pilot compliance to

collision avoidance advisories," in *Advances in Human Aspects of Aviation*, S. J. Landry, Ed., Boca Raton, FL, USA: CRC Press, 2012, pp. 214–223.
[30] N. B. Sarter and D. D. Woods, "Pilot interaction with cockpit automation: Operational experiences with the flight management system," *Int. J. Aviation Psychol.*, vol. 2, no. 4, pp. 303–321, 1992.
[31] N. B. Sarter and D. D. Woods, "Pilot interaction with cockpit automation II: An experimental study of pilots' model and awareness of the flight management system," *Int. J. Aviation Psychol.*, vol. 4, no. 1, pp. 1–28, 1994.
[32] P. C. Shutte, "Definitions of tactical and strategic: An informal study," NASA Tech. Rep., NASA/TM-2004-213024, Nov. 2004.
[33] A. B. M. Skjerve and G. Skraaning, Jr., "The quality of human-automation co-operation in human-system interface for nuclear power plants," *Int. J. Human-Comput. Stud.*, vol. 61, pp. 649–677, 2004.
[34] R. L. Small and E. J. Bass, "Certify for success: A methodology for human-centered certification of advanced aviation systems," in *Human Factors in Certification*, J. A. Wise and V. D. Hopkin, Eds. Mahwah, NJ, USA: Lawrence Earlbaum Associates, 2000, pp. 139–150.
[35] R. D. Sorkin and D. D. Woods, "Systems with human monitors: A signal detection analysis," *Human-Comput. Interaction*, vol. 1, no. 1, pp. 49–75, 1985.
[36] S. V. M. V. Clari, R. C. J Ruigrok, B. W. M. Heesben, and J. Groeneweg, "Advanced aviation concepts via simulation: Research flight simulation of future autonomous aircraft operations," in *Proc. Winter Simul. Conf.*, 2002, pp. 1226–12234.
[37] E. L. Wiener and R. E. Curry, "Flight deck automation: Promises and Problems," *Ergonomics*, vol. 23, no. 10, pp. 995–1011, 1980.
[38] D. J. Wing, K. Krishnamurthy, R. Barhydt, and B. Barmore, "Pilot interactions in an over-constrained conflict scenario as studied in a piloted simulation of autonomous aircraft operations," presented at the 5th USA/Eur. Air Traffic Management R&D Semin., Budapest, Hungary, 2003.
[39] L. C. Yang and J. K. Kuchar, "Performance metric alerting: A new design approach for complex alerting problems," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 32, no. 1, pp. 123–134, Jan. 2002.
[40] Yourdon, Inc., *Yourdon Systems Methods.* Englewood Cliffs, NJ, USA: Yourdon Press, 1993.
[41] A. P. Sage, and W. B. Rouse, *Handbook of Systems Engineering and Management*. New York, NY, USA: John Wiley, 2011.
[42] D. M. Buede, *The Engineering Design of Systems: Methods and Models*. 2nd ed. (Wiley Series in Systems Engineering and Management), New York, NY, USA: John Wiley, 2009.
[43] A. R. Pritchett, and L. J. Yankosky, "Pilot-performed in-trail spacing and merging: An experimental study," *J. Guidance, Control, Dynamics*, vol. 26, no. 1, pp. 143–150, 2003.

Matthew L. Bolton (S'05–M'10) received the Ph.D. degree from the University of Virginia, VA, USA.

He is an Assistant Professor of industrial engineering with the Department of Mechanical and Industrial Engineering, University of Illinois at Chicago, Chicago, IL, USA. His research interests include the development of tools and techniques for using human performance modeling, task analysis, and formal methods to analyze, design, and evaluate complex, safety-critical systems.

Sinan Göknur received the B.S. and M.S degrees in systems engineering from the University of Virginia, Charlottesville, VA, USA, and the M.S. degree in computer science from the University of Minnesota, Minneapolis, MN, USA.

He joined Art, Art History and Visual Studies Department, Duke University, Durham, NC, USA in 2013 for a practice based Ph.D. in Media and Visual studies. His research areas include interactive digital art in public places, performance and social art practices.

Ellen J. Bass (M'98–SM'03) received the Ph.D. degree from the Georgia Institute of Technology, USA.

She is a Professor with the College of Computing and Informatics and the College of Nursing and Health Professions, Drexel University, Philadelphia, PA, USA. She has 30 years of industry and research experience in human-centered systems engineering in the domains of air transportation, meteorology, healthcare and informatics. Her research interests include to develop theories of human performance, quantitative modeling methodologies, and associated experimental designs that can be used to evaluate human–automation interaction in the context of total system performance.