Function allocation is the design decision in which work functions are assigned to all agents in a team, both human and automated. Building on the preceding companion papers’ review of the requirements of effective function allocation and discussion of a computational framework for modeling function allocation, in this paper, we develop specific metrics of function allocation that can be derived from such models as well as from observations in high-fidelity human-in-the-loop simulations or real operations. These metrics span eight issues with function allocation: (a) workload, (b) stability of the work environment, (c) mismatches between responsibility and authority, (d) incoherency in function allocations, (e) interruptive automation, (f) automation’s boundary conditions, (g) function allocations limiting human adaptation to context, (h) and mission performance. Some of the metrics measure distinct issues whereas others assess different causes of issues that can manifest in similar ways; collectively, they are intended to be comprehensive in their ability to discriminate for a range of issues. Trade-offs may exist between these metrics, and they need to be examined collectively to identify potential trade-offs or conflicts between them. This paper continues the example given in the preceding companion paper, demonstrating how these metrics of function allocation can be assessed from computational simulations of an air transport flight deck through the descent phase of flight.

Keywords: function allocation, metrics, simulation, cognitive systems engineering

INTRODUCTION

Function allocation is the design decision in which work functions are assigned to all agents in a team, both human and automated. Function allocation serves to divvy up between agents the taskwork required to act upon the collective work environment in such a manner to create and maintain the desired conditions for mission performance. In dividing up the work, function allocation also establishes the need for additional teamwork functions by which the agents interact or monitor each other.

Function allocation is an important and difficult aspect of designing teams, to which the additional concerns of human–automation interaction must also be integrated as machines assume the functions normally attributed to humans. Not only should function allocation be carefully considered at the earliest stages of design, but it is also often the only issue with human–automation interaction that can be addressed at the earliest design stages, that is, before the interface and machine logic have been established.

However, the lack of established measures of a proposed function allocation has led to musings that allocation is, and perhaps will forever be, an art (Sheridan, 1998). Current human factors descriptions of function allocation are generally too abstract or conceptual to guide specific design decisions. For example, desired attributes of automation include that it should be a “good team member” and “not clumsy” (with some exceptions; for example, see Pritchett, 2005, for a discussion of when the purpose of alerting systems is to be clumsy). Parasuraman, Sheridan, and Wickens (2000) examined allocating functions via a finer characterization of the automation’s capabilities referenced to a model of human information processing, providing several primary evaluative criteria (predictions of cognitive workload, situation awareness, complacency, and skill degradation) and secondary criteria (automation reliability and costs of action outcomes). Unfortunately, only general heuristics are used to evaluate the criteria, such as a general discussion of how automation can decrease or increase workload depending on circumstances, rather than a detailed analysis in
context. Therefore, this and other studies have called for further quantitative assessment (e.g., Pew & Mavor, 1998).

Such quantitative assessment must provide sufficient resolution to recognize the trade-offs that can occur between measures of good function allocation. For example, automating as many functions as possible may reduce the workload of human agents in the aggregate but may also have numerous other effects, such as increasing the monitoring work required by human supervisors of the automation, reducing the coherency of the “leftover” work allocated to the humans, and increasing the unpredictability of the humans’ work environment (Bailey, 1982; Bainbridge, 1983; Dekker & Woods, 2002; Miller & Parasuraman, 2007).

Thus, a reasonably detailed and comprehensive set of metrics of function allocation should be assessed during design. These metrics are intended to discriminate for a range of issues as identified in the literature and in operations. Some of the metrics measure distinct issues, whereas others assess different causes of issues that can manifest in similar ways and thus are not necessarily orthogonal, that is, they measure similar or overlapping effects; collectively, they are intended to be comprehensive in their ability to discriminate for a range of issues. Trade-offs may exist between these metrics, and they need to be examined collectively for trade-offs or conflicts between them.

In addition, when they may be collected during real operations, these measures could also provide a rigorous basis for dynamic function allocation. These measures may inform the automation to change its function allocation (the adaptive automation concept), or they may be presented to human team members or to supervisors (the adaptable automation concept or other forms of team flexibility, including changing the roles and distribution of functions between human team members; see Feigh, Dorneich, & Hayes, 2012, and Miller & Parasuraman, 2007, for a discussion). In such cases, methods would also be required to value the trade-offs between measures.

In this paper, we start by reviewing the requirements for effective function allocation detailed in the preceding companion papers and note issues in meeting these requirements as identified in the literature. From these requirements, metrics of function allocation are proposed to capture the issues. We then demonstrate how these metrics can be predicted during design with detailed simulations of the collective work of the human–automation team, using the arrival and approach phases of flight in an air transport flight flight deck as a case study. We conclude by discussing how these metrics can be considered collectively by designers of function allocation and how they may be applied to adaptive and adaptable automation.

REQUIREMENTS FOR EFFECTIVE FUNCTION ALLOCATION

Function allocation distributes work between agents, human and automated, within a team. Although there is no universally accepted method for determining function allocations, the following requirements for effective function allocation can be noted from first principles and the literature. In the following subsections, we outline the requirements proposed in the preceding companion paper, adapting the discussion here to issues with function allocation that warrant metrics during design.

Requirement 1: Each Agent Must Be Allocated Functions That It Is Capable of Performing

The metric for success for this requirement is a function allocation in which every agent has the skills to perform each of the functions assigned to him/her/it, viewing each function in isolation. A common method of addressing this requirement follows what Bailey (1982) has termed a comparison strategy for function allocation: Each function is individually compared to the capabilities of each agent and assigned to the most capable. In a very coarse sense, such a strategy is supported by assessments of what “Men Are Better At” to what “Machines Are Better At,” as articulated in the “MABA-MABA” list (Fitts, 1951). From this perspective, automation can serve to provide functions that a human cannot perform at all or with sufficient reliability and, in doing so, can change requirements for personnel selection, staffing levels, and training.
The preceding companion paper also discusses how this requirement is often assumed to be met if the automation can perform its required functions, implicitly applying a function allocation strategy termed by Bailey (1982) as *leftover allocation*. This strategy automates as many functions as technology will permit and assumes the human will pick up whichever functions are leftover. Functions that can be commonly automated are those that typify standard processes within conditions that are clearly predefined, such as nominal operations or established emergency shutdown procedures.

However, such automation-centered strategies for function allocation are prone to several issues that warrant metrics for prediction during design and operation. The first is the potential for *brittle* automation that can operate reliably only within a set of boundary conditions; when placed outside its boundary conditions, such automation appears to its operator to fail. Such situations imply that one (automated) member of the team may fail to perform adequately in certain situations; these situations typically are those off-nominal conditions in which the (human) team members need the most support (Norman, 1990). Unfortunately, such boundary conditions may not be portrayed to (or known by) the human operator, particularly when they represent rare or unusual conditions (Javaux, 1998). Thus, a prediction of whether the automation will be placed outside its boundary conditions is itself a valuable metric that implies potential concerns with the resilient performance of the team. This topic will be addressed later in this paper under Metric 6: Automation Boundary Conditions.

Likewise, human–automation function allocation often considers only the agents’ capability to assume authority for a function, unlike human–human function allocation that also examines the capability to assume responsibility. Specifically, whereas *authority* is generally used to describe who is assigned the execution of a function in operational sense, *responsibility* identifies who will be held accountable in an organizational and legal sense for the outcome. Except when automation is proven to provide safety in all foreseeable operating conditions, humans remain vested with the responsibility for the outcome of automation’s actions, a situation termed the “responsibility-authority double-bind” (Woods, 1985). If the human cannot knowledgably oversee the automation, he or she is forced to “trust” the automation. However, without a concrete basis for assessing whether the automation is correct, humans often over- and undertrust the automation (Parasuraman & Riley, 1997); either way, incorrect trust is viewed as human error, despite its basis in the function allocation. Thus, identification of mismatches between responsibility and authority is itself a valuable metric that implies potential concerns with trust, reliance, and monitoring. This topic will be addressed later in this paper under Metric 2: Mismatches Between Responsibility and Authority.

Although each of these issues—automation that can be placed beyond its boundary conditions, and mismatches between responsibility and authority—can result in several adverse outcomes, one manifestation is a requirement for humans to monitor automation (Bainbridge, 1983; Wiener & Curry, 1980), despite consistent findings that humans are ineffective at this task (Lee & Molloy, 1992; Molloy & Parasuraman, 1996). Indeed, the report authoring the MABA-MABA list suggested instead that “machines should monitor men” (Fitts, 1951). Thus, assessments of the taskload or workload of the human operator should consider the monitoring functions explicitly or implicitly assigned to them both as important, sometimes unpredicted, contributors to workload and as indicators of underlying concerns with responsibility and authority. In addition to the specific effects noted later in this paper under Metric 6: Automation Boundary Conditions and Metric 2: Mismatches in Responsibility and Authority, this monitoring workload will also contribute to Metric 1: Workload/Taskload.

**Requirement 2: Each Agent Must Be Capable of Performing Its Collective Set of Functions**

The metric for success for this requirement is whether each agent can perform his/her/its collective set of functions under realistic operating conditions. This emphasis on each agent’s collective set of functions underlies methods for
global function allocation and for understanding the agents as limited resources (see Dearden, Harrison, & Wright, 2000, for a review). One potential obstruction may be that the set of functions is too large, that is, the taskload demanded of the human agents corresponds to them experiencing excessive workload.

In regard to concerns with human workload in human–automation teams, automation is generally good at reducing the overall workload of human operators on average, often by reducing the manual control or execution functions that the humans must perform. This function has been touted as a contribution of automation and has been used to reduce staffing requirements, such as the reduction of flight crew in the flight deck from three (or more) to two. However, operational studies also note the unfortunate prevalence of cases in which automation creates a workload spike for a human team member (Bainbridge, 1983; Billings, 1997; Wiener, 1985). This issue can be correlated with the issue of off-nominal situations outside automation’s boundary conditions as noted in Requirement 1 but may also potentially occur in nominal conditions whereby multiple tasks are triggered close together, perhaps by the same circumstances. For example, highly automated autoflight systems can suddenly require a significant amount of programming from their human flight crew in response to a reroute commanded by an air traffic controller, even as the flight crew must also respond to the controller and execute other tasks, such as finding charts appropriate to the new routing.

The corollary of workload spikes is the problem of excessively low workload during “normal” operations in between the spikes. Low workload by itself corresponds to concerns with task engagement, boredom, and the human becoming out of the loop (Bainbridge, 1983; Endsley & Kiris, 1995). Further, this low workload often typifies switching from active engagement to passive activities, such as monitoring, discussed earlier as being not well suited to the human.

Estimates of the workload of any human agent in the team also must make sure to weigh properly the cognitive workload allocated to the human even as the manual workload is allocated to the machine (Bainbridge, 1983; Billings, 1997; Wiener, 1985). These cognitive functions can demand significant information gathering, judgment, detection, and decision-making activities, sometimes even up to the point of second-guessing the output of the automation. Estimates of the workload incurred by such cognitive functions need to account for myriad effects. One is the human’s expertise at the task and thus his or her ability to apply less-effortful strategies, rules, and skills while maintaining performance (Rasmussen, 1983). Also, nonlinear effects arise when several cognitive functions rely on awareness of the same information or on interdependent judgments of similar phenomena. In such cases, adding or removing one cognitive function may seem to have little impact, yet another function may implicitly require a significant, unique set of supporting activities and thus appear to require a disproportionate amount of effort.

Thus, prediction of the taskload placed on the human operators—or, when possible, workload experienced by the human operators—is a valuable metric of function allocations. To fully address known issues with workload corresponding to function allocation, such assessments must involve consideration of the full range of activities required, including underlying cognitive activities around information gathering and judgment, and requirements to monitor automation, in addition to explicit manual activities. Further, metrics of workload should involve consideration of not only aggregate or average workload but also workload spikes and periods of complacency. These concerns are discussed later in this paper under Metric 1: Workload/Taskload.

Interdependence between tasks further suggests that Requirement 2 can be better achieved when the function allocations establish coherent roles within the team. One attribute of a coherent function allocation can be viewed from the bottom up: Within each agent, functions share (and build upon) obvious, common constructs underlying all activities, such as a shared information and knowledge basis, and the allocation prevents conflicts between actions and resources between agents. Another attribute can be viewed from the top down: The functions collectively
contribute toward work goals in a manner that is not only apparent to the human but that can be purposefully coordinated and adapted in response to context. As noted earlier, however, with function allocation strategies that focus on automating as much as possible, the human’s allocation tends to be whatever is leftover, and the human must pick up any remaining functions throughout the work domain. When these leftovers form an incoherent set, the human can neither coordinate underlying information search and judgment activities nor purposefully adapt their functions toward improved performance toward work goals (Dekker & Woods, 2002). Therefore, the general problem of incoherent roles has its basis in a team design that does not allocate functions according to clearly specified roles. Thus, the coherence of the functions allocated to each human is itself intrinsically an important construct meriting metrics, in addition to its potential manifestation in metrics of other issues, such as taskload. This will be discussed in detail later in this paper under Metric 4: Coherency of a Function Allocation.

Requirement 3: The Function Allocation Must Be Realizable With Reasonable Teamwork

Each different function allocation of the same taskwork demands its own unique set of teamwork functions to coordinate the taskwork, including functions for human–automation interaction and for human–human coordination. The impact of this teamwork must then be considered from the perspective of the previous two concepts, that is, can each agent perform each of his/her/its teamwork activities in isolation, and can each agent perform his/her/its assigned set of both taskwork and teamwork functions? Thus, metrics of the taskload placed on the human operators should also incorporate the teamwork functions explicitly or implicitly required of human team members, which may include overt interactions or more passive monitoring activities. As such, the later discussion of Metric 1: Workload/Taskload will include these functions.

A team perspective on function allocation involves considering automation as a team member. However, automation does not have the same teamwork skills that humans naturally have. Of note, when automation is placed outside its boundary conditions, it cannot function properly, unlike a human team member, who will generally continue to attempt effective performance in unfamiliar circumstances. Thus, metrics assessing whether the automation will be placed outside its boundary conditions are required to address a critical aspect of human–automation teams that has not historically manifested as profoundly in teams of humans. This aspect of human–automation interaction will be discussed later in this paper in Metric 6: Automation Boundary Conditions.

Likewise, members of good teams are able to anticipate each other’s information needs and provide information at useful, noninterruptive times (Entin & Entin, 2001; Hollenbeck et al., 1995; Hutchins, 1995). The function allocation itself may demand interruptions when team members’ functions are interleaved, in a manner that may be useful or may be disruptive. Further, too often automation is “clumsy”: It unduly interrupts its human team members because, although humans can implicitly sense information about whether other team members would benefit from an interruption, automation historically cannot (Christoffersen & Woods, 2002). Thus, the potential for a function allocation to cause agents to interrupt each other is an important construct meriting a metric. In some cases, such as poorly timed output from automation, such interruptions may be unwarranted; in other cases, different function allocations may require agents to interrupt each other more or less depending on how their functions are allocated and, perhaps, interleaved. This topic will be discussed later in this paper in Metric 5: Interruptions.

Requirement 4: The Function Allocation Must Support the Dynamics of the Work

From an ecological view, work is an ongoing response to, and action upon, the work environment. This definition can be viewed from the perspective of each agent and of the team as a whole. The physical environment has dynamics that can drive taskwork; likewise, the teamwork actions of each agent modify the environment of other agents, creating a dynamic interplay. In the preceding companion paper, we discussed
how models of function allocation often require simulation or analysis of observed dynamics to identify situations in which, for example, the interleaving of functions assigned to disparate agents requires significant coordination or idling as one waits on another, or when workload may accumulate, or when one agent will be unduly interrupting another, or when executing prescribed procedures may conflict with other work demands, or when automation may be placed outside its boundary conditions. Metrics of these dynamic manifestations have been discussed in the preceding sections.

Further consideration of Requirement 4 identifies additional issues. First, resilience is fostered when a human agent may select strategies (courses of action) appropriate to the state of the environment and their own capabilities (Pritchett, 2010). For example, with his concept of cognitive control, Hollnagel describes how humans select their activities (and sequence them) in response to their competency and their perception of resources available to them (such as information availability) and demands on them (such as subjective available time) (Feigh & Pritchett, 2006; Hollnagel, 1993). Hollnagel suggested that cognitive control may be varied along a continuum spanning several cognitive control modes. The ability of each human in the team to adapt his or her cognitive control mode to immediate context has been found to reflect a good balance between the demands on the human and the resources available to him or her in terms of information, knowledge, and time available (Feigh & Pritchett, 2006). However, such adaptation can be constrained or eliminated by an overly prescribed (or proscribed) function allocation, particularly when human–automation interaction dictates a specific sequence of activities from the human (Feigh, 2011). The adverse effects of such overly prescribed function allocations have been found to manifest in work-arounds or disuse of automation (Feigh & Pritchett, 2010; Kirlik, 1993; Parasuraman & Riley, 1997). Thus, the ability to which a function allocation can accommodate a reasonable variety of human adaptations of cognitive control and/or strategies merits a metric, discussed later in this paper under Metric 8: Human Adaptation to Context.

A second issue involves recognizing that humans work to develop and maintain a stable work environment. If the work environment maintains a certain level of regularity, the human can predict its dynamics and tailor his or her activity to the needs in the environment (Hollnagel, 2004). This issue also corresponds to the construct of task management, by which human team members may choose to time and order their activities to prevent workload spikes, to conduct additional planning or safety-promoting behaviors during low-tempo activities, and to maintain options for future activities (Funk, 1991; Pritchett, 2010). However, environmental predictability decreases in the face of unexpected events, requiring the humans to spend more time reacting to events (Hollnagel, 2002). Further, when the human’s actions do not consistently create the desired outcomes, the need to revisit and revise these actions can be unpredictable. A function allocation may aggravate inherent environmental unpredictability by, for example, limiting human agents’ ability to view important aspects of the environment or by distributing functions in a way such that one agent will trigger the requirement for another to act. In addition, a trade-off exists when designing function allocations between maintaining predictability versus dynamically allocating functions (Miller & Parasuraman, 2007). Thus, humans’ ability to predict their activities has intrinsic value meriting a metric; although some unpredictability may be inherent to the work environment, a function allocation may further limit agents’ ability to predict and schedule their own activities. This topic will be detailed later in this paper under Metric 4: Stability of the Human’s Work Environment.

**Requirement 5: The Function Allocation Should Be the Result of Deliberate Design Decisions**

In the preceding companion paper, we proposed that function allocation can and should be sensibly integrated into the broader engineering design process. In many cases, initial specifications of function allocation arise from changes in the high-level concept of operation. Such changes may be incremental and constrained by current-day technologies, procedures, personnel,
and/or policies; in other cases, such changes in concepts of operation may represent significant innovations in which constructs such as common work practices and relationships between tasks and tools must be significantly altered. This is particularly the case if metrics are available by which designers can simultaneously consider the economic and safety metrics by which the total system will be evaluated, the potential contributions of (or constraints on) technology and human performance, and the regulatory, policy, and procedural considerations in allowing access to—and defining interaction within—the collaborative functioning of the system. Thus, metrics of function allocation should not only involve consideration of aspects of each agent’s experience of his/her/its allocated work but also simultaneously predict metrics of cost and performance resulting from the combined efforts of the human–automated team. This topic will be discussed later in this paper under Metric 7: System Cost and Performance.

**METRICS OF FUNCTION ALLOCATION**

The requirements of a good function allocation contain several observable aspects warranting metrics. In total, these metrics fall into eight categories: (a) workload/taskload arising from all sources, (b) mismatches between responsibility and authority, (c) stability of the humans’ work environment, (d) coherency of the function allocation, (e) interruptions, (f) automation boundary conditions, (g) system cost and performance, and (h) humans’ ability to adapt to context. In each of the following subsections, we discuss metrics of these constructs in more detail, as they may be identified from detailed models of work, from computational fast-time simulations, and from assessments in simulated or actual operations.

**Metric 1: Workload/Taskload**

Work models and simulation of work, such as those described in the preceding companion paper, can serve to identify the actions that each human in a team will be required to perform. The type of taskload should be further detailed by examining its components: *taskwork* on the work environment contributing directly to mission performance, *teamwork* due to interaction with the automation, and *teamwork* due to monitoring demands (whether created explicitly or created implicitly by a mismatch between responsibility and authority). Total taskload, that is, all the actions demanded of an agent given a specific function allocation, is the sum of these components.

Of the requirements for effective function allocation noted in the previous section, the primary concern is Requirement 2, that is, that each (human) agent must be capable of performing his or her collective set of functions, taken together. In addition, important aspects of this workload—its origin and the type of workload, such monitoring or cognitive workload versus manual or physical workload—relate to additional requirements. For example, as discussed earlier under Requirement 1, if the automation can perform its functions only within a limited set of boundary conditions or otherwise might fail, and if there is a mismatch in which the human remains responsible for functions for which the automation has authority, then the human will have additional monitoring actions that are useful to delineate as a contributor to workload. Likewise, as discussed under Requirement 3, teamwork is another contributor to workload worth inclusion in workload metrics and its own delineation.

Static analysis of potential taskload just counts the total number of tasks potentially required of each agent in a given function allocation. This analysis provides a quick assessment to identify gross concerns with a function allocation and can highlight requirements for training and personnel selection in terms of the functions expected of each agent. Because these assessments of taskload originate from detailed work models, they can provide a quick baseline inclusive of the nonlinear additions of load resulting from different function allocations.

As noted earlier, however, taskload can be an emergent construct that is created by the evolving demands of the team’s work environment; an air traffic controller’s taskload, for example, is a product of how many aircraft are in his or her sector at any time. Thus, the dynamics of taskload are important. Early in design, taskload can be predicted by computational simulations.
of the team’s taskwork and teamwork, particularly when situated in a realistic model of the operating environment and applied to a large number of potential operating scenarios. Once a system is designed, real-time observations of human-in-the-loop simulation trials and/or actual observations can further refine estimates of this measure.

Key metrics of taskload include counting occurrences in which immediate workload or taskload exceeds some threshold, defining brief workload spikes or longer-duration periods of workload saturation. Similarly, dynamic simulation may also involve examining for situations reasonably hypothesized to lead to concerns with vigilance, complacency, and low engagement: Task engagement may be estimated by counting or timing situations in which the overall workload drops below a set minimum threshold, and concerns with vigilance may be assessed by identifying situations in which the human’s taskload comprises only monitoring actions.

Taskload is a fairly crude measure; a count of actions at any one time assumes that each has the same impact on the human executing them. To be more precise, workload can be defined as an intervening variable that indicates the relationship between this taskload and the capability of the human executing the actions (Kantowitz, 2000). In real operations or human-in-the-loop experiments, a number of methods may be used to assess the workload associated with a given function allocation. These methods include subjective ratings in multiple dimensions, such as measuring via psychophysical scaling (see, for example, the method applied by Dixon, Wickens, & Chang, 2005), multidimensional rating systems (Hart & Staveland, 1998; Potter & Bressler, 1989), and a combination of each to compensate and, sometimes, to strengthen effectiveness of the resulting measurements (Cegarra & Chevalier, 2008).

In both static work models and computational simulations, each action can be annotated by the workload it imposes. A univariate representation of workload requirements, although it faces the difficulty of validating each action’s workload requirements. Thus, the sum of workload requirements of all actions being executed by an agent at any point in time can predict workload throughout time and then be examined for constructs including workload saturation, workload spikes, and low-workload intervals promoting complacency.

**Metric 2: Mismatches Between Responsibility and Authority**

In a team of multiple agents working together, the responsibility for the outcome of a function must be considered relative to the authority to perform it. This issue was noted earlier, in the discussion of Requirement 1. Several problems can intrinsically arise when the responsibility and authority are not assigned to the same agent, including the additional monitoring actions the responsible agent must perform. Therefore, the metric of mismatches between responsibility and authority can be quantified by work models by the number of functions with mismatches between responsibility and authority (a static measure of a given function allocation). Further, an important distinction of the monitoring caused by such a mismatch suggests a finer-grained metric identifying the additional taskload/workload induced by the monitoring associated with these mismatches (a static measure of the different monitoring tasks a human may be required to perform). Further, through computational simulations of work models, or observations in real-time human-in-the-loop simulations or actual operations, the number and combined duration of the induced monitoring actions provide a dynamic measure of the effect of mismatches between responsibility and authority.

**Metric 3: Stability of the Human’s Work Environment**

Underlying concerns with the stability of the human’s work environment, as described earlier in Requirement 4, include the extent to which the function allocation allows human team members to predict (and potentially plan for) upcoming actions. Issues with stability may then be assessed by examining instances
of unpredictability stemming from changes in the actions required of a human due to several factors: changes in the team design (including dynamic, unpredicted changes in function allocation), the automation or other teammates unexpectedly requiring new actions of the human, or exogenous inputs to the work domain.

This unpredictability can manifest anywhere between the extremes of perfect predictability (which occurs when the agent has perfect control over when an action will be completed) to completely unpredicted (in which the agent can have acquired no previous knowledge of what to do). We propose metrics of unpredictability that count instances when the human cannot predict when or if he or she will need to act, limiting the ability to proactively manage and prepare for upcoming tasks. At the most extreme, humans may be asked to perform actions that they cannot predict with certainty will happen at all (which we will define as Type 1 unpredictability). For example, during the arrival and approach phases of flight, flight crew may be asked to quickly respond to an unanticipated reroute issued by an air traffic controller. Such occurrences may be further categorized by their severity relative to the operator’s ability to identify the correct response to the situation; regardless, all occurrences of Type 1 unpredictability represent situations unaccounted for in specific plans for that day’s operation.

As an intermediate level of predictability, human workers may expect some actions will be required at some time but cannot exactly predict or determine exactly when (which we will define as Type 2 unpredictability). For example, during an arrival, the flight crew can predict that the autoflight system may require (or establish the conditions for) a change in the aircraft configuration (speedbrakes, flaps, and gear) but must wait for some indication to implement these changes in aircraft configuration. Type 2 unpredictability occurrences may or may not come at inopportune times; either way, their operational impact is that they require human workers to be reactive and, thus, limit their ability to actively manage their work activities.

Thus, a metric of unpredictability involves examining those actions that the human team members could not predict. These may be identified through observations in simulated human-in-the-loop studies or actual operations. Additionally, these can also be identified in work models by examining the triggers of the actions the human will perform, such as the following: (a) Those actions that the human can choose to initiate or predict well when they will be required can be judged as predictable, (b) those that the human can predict needing to perform but with timing controlled by exogenous agents or circumstances can be described as Type 2 unpredictable, and (c) those that the human would not anticipate needing to perform in nominal operations (i.e., reflecting unexpected events) can be described as Type 1 unpredictable. The exact metric applied may be a count of the these actions, a comparison of the relative proportion of the human’s activity that each composing, or potentially a more detailed assessment of their impact.

**Metric 4: Coherency of a Function Allocation**

Incoherent function allocations do not establish clear roles and efficient work practices for all team members and may lead to interdependent or conflicting activities between agents. Such incoherency was discussed earlier in Requirement 2 as an obstacle to each agent’s being able to perform his/her/its collective set of assigned functions. As a bottom-up metric, the coherency of a function allocation can be assessed dynamically during simulations or observations of real operations by counting resource conflicts whereby two agents both try to set the values of the same resources within the environment. Such situations highlight where agents’ function allocation may overlap or require detailed coordination.

Additionally, this incoherence is inherently an aspect of the function allocation and team design that can be analyzed by top-down static measures examining the distribution of functions by function allocation across an abstraction hierarchy of the work, informed when possible by interviews, surveys, and observations. The abstraction hierarchy uses a means–ends decomposition to describe the functions needed to characterize a work domain (Bisantz & Roth, 2003).
Thus, the static measure **coherency level** quantifies the coherency of a given function allocation by measuring how many levels up the abstraction hierarchy model one can find a full description of the functions assigned to one agent. Although this metric provides a general indication of coherency, it is fairly coarse. Therefore, an additional metric, the **coherency percentage**, can also be measured as the percentage of the functions in the work model that are judged to be assigned entirely to only one agent (human or automation) compared to the total number of functions required to describe all the work conducted by the team. Thus, having functions within a work model that are clearly allocated to one agent is seen as increasing coherency. This aspect mirrors the one possible strategy for team design noted by Gualtieri, Roth, and Eggleston (2000) valuing the integrating of related goals within individual team members. This strategy has the benefit of allocating internally consistent tasks and minimizing the extent that tasks are interleaved between team members. Of course, this assessment of coherency levels and coherency percentages depends on the structure of the work model; the absolute values represent as much the modeler’s decisions in forming the abstraction hierarchy as the function allocation. However, as long as the model is based on work-relevant means–end relationships, relative values of this metric provide a systematic basis to examine for obvious effects that break up an agent’s work in a manner that cannot be sensibly abstracted.

**Metric 5: Interruptions**

As noted earlier in the teamwork discussion in Requirement 3, interruptions are generally disruptive to human work. Some interruptions may be necessary, particularly when unexpected situations suddenly arise and require immediate information sharing or response. However, function allocations should also be assessed for whether they divide functions between agents such that they create the need for interruptions. Further, automation lacks the ability of a human team member to assess whether an interruption is warranted relative the other dynamics of a situation.

Thus, a metric of interruptions is the count of interruptions experienced by each human agent, as indicated by the issuing of information presented with sufficient salience to interrupt immediate perception and attention, regardless of whether the recipient will subsequently decide to act on it. This metric is inherently dynamic, as it is driven by the timing of the “interrupter” relative to the “interruptee’s” other activities at that instant. This measure can be recorded by observations of apparent interruptions during real-time simulations or real operations or counted in computational fast-time simulations of work. When possible, expert interpretation of each event may identify whether the interruption was desired and/or warranted and whether the interruption was created by the function allocation. Further, the impact of the interruption can be assessed qualitatively or by quantitatively measuring its impact on performance.

**Metric 6: Automation Boundary Conditions**

The earlier discussion of Requirement 1 noted that boundary conditions represent a limit on automation’s ability to perform any of its assigned functions. One metric of automation boundary conditions recognizes when the immediate situation violates the fixed set of boundary conditions in which the automation is operable and thus is appropriate to use. One specific metric may count the instances (or durations) in which the automation is placed outside of its boundary conditions and assesses their impact. However, this metric requires that the boundary conditions are known and any exceedance can be measured. In the case of complex automation, boundary conditions are not always publicly documented and may, besides, represent multivariate circumstances that are difficult to recognize. Similarly, this metric alone does not capture the result of exceeding these boundary conditions.
Thus, another metric identifies when automation must have been placed outside its boundary conditions by noting when automation does not achieve its targets while operating according to its specification. This metric thus identifies situations in which automation can fail to provide expected outcomes, which provides a more direct link to concerns with resilience and system performance.

**Metric 7: System Cost and Performance**

Ultimately, as discussed in Requirement 5, the cost and performance of a function allocation should be considered within the design process. The team’s operating costs and collective performance can be measured via simulations or assessed in actual operations. The definition of operating cost and performance is dependent on the domain and the team’s objectives. In some cases, performance includes measures of the safety and robustness in off-nominal scenarios. Likewise, costs can often be assessed in terms of resources required, such as the fuel burn associated with a flight.

In computational simulations of models of function allocation, the choice of agent model provides an interesting distinction in the assessment of performance. Although intricate agent models can be used to predict the impact of human performance on ultimate mission performance, predictions of low system performance are often blamed on the human workers. Thus, there is also benefit in simulating simple agent models that execute all their work immediately and perfectly to capture the extent to which a function allocation inherently can (or cannot) meet targets for cost and performance.

**Metric 8: Human’s Ability to Adapt to Context**

In the earlier discussion of Requirement 4, we noted the need for a function allocation to support the dynamics of work. This metric specifically involves examining for situations in which a function allocation overspecifies any human team member’s behavior such that he or she cannot adapt to context. In real-time operations or human-in-the-loop simulations, such situations may be observed when the human’s behavior does not meet that expected by the function allocation—or where he or she effectively changes the function allocation by behaviors, such as refusing to use an automated capability because it is not consonant with his or her immediate needs.

Further, adaptation of context can also take the form of cognitive control modes. Observations can categorize human team members’ behavior according to observable markers of strategic behavior (in which all necessary monitoring behaviors are included, and actions are predicted and planned), tactical behavior (in which periodic monitoring checks are conducted), or opportunistic behavior (in which monitoring is conducted only when necessary to complete assigned taskwork) (Feigh, 2008; Feigh & Pritchett, 2010).

Cognitive control modes can also be examined in computational fast-time simulations of work. For example, in the preceding companion paper, we detail how strategic behavior can be modeled as containing substantial monitoring and planning actions, each scheduled according to expectations of when the actions will be warranted within the dynamics of the work environment. Likewise, tactical behavior can be modeled as containing more procedural or rule-based monitoring actions that are scheduled periodically, and opportunistic behaviors contain the minimum monitoring behaviors required to perform taskwork. Thus, these behaviors can interact strongly with assumptions about any human agent’s monitoring activities.

**EXAMPLE: MEASURING FUNCTION ALLOCATION IN THE FLIGHT DECK**

**Representative Function Allocations**

To illustrate a complex work domain where function allocation is a crucial concern, consider the flight deck of an air transport aircraft during descent. A more complete description is provided in the preceding companion paper, which uses the same example to illustrate how function allocation can be described within work models and fast-time simulations of the team’s collective taskwork and teamwork.

Some of flight deck functions, at this time, can be performed only by the team of two flight crew members, who collectively perform closely...
coupled tasks. Functions that must be performed by the flight crew include executing the checklists and otherwise ensuring proper systems management (e.g., configuration of gear, flaps, speedbrakes, landing lights, “fasten seatbelt” cabin light, and wheel braking settings for landing, and status checks of the electrical, hydraulic, propulsive, and control actuator systems) and communication management with the air traffic controller (although some communications, such as receipt of commanded trajectories, may be automated in the foreseeable future).

In contrast, automation for the aircraft control function, in the form of an autopilot, is a fairly mature and common technology. The aircraft control function includes two elements: (a) determining actuator settings (the engine and control surfaces of the aircraft) to achieve targets for heading, airspeed/thrust, and altitude/vertical speed and (b) moving the actuators appropriately. This function is closely tied to the flight crew’s systems management function. For example, the system management function can safely extend the flaps only within a particular speed range as established by the aircraft control function, and then the aircraft control function must respond to the changes in aircraft aerodynamics created by the flap extension.

The trajectory management function defines the heading, altitude, and speed targets that will cause the aircraft to fly to assigned waypoints while meeting altitude and speed restrictions and other air traffic controller instructions. Flight crew historically performed this function with reference to aviation charts and displays of tracking to or from ground-based navigation displays, and more recently, the flight deck provides moving map navigation displays. The automated autoflight system can now also guide the aircraft along a specified trajectory.

In summary, several high-level functions are part of the common parlance of the flight deck, including systems management, communication management, aircraft control, and trajectory management. However, function allocations cannot be simply defined by these high-level functions because of other operational considerations. For example, even when the autoflight system is executing the trajectory management function and the autopilot is executing the aircraft control function, every altitude clearance must be entered by the flight crew as an altitude target in the Mode Control Panel (MCP). This altitude target then serves as a visible reminder to the flight crew of the latest air traffic control clearance and as a constraint on the autoflight system’s vertical trajectory. As a result, broad-brush descriptions of function allocation, such as the levels of automation construct, cannot describe the interleaving between these functions.

In addition, the flight crew maintain responsibility for the safety of flight, even as the autopilot and autoflight systems may exercise significant authority over important actions within the aircraft control and trajectory management functions, suggesting an issue with mismatches between responsibility and authority. Flight crew’s responses to such situations can demand significant adaptations to behavior between nominal flight—characterized by low tempo, automation within boundary conditions, and significant look-ahead and planning—and off-nominal situations, whereby the pilot must quickly respond to the unexpected while also monitoring whether automation is appropriate to the situation. As such, the functions actually undertaken by the flight crew can vary significantly not only due to an official designation of function allocation (such as engaging the autoflight system) but also due to contexts ranging from time-stressed situations to those in which the flight crew has time to monitor the automation.

Thus, our models of different function allocations in the flight deck have been sufficiently detailed to capture the specific activities assigned to—or reasonably expected of—the flight crew and automation. In the following demonstration of metrics, we will examine four function allocations:

- **FA1**: The most-automated function allocation, in which the automation is allocated authority for receiving air traffic control communications dictating the route of the aircraft, for trajectory management, and for aircraft control. The flight crew are allocated the functions for systems management and for interacting with the autoflight system on communication management to confirm that air traffic control instructions are feasible and that they are properly followed by the autoflight system. This function allocation is not currently implemented in air transport aircraft, although it is technically feasible.
• FA2: The most highly automated function allocation currently implemented, in which the automation is allocated authority for trajectory management and for aircraft control. The flight crew are allocated the functions for communication management and systems management and must interact with the autoflight system to describe any changes to the planned trajectory.

• FA3: A mixed function allocation, in which the automation is allocated authority for managing only the lateral component of the trajectory and for aircraft control. The flight crew are allocated the functions for communication management and for managing the vertical component of the trajectory by determining the autopilot targets for airspeed and vertical path. Further, the flight crew must enter changes to the planned lateral trajectory to the autoflight system.

• FA4: The least-automated function allocation, in which the automation is allocated authority only for aircraft control. The flight crew are allocated the functions for communication management and for system management. Further, the flight crew are allocated the trajectory management function, which they execute by entering into the autopilot the targets for heading/course, airspeed, and vertical path.

In all cases, the pilot remains responsible for all aspects of the flight including meeting all restrictions on allowable airspeed and altitude as the aircraft crosses waypoints. This assignment of responsibility can require substantial monitoring actions.

Four scenarios were examined in dynamic simulations of these function allocations: nominal, late descent, unpredicted rerouting, and unexpected tail wind. These scenarios were purposefully designed to exercise the different function allocations.

• In the nominal scenario, the flight crew could predict which actions would be required of them (although they could not always control when these actions would be required), and the automation was never placed outside its boundary conditions.

• In the late descent scenario, the clearance to start the descent was given late by the air traffic controller. The duration by which this clearance was late was set at three different levels, which established conditions by which the aircraft could not meet the altitude crossing restrictions on the next one, two, or three waypoints, respectively. This scenario requires intervention by the flight crew over the trajectory management function being performed by the autoflight system in FA1, FA2, and FA3 and requires communication with the air traffic controller.

• In the unpredicted rerouting scenario, an unpredicted air traffic control clearance changed the required trajectory from that entered into the autoflight system. Three variants were examined within this scenario, each representing a reroute causing progressively larger deviations from the planned route. With FA1, FA2, and FA3, the flight crew needed to enter the revised trajectory into the autoflight system, and with FA3 and FA4, the flight crew needed to identify the airspeed and vertical profile (and, in FA4, heading/course) that would achieve this new trajectory.

• In the unexpected tailwind scenario, also with three levels representing three severities of tailwind, the planned vertical profile could not be maintained. The aircraft deviated above the planned profile, requiring changes in how the trajectory was managed by the flight crew or autoflight system and, when altitude restrictions could not be met, communications with the air traffic controller.

In the following subsections, we describe how the metrics of function allocation were assessed from computational simulations. Each simulation predicted one descent as just described from before top of descent to just before landing flare. A detailed work model was used by the simulation as described in the preceding companion paper, which represented both the actions of the flight crew (represented as taskwork, interactions with the autoflight system, and monitoring actions). The actions of the autoflight system and aircraft were created from detailed 6-degree-of-freedom models of the aircraft dynamic. The complete set of runs comprised all 120 combinations of the four function allocations, 10 scenario descriptions (one nominal and three levels within each of the three off-nominal scenarios), and three flight crew behaviors modeling strategic, tactical, and opportunistic contextual control modes.
A dynamic estimate of taskload was assessed as the number of actions and their combined duration, shown in Figures 1 and 2, respectively, for those cases in which the flight crew was modeled in the strategic contextual control mode. Consider FA3, the mixed function allocation, in which the flight crew manages the vertical profile while the flight deck automation manages the lateral profile. This function allocation has the interaction demands of FA4 and the monitoring demands of FA2. Therefore, the flight crew experienced the highest taskload in FA3. A one-way ANOVA for a fixed effect due to function allocation revealed that the number of actions varied significantly \((p = .010)\), as did the combined duration of taskload \((p = .042)\).

Post hoc comparisons using the Tukey HSD test indicated that the number of flight crew actions with FA1 and FA2 were significantly lower than with FA3 \((p < .05)\) but did not differ significantly from FA4.

This taskload may also be broken down further according to the type of activity it represents: taskwork, flight crew interaction with the autoflight systems, and monitoring. For example, with FA2 and FA3, the reroute scenario required the flight crew to enter the new route into the autoflight system (in FA1, it was communicated directly to the autoflight system by air traffic control, and the flight crew was expected to read and accept/reject it), but the corresponding increase in the interaction component of taskload is small and is offset by this scenario’s new, more-direct routing passing
through fewer waypoints and, thus, having fewer of the taskwork and monitoring actions associated with waypoint passage.

Further, the results in Figures 1 and 2 represent the taskload of flight crew in the strategic contextual control mode. As noted earlier, this mode is typified by extensive planning and monitoring activities, as is evident in these results. On the other hand, these results also portray the extent to which the most-automated function allocations, FA1 and FA2, leave the flight crew with almost exclusively monitoring work, which may lead to concerns with complacency and lack of engagement.

Conversely, the trace through time can also be examined for instances in which the number of actions concurrently demanded of the flight crew surpasses a criterion representing hypothetical limits on the flight crew’s capability. In this case, this criterion was set to identify when three or more actions were required simultaneously. Figure 3 shows both the count of these instances of saturation, representing number of workload spikes, and their average duration; when data exist for more sophisticated criteria for saturation, they may be applied. With the criterion used here, even the nominal scenario had a notable number of workload spikes and duration of saturation. Each scenario contributes its own dynamic. For example, the late descent scenario required extra responses only around the time of top of descent, whereas the reroute scenario bypassed several waypoints and the flight crew actions associated with each. Further, whereas the number of workload spikes was not statistically different between function allocations, the duration of saturation was noticeably higher with the less-automated FA3 and FA4 because of the greater interaction with the autoflight systems at the time of saturation, requiring activities that had a longer duration than the monitoring of the autoflight system required with the more-automated function allocations.

Metric 2: Mismatches between Responsibility and Authority

The metric of mismatches between responsibility and authority can be quantified conceptually by the number of functions with mismatches between responsibility and authority. In this example, if we assume that the autoflight system is certified (and thus essentially responsible) only for the functions allocated to the automation in FA4 (controlling heading, airspeed, and vertical speed), then the measure of mismatch for FA4 is zero. As more functions are allocated in FA3, FA2, and ultimately, in FA1, the number of mismatched functions increases, as illustrated in Tables 1, 2, and 3. These mismatches can be counted on the tables as a static measure.

Additionally, these mismatches require the flight crew to monitor the autoflight system. Thus, the number and combined duration of the induced monitoring actions as observed in actual
**TABLE 1:** Functions Within the Most-Automated Function Allocation, FA1, for Which the Automation Has Authority but the Human Flight Crew Has Responsibility, and Constituent Actions Required of Each Given This Mismatch

<table>
<thead>
<tr>
<th>Function</th>
<th>Flight Crew’s Actions</th>
<th>Automation’s Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Control vertical profile</td>
<td>Modify CDU pages</td>
<td>Manage waypoint progress</td>
</tr>
<tr>
<td></td>
<td>Reduce airspeed for late descent</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Confirm target altitude</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Confirm target speed</td>
<td></td>
</tr>
<tr>
<td>Control waypoints</td>
<td>Modify CDU pages</td>
<td>Calculate dist current waypoint</td>
</tr>
<tr>
<td></td>
<td>Monitor waypoint progress</td>
<td>Evaluate flight phase</td>
</tr>
<tr>
<td></td>
<td>Monitor dist active waypoint</td>
<td>Manage waypoint progress</td>
</tr>
<tr>
<td></td>
<td>Confirm active waypoint</td>
<td>Direct to waypoint</td>
</tr>
<tr>
<td>Control communication with air</td>
<td>Respond to hand off</td>
<td>Receive altitude clearance</td>
</tr>
<tr>
<td>traffic control</td>
<td>Confirm data communication</td>
<td>Receive ILS clearance</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Receive waypoint clearance</td>
</tr>
</tbody>
</table>

Note. CDU = Control Display Unit; dist = distance; ILS = Instrument Landing System.

**TABLE 2:** Functions Within the Function Allocation FA2 for Which the Automation Has Authority but the Human Flight Crew Has Responsibility, and Constituent Actions Required of Each Given This Mismatch

<table>
<thead>
<tr>
<th>Function</th>
<th>Flight Crew’s Actions</th>
<th>Automation’s Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Control vertical profile</td>
<td>Modify CDU pages</td>
<td>Manage waypoint progress</td>
</tr>
<tr>
<td></td>
<td>Reduce airspeed for late descent</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Confirm target altitude</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Confirm target speed</td>
<td></td>
</tr>
<tr>
<td>Control waypoints</td>
<td>Modify CDU pages</td>
<td>Calculate dist current waypoint</td>
</tr>
<tr>
<td></td>
<td>Monitor waypoint progress</td>
<td>Evaluate flight phase</td>
</tr>
<tr>
<td></td>
<td>Monitor dist active waypoint</td>
<td>Manage waypoint progress</td>
</tr>
<tr>
<td></td>
<td>Confirm active waypoint</td>
<td>Direct to waypoint</td>
</tr>
</tbody>
</table>

Note. CDU = Control Display Unit; dist = distance.

**TABLE 3:** Function Within the Mixed Function Allocation, FA3, for Which the Automation Has Authority but the Human Flight Crew Has Responsibility, and Constituent Actions Required of Each Given This Mismatch

<table>
<thead>
<tr>
<th>Function</th>
<th>Flight Crew’s Actions</th>
<th>Automation’s Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Control waypoints</td>
<td>Manage waypoint progress</td>
<td>Calculate dist current waypoint</td>
</tr>
<tr>
<td></td>
<td>Monitor waypoint progress</td>
<td>Evaluate flight phase</td>
</tr>
<tr>
<td></td>
<td>Monitor dist active waypoint</td>
<td>Direct to waypoint</td>
</tr>
<tr>
<td></td>
<td>Confirm waypoint target</td>
<td></td>
</tr>
</tbody>
</table>

Note. dist = distance.
operations, in real-time human-in-the-loop simulations, or in fast-time computational simulation provide a dynamic measure of an important manifestation of mismatches between responsibility and authority. For example, the number of times the flight crew are called to perform these actions in computational simulations of each of the scenarios in the arrival and approach simulations are shown in Figure 4. Mirroring the static measure of mismatch inferred from Tables 1, 2, and 3, more mismatch-induced monitoring actions were demanded by the highly automated function allocations, FA1 and FA2.

**Metric 3: Stability of the Human’s Work Environment**

Underlying concerns with the stability of the human’s work environment, as described earlier, includes the extent to which the function allocation allows human team members to predict (and potentially plan for, or plan around) upcoming actions. Thus, the metric of the stability of each human’s work environment are counts of unpredicted actions of both Type 1 and Type 2. Figure 5 illustrates the average percentage of the flight crew’s actions that were unpredicted. Some aspects vary with scenario: Type 1 unpredictability was found only in the unpredicted reroute scenario. The more manual function allocations, FA3 and FA4, provide lower unpredictability levels compared to the more highly automated function allocations, FA1 and FA2, mirroring the trend between the human’s predictability of the environment and degree of automation suggested by Miller and Parasuraman (2007). However, some measure
of unpredictability may be inherent in the work: Even in the most manual function allocation, FA4, and the nominal scenario, roughly 3% of the pilot’s actions in the model have Type 2 unpredictability, that is, their timing cannot be exactly determined by the flight crew.

**Metric 4: Coherency of a Function Allocation**

The coherency of a function allocation involves examining how the work is distributed between agents by looking at static representations of the work, such as those provided by the abstraction hierarchy used in cognitive work analysis. Between Figures 6 and 7, the allocation of function varies, but underlying task-work is the same (the figures contain the same functions). One metric of coherency may be expressed by counting up from the lowest level to the highest level, providing a full description of the functions assigned to one agent. With the highly automated FA1 shown in Figure 6, the flight crew and automation are each assigned a smattering of functions across the work domain, and thus, the coherency for either agent can be expressed only as Level 2. In contrast, in the mostly manual FA4 shown in Figure 7, the flight crew are given the entire set of functions for the priorities and values functions “maintain flight rules and regulations” and “maintain interaction with air traffic systems,” and thus, the flight crew’s coherency can be expressed as Level 3. This metric, however, is fairly coarse: Three function allocations (FA1, FA2, and FA3) are all assessed at Level 2, even as they assign different low-level functions to the automation.

Using a finer-grained metric, the coherency percentage, one assumes that having more functions clearly allocated to individual agents reflects greater coherency. For example, the function allocation FA1 shown in Figure 6 has a coherency percentage of 65% (13 functions out of 20 are each assigned to one agent). In FA2, the communication management generalized function (and all its contributing lower-level functions) are assigned to the flight crew, increasing the coherence percentage to 70%, whereas in FA3, the coherence percentage is reduced to 60% because the trajectory management generalized function is distributed between automation and flight crew. The coherency percentage of FA4 increases to 85% as entire Level 3 priorities and values functions are allocated to the flight crew.
Metric 5: Interruptions

As noted earlier, interruptions are generally disruptive to human work. A metric of interruptions is the count of interruptions experienced by each human agent, as recorded by observations of apparent interruptions observed during real-time simulations or real operations, or counted in computational fast-time simulations of work.

The average number of interruptions of the flight crew during descent is shown in Figure 8 by scenario and function allocation. The number of interruptions with the less-automated function allocations, FA3 and FA4, is more sensitive to the different scenarios than is that with the more-automated function allocations. This interaction is due in large part to the three types of interruptions captured in the simulated work model:

- First, if the autopilot’s altitude target is not lower than the cruise altitude upon reaching 10 nm of the automation’s estimate of the optimal top of descent point, the automation displays “RESET MCP ALTITUDE.” This interruption was triggered only in the late descent scenario and only with the more highly automated function allocations, FA1 and FA2.
- Second, an altitude alert sounds when the aircraft reaches within 1,000 ft of the autopilot altitude target. This interruption is, therefore, given once per entry of a new altitude target and thus reflects how often each scenario requires altitude changes (the reroute scenario bypasses many waypoints and has thus has fewer altitude constraints) and more-manual function allocations, in which the flight crew must manually enter each new altitude target.
- Third, if the airspeed increases to 10 knots higher than the planned descent airspeed, the automation displays “DRAG REQUIRED” to the flight crew. This interruption is therefore a reflection of the speed tracking achieved by the function allocation and whether the scenario inherently requires a steeper descent due to a late descent clearance or tailwind.

Metric 6: Automation Boundary Conditions

Air traffic procedures commonly establish a nominal descent trajectory, including a vertical profile, that is entered into the autoflight system. However, there are environmental conditions purposefully created by the scenarios examined here—the unexpected tailwind and the late descent clearance—in which it is physically
difficult to meet the vertical profiles. This difficulty can be exacerbated by the function allocation, both in terms of how the aircraft is flown and in terms of when the flight crew recognize a deviation that requires notifying the air traffic controller.

Thus, one automation boundary condition metric is the duration of vertical deviations from the planned vertical profile greater than 400 ft. As shown in Figure 9, this metric shows strong interactions between function allocations and scenarios. The nominal scenario did not show any vertical deviation, and the reroute scenario is not shown because the reroute voided the altitude and airspeed restrictions defining a target for comparison. However, in the late descent scenario, the more-automated function allocations, FA1 and FA2, have longer durations of vertical deviations than the more-manual function allocations, FA3 and FA4. Because variance was not homogeneous between function allocations, the nonparametric Kruskal-Wallis test was applied, and this test showed significant differences between the function allocations ($p < .0005$). Post hoc comparisons using the Tukey HSD test indicated that the average integrated duration of vertical deviation for FA1 and FA2 was significantly higher than FA3 and FA4 ($p < .05$). On the other hand, in the tailwind scenario, the more-manual function allocations (FA3 and FA4) had longer durations of vertical deviations than the more-automated function allocations (FA1 and FA2).
Metric 7: System Cost and Performance

Ultimately, the team’s operating costs and collective performance can be measured via simulations or assessed in actual operations. In computational simulations of descent, assuming perfect execution of the actions by the agent models, mission performance has three aspects: two aspects of efficiency (average thrust as a surrogate for fuel burn and time to land) and a safety aspect (average number of air traffic restrictions violated, which can place the aircraft into opposing traffic flows). Neither efficiency measure shows any distinctive differences as a function of function allocation; instead the measures are driven by the inherent duration of each scenario.

The safety aspect, violations of air traffic restrictions, demonstrates an interaction between function allocation and scenario, as shown in Figure 10. The more highly automated function allocations, FA1 and FA2, perform better in the tailwind scenario, which benefits from the autoflight system’s ability to constantly adjust the vertical speed. On the other hand, the more-manual function allocations, FA3 and FA4, perform better in the late descent scenario, in which the autoflight system’s descent rate in FA1 and FA2 is constrained to lie within a range that cannot reliably meet the air traffic restriction.

Metric 8: Human’s Ability to Adapt to Context

This metric is derived from what-if simulations of different human behaviors, represented as different cognitive control modes. Three cognitive control modes were examined here, as detailed in the preceding companion paper: strategic (in which all necessary monitoring behaviors are included, and actions are predicted and planned), tactical (in which periodic monitoring checks are conducted), and opportunistic (in which monitoring is conducted only when necessary to complete assigned taskwork).

Figure 11 illustrates the impact of these cognitive control modes on taskload. The opportunistic cognitive control mode has similar monitoring demands regardless of function allocation, although more taskwork and interaction actions are required in the more-manual function allocations, FA3 and FA4, compared to the more highly automated function allocations, FA1 and FA2. The tactical mode shows increased monitoring demands in all four function allocations, with a greater increase in FA1 and FA2, whereas taskwork and interaction demands are similar compared to the opportunistic mode. The strategic mode shows the highest monitoring demands and interaction demands across all function allocations. Thus, should the human ever reach workload saturation in the strategic control
mode, a transition to a tactical or opportunistic control mode will likely result; such a transition will manifest in less monitoring and interaction with the automation than generally assumed when this function allocation was designed.

Because monitoring can help the flight crew predict upcoming actions and thus stabilize their work environment, we also examined the interaction between function allocation and cognitive control mode within the metric of action unpredictability, as shown in Figure 12. One-way ANOVA found that the unpredictability varied significantly with cognitive control mode ($p < .0005$): A pilot operating in the strategic cognitive control mode is predicted to experience less unpredictability than the pilot in the tactical cognitive control mode, and again, a pilot in the tactical cognitive control mode is predicted to experience less unpredictability level than in the opportunistic cognitive control mode. This difference across the cognitive control modes stems from the prevention of unpredicted actions by better management of flight route provided by the tactical and, especially, strategic cognitive control modes. These effects are more profound with the highly automated function allocations, FA1 and FA2, suggesting that if the human cannot operate strategically in highly automated function allocations, he or she will need to also behave more reactively to the automation and to events in the environment.

**Figure 11.** Number of actions (taskload) per simulated flight associated with each cognitive control mode by function allocation, averaged across all scenarios with no limitations on flight crew capability for performing multiple tasks at the same time.

**Figure 12.** Average unpredictability per simulated flight by cognitive control mode and function allocation, averaged across all scenarios with no limitations on flight crew capability for performing multiple tasks at the same time.
The simulated cognitive control modes also had a significant effect on the number of air traffic restrictions violated. Figure 13 illustrates the average number of air traffic restrictions violated by function allocation and cognitive control mode in the late descent scenario. The flight crew’s monitoring in the strategic cognitive control mode included a predictive behavior whereby the flight crew would monitor for a late descent clearance and, if detected, decrease the speed by 0.02 Mach to make the aircraft energy more manageable once the descent instruction was finally given. Thus, the strategic mode shows better performance (fewer violated air traffic restrictions) with the most-automated function allocations, FA1 and FA2. One-way ANOVA of the number of violated air traffic restrictions found that the number of violations varies significantly between different cognitive control modes ($p < .0005$). This sensitivity to different flight crew behavior is more profound for the most-automated function allocations.

**Considering the Measures as a Set**

The introduction of this paper noted the need for metrics of function allocation capable of highlighting trade-offs between different function allocations. Figure 14 illustrates the metrics of the arrival and approach example as a collective set. Each represents a percentage of the best possible value, that is, each was normalized by its highest possible value or the highest value recorded (e.g., for the coherency level, 4 is the possible highest value) and scaled such that a higher score indicates “better” results (e.g., the unpredictability measure is converted to a predictability measure).

Certain trade-offs may be observed between function allocations. With the more-manual function allocations, FA3 and FA4, the metrics of predictability and combined duration of mismatch-induced actions tend to have higher (better) values, but the combined durations of taskwork and of interaction work tend to have lower (worse) values. With the more highly automated function allocations, FA1 and FA2, the reverse relationship is found. Thus, there is no one best function allocation by all metrics.

In some cases, measures of the function allocations are also very sensitive to assumptions about how the human agents will act in context, modeled here as different contextual control modes. These measures particularly capture the effect of monitoring workload that is maintained in strategic cognitive control modes and may be shed in more time-stressed situations.

Finally, it is worth noting that some measures—those on the extreme right and left of Figure 14—do not vary appreciably with function allocation or cognitive control mode. Some, such as the performance measure time to land, are consistently high; others, such as duration of vertical deviation, are consistently low. Either way, these measures appear to be inherent to the
work domain and specific scenarios simulated here; thus, a low score cannot be improved by function allocation alone.

**DISCUSSION AND CONCLUSIONS**

In this paper, we identified eight types of metrics, with each type addressing an important issue with function allocation identified in the literature as an impediment to meeting requirements for effective function allocation. These metrics span a reasonably comprehensive and detailed set of metrics. Some of the metrics are well established, such as measures of workload and taskload. Others are proposed here as new measures of phenomena, including responsibility–authority mismatches, predictability of the work environment, and coherency of the function allocation. For these concerns, several measures have been proposed, ranging from fairly coarse, static assessments of coherence and flags of mismatches between responsibility and authority to detailed assessments of the impact of assumptions about human behavior in context.

These metrics not only target specific concerns but also collectively enable the designers of a function allocation for trade-offs between metrics. For example, our computational simulations of the work and dynamics of an air transport flight deck during descent revealed a trade-off between reducing flight crew taskwork by allocating more functions to the autoflight system, at the expense of requiring more monitoring and reducing the flight crew’s ability to predict upcoming actions.

Thus, these metrics dovetail with a design process that proceeds from conceptual to detailed design, particularly when informed by a model of the team’s collective work as described in the preceding companion paper. Early in design, a model of prototype function allocations can be laid out at the granular level of high-level functions. From this model, static measures of the coherency of a function allocation and of mismatches between responsibility and authority can be identified and considered even in conceptual design.

As the design process proceeds further and specific actions can be modeled in detail, other static metrics can be assessed. For example, the static measure of workload identifies the actions required from the human and from the automation that the design must support. Also, the

---

*Figure 14. All function allocation metrics by function allocation and scenario, each normalized as a percentage of the measures’ best possible value.*
information the human will require to perform monitoring actions due to mismatches between responsibility and authority and due to automation boundary conditions can be identified.

Finally, as the design matures and the work model is ready, then the designer needs to consider potential operating conditions, represented as scenarios to be simulated. As the designer defines the intended operating conditions of the combined human–automation system, predicted versus unpredicted actions and potential violations of the automation boundary conditions can be estimated, providing further insight into potential issues with proposed function allocations.

Once a detailed work model and representative operating conditions are identified, the metrics can be assessed from computational simulations to confirm the design or to identify problematic aspects of the design. Because the model can be easily changed to reflect different function allocations (or details of the automation’s functioning), these models and metrics can be applied as part of an iterative design process that involves exploring potential mitigations to issues with function allocation as they are identified. Ideally, these computational predictions will also inform real-time human-in-the-loop simulations and observations of real operations as they are implemented.

Ultimately, these metrics may not only be gathered in real time, but they may be then applied to dynamic function allocation decisions. Several mechanisms may be envisioned: the adaptive automation construct, whereby the automation compares metrics of current operation with those predicted with different function allocations to automatically assume new function allocations; the adaptable automation construct, in which these metrics may, in some form, be presented to a human team member for consideration as to the potentiality of new function allocations relative to the present; and the use of these metrics for dynamically adapting function allocations in teams in a broader sense than only that between a human and an automated system.

ACKNOWLEDGMENTS

The models and simulation framework discussed here have been developed through cooperative agreements with NASA’s Aviation Safety Program’s Integrated, Intelligent Flight Deck project and System-wide Safety and Assurance project, under the technical supervision of Michael Feary, Paul Schutte, Michael Schafto, and Guillaume Brat.

REFERENCES


Amy R. Pritchett is the David S. Lewis Associate Professor in the Georgia Institute of Technology’s School of Aerospace Engineering and director of the Georgia Tech Cognitive Engineering Center. She earned an ScD in aeronautics and astronautics from MIT in 1997.

So Young Kim is a user experience/human factors researcher at the UX Innovation Lab in GE Global Research, located in the Bay Area, California. She received her PhD in aerospace engineering from the Georgia Institute of Technology in 2011, focused on human–automation function allocation.

Karen M. Feigh is an assistant professor in the Georgia Institute of Technology’s School of Aerospace Engineering. She earned a PhD in industrial and systems engineering from the Georgia Institute of Technology in 2008.