Controller-Automation Interaction in NextGen: A New Paradigm

Julian Sanchez, Cole J. Krumbholz, Marcio O. Silva

Abstract

As conventional Air Traffic Control (ATC) environments transition towards the Next Generation Air Transportation System (NextGen), new tools and capabilities, such as Area Navigation (RNAV), Required Navigation Performance (RNP), data communication, Automatic Dependent Surveillance-Broadcast (ADS-B), and automated decision making tools for monitoring, merging, and spacing are expected to be introduced. These new capabilities may drastically change the nature of the controller task, presumably toward a more supervisory one (JPDO, 2007; Kopardekar, Bilimoria, & Sridhar, 2008). The objective of this paper is to describe a NextGen ATC display concept that supports an automation-rich ATC environment, while adhering to a human-centered design philosophy and key human-automation interaction design guidelines. The underlying design philosophy of this concept is to maximize the overall performance of the ATC system, by ensuring that both controller and automation work in harmony and make optimal contributions. The design philosophy is based on the broader research findings and guidelines in the area of human-automation interaction. As part of this concept, a proposed model of controller-ATC automation is also described.

Introduction

As industry, government, and research institutions continue to develop advanced concepts and technologies to make the Next Generation Air Transportation System (NextGen) a reality, it is critical to proactively examine Human Factors-related challenges at an early stage (Pritchett, 2008). One area that is likely to experience considerable changes with NextGen, is Air Traffic Control (ATC). Two key challenges for ATC in NextGen include (1) clearly defining the controllers' tasks and responsibilities in highly automated environments, and (2) implementing intuitive human-machine interfaces that advocate high controller and system performance.

In general, most descriptions of NextGen, such as the Joint Planning and Development Office (JPDO) NextGen Concept of Operations (2007) describe mechanisms and architectures within aviation that will yield critical operational benefits. These descriptions/roadmaps of NextGen also highlight opportunities for candidate technologies/systems to help meet all of the projected operational challenges. As these candidate technologies evolve, there is a valuable opportunity for Human Factors issues to be key drivers of NextGen requirements.

The main objective of this paper is to take a step toward defining the roles and responsibilities of air traffic controllers in NextGen by describing a user-centered, ATC display concept that supports an automation rich environment. The scope of the ATC display prototype includes both the Terminal Area Approach Control (TRACON) and En Route domains. A review of the broader human-automation interaction research is provided to highlight major challenges in introducing high levels of automation to environments where humans are expected to play a central role. Key design guidelines from the human-automation interaction literature are extrapolated. Based on these design guidelines, a conceptual model of controller-ATC automation interaction is proposed, and a design philosophy for an intuitive ATC display is presented. The ATC display concept is strictly a human-computer interaction prototype and it is partially agnostic to the automation algorithms that could „drive' it. The main assumptions underlying the design of the ATC display, and its design philosophy, are that the ATC automation will be robust, reliable, and operate at a Stage 3 and 4 levels (Parasuraman, Sheridan, and Wickens, 2000). Stage 3 and 4 ATC automation is capable of respectively making decisions about aircraft separation and efficiency, and executing actions such as sending commands directly to aircraft. Stage 1 automation is used strictly for sensing/detection tasks, while Stage 2 automation can assimilate raw data to represent a processed version.

The Air Traffic Control Task (Today)

Today's ATC task is primarily a „manual', tactical task in which controllers visually scan the surveillance display to gather information about aircraft such as their location, altitude, speed, and type/model (Danaher, 1980). Controllers use this information to update their mental model of the airspace, as well as plan and execute changes, such as merging aircraft, or increasing in-trail spacing (Rodgers and Drechsler, 1993). Performing these tasks requires a considerable amount of mental workload (Hilburn and Jorna, 2001). Figure 1 illustrates a conceptual representation of today's Air Traffic Control (ATC) task. This conceptual model illustrates the primary role of automation in today's ATC environment, which is to serve as an information-processing tool, mostly to display surveillance information (i.e., Data Sources 3 & 4, which could be the speed and location of an aircraft). Controllers also gather and integrate information from other sources (i.e., Data Sources 1 & 2), such as verbal communication with pilots, as well as other controllers. While the model in Figure 1 appears „simple', it is worth noting that the ATC task in today's environment is very complex, and several years of training and experience are required to acquire the declarative and procedural knowledge necessary for satisfactory levels of performance. The main objective of Figure 1 is to serve as a „baseline' representation of today's ATC task; this way, the changes that accompany NextGen can be more readily identified and analyzed.

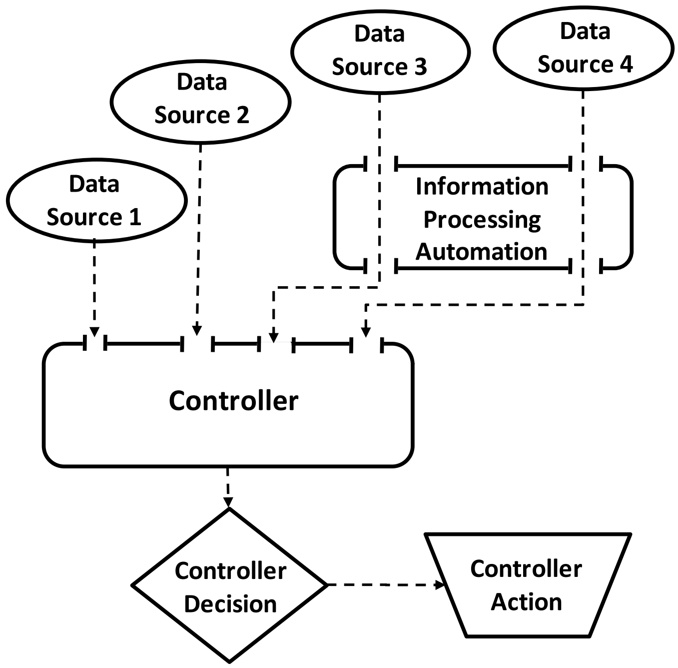

Figure 1. Conceptual model of today's ATC task flow

Examples of ATC Automation Systems

There are several examples of automated systems in today's ATC environment. Two of those systems, which have a controller Graphical User Interface (GUI), are the User Request Evaluation Tool (URET) and the passive Final Approach Spacing Tool (pFAST). URET is a decision support tool used in the En Route environment to help controllers detect and resolve conflicts, as well as engage in trial planning (Brudnicki, McFarland and Schultheis, 1996). The URET user interface consists of a secondary display, where the controller can evaluate the advice and feedback provided by the automation without obscuring any of the real-time surveillance information. The URET display provides textual and graphical information. The implementation of URET does usually entail a second controller (D-Side) to use it. However, the responsibilities of a controller who has access to it are not much different to those of a controller who does not, although there is evidence that ATC performance is positively impacted in many cases (Kerns and McFarland, 1998). However, it is up to the controller to determine how the tool is best utilized and incorporated into the ATC task.

The pFAST system, which was evaluated in the Dallas/Ft. Worth TRACON facility back in 1996 (Lee and Sanford, 1998), suggests runway assignments for arriving aircraft, as well as an arrival sequence aimed at optimizing the flow into multiple runways. Therefore, the controller remained responsible spacing and separation, as well as the strategy to accomplish the recommendations given by pFAST.

The user interface implementation of pFAST consists of runway assignment and sequencing information integrated into the data block. Like URET, the roles and responsibilities of the controller remain the same. One of the potential fallbacks of pFAST is that the controller does not get any insight into the logic of the automation (i.e. why specific suggestions are being made). Controller feedback about pFAST suggests they were generally concerned about the accuracy of the sequence advisories, even when the automation was deemed to be correct (Lee and Sanford, 1998). This issue of „automation transparency' is discussed later in this paper, and design guidelines to help promote appropriate trust in automation are presented.

The Air Traffic Control Task: Changing Roles and Responsibilities

While today's ATC environment has a considerable amount of automation, it is primarily Stage 1 & 2 (Parasuraman et al., 2000), which is used for sensing, detecting, and processing surveillance information (i.e., location, speed, and direction of aircraft). As mentioned previously, there are some examples of Stage 3 automation in today's environment, such as URET. However, in NextGen, solutions for ATC tasks such as spacing and separation may not stem solely from the skill/knowledge of controllers to vector aircraft or manage aircraft in RNAV environments. Solutions might also be derived from the algorithms of an automated system. Similarly, Data Communications (DataCom), which is a collection of technologies designed to transmit data between ATC and aircraft via digital text format, may significantly impact the prevalence of radio communications between controllers and pilots. As the NAS transitions toward NextGen, the broader implementation of Stage 3 & 4 automation is becoming a more tangible reality. An example for the desire of the trend toward Stage 3 & 4 automation can be found in the JPDO NextGen Concept of Operations (2007), which provides the following statement about the future of the ATC task:

> "automation supports the migration from tactical to strategic decision making by assimilating data and supplying information, as well as by performing many routine tasks" (p. 2-11).

The incorporation of Stage 3 & 4 automation into the ATC system to perform „routine' tasks is expected to carry numerous benefits. Those benefits include the increased predictability of operations within the NAS, which is likely to have a positive impact on efficiency and safety. However, the introduction of higher levels of automation can also affect the behavior of controllers, and ultimately, the performance of the ATC system. Overall, the introduction of new automation into systems has a number of positive implications; mainly, humans are able to optimize their efforts by reallocating their cognitive, perceptual and physical resources away from tasks where the automation is providing assistance, onto other tasks that can benefit from additional attention (Shepley, Johnson, Sanchez and Smith, 2009). In most cases, this shift in attention has the potential to increase the overall performance of the system, as long as the automation is highly reliably (Dixon & Wickens, 2004; Maltz and Meyer, 2001; St. John and Manes, 2002). For example, if an ATC automation aid provides accurate aircraft sequencing support, the cognitive/perceptual demands on the controller will decrease (Sanchez and Zakrzewski, 2009). With decreased cognitive/perceptual demands, the controller has „spare' resources that could be used to manage additional air traffic.

While highly reliable automation leads to improved overall system performance, it also increases the tendency of humans to over rely on automation. Therefore, if the automation does fail (e.g., it misses a potential conflict), humans who are relying on said automation are not likely to detect it. The results of Skitka, Mosier, and Burdick (1999) exemplify the paradoxical nature of the issue of automation reliability and human reliance. Skitka et al found that the presence of a highly reliable decision support aid in a system failure detection task, led to an increase in the number of correct responses, when compared to the non-automated condition. However, in instances when the automation was unreliable by missing system failures, participants in the non-automated condition were significantly better at detecting those failures than those aided by automation. Ultimately, the pivotal issue around the subject of introducing automation into any system comes down to a matter of reliance of the human on automation.

To Rely or Not Rely?

Over the last 20 years, research in the area of human-automation interaction has yielded critical insights about variables that impact human-automation systems, such as level of automation (Endsley and Kaber, 1999; Moray, Inagaki, and Itoh, 2000), vigilance-related issues (Molloy and Parasuraman, 1996), and the impacts of automation reliability on trust (Dixon and Wickens, 2003; Sanchez, Fisk, and Rogers, 2004). Overall, the majority of research in this field, has focused on the „rely or not rely on automation' dilemma. The crux of the dilemma is that too much reliance (a.k.a. over reliance, over trust, over use) on the automation, results in a degraded ability of the human to serve as a redundancy option if/when the automation fails (Parasuraman, Molloy, & Singh, 1993). However, assuming the automation is more effective at performing the task than the human, not relying on the automation usually results a lower overall system performance (Skitka et al., 1999). This dilemma creates a paradoxical relationship between automation reliability and system performance. On one hand, automation is introduced to improve system performance, and hence, the human should rely on it. On the other, once the human interacts with automation of high reliability for extended periods of time, over reliance begins to take place, which means if/when the automation fails, the human is not likely to detect the error.

In an attempt to offer a solution to the „rely or not rely on automation' dilemma, the concept of *appropriate reliance* (a.k.a. *appropriate trust)* was proposed by Lee and See (2004). Appropriate reliance describes a match between the perceived capabilities of the automation by the human and the actual capabilities of the automation (Lee and Moray, 1994; Lee and See, 2004). The idea is the human should rely (or not rely) on the automation based on when the automation is likely to be reliable (or not reliable). For example, if a collision avoidance system of an automobile has a high false alarm rate during low visibility conditions, appropriate reliance, on the part of the human, would consist of frequently checking the validity of automated alarms during low visibility, while relying on the automated alarms during high visibility. To achieve appropriate reliance, Lee and See suggested that automated systems should be „transparent', so the human can determine when it is appropriate to rely on it.

While the goal of achieving appropriate reliance through transparency has been shown to be effective (Skjerve and Skranning, 2004), the central issue of the „rely or not rely on automation' dilemma remains embedded within the common mistake of assigning the human with the task and responsibility to serve as a backup to the automation. The statement "the human is there in case the automation fails, or makes a mistake" is used too often as a design crutch when introducing automation into a system. In <u>most</u> cases, this statement violates the logic and negates the benefits of introducing automation into a system, which is to relieve the human of a task or tasks. When an automated agent is introduced into a system, it should not require a human to act as its backup. The human's contribution should not be redundancy, but rather, robustness to the system. This seems like a slight distinction, but it is a critical one. The human's task should be to complement the automation by gathering data the automation cannot sense, and processing information outside the boundaries of the automation's 'intelligence.' The reason to make automation transparent, as Lee and See suggest, should not be so the human can verify the automation's performance in the task it was designed to do. Automation transparency should simply help the human understand, at a high level, the „reasoning' used to generate solutions. This way, the human can evaluate the automation's aid against other information sources, as well as the context of the circumstances.

The following example helps describe the critical difference between having a human act as a backup to the automation and having the human in the loop to inject robustness into the system. A hypothetical ATC automated decision support aid for TRACON airspace operations detects a potential conflict between two aircraft 10 minutes into the future and generates a solution, which is presented to the controller. The solution calls for a minor speed decrease of aircraft A, and a minor speed increase for aircraft B. If these speed changes are executed within the next minute, the future conflict will disappear. In this example, the ATC automation is responsible for the task of detecting future conflicts (more than five minutes into the future), and generating safe solutions that have a minimal impact on the rest of the airspace. One of the controller's tasks in this hypothetical environment is to evaluate automation-generated solutions and accept/reject/modify them. This is the point in a design philosophy of a human-automation system where one can fall into the „human is a backup to the automation' fallacy. If the fundamental tasks of the controller are to verify that aircraft A and B will in fact be in a future conflict and ensure the automation-generated solution results in sufficient separation, then, the controller is simply acting as a backup to the automation. To be an effective backup, the controller would need to allocate a considerable amount of cognitive resources to validate the existence of future conflicts predicted by the automation, and search for other conflicts the automation might fail to detect. Performing this „backup' task would not be much different from the manual (without automation) workload requirements, and therefore, would largely negate the potential benefits yielded by the automation (Hilburn, Jorna, and Parasuraman, 1995). The other potential drawback of making the human the backup is that over time, if the ATC automation behaves reliably, the controller becomes complacent in his/her role as a backup and the probability of catching an automation mistake would be very low (Molloy and Parasuraman, 1996; Metzger and Parasuraman, 2005; Sethumadhavan, 2009).

Assume instead that the fundamental task of controller in the example above is not simply to be a backup to the automation. This means the controller is not responsible for detecting future conflicts (that is a responsibility of the automation), and when an automated solution is presented, he/she does not need to evaluate whether the solution complies with separation standards. Instead, the controller's task is to accept/reject/modify the solution based on contextual information that the automation does not have access to. For example, the controller may know that the busiest airport in the TRACON is considering a change in runway configuration, which would require aircraft A to transition to a new arrival procedure. Given this information, the controller may choose to reject, or modify the solution offered by the automation. The point is, while the controller still has final authority over the execution of a solution, the judgment used to make that decision is based on different criteria than those the automation is using to generate its solutions. This collaboration between controller and ATC automation adds robustness to the system, while allowing the true benefits of having automation to be realized.

What Should the Human and the Automation Do?

Another key thread of work in the human-automation interaction field, is the topic of function allocation between humans and automation (machines), which has been analyzed for several decades, dating back to Fitts (1951) list “humans are better at, machines are better at.” The underlying premise of the Fitts’ list, as well as other similar lists, is that assigning functions to each agent, based on their skills/capabilities, is a key to achieving optimal system performance. Of course, with continuous advances in technology, the functions that are effectively performed by automation keep changing and growing. For example, in domains like agriculture, 10 years ago, the task of steering a tractor was on the “humans are better at” side of the list (Sanchez & Duncan, 2009). Today, GPS-based automated steering systems are significantly more accurate than manual driving, so steering agricultural vehicles has become a “machines are better at” task. The aviation domain has also experienced a similar trend toward automation, especially in the flight deck, where tasks like landing an aircraft are being transitioned to automation in situations like CAT III operations.

As the capabilities of ATC automation continue to improve, there will likely be a trend toward reallocating some of the functions currently performed by the controller. Therefore, what should ATC automation do in NextGen? An overly simplistic answer to this question is „it should do whatever it can do better than a controller.’ However, one of the many reasons this answer is overly simplistic is because as functions are allocated to the automation, the nature of the ATC task will change, along with the skills/knowledge required to effectively manage traffic. This change in the ATC task yields the next crucial question: „is it reasonable to expect the controller to perform these new tasks?’ A „reasonable’ task is one that falls within the envelope of human performance capabilities. For example, it is not reasonable to expect a controller to sit in front of an ATC display, not perform any actions or make any decisions of any kind, and simply wait for a rare event to arise. Decades of basic vigilance research suggest that is a formula for failure (Warm, 1984). Similarly, asking the controller to act strictly as a backup to the automation carries numerous disadvantages. Therefore, the challenges in successful integration of ATC automation into the NAS are to clearly define the tasks the controller is expected to perform and assess if those tasks reasonably fall within the envelope of human performance capabilities. One way to satisfy the latter requirement is to develop a human-automation interaction environment that embodies a human-centered design philosophy. In other words, the initial assessment of the new controller tasks in an environment with more automation may suggest they are not reasonable, but this may be partly the result of a User Interface (UI) platform that simply does not support these new tasks. Later in this paper, a design philosophy for a highly automated ATC environment is outlined.

The Air Traffic Control Task (NextGen)

Figure 2 is a conceptual model, which outlines the ATC tasks during NextGen. The purpose of this model is to outline the interaction between the controller and the automation, which includes roles and responsibilities of each agent, as well as the exchange of information between the two. The main assumption underlying this model is that in NextGen ATC automation will be capable of generating safe and effective solutions about spacing and separation of aircraft. Therefore, responsibility for these tasks will be partially allocated to the automation. This is a key shift from today’s ATC environment, where the controller is the sole agent responsible for those decisions. Another important assumption is that ATC automation will be able to send most instructions to the flight deck via DataCom.

The introduction of reliable automation to the ATC environment should reduce task load demands on the controller (Kopardekar, Prevot, and Jastrzebski, 2009; Metzger and Parasuraman, 2005). With reliable automation and an intuitive user interface concept, one controller should be able to effectively manage more aircraft than in today's ATC environment. The controller-automation interaction depicted in Figure 2 outlines a „side-by-side' collaboration of the two agents, where each is responsible for performing tasks they are „better at' than the other agent. Furthermore, there is transparency between the controller and the automation, such that each agent can communicate pertinent information to the other.

One of the key tasks of the controller in NextGen will be data acquisition in areas where the automation's sensing and perception capabilities are not developed. For example, *Data Source 1* (Figure 2) could represent a radio communication from a pilot who is describing a weather status update, such as turbulence or icing. A NextGen ATC display should provide the controller an easy way to communicate these types of information to the automation (i.e., *Data Source 1'*), which in turn, helps the automation generate context-driven solutions. In some cases, the controller and the automation can have access to the same data source, but may sense and process it differently. For example, *Data Source 3a* might represent the altitude of a specific aircraft, which the controller can view in the surveillance radar display. Meanwhile, the automation has access to the same information (*Data Source 3b*), but it gets it directly from ADS-B.

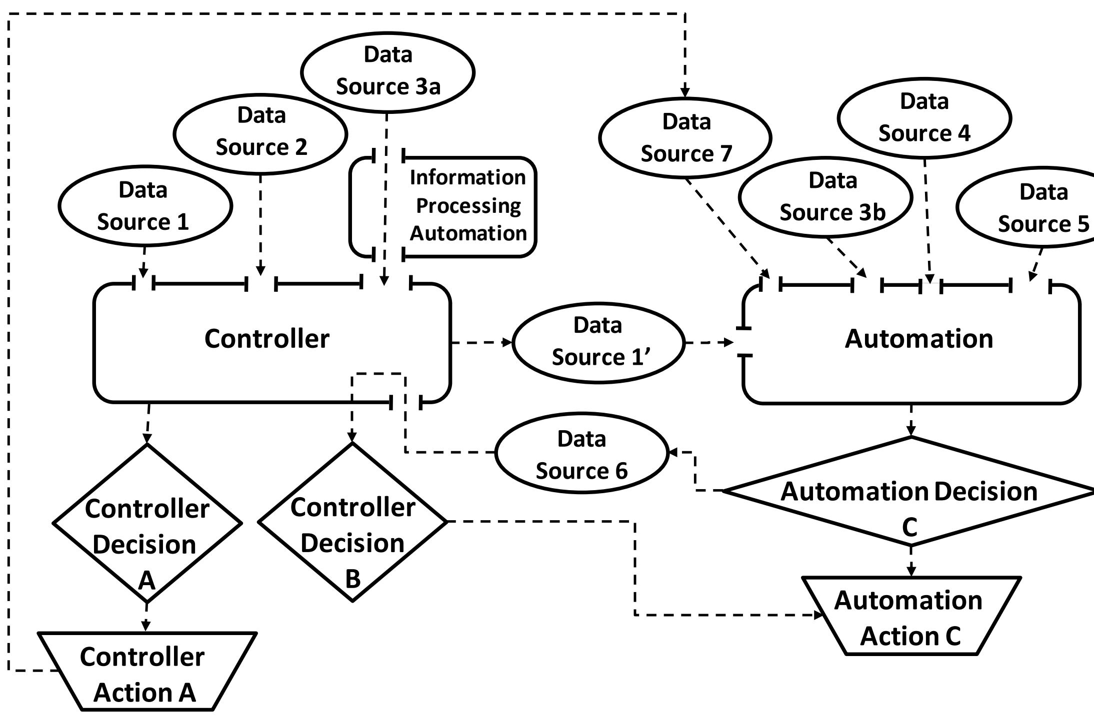

Figure 2. Conceptual model of the Human-Automation Interaction paradigm embodied in the N-CAID prototype

In NextGen, ATC automation will likely perform most tactical tasks, such as generating solutions to ensure all aircraft maintain minimum separation standards, while maximizing airspace efficiency (*Automation Decision C*, Figure 2). Those decisions will be shared with the controller (*Data Source 6,* Figure 2) before being executed. The controller evaluates the automation decisions, and has the authority to override them, especially when there is other information (e.g., *Data Source 2,* Figure 2) that puts into question the effectiveness of those decisions. For example, to maximize efficiency, the automation may decide to increase the speed of two arriving aircraft that are 25 miles from the airport. As it communicates this decision/suggestion to the controller, he/she is also informed by a supervisor that there may be a shift in airport runway configuration due an unforeseen event. The controller decides it may not be a good time to ‚push the limits' of the airport, given the upcoming runway configuration change, so he/she declines the automation-generated decision to speed up the aircraft. Evaluating automation-generated decisions against contextual information is a key role of the controller in NextGen. It is worth highlighting that the controller is not acting as a backup to the automation by evaluating its solutions against basic criteria such as meeting minimum separation standards. Instead, the controller evaluates the solution provided by the automation against contextual information that the automation in unable to sense, detect, and/or process. Rather than a backup, the controller collaborates with the automation to ensure its solutions are effective given the state of the airspace. At this point in time it is not feasible to exactly define the capabilities of an automation system that does not yet exist, which makes it difficult to clearly outline the responsibilities of the controller and the information he/she is supposed to use to accomplish those responsibilities. However, not making the controller a backup to the automation should be the underlying design philosophy.

In cases when the controller approves the automation-generated decisions, the automation executes those decisions by transmitting the appropriate commands to aircraft via Data Communications. It is likely that a NextGen ATC will have to support both active and passive controller approval strategies. Under active approval, the controller has to explicitly consent to every decision by the automation before it is executed. In a passive approval system, the controller has the authority to override any decision, but does not have to explicitly consent to every decision for the automation to execute it. The design decision to implement either active or passive approval needs to be carefully examined, and should be driven in part by empirical data.

In addition to evaluating and approving automation-generated solutions, the controller should also have the ability to make tactical decisions (i.e., *Controller Decision A,* Figure 2) and instruct the automation to execute those actions (*Controller Action A,* Figure 2). The controller's actions would be immediately passed on to the automation (i.e., *Data Source 7,* Figure 2) to ensure its future decisions account for the most recent circumstances. An example of this task flow is a controller's ability to expedite a specific aircraft's arrival in an emergency situation. The controller would be able to simply state intent, and the automation would support it by generating solutions to help accomplish that intent. Again, it is important to highlight that the controller is the supervisor in the human-automation relationship.

NextGen Controller-Automation Interaction Display (N-CAID)

Toward a User Interface Design Philosophy for Controller-Automation Interaction

Once the roles and responsibilities of the controller and the ATC automation are defined, it is important to identify a design philosophy that promotes a collaborative environment between the two agents. The crux of the issue with human-automation systems is finding a balance between an environment where the human is not engaged and an environment where the human is asked to act as a backup to the automation. The design of the interface between the two agents provides a great opportunity to ensure successful collaboration. The following guiding principles were used as the basis for the design philosophy of the ATC NextGen display prototype, which is referred to as *NextGen Controller-Automation Interaction Display* (N-CAID) for the remainder of this paper.

1. Make the automation-generated solutions transparent to the controller. Automation transparency allows the controller to evaluate the automation's solutions against contextual information the automation is not designed to incorporate in its decision making. It also promotes appropriate reliance, by allowing the controller to develop a mental model of the automation's reliability. This means, over time, the controller will learn the relationship between context and the quality of the decisions by the automation.
2. Balance automation transparency with providing only the right amount of information. A design philosophy strictly based on transparency can easily degrade into a platform that reveals too many details about what the automation's 'thought process.' Transparency should be task-centric, so the controller gets the right information, not all possible information.
3. In a supervisory role, a controller should have the ability to easily express strategic and tactical goals to the automation. This will help the controller feel like they have control, as opposed to being "driven" by the automation (Hancock and Scallen, 1996; Whitfield, Ball, and Ord, 1980).
4. Promote an intuitive, direct mechanism for the controller to communicate with the automation. This entails eliminating, or minimizing the number of „hidden' commands and features of the graphical user interface. It also strives toward an input device that promotes an optimal balance between speed of entry and errors.
5. Similar to a dialogue between humans, the automation should convey tone and emphasis when communicating with the controller. This means the automation should be able to express the situational and temporal urgency of its messages in a way that is intuitive to the controller.

N-CAID Prototype Description

A picture of the N-CAID prototype is shown in Figure 3. The N-CAID prototype serves as an instantiation of the controller-automation interaction concept shown in Figure 2, where both controller and ATC automation work collaboratively to manage the airspace. The design philosophy of N-CAID also encompasses the principles discussed in this paper, such as a clear division of tasks, and an appropriate level of transparency. The prototype is composed of two physical displays, which help provide a meaningful, task-based division of functions for the controller. Again, N-CAID is a UI concept, not an automation prototype.

The primary display (bottom of Figure 3) provides a real time visualization of surveillance information in the controller's assigned airspace. The primary display has two functions. The first is to render a real-time picture of the airspace, much like today's surveillance radar display. This area is never obstructed or locked by dialogue windows, or any other type information that is not directly relevant to what is occurring in real-time. The second function is to provide the controller with information about the „thoughts' and intentions of the ATC automation. This function is housed in the ATC Automation Window on the right side of the display.

Through the ATC Automation Window, the automation can notify the controller that it has a solution to a conflict, or a suggestion to optimize the airspace. The bottom-horizontal edge of the ATC automation window represents real-time, and each horizontal line above it an ascending unit of time (e.g., one minute). When the ATC automation generates a solution, it communicates it to the controller by creating a solution rectangle within the ATC Automation Window. The distance from the bottom horizontal edge of the solution rectangle, to the bottom horizontal edge of the ATC Automation Window represents the amount of time before the first action associated with that solution needs to be executed.

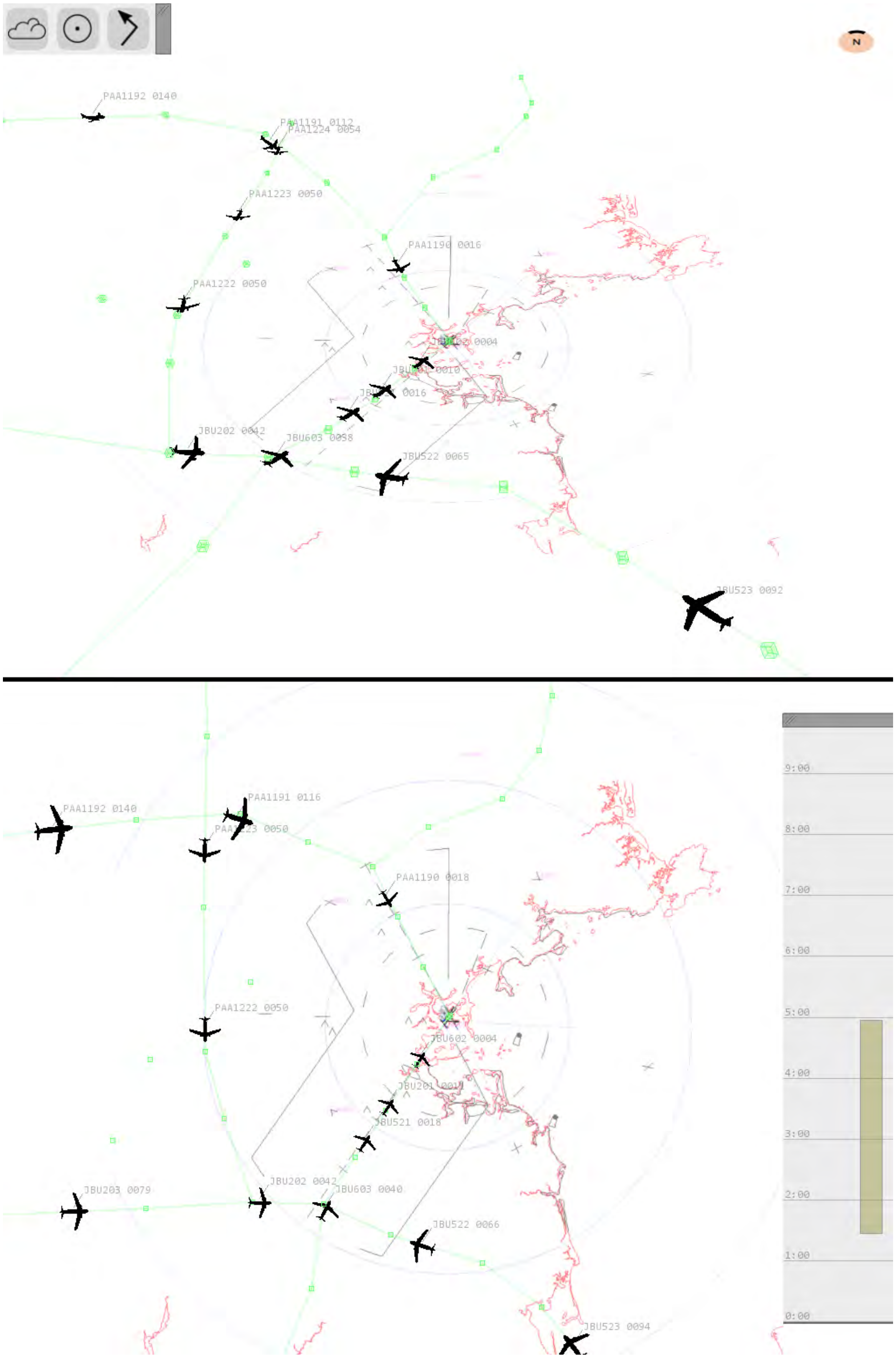

Figure 3. N-CAID prototype

Once a solution rectangle is generated, it continuously moves down (as time progresses). The length of the solution rectangle represents the estimated time it will take the solution to unfold. The color of the solution rectangle represents whether it is an active or passive approval solution, although color could be used to communicate other variables. When the controller hovers over the solution rectangle, the aircraft involved in the solution are highlighted in the primary and secondary displays. Overall, the physical characteristics of the solution rectangles can be manipulated to provide any information about the automation-generated solutions, even to the extent that the shapes of the solutions do not necessarily have to be rectangles. The goal of the ATC Automation Window is to provide a meaningful, high level of automation transparency to the controller. By visually representing the temporal characteristics of the ATC solutions (e.g., start time and duration of solution) along with other information about the solutions (e.g., active versus passive solutions), the controller gets a quick, easy look at the „thoughts' and intentions of the automation.

The secondary display (top of Figure 3) hosts a several capabilities, which allow the controller to (1) view/explore detailed views of the ATC automation's solutions; (2) modify those solutions, if necessary; (3) view/explore the modified solutions; (4) communicate new information to the ATC automation; (5) evaluate the impact of the solutions on variables like fuel consumption. This display is the main channel of communication between the controller and the ATC automation. Given the number of capabilities hosted in the secondary display, the most effective method to describe its functionality is to provide a task-based scenario, which offers a clear portrayal of the roles of the controller and the ATC automation, as well as the collaboration between the two agents.

Sample Scenario. The ATC automation identifies a future conflict and generates a solution, which appears on the ATC Automation Window. The controller detects the solution rectangle and notices, based on the location of the solution rectangle, that he has approximately two minutes before the first action of this solution needs to be implemented. The controller can also tell by the color of the rectangle that it is a solution requiring active approval. By placing his/her finger (or direct input device) over the solution rectangle, the aircraft affected by the solution are highlighted. Therefore, even before exploring further details of this solution, the controller is able to develop an initial mental model of the situation by seeing the aircraft involved in the automation solution, and understanding some of the high-level characteristics of the solution (i.e., expected start time, duration, active/passive, etc).

To explore further details of an automation-generated solution, the controller clicks/touches the solution rectangle, which triggers the timeline evaluation mode of the secondary display. In this mode, the controller can drag a scroll bar within the horizontal timeline located at the bottom of the display. As the scroll bar slides to the right, the automation generates a projected future view, which provides the controller with a visualization of how the solution will unfold as a function of time and within the context of the airspace he is responsible for managing (i.e., all of the aircraft in the airspace are included in this projection). The aircraft affected by the solution are highlighted while the controller views the solution (Figure 4). The timeline evaluation mode allows to the controller to build a mental model of the airspace in the projected future. As discussed earlier, it is not the task of the controller to make sure a solution is valid in terms of complying with separation standards, etc. The controller's task is to ensure the solution is not in conflict with other information the automation is not capable of sensing, detecting, and/or processing. This change in paradigm will require a re-definition of Operational Errors (OEs) and Pilot Deviations (PDs). It will also call for the creation of a category that encompasses errors by the automation (AEs).

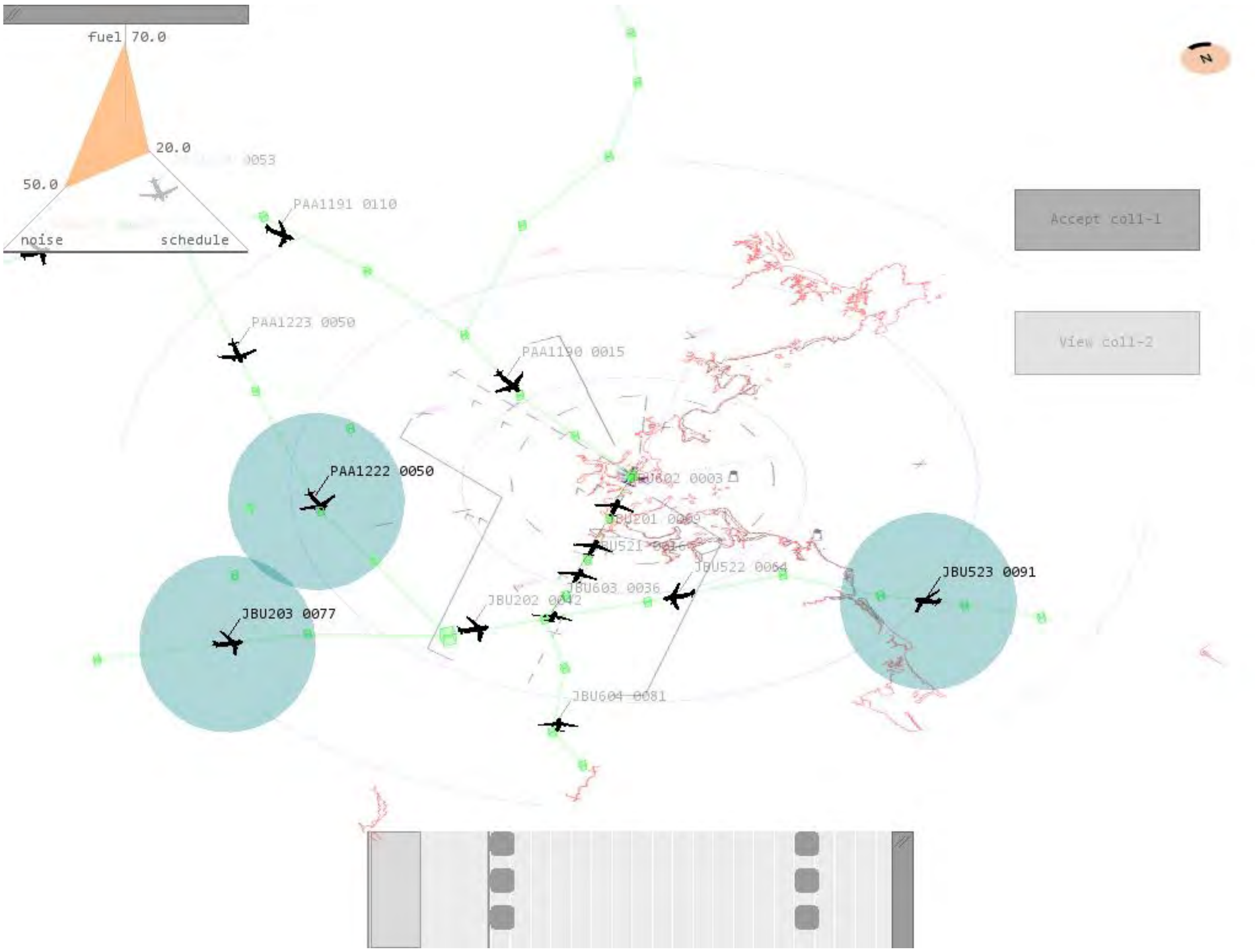

Figure 4. Secondary display: Aircraft involved in solution are highlighted while the controller views the solution by sliding the time bar

If the controller decides the solution is appropriate, the “Accept col1-1” button is pressed. This authorizes the automation to execute that solution, and DataCom messages are sent to the respective aircraft at the appropriate time. If the controller decides the solution is not appropriate (e.g., it gets an aircraft too close to restricted airspace), there are two options. First, the controller can request the automation to calculate and display a new solution, which can be evaluated in the same manner as the first one (this is done by clicking the button labeled “view col1-2”). Alternatively, the controller can make slight modifications to any solution. The timeline is populated by markers that represent instances where there are opportunities to make changes to the trajectory and speed of aircraft (Figure 5). For example, if the first action of an automation-generated solution is to instruct Flight JBU208 to change its speed to 250 knots, there would be a marker on the timeline showing this request. The controller can click on any of the markers and make adjustments to the heading, altitude, and speed of the aircraft. Once these changes are made, the controller can explore the modified solution the same way any other solution is explored. This provides an option to take an “almost perfect” solution, from the controller’s perspective, modify it, and explore it.

The timeline evaluation mode is intended to provide an optimal level of automation transparency. The controller is able to see exactly what the automation thinks will happen when it generates a suggestion to resolve a conflict or optimize the use of the airspace. The controller is also able to evaluate the impact of the solution within the context of the broader picture.

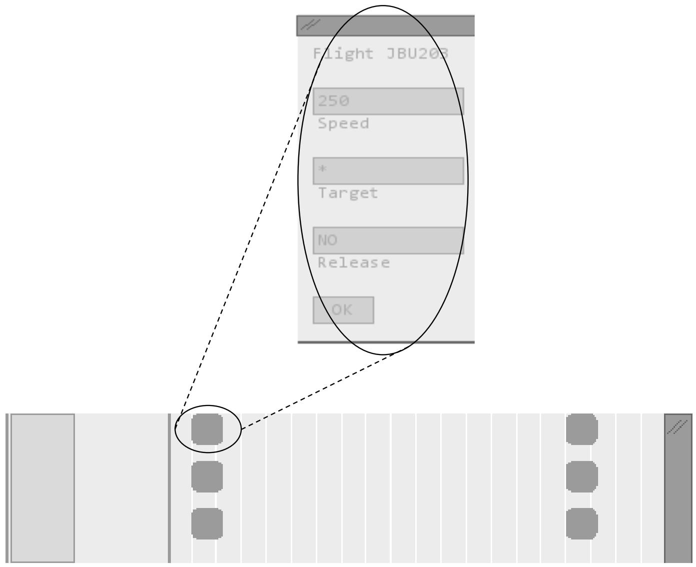

Figure 5. Markers on timeline represent commands that will be uplinked to aircraft. When the controller touches any of these they will see the specific information about the aircraft and have the ability to modify it and view it before accepting it.

As the controller evaluates each solution set, information about the impact of the solution on variables of interest is made easily available. On the top left corner of the secondary display, there is a notional representation of how the solution affects three variables (Figure 6). This notional representation uses emergent display features to convey information in an intuitive manner. In an emergent display, multiple elements are grouped such that its emerging features convey information about the individual elements. A unique characteristic of emergent displays is the reduction of information processing time, because the human does not have to sense and interpret each individual element, but rather, can use the overall emerging patterns to draw information about the system (Treisman, 1986). In this example, the concept of emergent displays was used to represent the impact of the solution on fuel, noise, and schedule compliance. Each variable is plotted along an axis, and the three axes are connected in a triangular shape. As the value of any of these variables changes, the connecting line adjusts accordingly. Therefore, the controller is able to quickly assess and compare each solution by simply looking and interpreting the emerging shapes of the triangles. Over time, a controller can develop heuristics that link the shape of various triangles to the effectiveness of solutions (e.g., an acute triangle suggests that one of the variables is being affected adversely in favor of the other two). The goal of the notional information display is to provide the controller with easily accessible information that facilitates the task of evaluating automation-generated solution against the broader context. In today's operational environment, these types of criteria are not critical to controllers' decision-making; however, they are usually important to airlines/operators. In a NextGen environment, the role of monitoring these types of contextual criteria may become an important function of controllers.

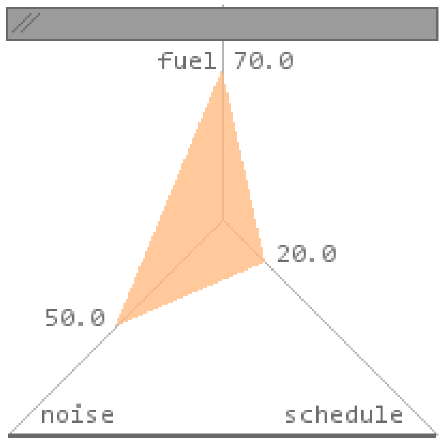

Figure 6. Notional representation of the impact of a solution on variables of interest to the controller.

Another function of the secondary display is to support the ability of the controller to communicate with the automation. As illustrated in Figure 2, the controller may have acquired information that will affect the quality of the solutions generated by the automation. Therefore, it is important the controller has the ability to pass this information onto the automation. The icons illustrated in Figure 7 are examples of the types of information a controller may need to convey to the automation. For example, the cloud icon allows the controller to outline an area on the secondary display where there is undesirable weather. By outlining the weather area, and specifying an approximate altitude, the automation will help the controller generate solutions to route traffic around the weather. The controller can also use these icons to express intent to the automation. For example, the controller can select the arrow icon, then select an aircraft and draw a line vector to a specific waypoint, or location. This expresses to the automation that the controller would like to send that aircraft to that point, and the automation would generate a solution to help carry out that intent.

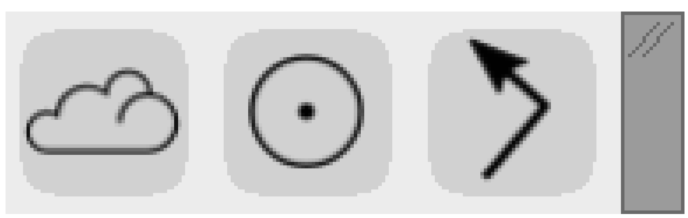

Figure 7. Examples of icons that represent types of information that a controller may need to convey to the automation

3D Manipulation of the Airspace

The secondary display also offers the capability of exploring the airspace in a 3D view. The use of 3D displays to visualize ATC information has been researched in the past with mixed results. For example, Burnett and Barfield (1991) found that controllers preferred a perspective rendering at a 45 degree elevation angle over the standard, planar ATC display. Conversely, Wickens (1995) concluded that the collective results of a research program aimed at identifying the potential benefits of 3D visualization for ATC did not yield any clear support for this approach over the traditional 2D views. The main source of the costs associated with the 3D views is related to line-of-sight ambiguity (Tham and Wickens, 1993). Even the use of stereoscopic displays does not appear to add any meaningful benefit to the use of a 3D environment. However, most of the previous research aimed at understanding the benefits/costs of 3D for air traffic control has been conducted using today's ATC task. The use of a 3D perspective to gain and maintain a "bigger picture" by the controller has not been one of the tasks considered when evaluating its benefits. Therefore, the ability to view the airspace in 3D is a feature of the N-CAID prototype, which may prove useful if the controller's is considerably different from today's task.

Conclusions

As new ATC automation concepts are developed, evaluated, and implemented, the roles and responsibilities of the controller will likely change. ATC automation will likely assume greater roles in decision-making and solution implementation. As these changes occur, it is critical to ensure that the controller does not simply become a backup to the automation, forcing him/her to engage in vigilance tasks. Instead, controllers should be used to boost the robustness of the ATC system by performing tasks where the automation lacks proficiency. For example, controllers can serve as valuable sensors in the system by gathering information related to weather, airport runway configurations, etc. and passing on that information to the automation, or in some cases setting the high-level strategy for the automation. Other tasks for the controller include monitoring the solutions generated by the automation, approving or disapproving those solutions as necessary, and informing the automation of events or restrictions it should incorporate into its future solutions. The controller continues to be a key part of the ATC system, and his/her role is <u>not</u> to be a backup to the automation, but rather, a partner of the automation. N-CAID offers a paradigm for controller-automation interaction that will maximize the overall performance of the ATC system and support an effective controller-automation partnership.

References

Burnett, M. S., and Barfield, W. (1991). Perspective versus plan view air traffic control (ATC) displays: Survey and empirical results. Proceedings of the Sixth International Symposium on Aviation Psychology (pp. 448-453). Columbus, OH.

Brudnicki, D. J., McFarland, A. L., & Schultheis, S. M. (1996). Conflict probe benefits to controllers and users. Technical Report (MP96W0000194). McLean, VA: MITRE.

Church, G. (2008). NextGen: New role for air traffic controllers. *Air Traffic Control*, 50, 13 – 15.

Danaher, J. W. (1980). Human error in ATC system operations. *Ergonomics*, 36, 1111-1120.

Dixon, S., & Wickens, C. D. (2003). *Imperfect automation in unmanned aerial vehicle flight control*. Technical Report AHFD-03-17/MAAD-03-2. Savoy, IL: University of Illinois, Aviation Human Factors Division.

Dixon, S. R., & Wickens, C. D. (2004). *Reliability in automated aids for unmanned aerial vehicle flight control: Evaluating a model of automation dependence in high workload.* Technical Report AHFD-04-05/MAAD-04-1. Savoy, IL: University of Illinois, Aviation Human Factors Division.

Endsley, M. R., & Kaber, D. B. (1999). Level of automation effects on performance, situation awareness and workload in a dynamic control task. *Ergonomics*, 42, 462 – 492.

Fitts, P. M. (1951). Human engineering for an effective air navigation and traffic control system. Ohio State University Foundation report. Columbus, OH.

Hancock, P. A., & Scallen, S. F., (1996). The future of function allocation. *Ergonomics in Design*, 4, 24-29.

Hilburn, B., & Jorna, P. G. A. M. (2001). Workload and air traffic control. In P. A. Hancock and P. A. Desmond, (Eds). Stress, Workload, and Fatigue. Human Factors in Transportation, Lawrence Erlbaum Associates: Mahwah, NJ

Hilburn, B., Jorna, P. G. A. M., & Parasuraman, R. (1995). The effect of advanced ATC strategic decision aiding automation on mental workload and monitoring performance: An empirical investigation in simulated Dutch airspace. In *Proceedings of the eight international symposium on Aviation Psychology*. Columbus: Ohio State University

Joint Planning and Development Office (2007). Concept of Operations for the Next Generation Air Transportation System, Version 2.0.

Kerns, K., & McFarland, A. L. (1998). Conflict probe operational evaluation and benefits assessment. Technical Report (MP98W0000239). McLean, VA: MITRE.

Kopardekar, P., Bilimoria, K., D., & Sridhar, B. (2008). Airspace configuration concepts for the Next Generation Air Transportation System. *Air Traffic Control Quarterly*, 16, 313 – 336.

Kopardekar, P., Prevot, T., & Jastrzebski, M. (2009). Traffic complexity measurement under high levels of automation and higher traffic densities. *Air Traffic Control Quarterly*, 17, 125-148.

Lee, J. D., & Moray, N. (1994). Trust, self-confidence and operator's adaptation to automation. *International Journal of Human Computer Studies, 40*, 153-184.

Lee, J. D., & See, K. A. (2004). Trust in automation: Designing for appropriate reliance. *Human Factors, 46*, 50-80.

Lee, K. K. & Sanford, B. D. (1998). The passive Final Approach Spacing Tool (pFAST) Human Factors operational assessment. *Proceedings of the 2nd USA/Europe Air Traffic Management R&D seminar*. Orlando, FL.

Maltz, M., & Meyer, M. (2001). Use of warnings in an attentionally demanding detection task. *Human Factors*, *43*, 217 – 226.

Metzger, U. & Parasuraman, R. (2005). Automation in future air traffic management: Effects of decision aid reliability on controller performance and mental workload. *Human Factors, 47*, 35-49.

Molloy, R., & Parasuraman, R. (1996). Monitoring an automated system for a single failure: Vigilance and task complexity effects. *Human Factors, 38*, 311-322.

Moray, N., Inagaki, T., & Itoh, M. (2000). Adaptive atomation, trust, and self-confidence in fault management of time-critical tasks. *Journal of Experimental Psychology: Applied*, *6*, 44-58.

Parasuraman, R., Molloy, R., & Singh, I. (1993). Performance consequences of automation-induced "complacency." *The International Journal of Aviation Psychology*, 3, 1-23.

Parasuraman, R., Sheridan, T. B., & Wickens, C. D. (2000). A model for types and levels of human interaction with automation. *IEEE Transactions on Systems, Man, & Cybernetics,* 30, 286-297.

Pritchett, A. R. (2008). NextGen Human Factors research and engineering requirements. *Proceedings of the Human Factors & Ergonomics Society 52nd Annual Meeting*. Santa Monica, CA

Rodgers, M. D. & Dreschsler, G. K. (1993). Conversion of the CTA Inc., En Route Operations Cocnepts database into a formal sentence outline job task taxonomy. DOT/FAA/AM-93/1. Washington, D.C., Department of Transportation, Federal Aviation Administration, Office of Aviation Medicine.

Sanchez, J., & Duncan, J. R. (2009). Growing automation in agricultural vehicles yields new operator requirements. *Ergonomics in Design, 17*, 14 – 19.

Sanchez, J., Fisk, A. D., & Rogers, W. A. (2004). Reliability and age-related effects on trust and reliance of a decision support aid. *Proceedings of the Human Factors & Ergonomics Society 48th Annual Meeting*. Santa Monica, CA

Sanchez, J., & Zakrzewski, E. (2009). The impact of ATC automation on a controller's visual attention. *IEEE Proceedings of the Digital Avionics and Surveillance Communications (DASC) Conference*. Orlando, Fl.

Shepley, J. P., Johnson, C. M., Sanchez, J., & Smith, E. C. (2009). Eye-tracking analysis of near-term terminal automation for arrival coordination. *Proceedings of 9th American Institute of Aeronautics and Astronautics (AIAA) Aviation, Technology, Integration, and Operations Conference (ATIO)*, Hilton Head, SC.

Skerve, A. B. M., & Skraaning, G. (2004). The quality of human-automation cooperation in human-system interface for nuclear power plants. International Journal *of Human-Computer Studies*, 61, 649 – 677.

Skitka, L. J., Mosier, K. L., & Burdick, M. (1999). Does automation bias decision-making? *International Journal of Human-Computer Studies*, *51*, 991-1006.

St. John, M., & Manes, D. I. (2002). Making unreliable automation useful. *Proceedings of the Human Factors and Ergonomics Society 46th Annual Meeting,* Santa Monica, CA: Human Factors and Ergonomics Society.

Tham, M., and Wickens, C. D. (1993). Evaluation of perspective and stereotopic displays as alternatives to plan view displays in air traffic control. University of Illinois Institute of Aviation Technical Report (ARL-93-4/FAA-93-1). Savoy, IL: Aviation Research Lab.

Treisman, A. (1986). Features and objects in visual processing. *Scientific American, 255*, 114-125.

Whitfield, D., Ball, R. G., & Ord, G. (1980). Some human factors aspects of computer-aiding concepts for air traffic controllers. *Human Factors, 22*, 569-580.

Warm, J. S. (1984). An introduction to vigilance. In J. S. Warm (Eds.). *Sustained attention in human performance.* (pp. 1 – 10). New York, NY: John Wiley & Sons.

Wickens, C. D. (1995). Display integration of air traffic control information: 3D displays and proximity compatibility. University of Illinois Institute of Aviation Technical Report (ARL-95-2/FAA-95-2). Savoy, IL: Aviation Research Lab.