Methodology to Support Dynamic Function Allocation Policies Between Humans and Flight Deck Automation

Final Report
Cooperative Agreement NNL06AA22A

submitted to
NASA Langley Research Center
Hampton VA 23681-2199

for the period
October 1, 2006 – September 30, 2010.

Attention: Paul C. Schutte, Technical Monitor

Eric N. Johnson, Ph.D., Principal Investigator
Schools of Aerospace Engineering
Georgia Institute of Technology
Atlanta, GA 30332-0150
(Tel) 404-385-2519
(Fax) 404-894-2760
Eric.Johnson@ae.gatech.edu

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

Function allocation assigns work functions to all agents in a team, both human and automation. Efforts to guide function allocation systematically have been studied in many fields such as engineering, human factors, team and organization design, management science, cognitive systems engineering. Each field focuses on certain aspects of function allocation, but not all; thus, an independent discussion of each does not address all necessary aspects of function allocation. Four distinctive perspectives have emerged from this comprehensive review of literature on those fields: the technology-centered, human-centered, team-oriented, and work-oriented perspectives. Each perspective focuses on different aspects of function allocation: capabilities and characteristics of agents (automation or human), structure and strategy of a team, and work structure and environment.

Together, these perspectives identify the following eight issues with function allocation:

1) *Workload*
2) *Incoherency in function allocations*
3) *Mismatches between responsibility and authority*
4) *Interruptive automation*
5) *Automation boundary conditions*
6) *Function allocation preventing human adaptation to context*
7) *Function allocation destabilizing the humans' work environment*
8) *Mission Performance*

To address these issues systematically requires formal models and simulations that include all necessary aspects of human-automation function allocation: work, environment, agents, their inherent dynamics, and the relationships among them. Also, to address these issues requires not only a (static) work model that describes the structure of

the work and the relationships among them, but also a (dynamic) simulation that captures temporal aspects such as the timing of actions and their impact on the environment. Therefore, with properly modeled work as described by the environment, agents, their inherent dynamics, and relationships among them, the framework which includes a static work model and a dynamic simulation can capture occurrences of the previously-identified issues with function allocation.

Then, based on the eight issues, eight types of metrics are established. The purpose of these metrics is to assess the extent to which each of issues exist on a given function allocation. Specifically, the eight types of metrics assess workload, incoherency in a function allocation, mismatches between responsibility and authority, interruptive automation, automation boundary conditions, human adaptation to context, stability of the human's work environment, and mission performance.

Finally, to validate the modeling framework and the metrics, a case study modeled four different function allocations between a pilot and flight deck automation during the arrival and approach phases of flight. A range of pilot cognitive control modes and maximum human taskload capacities were also included in the model. The metrics for the four function allocations were assessed and analyzed to validate their capability to identify important issues in function allocation.

This report concludes with a discussion of mechanisms for further validating the modeling framework and function allocation metrics developed here, and highlights where these developments can be applied in research and in the design of function allocations in complex work environments including aviation operations.

Contributors to the project at Georgia Tech included: So-Young Kim, Amy Pritchett (original P.I.), Seung-Man Lee, Karen Feigh, Suresh Kannan, Matt Bigelow, and H. Claus Christmann.

CHAPTER 1

INTRODUCTION

Function allocation refers to the distribution of functions among humans and machines in complex systems (Sherry & Ritter, 2002). Thus, function allocation is the design decision which assigns work functions to all agents in a team, both human and automation. The function allocation for a human-automated system should be designed depending on the context in which the system is operated. If functions are allocated properly, it maximizes mission performance by best utilizing the capabilities of each agent and provides the environment that fosters their individual performance and that promotes effective interactions within the team; thus, human and automated team members can each best contribute to the overall goals of their collective work.

Function allocation, in some situations, may be represented as broad specifications of high-level responsibilities. However, in situations such as the flight deck operations, detailed function allocations may need to capture intricate couplings between low-level tasks, such as the inter-relation between a pilot's control of pitch together with an autothrottle's control of speed.

As an additional distinction, function allocations may be static or dynamic. At one extreme, a single function allocation may dictate a fixed set of functions for all team members from which no deviation is tolerated. At the other extreme, any function may be allocated dynamically at any time to any agent in response to agent capabilities and availability, and in response to events in the environment. In between these extremes, a set of function allocations may be pre-determined for agents in the team to invoke as appropriate to the situation.

1.1 Problem Statement

While several guidelines for function allocation have been proposed over the last decade, each represents a limited perspective. The Fitts List (Chapanis et al., 1951), for example, focuses on the capabilities of the human and automation without intrinsically examining the coherency of the allocation, the ultimate responsibility for outcomes, the team interactions, or the overall relationship to mission goals.

In addition, current human factors guidelines for function allocation are comparatively abstract. For example, desired attributes of automation include that it should be a “good team member” and “not clumsy.” While these attributes are generally agreed to be necessary (with some exceptions, for example, see Pritchett, 2001 for a discussion of when the purpose of alerting systems is to be clumsy), they are not sufficiently specific to enable comparison of the merits of similar function allocations. Such comparison is not only necessary during design, but, if feasible during operations, could provide a rigorous basis for dynamic function allocation. Thus, a purpose of this effort is to provide metrics of function allocation specific enough to enable comparison of function allocations, especially during design and during operations with dynamic function allocation. These metrics must be sufficiently comprehensive to identify key issues with function allocation that cannot be observed from a single perspective.

1.2 Objectives

The first and foremost objective of the effort was to establish metrics of human-automation function allocation that can predict a comprehensive set of known issues with function allocation. These metrics must be sufficiently specific to guide designs and dynamic function allocation.

These metrics require a model of the team and its work that instigates the second objective of this effort: to develop a modeling framework by which function allocation can be modeled and from which the metrics can be assessed.

The third and last objective of the effort was to validate the metrics and the modeling framework via a case study. The metric set is considered to be validated if it accurately captures key issues with different function allocations.

1.3 Report Overview

The report is structured as follows: this chapter introduces the motivation, problem statement, and objectives. Chapter 2, first, illustrates how human and automation can be allocated functions in a flight deck during arrival and approach phases and, second, discusses four perspectives on human-automation function allocation (technology-centered, human-centered, team-oriented, and work-oriented perspectives), identifying the key issues with function allocation that each reveals. These key issues are then summarized into eight categories that span the various perspectives, which then identify the need for eight types of metrics of function allocation.

Chapter 3 describes the requirements for a modeling framework suitable for assessing these metrics of function allocation, Work Model that Computes (WMC). WMC is built on cognitive engineering principles to generate analytic and computational representations of the tasks, their allocation, and the broader operating environment, and to incorporate a computational human performance model capable of predicting and quantifying the performance and safety impact of function allocation designs.

Chapter 4 builds on the previous two chapters, illustrating specifically how metrics of the FA issues identified in Chapter 2 can be systematically and unambiguously evaluated by static and dynamic measures of models developed in the framework described in Chapter 3.

Finally, Chapter 5 describes the case study of aircraft arrivals and approaches with a range of current and near-term function allocations, assessing the function allocation metrics with each. The experiment's four independent variables are the scenarios, the function allocations, the pilot's cognitive control modes, and the maximum capacity of humans. The experiment's dependent variables are the function allocation metrics proposed in Chapter 3. The chapter ends with a discussion of the extent to which the metrics and modeling framework capture key issues with function allocation.

Finally, Chapter 6 concludes the report by summarizing the developments across the report. The contributions of the report are discussed, highlighting how the results contribute to models of the joint work of humans and automation, to scientific understanding of issues with function allocation, and to designers in specifying function allocations. Finally, recommendations are provided for future work.

CHAPTER 2

HUMAN-AUTOMATION FUNCTION ALLOCATION

Function allocation is the design decision that assigns work functions to all of the agents in a team, both human and automated. This chapter demonstrates function allocation using an example of a range of function allocations in a flight deck during the arrival and approach phases of flight. This example is particularly relevant because it has historically experienced multiple issues with function allocation and, also, because it serves as the case study examined in Chapter 5. This chapter, next, provides a broad review of function allocation from four perspectives that emerged from the literature: the technology-centered, human-centered, team-oriented, and work-oriented perspectives. From this discussion, eight issues with function allocation are identified, many of which span findings from multiple perspectives. These issues can be described or predicted via models that will be discussed in Chapter 3. Then, Chapter 4 will define specific metrics of function allocation that can assess these issues from the models or from operational data.

2.1 Flight Deck Function Allocation for Flight Path Management During the Arrival and Approach Phases of Flight

A commercial flight is generally composed of six phases: takeoff, departure (climb), cruise, arrival (descent), approach, and landing. Each phase requires a different set of functions to achieve its goals. These functions may be allocated between pilots and flight deck automation. Among the flight phases, the arrival and approach phases are usually considered the most difficult ones for pilots to fly well (Casner, 2001) because these phases have the tightest requirements on performance, requiring intimate teamwork between the pilot and the flight deck automation. The goal of the arrival and approach phases is to descend the aircraft while maintaining a stable energy profile to establish the aircraft at the right location, altitude, and airspeed for the final approach.

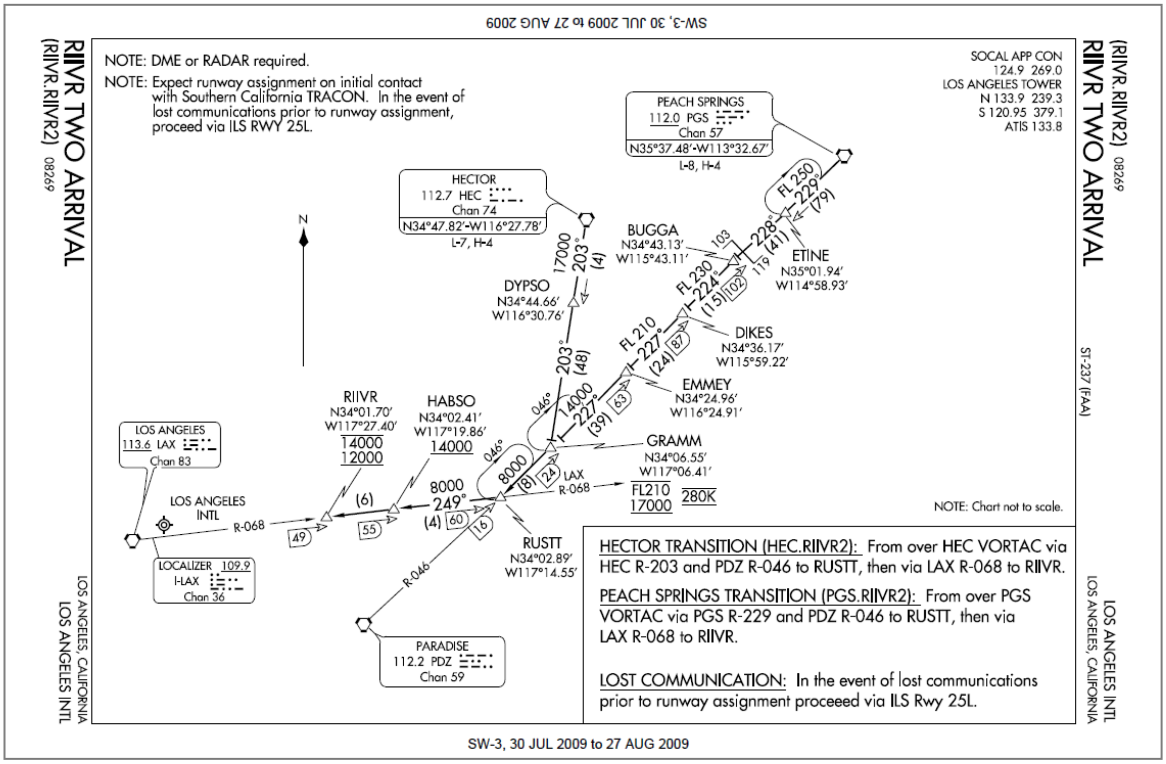

Figure 1. An example of a STAR chart, RIIVR TWO ARRIVAL towards LAX

To facilitate the work of the pilot and the flight deck automation during these phases, standard terminal arrival routes (STAR) and an instrument approach procedures (e.g., as illustrated in the STAR chart and approach plate shown in Figure 1 and Figure 2, respectively) are provided as established standard operating procedures. These standard operating procedures predefine many aspects of the arrival and approach phases such as waypoints, heading, airspeed, altitude, etc. When air traffic controllers instruct aircraft to follow the RIIVR TWO ARRIVAL in Figure 1, for example, the pilot and the flight deck automation are effectively cleared through a series of waypoints, each of which may have corresponding altitude and/or speed restrictions, without requiring communication of the details of the procedure.

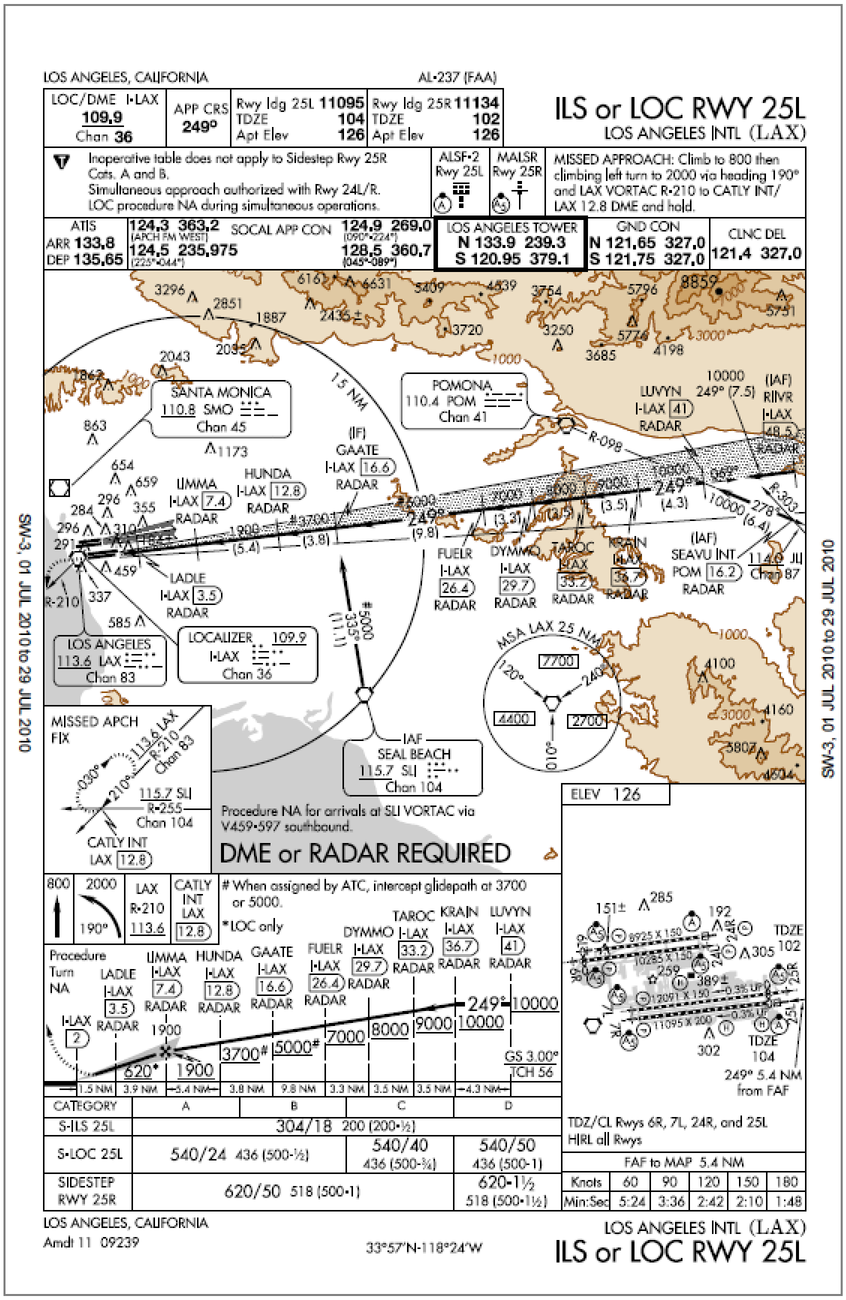

Figure 2. An example of an instrument approach plate, ILS or LOC RWY 25L at LAX

During the arrival and approach phases, the pilot and the flight deck automation, together, need to fly the aircraft in a timely manner, to minimize fuel burn, and to maintain flight safety. Achieving these goals requires several tasks: aircraft control, trajectory management, communication (with air traffic controllers) management, and flight regulation management (i.e., ensuring that the flight trajectory of the aircraft is within the allowed path and that it is achievable without compromising the flight safety). The aircraft control task includes determining actuator settings (the control surfaces of the aircraft) and engine settings to achieve targets for heading, airspeed/thrust, and altitude/vertical speed. These targets are calculated to follow the assigned flight route or air traffic instructions. The pilot and the flight deck automation are also required to manage the trajectory (ensuring the trajectory follows the assigned flight path) while interacting with air traffic controllers. In addition, flight safety requires that all the aircraft systems are managed correctly and that safe separation is maintained from other aircraft.

The arrival and approach phases are initiated when the aircraft reaches the top of descent, which is a calculated position where the aircraft can perform a descent while achieving an optimal fuel usage, an expected time of arrival, or both. The air traffic controller clears the aircraft for descent-via instructions that may specify the entire arrival route, a certain waypoint, or simply a lower altitude. The pilot and the flight deck automation then initiate the descent to achieve the targets for heading, airspeed/throttle, altitude/vertical speed, and waypoints. Achieving these targets requires functions that manage aircraft energy not only by controlling the control surfaces and throttle, but also by managing the aircraft configuration (e.g., flaps, gears, and speed brakes). Although many of the functions required during the arrival and approach phases can be allocated to the flight deck automation or to the pilot, some functions can only be assigned to the pilot for technical and regulatory reasons: for example, deploying flaps, gear, and speed brakes can only be done by the pilot because these functions are currently not automated.

In addition, a safety-ensuring mechanism has been designed into the flight deck: every altitude clearance must be entered by the pilot as an altitude target in the mode control panel (MCP, which will be described in detail in Section 2.1.1). This altitude target then serves as a visible reminder to the pilot and as the altitude target to the flight deck automation so that the aircraft will not descend below the assigned altitude (which may reflect a minimum safe altitude).

As another mechanism to ensure flight safety, multiple operating procedures have been established. The descent checklist, approach checklist, and landing checklist are composed of multiple steps ensuring the flight deck systems are configured properly, and the pilot and cabin crews are "briefed" for upcoming phases. These operating procedures can be only performed by the pilot because many of the flight deck systems must be monitored and configured manually, and the pilots could rehearse the upcoming route of flight and likely events.

As described above, these phases of flight require multiple functions. These functions can be allocated to the pilot or the flight deck automation. The following section 2.1.1 describes different function allocations available in the flight deck, focusing on flight path management.

2.1.1 Available Function Allocations Between and Flight Deck Automation During the Arrival and Approach Phases of Flight

Current flight deck automation includes a flight management system (FMS), an autopilot system, and an autothrottle system. The FMS determines a trajectory by a set of waypoints, some with altitude and/or speed restrictions. (At any given time during the flight, the pilot may enter new [or modified] waypoints and restrictions.) The FMS is capable of calculating an optimal trajectory that can satisfy these restrictions. This specification for a trajectory is, then, translated into immediate targets for heading, altitude/ vertical speed, and throttle/airspeed.

The autopilot and autothrottle systems (together commonly referred to as the autoflight system) take these targets for heading, altitude/vertical speed, and throttle/airspeed and employ specific "control modes" to determine actuator settings. The control modes specify the autoflight system's behavior in terms of how to track which target. Different control modes may be appropriate at different times: the "Vertical Speed" mode, for example, tracks a given vertical speed target using pitch; the "Altitude Capture" mode identifies where the autoflight system should initiate a level-off using pitch; and the "Altitude Hold" mode maintains the target altitude using pitch.

A modern autoflight system may encompass hundreds of control modes, some of which differ subtly in their behaviors. Therefore, the flight deck automation provides pilots with flight mode annunciators (FMAs) indicating the pitch, roll, and thrust control modes, target values commanded to the autoflight system, and current flight route information. The FMAs and targets are provided throughout the multiple interfaces in the flight deck, including the navigation display (ND) and the primary flight display (PFD), shown in Figure 3 and Figure 4, respectively.

Figure 3. Navigation display (photo retrieved from www.meriweather.com/747/fd-747.html)

The ND provides a horizontal planar view of the area ahead of the aircraft, its heading, and the waypoints defining the flight route currently "programmed" in the FMS.

For example, Figure 3 shows the track heading (066°) in magenta at the top center, the ground speed (480knots) and the true air speed (350knots) in magenta and blue at the top left corner, the path to the waypoint KYIG in the center, and also the vertical deviation indicator at the bottom right corner.

Figure 4. Primary flight display (photo retrieved from www.airliners.net)

The PFD portrays the basic states of the aircraft including attitude, altitude, speed, vertical speed, and heading. Of special interest in this interface is the addition of target values and FMAs. In Figure 4, for example, the FMAs are shown at the top: this aircraft is in SPD, LNAV, VNAV PTH modes. The heading, altitude, and speed targets of the autoflight system are displayed regardless of where the targets are determined (i.e., in the FMS or programmed by the pilot into the MCP). In Figure 4, for example, these targets are shown explicitly in magenta on the heading indicator at the center bottom (143°), on the airspeed tape on the left (a pointer to 304knots as well as text on the top of the tape implicating Mach 0.85), and on the altitude tape on the right (a bracket 33,000, as well as text above the tape indicating the same 33,000). The following table provides an overview of control modes and corresponding FMAs commonly used during the arrival and approach phases.

Table 1. Boeing 747-400 fight mode annunciators (adapted from Casner 2001)

Guidance Function	How it works	Flight Mode Annunciations		
		Roll	Pitch	Thrust
LNAV	Roll is used to track the waypoints in the flight route that defined in the CDU.	LNAV		
Heading Hold	Roll is used to maintain the heading dialed into the heading window in the MCP.	HDG HOLD		
VNAV (During descent)	Thrust is idle. Pitch is tracking the planned vertical profile.		VNAV PTH	HOLD
	Thrust is idle. Pitch is used to track the descent airspeed.		VNAV SPD	THR
	Pitch is used to maintain the altitude dialed into the altitude window in the MCP (only when the next target altitude is lower than the altitude indicated in the MCP).		VNAV ALT	SPD
Vertical Speed	Thrust is used to maintain the speed dialed in the speed window in the MCP. Pitch is used to maintain the vertical speed dialed in the vertical speed in the MCP.		V/S	SPD
Flight Level Change	Thrust is idle. Pitch is used to maintain speed dialed in the speed window in the MCP, a vertical speed results descent (or climb) to a new flight level.		FLCH SPD	HOLD
Altitude Hold	Thrust is used to maintain the speed dialed in the speed window in the MCP. Pitch is used to maintain the altitude dialed into the altitude window in the MCP.		ALT	SPD

Finally, the following sections (2.1.1.1 to 2.1.1.4) describe four different function allocations of the flight path management between the pilot and the flight deck automation that are either currently available or foreseeable in the near future, ranging from highly-automated to mostly-automated, mixed, and mostly-manual ones.

2.1.1.1 Pilot Using LNAV/VNAV with Air Traffic Instructions Directly Processed by the Flight Deck Automation

This represents a "highly-automated" function allocation that has not yet been implemented. This function allocation assumes a new concept of operation in which air traffic instructions, in the form of an assigned trajectory, can be communicated from the air traffic controllers directly into the FMS using digital data link (i.e., data communication to the flight deck automation as opposed to voice communication to the pilot).

In this function allocation, the flight deck automation is assigned to controlling the aircraft and managing the trajectory (i.e., calculating the autoflight system targets). Meanwhile, the pilot is assigned to managing aircraft configuration and performing operating procedures. In addition, although not explicitly assigned, the pilot is expected to remain vigilant, verifying the aircraft states, monitoring the flight deck automation's ability to satisfy air traffic restrictions, and ensuring that the flight deck automation is acting upon the proper data. (For example, verifying whether the correct arrival and approach are "programmed" into the FMS.) These implicit monitoring functions assigned to the pilot are mostly aided by the flight deck automation displaying the aircraft states and other environmental information. However, the responsibility to monitor and identify an abnormality of the flight deck remains with the pilot.

This function allocation allows (or shapes) the interactions between the pilot and the flight deck automation to operate as follows: the flight deck automation calculates its anticipated top of descent point (T/D point, specified by altitude, latitude, and longitude as the optimal position to initiate the descent). Usually, the flight deck automation receives an air traffic instruction to start the descent to a lower altitude before the aircraft reaches the T/D point. The flight deck automation then processes the altitude instruction and updates the autoflight system's target altitude. This new target altitude serves as an immediate restriction for the autoflight system to capture. The automation engages the VNAV PTH control mode with idle thrust, and the aircraft starts descending. If the air traffic controller instructions require an earlier descent, the trajectories may be required to "step-down" via a series of assigned altitudes, or may follow a continuous path that is shallower and slower than optimal. Conversely, if the controller instructions require a later descent, the flight deck automation may not be able to meet all its air traffic restrictions without the pilot intervening with speed brakes. Figure 5 illustrates these potential cases of initial descent: earlier, planned (optimal), and later descent.

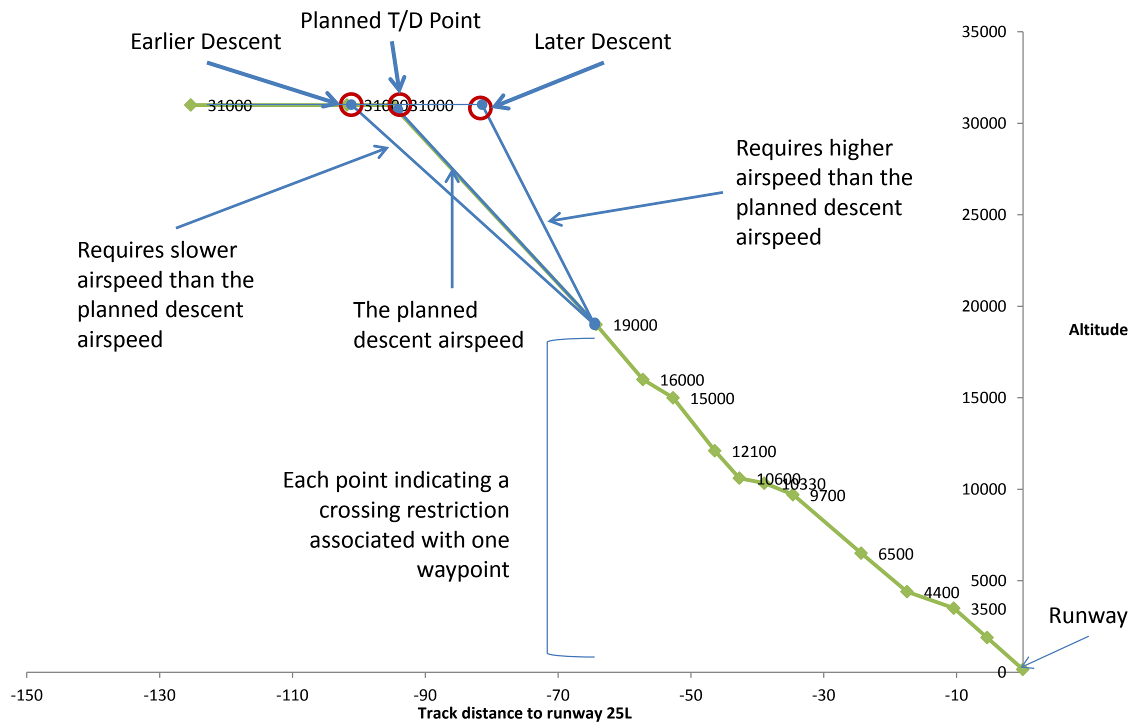

Figure 5. An example of a vertical profile of the arrival and approach phases with three potential initiation of descent (early, normal, and late descents) assuming the air traffic instruction is given as "descend to flight level 190")

Figure 6. ND with vertical deviation indicator highlighted in red box (photo retrieved and adapted from http://www.meriweather.com/747/fd-747.html)

When the aircraft descends in the VNAV PATH control mode, a vertical deviation indicator appears at the bottom right corner of the ND, highlighted with a red box in Figure 6. The indicator places the diamond on the center of the scale while the aircraft is on profile, and the upper and lower bars display a range of ± 400ft above/below profile, respectively.

Because the flight deck automation is assigned to managing the lateral and vertical profiles, it is responsible for monitoring the environmental factors that perturb them. If the airspeed falls 15 knots below the planned descent airspeed due to an unanticipated headwind, the flight deck automation responds by commanding higher thrust to the autothrottle (Casner, 2001; Stimpson, 2010). As the airspeed increases, the aircraft recaptures the planned vertical profile. On the other hand, if there is an unanticipated tailwind, the aircraft will drift above the planned vertical profile. The flight deck automation responds by commanding a pitch-down maneuver to the autoflight system, which then causes the airspeed to increase. When the airspeed is 10 knots higher than the planned descent airspeed, the FMS requests the pilot's intervention by displaying "DRAG REQUIRED" (Stimpson, 2010). The pilot is, then, required to deploy the speed brakes. If the aircraft cannot capture the planned vertical profile even with the additional drag from the speed brakes, and the deviation from the vertical profile becomes more than 400ft, a VNAV SPD control mode is triggered, tracking the target airspeed instead the vertical profile and thus ignoring any air traffic restrictions required by an air traffic controller. Thus, the pilot's task in this function allocation focuses on monitoring the behavior of the aircraft and the flight deck automation. If the flight deck automation cannot satisfy air traffic restrictions, then the pilot is responsible for reporting this situation to the air traffic controllers.

2.1.1.2 Pilot Using LNAV/VNAV with Pilot Receiving Air Traffic Instructions and Programming the Autoflight System

This function allocation represents the "mostly-automated" one in current operations. Compared to the highly-automated function allocation, the pilot is now responsible for monitoring for and receiving air traffic instructions and programming them into the autoflight system.

In this function allocation, the flight deck automation is assigned to controlling aircraft and managing trajectory. Meanwhile, the pilot is assigned to managing aircraft systems and managing communication with air traffic controllers. In addition, the pilot is assigned to the implicit functions of monitoring and verifying information in the flight deck and the environment.

Figure 7. Boeing 747-400 CDU

This function allocation allows (or shapes) the interactions between the pilot and the flight deck automation to be different from to those with the highly-automation function allocation. The difference is in how air traffic instructions are programmed into the autoflight system (i.e., in this function allocation, the pilot programs air traffic instructions into the autoflight system). The pilot is still required to monitor the flight deck automation, managing the aircraft configuration by deploying flaps and speed brakes when necessary, managing operating procedures in the flight deck, and verifying and confirming information provided to and from the FMS. To facilitate the interaction between the pilot and the flight deck automation, an interface that allows the pilot to coordinate and communicate with the FMS is provided: the control display unit (CDU). The CDU incorporates a screen and a keyboard (or a touch-screen) as shown in Figure 7 by which the pilot can program waypoints and their restrictions into the FMS.

2.1.1.3 Pilot Programming the Vertical Targets of Autoflight System and Receiving Air Traffic Instructions, and the FMS Commanding the Lateral Autoflight Targets

This function allocation represents a "mixed" case in that the flight path management task is "distributed" between the pilot and the flight deck automation. (The previous two function allocations assign this task entirely to the flight deck automation.)

In this function allocation, the flight deck automation is assigned to controlling the aircraft and managing the lateral trajectory. Meanwhile, the pilot is assigned to managing aircraft systems, communicating with air traffic controllers, and managing the vertical profile. Thus, the pilot is responsible for calculating target altitude and speed and engaging the appropriate control modes in the autoflight system. In addition, the pilot is assigned to the implicit functions of verifying and monitoring adherence to the required trajectory except that no vertical deviation indicator is provided to the pilot. Instead, the pilot must directly estimate the vertical profile and predict any violations of air traffic restrictions.

This function allocation establishes interactions between the pilot and the flight deck automation as follows: when the air traffic controller clears the pilot to initiate the descent, the pilot updates the target altitude and speed of the autoflight system via the MCP while the FMS commands the target heading to the autoflight systems directly based on the target waypoint programmed in the FMS. Figure 8 provides an example of an MCP. Whenever the air traffic controller instructs new altitudes and airspeeds, the pilot needs to update the altitude and airspeed targets using the MCP and to track them using guidance functions provided in the MCP. If the air traffic controller instructs changes to the lateral path, the pilot is responsible for programming it into the CDU, and the autoflight system translates this changed route information into a target heading for tracking the lateral flight path.

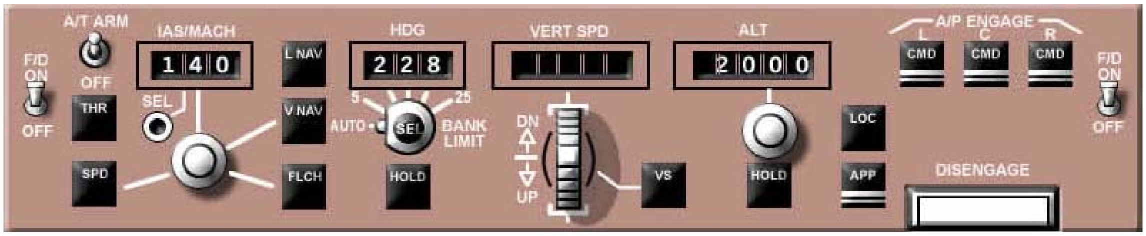

Figure 8. Mode control panel (MCP) in Boeing 747 (photo retrieved from http://www.meriweather.com/747/fd-747.html)

2.1.1.4 Pilot Programming the Targets of the Autoflight System and Receiving Air Traffic Instructions

This function allocation is "mostly-manual" as the entire flight path management task is assigned to the pilot. The flight deck automation is assigned to controlling the aircraft. Thus, the pilot is assigned to managing aircraft systems, communicating with air traffic controllers, and managing the flight path. Thus, the pilot is responsible for calculating target heading, altitude, and speed and engaging the appropriate control mode in the autoflight system via the MCP. In addition, the pilot is assigned to the implicit functions of verifying and monitoring adherence to the required trajectory.

This function allocation establishes the interactions between the pilot and the flight deck automation as follows: the pilot manages the flight path by programming all target values of the autoflight system using the MCP, including heading to follow the trajectory specified by the arrival and approach procedures and by air traffic instructions. Thus, when the air traffic controller instructs the initiation of the descent, the pilot needs to update the target altitude, speed, and heading into the MCP and to engage proper control modes to track those target values. Whenever the aircraft reaches a waypoint (or other transition), the pilot needs to update the targets and control modes. In addition, the pilot also remains responsible for monitoring the autoflight system and all the other management tasks assigned to the pilot with the previous function allocations.

2.1.2 Operational Issues with Function Allocation during Arrival and Approach

Since the introduction of automated systems in the flight deck, many operational issues have been observed. Of particular interest here are the issues with flight deck automation observed during (or relevant to) the arrival and approach phases of flight.

One of the most apparent issues with flight deck automation has been workload. Wiener and Curry (1980) described the issues with workload explicitly in their observational study of automation use in aviation. They noted that, although the "manual" workload (i.e., workload due to manual functions such as moving control yokes or throttle levers) decreased with the implementation of automation, a different, more cognitive type of workload had been introduced: therefore, the total workload that pilots experienced had increased. Wiener (1989b) also conducted a survey study with pilots who have been flying aircraft equipped with advanced flight deck automation. This study showed that the workload had indeed increased. More than half of the pilots who participated in the survey agreed with the statement, "Automation did not reduce total workload." In fact, the pilots believed that the introduction of automation in flight deck increased the workload due to the requirement of reprogramming the FMS (Wiener, 1985). Worse, these demands from the automation increase precisely at the phases of flight (such as arrival and approach) when the demands from other tasks increased (Parasuraman & Riley, 1997), resulting in workload spikes (for short-term demands) or workload saturation (for longer-term demands).

The next issue observed from operations is that pilots do not have the appropriate level of understanding of the automation's capabilities and limitations (i.e., boundary conditions). For example, in 1994, an A300 crashed in Nagoya due to the conflicting actions between the autopilot and the pilot flying (the first officer). During the approach phase of the flight, the first officer inadvertently activated the "Go-Around" control mode which caused the autoflight system to halt the approach and initiate a climb by increasing thrust and setting horizontal stabilizer to nose-up trim. However, the first officer did not disengage this incorrect mode although the captain recognized and called out that the Go-Around control mode was engaged. The first officer instead maneuvered the control wheel to achieve the nose-down pitch to continue the approach. With the autopilot being engaged to the Go-Around control mode, the first officer was commanding the elevators, and the autopilot was commanding the thrust and the horizontal stabilizer to achieve conflicting goals (one being attempting to continue the approach, the other being attempting to halt the approach and climb up). At this point, the first officer felt significant resistance on the control column. Unknown to the first officer, this resistive force was the indication of the flight deck automation's intention to convey its goal to halt the approach. Likewise, the first officer was pushing hard on the controls (which could be the indication of his goal), but the autopilot did not recognize the need to disengage to allow the first officer to achieve his goals. The first officer's lack of understanding of the characteristics of the flight deck automation and the flight deck automation's lack of capability to interpret his intention directly led to the resulting crash and fatal casualties (description adapted from Leiden, Keller & French, 2002).

The Nagoya accident of 1994 is an example of incidents caused by pilots' (lack of) understanding of the flight deck automation's behavior in the flight deck (Abbott, Slotte & Stimson, 1996; Funk & Lyall, 1998, 2000). In addition, the actions of the flight deck automation were not apparent to the pilots which negatively contributed to the pilot's understanding of the flight deck automation's behavior (Funk & Lyall, 2000). The underlying causes of this accident include issues with functions allocation between the pilot and the automation: the function allocation did not allow the pilot to form a coherent description of the work distributed between them. (Also, the causes include "interface" issues outside the scope of this report, i.e., function allocation.)

In addition, automation may be too complex for pilots to understand and monitor (Abbott, et al., 1996; Funk & Lyall, 2000; Javaux, 2002; Wiener & Curry, 1980). Automation became capable of more "control modes." However, this increased capability created a new type of undesirable human-automation interaction termed as "mode confusion" in which pilots are "confused" by uncommanded transitions in mode or unintended outcome of the modes (Sarter & Woods, 1992, 1994, 1995); thus, the automation behaves in a different manner than one that the pilots were expecting (Sarter, Woods & Billings, 1997).

This complexity of the flight deck automation is well represented in the VNAV control mode. One button engages the VNAV control mode of the autoflight system. Yet, engaging VNAV control mode could result in many different behaviors as noted earlier in Table 1 and shown in Figure 9: the VNAV mode tracks a target airspeed using pitch, and, at other times, it tracks a target altitude with pitch, depending on the position of the aircraft relative to the FMS planned vertical profile, the ATC clearance altitude, holding pattern at a fix, etc. (Sherry, Feary, Polson, Mumaw & Palmer, 2001). Although pilot interpretation of the VNAV control mode is aided by the flight deck automation providing the FMAs, the FMAs represent only partial information about which actuator is being used to track a target and which targets it is tracking.

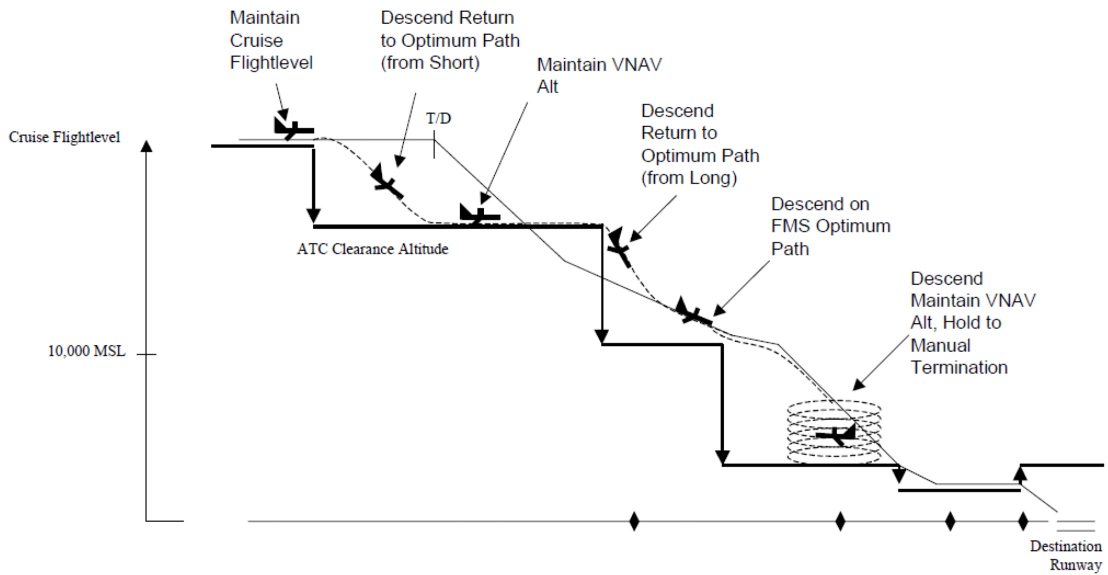

Figure 9. Selecting the VNAV button during a STAR invokes one of the VNAV commanded behaviors depending on the position of the aircraft relative to the FMS optimal path, the ATC clearance altitude, holding pattern at a fix, etc. (figure copied from Sherry, et al., 2001)

The complexity beneath the mode confusions spans not only understanding the current modes but also being able to modify them appropriately in context (Javaux, 1998). Current operations rely on pilots' monitoring and interpreting skills to cope with mode confusion. In addition, pilots are required to monitor a significant amount of other information to ensure proper flight path management, including satisfying air traffic restrictions such as crossing fixes (e.g., waypoints) at a certain altitude or certain airspeed.

In conjunction with this monitoring burden, the information needs to be sampled is scattered across the flight deck displays, and is sometimes not clearly represented (Funk & Lyall, 2000). For example, an American Airlines B757 crashed in Cali, Columbia in 1995. The flight route representations provided in the chart and in the database in FMS used same one character representation to refer to two different waypoints. Therefore, when the air traffic controller instructed the pilots direct to the one of those two waypoint, the pilots selected the incorrect one, and the aircraft followed an incorrect flight path into mountains terrain (accident description adapted from Leiden, et al., 2002).

This example may also reflect another problem with flight deck automation: complacency. Although complacency has many factors, an underlying problem is an allocation of functions that does not highlight a clear relationship to responsibility. Allocating the flight path management to the FMS was clear for example; however, the responsibility to monitor and ensure safety remained with the pilots, requiring monitoring the flight path relative to terrain. However, the pilots in this case abrogated their responsibility for flight safety, either due to competing task demands or a false trust in the automation.

These multiple issues observed during real operations are, in fact, due to common underlying issues of function allocation. As briefly discussed throughout this section, many of these seemingly different issues stem from common ground: how work is distributed between pilots and flight deck automation (i.e., functions allocation) and their teamwork via current flight deck automation interfaces. Building on these findings from observed operational issues, the following sections examine function allocation in a much broader sense including not only pilot-flight deck automation, but also other time-critical and safety-critical human-automated systems.

2.2 Perspectives on Function Allocation

Including the issues described in the previous section, many issues with human-automation function allocation have been raised by studies in multiple fields. Each field focuses on certain aspects of function allocation, but not all; thus, an independent discussion of each does not address all necessary aspects of function allocation. Therefore, it is necessary to review the range of perspectives of function allocation in the literature, and organize the issues raised by them to identify underlying common issues. Four distinctive perspectives have emerged from this comprehensive review of literature on automation design, human factors, team and organizational design, and cognitive systems engineering. These perspectives are termed here as technology-centered, human-centered, team-oriented, and work-oriented perspectives, respectively. Each perspective focuses on different aspects of function allocation: capabilities and characteristics of agents (automation or human), structure and strategy of a team, and work structure and environment. Some of the perspectives have been widely used and have inspired multiple frameworks that attempt to guide and support function allocation. Likewise, some have established theoretical constructs and modeling frameworks that mimic a real system.

A historic basis for all these perspectives is the “Fitts List” (compiled in Chapanis, et al., 1951), shown in Table 2. The effort to develop a systematic approach to function allocation in aviation was initiated with the commonly-called “Fitts Report” edited by Fitts and his colleagues (Chapanis, et al., 1951). It provides a list comparing the capabilities of humans and machines. This list has been widely used to guide function allocation and, also, widely criticized. Therefore, throughout the following sections discussing each perspective, this list will be described to highlight differences the perspectives.

Table 2. Fitts list (table reformatted from Chapanis, et al., 1951)

Humans appear to surpass present-day machines with respect to the following:	
1	Ability to detect small amounts of visual or acoustic energy.
2	Ability to perceive patterns of light or sounds.
3	Ability to improvise and use flexible procedures.
4	Ability to store very large amounts of information for long periods and to recall relevant facts at appropriate time.
5	Ability to reason inductively.
6	Ability to exercise judgment.
Present-day (in 1950s) machines appear to surpass humans with respect to the following:	
1	Ability to respond quickly to control signals and to apply great forces smoothly and precisely.
2	Ability to perform repetitive, routine tasks.
3	Ability to store information briefly and then to erase it completely.
4	Ability to reason deductively, including computational ability.
5	Ability to handle highly complex operations, i.e., to do many different things at once.

The following sections discuss issues raised and highlighted by the four perspectives of function allocation. Each section describes issues observed and identified from each perspective and any models or frameworks used to address these issues. The descriptions given within each perspective tend to be expressed in their vernacular. Thus, although their findings are sometimes described using different terms, they have a basis in common underlying issues with function allocation. Therefore, each section details the issues with function allocation it reveals, which, then, the following sections will categorize into common underlying issues with human-automation function allocation.

2.2.1 Technology-centered Perspective

A technology-centered perspective defines function allocations according to automation's capabilities. This perspective is built upon several assumptions on humans and automation taken from one reading of the Fitts list: First, humans are inherently unreliable and inefficient, and, second, automation can substitute for humans at specific tasks without any impact on the overall performance. With these assumptions as a base, this perspective focuses on expanding machine capabilities to expand what functions can be allocated to automation, and, thus, it values increased machine "autonomy."

This perspective has been widely used in practice and in operation. An example of a design based on this perspective currently exists in modern aircraft, as noted in Section 2.1.2. Today's aircraft are designed to assign almost all possible functions to automation in nominal flight conditions, often improving fuel efficiency and navigation accuracy. However, pilots now have latent responsibilities that are not explicitly described: they must detect and respond to any off-nominal events that might occur with the automation and in the environment, and they must re-format air traffic instructions for data entry into the FMS.

More advanced autoflight systems are being developed (e.g., Johnson, Calise & de Blauwe, 2008). These systems are capable of dynamically responding to changes in the environment, extending the capabilities of the current autoflight system, which is preprogrammed for only small number of reasonably probable emergencies (e.g., engine out), although the flight safety under other adverse flight conditions still depends on pilots' skills and capabilities (Johnson, et al., 2008). Similar studies are seeking to increase overall aircraft safety through dramatic improvements of the autoflight system in terms of stability, maneuverability, and probability of safe landing in the presence of adverse conditions, such as faults, damage, and/or upsets (e.g., Totah, Krishnakumar & Vikien, 2007).

In designs based on this perspective, the humans' assigned functions are scattered across the flight deck and do not necessarily work to their strengths. As many operational studies noted (Bainbridge, 1983; Norman, 1990; Wiener & Curry, 1980), for example, current function allocations based on this perspective often result in designs in which humans are assigned to monitoring automation, despite consistent findings that humans are ineffective in monitoring automation (Lee & Moray, 1992; Molloy & Parasuraman, 1996).

Human operators working with automation also expect to clearly understand the functions assigned to automation. In current operations in flight deck, for example, the captain and the first officer expect an exact specification of the functions assigned to each other such as the function allocation dictated by the roles of “Pilot Flying” (PF) and “Pilot Not Flying” (PNF). However, the technology-centered perspective allocates functions to the automation based on its capabilities. The humans “pick up” the remaining functions that are scattered throughout the work domain. Thus, the structures of the tasks to be performed by the human are inefficient and incoherent, which may even make their overall role ambiguous. Therefore, this highlights an issue with function allocation: *incoherency in function allocation* in which the human “picks up” any functions beyond the automation’s capabilities.

Likewise, whereas “authority” is generally used to describe who is given the resources to perform a function in operational sense, “responsibility” is used to identify who will be held accountable in an organizational and legal sense for the outcome. A function allocation designed from the technology-centered perspective often disregards the necessity of aligning authority and responsibility. 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. This requires the humans to constantly judge whether automation behaves correctly. If the human cannot knowledgably oversee the automation, they need to “trust” the automation. However, without a concrete basis for assessing if the automation is correct humans often over- and under-trust the automation: either way, incorrect trust is viewed as "human error," despite its basis in the function allocation (Parasuraman & Riley, 1997).

Thus, authority and reasonability are often not aligned (i.e., the human who is held responsible does not have the resources and capability to act with authority) in a function allocation driven by the technology-centered perspective. Any mismatch between responsibility and authority will demand heavy monitoring and information seeking efforts from the humans. Further, in some situations, it is questionable whether the humans are given sufficient authority (i.e., the capabilities and the resources to judge and intervene) to override automation's functions if necessary. This situation is termed the "responsibility-authority double-bind" (Woods, 1985). Therefore, an issue with function allocation is highlighted: *mismatch between responsibility and authority* due to function allocation only considering the capabilities of automation.

Regardless of how advanced the technology is, automated systems are designed to operate within a fixed set of boundary conditions; when placed in an environment exceeding these boundary conditions, they can be brittle, appearing to fail in an unexpected manner (Norman, 1990). When the degradation is sharp or profound, the automation may need to be considered a weak link, and the humans are expected to monitor for and prevent its operations in the inappropriate conditions. Thus, automation tends to work well in nominal conditions (i.e., within expected operating conditions) whereas it often fails in an unexpected manner or provides little support in off-nominal conditions in which the human needs the most support. Therefore, the efficacy of automation essentially depends on immediate context. If a design assigns functions to automation without considering possible contexts, the resulting function allocation may result in situations in which humans inevitably face "brittle automation." This highlights a further issue with function allocation: function allocation creating the requirement for the human to monitor for *automation boundary conditions*.

The technology-centered perspective has inspired several frameworks to support and guide function allocation. The Fitts list is one of those frameworks along with categorizations such as "Levels of Automation" (Sheridan & Verplank, 1978) that provide different "categories" of capabilities that the designer can select, as shown in Table 3.

Table 3. Levels of automation (table reformatted from Sheridan & Verplank, 1978)

Levels of Automation	Description
1	Human does the whole job up to the point of turning it over to the computer to implement.
2	Computer helps by determining the options
3	Computer helps determine options and suggests one, which human need not follow.
4	Computer selects action and human may or may not do it.
5	Computer selects action and implements it if human approves.
6	Computer selects action, informs human in plenty time to stop it.
7	Computer does whole job and necessarily tells human what it did.
8	Computer does whole job and tells human what it did only if human explicitly asks.
9	Computer does whole job and tells human what it did and it, the computer, decides he should be told.
10	Computer does whole job it decides it should be done, and if so tells human, it decides he should be told.

This perspective takes the Fitts list as a challenge to increase machine autonomy to also assume the capabilities of the human. Although this perspective has been highly criticized from other perspectives, which will be discussed in the following sections, it is notable that this perspective indeed pushes the boundaries of automation technologies and has contributed to highly-reliable automated systems used in current operations and practices.

In summary, function allocations created from the technology-centered perspective highlight the following issues: 1) *incoherency in function allocations* in which the human "picks up" any functions beyond the automation's capabilities, 2) *mismatches between responsibility and authority* due to function allocation only considering the capabilities of automation, and 3) function allocation creating the requirement for the human to monitor for *automation boundary conditions*.

2.2.2 Human-centered Perspective

A human-centered perspective states that automation should be designed as a tool for humans, and automation designers should take an approach that best supports humans' needs. This perspective emerged from the human factors field as an effort to guide automation design to respond to the needs of humans rather than to the capabilities of automation (i.e., as a counter-point to the technology-centered perspective).

As also noted earlier in Section 2.1.2., issues with workload have been reported in human-automation interaction (Parasuraman & Riley, 1997; Weiner & Curry, 1980; Wiener, 1989a; Wiener, 1985). Although a human's average workload over a long period of time may seemingly be well within their capacity, workload spikes (shorter-duration demands) or periods of saturation (longer-duration demands) are commonly reported. Automation sometimes imposes more workload during high tempo operations, an effect termed "clumsy automation" (Billings, 1997; Parasuraman & Riley, 1997; Wiener & Curry, 1980). That is, automation works well in nominal conditions in which the workload of humans is already fairly regulated. However, in off-nominal conditions, which already tend to increase human workload, automation tends to also further require more work from humans. Additionally, if the environmental condition pushes the automation out of its boundary conditions, it may give up functioning or the human is assumed to "take-over," increasing the human's workload yet further.

More intricate, non-linear workload concerns may also arise: while the introduction of the automation took over the most of physical tasks such as, in the flight deck, controlling control surfaces and engine thrust, automation tends to only reduce “manual” workload, not accounting for cognitive workload (Wiener, 1985). Thus, the introduction of highly-automated systems did not reduce workload, but changed its nature. For example, Wiener’s (1989a) study of pilots flying an aircraft equipped with advanced flight deck automation, discussed in Section 2.1.2., revealed that the introduction of flight deck automation increased workload.

Thus, simply allocating functions to automation that were originally assigned to humans does not guarantee a reduction in workload. Instead, it may introduce more cognitive tasks in the form of human-automation interaction and human monitoring of the automation. Also, allocating functions to automation does not remove the need for the human to maintain situation awareness. This highlights an additional issue with function allocation: *workload* that is not decreased or is increased by the function allocation, workload spikes and saturation, clumsy automation, and changes in the nature of the workload.

The next issue identified from the human-centered perspective is the relationship between the human’s likely cognitive control modes in context and the actions a function allocation requires of the human. Hollnagel’s concept of “Cognitive Control” describes how humans select their activities (and sequence them) as a response to their competency and perception of resources available to them (such as information availability) and demands on them (such as subjective available time) (Hollnagel, 1993). Further, Hollnagel suggested that cognitive control may be varied along a continuum that transitions between four general “Cognitive Control Modes.” The first cognitive control mode, “Scrambled” (sometimes also called “Panic”), indicates the least controlled mode in which the humans are not able to exert any control over their work environment and work activities are chosen randomly with little relationship to the needs of the environment.

The opportunistic cognitive control mode is defined as “the case in which the next action is chosen from the current context alone and mainly based on the salient features rather than durable goals or intentions” (Hollnagel, 1993). Thus, the humans operating in this mode will select their next action in response to judgment and other assessment activities that are driven by salient features in the environment (Feigh, 2008).

The tactical cognitive control mode is defined as the case in which “the person’s event horizon goes beyond the dominant needs of the present, but the possible actions considered are still very much related to the immediate extrapolations from the context” (p. 170, Hollnagel, 1993). Thus, humans in this mode will demonstrate decision making guided by familiar rules or patterns of behavior (Feigh, 2008). For example, pilots monitoring for flight deck information would scan the flight deck using their trained “scan pattern” or “flow.”

The strategic cognitive control mode is defined as the case where “the person is using a wider event horizon and looking ahead at higher level goals...” (Hollnagel, 1993). Thus, humans in this mode will focus on planning their actions, including anticipating upcoming actions or abnormalities in the environment that may be prevented or pre-empted. For example, if pilots operating in this mode do not receive the descent clearance from the air traffic controller by their T/D point, they may decrease the current airspeed by 0.02 Mach (thus, decreasing aircraft energy without violating their current air traffic instruction) or query the air traffic controller.

Humans switch their cognitive control modes during operations in response to context. However, a function allocation may not support all likely cognitive control modes. A recent study in cognitive control modes and cognitive work support system design demonstrated this impact: when the humans were operating in a cognitive control mode different than that used by automation, the observed humans' activities were significantly disrupted and, in some cases, performance was adversely impacted (Feigh, K.M., 2010). On the other hand, when the human's cognitive control mode matches the behavior assumed in designing the automation, consistent, effective patterns of activities were observed. In the flight deck, the construct of cognitive control modes is a useful description of situations in which a "busy" pilot (in an opportunistic or tactical mode) is asked to spend significant time on fairly strategic activities such as reprogramming the FMS. This highlights an issue with function allocation: *function allocation preventing human adaptation to context* such as conflicts between their required actions and their cognitive control modes.

Building on the construct of cognitive control, humans work to develop and maintain a stable work environment. While maintaining a stable work environment requires some degree of control by the human, it can be fostered by the nature of the work environment and the function allocation. If the work environment maintains a certain level of regularity, the human can predict its dynamics and tailor his/her activity to the needs in the environment. This enables the human to work efficiently and is suggested to contribute to the robustness of the human-automated system (Hollnagel, 2004). However, if unexpected events occur often, environmental predictability decreases, and the humans are required to spend more time to reacting to events (Hollnagel, 2002). If automation's performance is not predictable, then its contributions to the human's work environment will make that environment appear unstable. More profoundly, dynamic function allocation changes not only the dynamics within the human's perceived work environment, but also changes the definition of what aspects of their environment they need to interact with and what actions they can apply to maintain stability. Therefore, a trade-off exists when designing function allocations between maintaining predictability vs. applying complex automated capabilities and dynamically allocating functions (Miller & Parasuraman, 2007). This highlights an issue with function allocation: *function allocation destabilizing the human's work environment* by reducing predictability.

Many human performance models have been developed to model human-automation interaction. Among many, four of them are briefly introduced here. Adaptive Control of Thought-Rational (ACT-R) is "a computational architecture designed to support modeling of human cognition and performance at a detailed temporal grain size" (Byrne, Kirlik & Fleetwood, 2007). Air Man-machine Design Integrated Design and Analysis System (Air MIDAS) is a modeling framework that predicts human performance in aviation-specific applications (Corker, Muraoka, Verma & Jadhav, 2007). The Distributed-Operator Model Architecture (D-OMAR) is an architecture for modeling multitask behaviors (Deutsch & Pew, 2007), mainly used for modeling pilot behaviors in aviation operations. Attention-Situation Awareness (A-SA) is "a two-component computational model of attention and situation awareness" (Wickens et al., 2007). A-SA comprises two modules: one manages attention allocation to events or channels available in the environment, and the other generates an inference about or situation awareness of the current and future states of the environment.

However, these tools are limited for addressing issues with function allocations between multiple agents in a complex work environment. In general, these tools focus on modeling human actions in detail without capturing the complex work environment with interaction between multiple agents, or they focus on one aspect of human behaviors in a very detailed manner. Therefore, while the human-centered perspective has highlighted significant issues with function allocation, current modeling frameworks have not yet been established that can fully address these issues.

Evaluative criteria of issues with function allocation are also required. 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. They also provided several primary evaluative criteria (cognitive workload, situation awareness, complacency, and skill degradation) and secondary criteria (automation reliability and costs of action outcomes). However, 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; this and other studies note the benefit of further quantitative modeling (e.g., Pew & Mavor, 1998).

> Function allocations created from the human-centered perspective considers the following issues: 1) *workload* that is not decreased or is increased by the function allocation, workload spikes and saturation, clumsy automation, and changes in the nature of the workload, 2) *function allocation preventing human adaptation to context* such as conflicts between their required actions and their cognitive control modes, and 3) *function allocation destabilizing the human's work environment* by reducing predictability.

2.2.3 Team-oriented Perspective

A team-oriented perspective considers automation as a team member and thus views human-automation interactions as similar to interactions within a human team. This perspective is inspired by many different fields including the team and organization design, and management science. Although these studies did not include automation as a focus of their studies, they have extensively studied how team members interact with each other. Extrapolating their insights about human teams to teams of human and automated agents is well-grounded in the human-automation research community. For example, Muir (1994; 1996) related models and measures of trust from the social sciences to human trust in automation; Bass and Pritchett (2008) modified social judgment theory to quantitatively model human interaction with automated judges; Pritchett (2001) proposed framing human interaction with alerting systems in terms of the same type of role descriptions used within human teams; and Sarter and Woods (2000) explicitly described flight path automation as a "poor team member." More explicitly, Woods (1985) and Woods and Hollnagel (2006) suggested that "good" automation should create a diverse joint human-machine cognitive system. Likewise, the strategy of "complementation" seeks to form a heterogeneous team in which automation and humans work together cooperatively, each contributing those strengths it can provide within its environment and context (Schutte, 1999; Schutte et al., 2007), and Miller and Parasuraman's "playbook" metaphor for assigning functions to automation is specifically described using delegation in human teams as a metaphor (2007).

Within the broad range of definitions of "team" provided in the literature (e.g., LaJoie, 1999), Salas and his colleagues provide a generally accepted definition of teams as "a collection of (two or more) individuals working together inter-dependently to achieve a common goal" (Salas, Dickinson, Converse & Tannenbaum, 1992). This definition of a team is also applicable to human-automation teams. With automation's increasing capabilities to support cognitive functions such as judgment, decision making and communication, humans and automation "work together" to achieve mission goals.

Team structure is concerned "with the lines of authority in the team and with how the team divides its tasks and responsibilities and controls its resources to perform its mission" (MacMillan, Entin & Serfaty, 2004). For a single human to perform an entire mission is not always possible. Therefore, a team is constituted and structured, and a careful delegation among team members is established. Delegation is the assignment of lines of authority and responsibility to another team member to perform specific activities although the ultimate responsibility (also called "accountability") for the outcome of the delegated work still remains with the original team member. However, assigning functions and delegating responsibility among human and automated agents adds several considerations. Automation does not have same “teamwork skills” that humans naturally have. The automation does not have a sense of responsibility (Sarter, et al., 1997). Automation does not “worry” about consequences. Automation has no motivation to live up to its obligations, does not experience shame or embarrassment, and cannot be assessed for attributes such as loyalty, benevolence and agreement in values (Lee & See, 2004; Pitt, 2004). When automation is placed outside its boundary conditions it cannot function properly, unlike a human team member who will continue to attempt effective performance in unfamiliar circumstances.

Therefore, it is desirable for an agent who is responsible for the outcome of a task to have the authority to execute the task, to avoid the so-called “responsibility-authority double-bind” (Woods, 1985) in which humans are only able to accept or override the automation’s functions. In human-automated teams, the responsibility for the final outcome of a mission is assigned to the human operators. Thus, they should be also given the resources and the capabilities to oversee and, if necessary, override the automation. However, too often human operators do not possess the authority “in a practical sense” to override the automation. If human operators do not have the knowledge or time to verify the automation’s activities, then the human operators may be “cognitively railroaded” into following its output exactly, abrogating his or her responsibility (Pritchett, 2001). This again highlights the issue with function allocation discussed in the technology-centered perspective: *mismatches between responsibility and authority* where a function allocation delegates authority without delegating responsibility.

Team structure is also concerned with how teams divide their tasks among team members and control relevant resources. Therefore, studies from team design seek to specify who controls resources, who takes actions, who uses information, who coordinates with whom, the tasks about which they coordinate, who communicates with whom, who is responsible for what, and who shall provide backup to whom (Szilagyi Jr & Wallace Jr, 1980). For example, in a flight deck, a captain and a first-officer are given a clear description of the team structure through the titles "Pilot Flying (PF)" and "Pilot Not Flying (PNF)" which detail who flies the aircraft, who monitors nearby traffic, who operates the MCP and CDU, and who communicates with air traffic controllers. Likewise, a human-automated team also needs to specify its structure. In the flight deck, for example, pilots should be given clear descriptions of the automation's roles and specific functions. However, current function allocations often do not (or cannot) provide clear descriptions of automation's roles because the functions assigned to automation are scattered throughout the work domain rather than "coherently" divided among humans and automation. This problem may be partly addressed by an interface that can illustrate the allocation, clearly coordinate human and automation tasks, and enable better monitoring of automation (Palmer, Rogers, Press, Latorella & Abbott, 1995). However, beyond an interface solution this problem has its basis in a team design that does not allocate functions according to clearly-specified roles. This highlights a further issue with function allocation: *incoherency in function allocations* compared to a clearly defined team structure.

Communication, in general, is seen as a vital component to a team working together, yet some communication patterns can disrupt individual task performance. Therefore, it is important that team members are able to predict each other's information needs and provide information at useful, non-interruptive times (Hollenbeck et al., 1995; Hutchins, 1995; Stout & Salas, 1993). In taxonomies such as that proposed by Entin and Entin (2001), a key measure of good communication is that team members "anticipate" the needs of team members by communicating information or transferring actions before specific requests are made. On the other hand, interruption is often considered to be bad, unless absolutely required. Interruption is "an externally generated, randomly occurring, discrete event that breaks the continuity of cognitive focus on a primary task" (Coraggio, 1990 cited in Speier, Valacich, & Vessey, 1999). Interruption is intrusive and distracting and breaks up workflow (Jett & George, 2003). However, an interruption may be demanded by circumstances, in which case it can spur knowledge acquisition (Zellmer-Bruhn, 2003) and facilitate decision-making performance (Speier, Valacich & Vessey, 1999). Therefore, a "good" team member knows when an interruption is warranted versus when it will be detrimental.

Unfortunately, too often automation unduly interrupts other human team members. Billings (1997) defined automation that interrupts when humans are experiencing high workload as "clumsy." Also, automation is considered to be a "powerful and independent agent" in which automation has powerful capabilities to act on its own, but it provides poor feedback and interrupts human team members without consideration of context or the status of the humans (Sarter & Woods, 1995). In addition, whereas humans can implicitly sense information about other team members (Christoffersen & Woods, 2002), automation cannot. Thus, when automation provides information, it cannot take into account whether this information warrants an interruption of its human team members. Therefore, another issue with function allocation is: *interruptive automation* compared to human-to-human communication.

In human teams, communication among team members adds "teamwork" demands in addition to "taskwork" demands. Taskwork refers to an individual's or a team's effort to understand and perform the requirements of the job, tasks, and equipment to be used, i.e., to meet mission goals (Arthur, Edwards, Bell, Villado & Bennett, 2005). On the other hand, interaction among team members is referred to as teamwork. In human-automation function allocation, this teamwork imposes two requirements. First, it requires the automation (or more specifically, the automation designer) to provide displays and interfaces. Second, it induces extra human-automation teamwork for the human to perform, especially when a function allocation assigns much of the taskwork to automation. As noted in Section 2.1.2, this added teamwork can create significant additional workload for the pilot, especially critical if it is induced in high-tempo periods of work. Therefore, this discussion also highlights an issue: *workload* through induced teamwork.

Similarly, coordination is also seen as a vital aspect of a team. Well-coordinated teams enable team members to anticipate changes in the environment and in the needs of other team members (Entin & Serfaty, 1999). Good teams can maintain their performance regardless of given stress or time pressure by changing the "mode of coordination" from explicit coordination to implicit coordination (Entin & Serfaty, 1999). That is, under high workload and/or time pressure, high-performing teams adopt a strategy of coordination that allows the team members to reduce communication and coordination overhead and to maintain performance. However, changes in coordination without strategies (explicit or implicit) can destabilize the work environment of the team members (Entin & Serfaty, 1999). Thus, human teams maintain implicit coordination strategies formed through training and shared experiences. Similarly, with human-automated teams, predefined sets of function allocations may serve as more explicit coordination strategies. For example, the playbook of pre-defined function allocations and coordination strategies by Miller and Parasuraman (2007) demonstrated improved performance by a human-automated team. Because this approach defines coordination strategies before actual operations, the human team member can prepare for a set of different coordination strategies in advance and select an appropriate one in context during operations. However, such "playbook metaphors" are not widely applied and automation typically has comparatively fixed functioning that cannot recognize and adapt to circumstances. Therefore, this highlights an issue with function allocation: *function allocation destabilizing the human's work environment* through poor adaptation of, or rigidity in, coordination strategies.

Several relevant models and methods have proved their usefulness in the team and organization design fields. Therefore, designing human-automation function allocation via a formal model and simulation can be beneficial. These models may be modifiable to account for the unique attributes of adding automated systems to teams. Indeed, some organizational models represent members at a coarse level of granularity that does not distinguish between human and automated system team members (Carley & Kamneva, 2004; Rasmussen, Pejtersen & Goodstein, 1994; Schraagen & Rasker, 2003). However, other studies have highlighted the need to consider additional behaviors within the team when the new team member is an automated system (Bowers, Oser, Salas & Cannon-Bowers, 1996; Paris, Salas & Cannon-Bowers, 2000).

Examining the Fitts List from the team-oriented perspective, it ascribes a range of functions to human and a different range to automation, but it considers neither automation as part of a team, nor considers team dynamics overall. Thus, designs based on the Fitts list inevitably ignore crucial aspects of human and automation working together: how the human and the automation communicate, coordinate with, support, and complement each other.

> Investigation of human-automation function allocation from the team-oriented perspective provides the following issues: 1) *mismatches between responsibility and authority* where a function allocation delegates authority without delegating responsibility, 2) *incoherency in function allocations* compared to a clearly defined team structure, 3) *interruptive automation* compared to human-to-human communication, 4) *workload* through induced teamwork, and 5) *function allocation destabilizing the human's work environment* through poor adaptation of, or rigidity in, coordination strategies.

2.2.4 Work-Oriented Perspective

A work-oriented perspective focuses on "work." The preceding perspectives have generally focused on "agents" in the system (in the technology- and human-centered perspectives) and on agent interactions (in the team-oriented perspective). However, before considering what agents can/should do, one must delineate what taskwork the work environment requires to achieve mission goals, and what teamwork is required to coordinate the taskwork when it is allocated across a team. Thus, this perspective aims to design function allocations that can support work in response to its dynamic work environment. This perspective emerged mainly from cognitive engineering in an attempt to answer how human-automated systems can improve ultimate work performance and maintain or improve safety.

Work is defined here as purposeful activity acting on a dynamic environment, and in response to the demands of this environment (Beyer & Holtzblatt, 1998; Rasmussen, et al., 1994; Vicente, 1999). The "environment" is the aggregation of physical and social/cultural/policy constructs required to describe, constrain, and structure the dynamics of the work (Pritchett, Kim, Kannan & Feigh, 2010). Work is a purposeful activity in that it is performed to cause outcomes in the environment that meet the mission goals. Work is an activity "on and in response to the environment". The environment may have inherent dynamics which expert-agents need to mirror, may provide affordances which need to be sensed and capitalized upon, and may constrain behavior. Thus, the work must mirror the inherent dynamics of the work environment as the human's (and automation's) behavior is driven by it, and the human-automation function allocation must support this work.

Within this perspective, a framework to analyze cognitive work systems has been established. Cognitive Work Analysis (CWA) (Rasmussen, Pejtersen, Schmidt & Risø, 1990) is a comprehensive modeling framework that encompasses several phases of analysis to uncover requirements, constraints, and affordance implied but hidden in the collective work environment (Roth & Bisantz, in press) that spans physical, procedural, and social environments. Thus, CWA provides methods to perform analyses of the work domain, cognitive tasks, strategies, social organization and cooperation, and worker competency.

Each aspect of work is covered by a different phase of analysis. First, Work Domain Analysis (WDA) identifies intrinsic constraints and information requirements in the work domain using an Abstraction Hierarchy (AH). Second, Control Task Analysis (CTA) identifies how a human agent processes the information available and produces output actions, represented in a Decision-Ladder (DL). Third, Strategy Analysis identifies different categories (i.e., strategies) to complete tasks using an Information Flow Map. Fourth, Social Organization and Cooperation Analysis identifies how the requirements (identified in previous phases) can be distributed across human and automated agents using the modeling tools provided from the previous phases. Lastly, Worker Competency Analysis identifies the competencies (capabilities) of agents required to effectively function in the work domain using the Skill, Rule, Knowledge (SRK) taxonomy.

Although CWA has not been commonly used to discuss human-automation function allocation, it provides useful discussions and modeling tools. WDA identifies what information and what functions are available and required to achieve work independent of function allocation via a multi-level goal-means representation, the AH. The AH illustrates taskwork demands in the work environment in response to inherent dynamics in the work environment. The AH places higher-level abstract functions within a work environment at the top and physical, concrete functions at the bottom. A common convention of the AH is comprised of five levels: Functional Purposes (representing mission goals of the system), Priorities and Values (representing principles or values that the system must follow or preserve), Generalized Functions (representing process descriptions entailed to achieve mission goals), Physical Functions (representing capabilities of agents and equipment), and Physical Form (representing physical characteristics of equipment). These five levels, each independently providing complete descriptions of work, are connected to each other through goal-means relationship. Thus, WDA describes taskwork functions in a context-independent manner, which then can be used as a baseline to allocate functions.

CTA identifies the requirements associated with the proficient control of specific tasks required within the broader work environment examined in the WDA. CTA focuses on the states of knowledge, information processing, and their connections to each other. By identifying these states of knowledge and how they are processed, CTA has been used to represent expertise as active, constructive processing. Thus, it provides a means to structure more efficient and proficient ways to achieve work. These findings are then used to identify whether any pre-defined procedures exist (established by the agents themselves as they lay out their work or by system designers as instructions) and whether they shape the behavior of agents. These pre-defined procedures may limit potential function allocations.

Strategy Analysis identifies multiple patterns of activities feasible to complete a task. Humans often perform a task by using different strategies, switching between them in response to contextual factors. Strategy Analysis uses an Information Flow Map, which represents idealized categories of task procedures and demonstrates how the work structure can potentially change in response to the dynamic work environment. That is, as agents in the work system select different strategies, the structure of their work changes accordingly. While not currently represented in Strategy Analysis, which typically examines only single agent's actions, the selection of a function allocation is itself a selection of a strategy, and, once implemented, a function allocation then structures the work in a manner that constrains the selection of other strategies in a range of tasks.

Social Organization and Cooperation Analysis provides a way to determine how agents communicate and coordinate to enhance the performance of the system as a whole. This analysis uses the modeling tools developed in the previous phases such as the AH and Information Flow Map. Therefore, this analysis attempts to allocate aspects of work across multiple agents, similar to function allocation; however, this analysis often allocates entire aspects of the work environment rather than allocating specific, inter-dependent functions.

Worker Competency Analysis provides a method to link the cognitive constraints of agents, as identified by SRK taxonomy, to system designs. Thus, it identifies the knowledge and skills that agents in the work system require to function properly within the system. Specifically, this analysis identifies which capabilities are required for agents to achieve work, and, from this, allocate functions based on the current capabilities of agents available. Although this process may seem similar to the capability-comparison inherent to the Fitts list, it should be noted that this phase is conducted based on the detailed analysis of the work, and at a finer resolution.

These efforts of modeling work, its environment, and agents focus on whether the work indeed meets its mission goals. That is, mission performance is only achievable by all agents meeting the taskwork demands of the environment, with their teamwork synchronized and executed to mirror these taskwork demands. Therefore, this highlights an issue with human-automation function allocation: *mission performance*.

As discussed in the CTA phase, how the work is to be performed is dictated by established procedures in some domains such as aviation, process control, and, in other domains, by less-formally defined work practices. Whether explicitly established or informally defined, a function allocation should not interrupt these procedures. These established procedures are proven to aid agent memory and guarantee consistency and safety (Ockerman & Pritchett, 2000). Interruption to these procedures could endanger system safety and, also, degrade mission performance. For example, checklists are developed to support pilots maintaining a seamless work flow in a flight deck: multiple studies have shown their efficacy and relationship to safety (see Degani & Wiener, 1990; Degani & Wiener, 1993; Mosier, Palmer & Degani, 1992; Palmer & Degani, 1991). However, when the execution of such checklists is interrupted, pilots are prone to skipping a step or sometimes dropping the rest of the steps in the checklist, resulting in compromised aircraft safety (see Damos & Tabachnick, 2001). Therefore, a function allocation should consider procedures and mirror their structures so that the resulting function allocation would not impede them. Therefore, an issue with function allocation is highlighted: *interruptive automation* relative to the established workflow.

Studies of resilience describes the ability to cope with unexpected situations as maintaining control of those situations (Hollnagel, 2004; Hollnagel, Woods & Leveson, 2006; Sheridan, 2008). Resilience enables organizations to cope with unexpected variabilities in the environment in a "robust yet flexible" manner in that the organizations use resources proactively to accommodate for external and internal disruptions (or threats), thus mitigating risks (Chialastri & Pozzi, 2008; Hollnagel, et al., 2006). However, the brittleness of automation (i.e., degradation in its performance when the environment exceeds their boundary conditions) discussed earlier in the technology-centered perspective section reflects how automation cannot contribute in these unexpected situations. Therefore, this highlights an issue with function allocation: *automation boundary conditions* as a limit to resilience.

Likewise, resilience is fostered when a human agent may employ a range of cognitive control modes so that they can adapt to the situations (Pritchett, 2010). This also relates to the insights of CWA's Strategy Analysis phase, which recognizes that workers select different strategies in response to context. However, a rigid function allocation may be fixed to one strategy. Therefore, an issue with function allocation is highlighted: *function allocation preventing human adaptation to context* by limiting strategy selection.

Social Organization and Cooperation Analysis entails two dimensions in terms of content and form. Content refers to division and coordination of work while form refers to structures such as authority and responsibility to establish a clearly defined chain of command. Thus, the content dimension focuses on how work is divided and coordinated, and covers many aspects of identifying feasible function allocations. An important note recognized in this analysis is that an agent or a group of agents assigned to a certain function requires access to the information needed to perform that function. Also, another important note is that division and coordination of work is dynamic, requiring multiple criteria for allocating functions among agents or groups of agents. However, discussions of function allocation in Social Organization and Cooperation Analysis are limited: they generally focus on largely-independent agents with clear goals. In domains such as aviation, the distributions of work are interdependent and heavily coupled. In addition, these couplings may be hidden within the context of the work environment. For example, in a function allocation with the pilot flying by controlling the control column and autothrottle on, the pilot controls elevator and the automation controls throttle, and pitch and speed are intrinsically coupled so that the actions of one will start to step on the actions of the other. Therefore, this highlights an issue with function allocation: *incoherency in function allocations* both in terms of clear role distribution and in terms of inter-dependencies where the actions of one may drive the actions of the other.

To summarize, CWA provides a means to identify hidden, complex relationships between goals, functions, information required, work environment, and agents. These phases of analyses provide useful tools to model human-automation function allocation. The AH can be used to identify and model what functions are needed in a work domain. Strategy Analysis can be used to model different function allocations as strategies, including how they may be selected in context and how they will structure other aspects of work. Social Organization and Cooperation Analysis provides a set of criteria for dividing and coordinating work across multiple agents. Worker Competency Analysis provides a guideline for identifying the capabilities and limitations of agents.

However, CWA does not fully address all aspects of function allocation. First, the models formed by CWA are static and qualitative. Second, CWA has no basis for validating suppositions based on its models other than "assuming" that the models are correct. Therefore, a model framework is needed that raise the level of validation possible in its models and commensurate insights into function allocation.

Examining the Fitts list from the work-oriented perspective, it does not address many important aspects of work that CWA attempts to identify: constraints in work domain, context (as noted in strategy selection), work division and coordination. The Fitts list only concerns the last phase of CWA, Worker Competency Analysis: as mentioned earlier, this analysis heavily relies on the outcomes of other analyses whereas the Fitts list only describes the capabilities of humans and machines independent of work domain and context. Therefore, designs based on the Fitts list as the sole or primary justification for the human-automated systems can result in function allocations that are piece-meal and incoherent, thus increasing possibility of agents interrupting workflow.

> The investigation of human-automation function allocation from the work-oriented perspective highlights the following issues: 1) *mission performance*, 2) *interruptive automation* relative to the established workflow, 3) *automation boundary conditions* as a limit to resilience, 4) *function allocation preventing human adaptation to context* by limiting strategy selection, and 5) *incoherency in function allocations* both in terms of clear role distribution and in terms of inter-dependencies where the actions of one may drive the actions of the other.

2.3 Issues with Function Allocation

The previous section described four perspectives of function allocation. The technology-centered perspective focuses on the capabilities of automation and function allocations that use those capabilities. The human-centered perspective focuses on human needs and function allocations that best support humans. The team-oriented perspective focuses on team interaction and function allocations that could support teamwork. The work-oriented perspective focuses on work and its environment and function allocation designs that could support effective work, including the ability to adapt to any changes in work environment.

Table 4. Four perspectives and the issues they identified with function allocation

Perspective	Issues identified
Technology-centered perspective	*Incoherency in function allocations* in which the human "picks up" any functions beyond the automation's capabilities
	Mismatches between responsibility and authority due to function allocation only considering the capabilities of automation
	Function allocation creating the requirement for the human to monitor for *automation boundary conditions*
Human-centered perspective	*workload* that is not decreased or is increased by the function allocation, workload spikes and saturation, clumsy automation, and changes in the nature of the workload
	Function allocation preventing human adaptation to context such as conflicts between their required actions and their cognitive control modes
	Function allocation destabilizing the human's work environment by reducing predictability
Team-oriented perspective	*Mismatches between responsibility and authority* where a function allocation delegates authority without delegating responsibility
	Incoherency in function allocations compared to a clearly defined team structure
	Interruptive automation compared to human-to-human communication
	Workload through induced teamwork
	Function allocation destabilizing the human's work environment through poor adaptation of, or rigidity in, coordination strategies
Work-oriented perspective	*Mission performance*
	Interruptive automation relative to the established workflow
	Automation boundary conditions as a limit to resilience
	Function allocation preventing human adaptation to context by limiting strategy selection
	Incoherency in function allocations both in terms of clear role distribution and in terms of inter-dependencies where the action of one may drive the actions of the other

The issues identified in each perspective are summarized in Table 4. Each perspective identified a subset of these issues. Thus, examining multiple perspectives provided a comprehensive review of these issues based on the findings throughout the literature. Examining the issues listed in Table 4, issues with human-automation function allocation can be summarized as follows:

1) *Workload*: Issues with workload include changes in the nature of the workload, workload spikes and saturation, and can result from not only the taskwork but also the additional of the teamwork (including human-automation interaction and monitoring) induced by a function allocation.
2) *Incoherency in function allocations*: Incoherent function allocations do not establish clear roles and efficient work practices for all team members, and may lead to inter-dependent or conflicting activities between agents.
3) *Mismatches between responsibility and authority*: Mismatches between the assignment of responsibility and authority leave the human responsible for the outcome of automated functions, and, thus, induce monitoring functions to supervise delegated functions.
4) *Interruptive automation*: Automated functions may unnecessarily interrupt humans or established procedures, especially compared to human-to-human interaction.
5) *Automation boundary conditions*: Function allocations can be contextually inappropriate where they place automation outside the boundary conditions in which it can effectively and reliably operate.
6) *Function allocation preventing human adaptation to context*: Function allocation may have implicit assumptions about human behavior as a fixed pattern, which may not hold as human team members adapt to context by selecting strategies or as part of cognitive control.

7) *Function allocation destabilizing the humans' work environment*: Predictability in the work environment allows humans to anticipate upcoming tasks; automation can add unpredicted behaviors to their work environment.
8) *Mission Performance*: Ultimately, function allocations should improve mission performance.

Function allocation is not the only issue with human-automation interaction. As Dekker and Woods (2002) highlighted, another issue is "how do we make them (human and automation) get along together?" Thus, other issues with human-automation interaction go beyond the issues with function allocation noted here. Specifically, function allocation can address the structure of communication and coordination, but interfaces and displays that enable this communication and coordination are also necessary aspects of effective human-automation interaction.

Thus, there are aspects of human-automation interaction that cannot be addressed only by the discussion of function allocation. However, the focus here on establishing a function allocation that addresses the issues identified by the four perspectives is a necessary condition. Not only does function allocation generally need to be addressed at the earliest stages of design, it also often is the only issue that can be addressed early (i.e., before the interface and machine logic have been established).

However, no one definitive phenomenon determines the success of a function allocation. Likewise, no one metric can address the range of issues with function allocation noted here. The lack of a formal approach to assess function allocation has led to musings that allocation is and perhaps forever will be an art (Parasuraman, et al., 2000; Sheridan, 1998).

To address this problem, this chapter identified issues with function allocation to extend the degree that it may be formally assessed. In addition, the findings also indicate needs for a systematic approach that applies models and for a comprehensive set of metrics to assess function allocation. The next two chapters, then, introduce a modeling framework comprising static work models and dynamics simulations and a set of metrics, respectively, addressing the issues identified and categorized in this chapter.

CHAPTER 3

MODELING FRAMEWORK TO ASSESS HUMAN-AUTOMATION FUNCTION ALLOCATION

Chapter 2 described, first, the issues with human-automation function allocation as seen from multiple perspectives and, second, the necessity of a systematic approach to assess these issues. These issues have a basis in the structure of the work jointly executed by humans and automation. For example, workload spikes, incoherent function allocation, problems with timing of actions and information availability, and undue interruptions are issues that arise out of joint human-automation work. To address these issues systematically requires formal models and simulations that include all necessary aspects of human-automation function allocation: work, environment, agents, their inherent dynamics, and the relationships among them. Also, to address these issues requires not only a (static) work model that describes the structure of the work and the relationships among them, but also a (dynamic) simulation that captures temporal aspects such as the timing of actions and their impact on the environment.

This chapter describes a modeling framework developed to provide the required static work model and dynamic simulation. First, the chapter identifies functional requirements for the static model and the dynamic simulation based on the underlying premises. Then, this chapter introduces and describes the constructs of the work model developed under this effort. Finally, the chapter details dynamic simulation of the work model. Therefore, with properly modeled work as described by the environment, agents, their inherent dynamics, and relationships among them, this framework can capture the previously-identified issues with function allocation.

3.1 Requirements for Modeling Human-Automation Function Allocation

The definitions of work, environment, and their inherent dynamics and relationships are discussed in 2.2.4. Work is defined as purposeful activity acting on a dynamic environment, in response to the demands of this environment. The environment includes physical aspects as well as procedural aspects – general work practice, defined procedures, regulations, letters of agreement for interaction, etc., that constrain and structure behavior. The inherent dynamics of the work environment may shape/constrain/structure behaviors of agents performing work. Thus, work is situated and embodied in the environment as a response to the immediate situation.

An agent performs work on the environment. That is, the agent executes work by choosing strategies to respond to the immediate context. A "strategy" is a sequence of actions that can be used to accomplish a more-aggregate description of high-level function (Roth & Bisantz, in press; Vicente, 1999). Thus, strategies to achieve the same goal are described as different sequences of same or different actions. Agents choose strategies in response to contextual factors that may include aspects of the work environment, the function allocation within the team, and agent status including expertise, the demands on the agent, and resources available to the agent such as time and information. Therefore, the work model should be able to represent the multiple strategies by which work is adapted to context.

Teamwork adds to the work of each agent and, thus, to each agent's view of the environment. While the overall environment refers to the surroundings of the team, each individual's perception of his/her work environment includes both part of the overall environment and the teamwork aspects created by his/her team members. Figure 10 (a), for example, depicts a team of two agents and the part of their overall environment that each interacts with. Thus, Figure 10 (a) illustrates the taskwork that each agent performs on the overall work environment. Figure 10 (b) additionally shows the teamwork created when the agents need to coordinate and communicate with each other. Through these distinctions between taskwork and teamwork, the environment of each agent can be identified thoroughly by examining both the taskwork and teamwork performed by the agent. Therefore, the model should be able to represent not only taskwork but also teamwork, and the team members' appropriate aspects within the work environment.

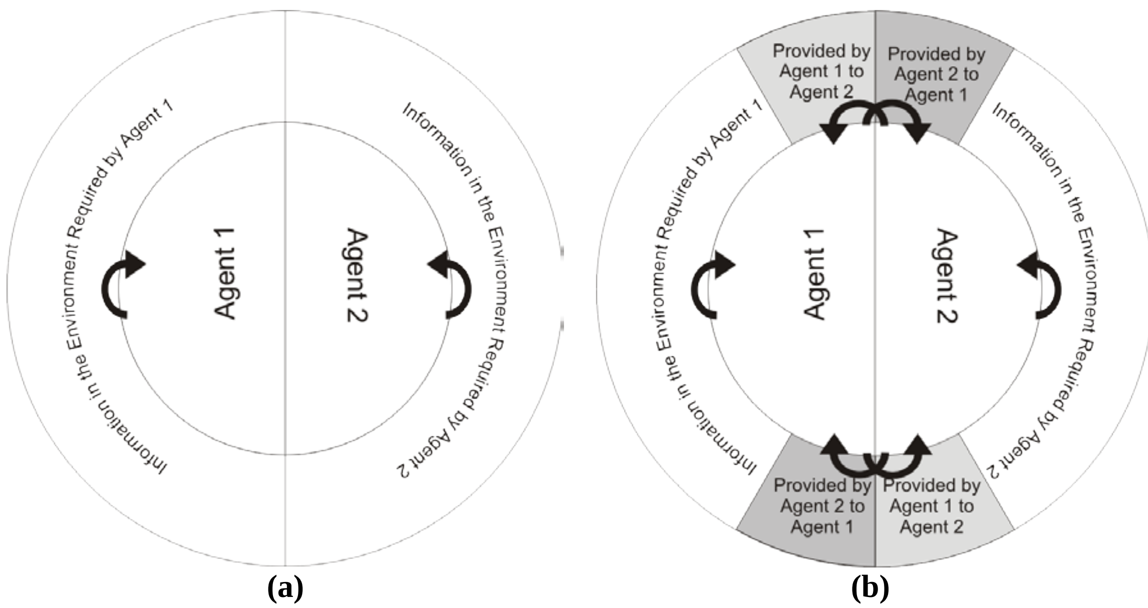

Figure 10. Agents working independently on taskwork only (a, on the left) and agents working together on taskwork and teamwork (b, on the right). The collective team environment in (a) is supplemented in (b) by individual environments that also include teamwork constructs.

Given that a realistic description of complex, heterogeneous dynamic systems generates a vast span of required activities, an additional requirement for the model is the ability to describe how agents might make sense of their complex work domain. Such multi-level modeling represents how an agent progressively abstracts the most detailed work activities into higher-level aggregations, both to develop succinct descriptions of higher-level functions that the work performs and to link them to mission goals. Likewise, the complexity of the work is reflected in the complexity of the work models. Thus, just as the agent needs to make sense of the work, the modeler needs to organize the work model. Therefore, the model should provide descriptions of agent making sense of their complex work domain as well as a mechanism for modelers to manage the complexity of a detailed work model.

Based on these premises of a work model including the discussions of work and environment in 2.2.4, the functional requirements of a work model to assess human-automation function allocation are:

- A model of human-automation function allocation should represent that work is a purposeful activity on the environment.
- It should represent the work that is situated and embodied in the environment and responds to the dynamics of the environment.
- It should allow for different strategies to be selected in response to context.
- It should capture the taskwork as well as the teamwork.
- It should capture the way agents abstract work.
- Its complexity should be manageable by the modeler.

3.2 Work Model that Computes: Constructs for Modeling Work

This section details the constructs of a work model meeting these functional requirements for modeling human-automation function allocation. The following section will describe how this work model can also be simulated.

3.2.1 Modeling Work

At the most atomic level, two constructs represent work: *resource* and *action*. A resource represents a tangible aspect of the environment. The collective set of resources represents the entire environment surrounding an agent or a team of agents; that is, the environment is composed of resources, and the current values of all resources represent the current state of the environment. A resource may represent a physical aspect of the environment with continuous dynamics, or may be a discrete value representing a categorization of the state of the environment or a policy decision such as specification of a particular function allocation between agents within the team.

An action represents an element of work performed by an agent. An action is temporally and organizationally atomic in that it represents a distinct work process performed by one agent at one instance in time. One type of action, temporal actions, samples the environment by *getting* resources and changes the environment by *setting* resources as shown in Figure 11. Actions represent the knowledge of work and are represented in the work model, but are not autonomous and may not execute by themselves – instead, they are passed to agent models. The level of detail at which an action is described can vary depending on the purpose of the modeling. For example, in modeling pilot-flight deck function allocation, a pilot dialing the MCP selectors (such as altitude selector, heading selector, etc.) or pushing the switches in the MCP should be represented at a relatively fine level with multiple actions. However, in this case, a detailed model of the air traffic controller's activities is unnecessary; thus, the air traffic controller can instead be modeled as a script that issues air traffic control commands at appropriate times (i.e., a relatively coarse level).

Temporal Action: **Control Airspeed**
Agent: Pilot
Next update : +2.0 seconds
Duration: 0.01 seconds

Gets

Sets

Resource: **Airspeed**
Value: 195 knots
Last update: 1:28:31.04

Figure 11. Action "Control Airspeed" gets and sets resource "Airspeed"

Also, a temporal action models two timing aspects: next update time and duration. The next update time identifies when the action must next be executed to accurately describe its dynamics. For models of continuous actions, this next update time may reflect an integration time-step, whereas, for discrete behaviors, this next update time reflects the next event of interest. The duration represents the time to finish an action, and it is used to track what actions each agent is working on through time. The calculations of either of these temporal aspects can be simple (e.g., a fixed time step between execution and duration) or may involve significant calculations that allow for varying effects in the environment (e.g. more precise, faster maneuvering phases of flight) and in the status of the agent performing the action (e.g., cognitive control mode or other workload effects).

3.2.2 Distinguishing Between Taskwork and Teamwork

Both teamwork and taskwork can be represented by actions and resources. For example, Figure 11 illustrates the taskwork of a pilot controlling airspeed directly – the action "Control Airspeed" works directly on the physical resource "Airspeed." Compare this method to the method of controlling airspeed shown in Figure 12 in which the function allocation includes both automation and pilot: their teamwork requires a second action "Update Autopilot Target Speed" to be executed by the pilot and a second resource

"Target Airspeed." Neither this new action nor this new resource are inherent to flight but are instead created to enable the function allocation. Thus, the environment of both the pilot and the automation is enlarged to include this teamwork resource. Note that agents do not act directly on each other. Therefore, when modeling human-automation function allocation, this distinction between taskwork and teamwork should be made explicitly, and the model should include all necessary actions induced by teamwork requirements.

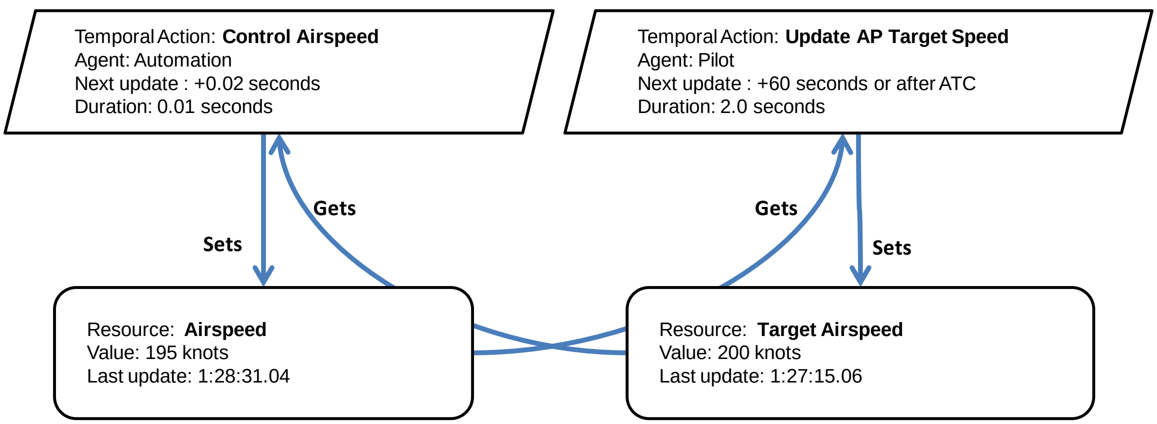

Figure 12. Teamwork action induced due to function allocation distributing work between pilot and automation

3.2.3 Modeling Work at Multiple Levels of Abstraction

A model to assess human-automation function allocation should provide a structure to represent how agents may make sense of how work activities relate to mission goals and for modelers to manage complexity. These requirements motivated "multi-level modeling," specifically the Abstraction Hierarchy (AH) used in Work Domain Analysis in Cognitive Work Analysis (Rasmussen, 1985; Rasmussen, et al., 1990). To remind as CWA was discussed extensively in 2.2.4, the AH is a multi-level means-end representation that places the abstract system purpose at the top and the concrete aspects of the environment at the bottom (Roth & Bisantz, in press), with intermediate levels allowing for progressive aggregation and abstraction of finer-ground work activities to describe the higher-level functions they perform.

The choice of functions at each level follows a means-end decomposition: any particular function is related to functions in the level above by answering the question of "why" the function is to be performed, and is related to functions in the level below it through the question of "how." Thus, the multi-level modeling links higher abstract mission goals to specific activities on the work environment.

This multi-level modeling is intended to mirror sense-making by agents of the relationship of their work activities relative to mission goals. Within the defined problem space, agents can understand and reason about the work domain at different levels of abstraction (Roth & Bisantz, in press). Thus, the multi-level model is a representation of how agents may reason about work, and how they may select strategies described at varying levels of abstraction (Vicente & Rasmussen, 1992).

Multi-level modeling also provides modelers with a mechanism for managing the complexity of the model. In theory, work could be described only at the lowest level of abstraction (i.e., by using only action and resource constructs). However, for complex work domains, the number of actions and their inter-relationships can become unmanageable without an organizing structure. By using the multi-level modeling technique with this organizing structure, the modeler is forced to reason about the work in detail and, as a side benefit, is fostered in estimating the abstractions an agent may employ when performing the work.

Therefore, the work model to assess function allocation should introduce a multi-level modeling structure into the framework. The *function* construct enables this multi-level modeling. A function aggregates elements of the work model into useful higher level abstractions. It may call upon other functions at lower levels of abstraction and, at the lowest level, may call temporal actions. The name of the function is selected to represent the purpose it achieves. Following common conventions (Naikar, 1999; Naikar, Pearce, Drumm & Sanderson, 2003; Vicente, 1999), the work model used here comprises four levels of abstraction: mission goals, priorities and values, generalized functions, and temporal functions. Note that, although this framework is inspired by the AH, it expanded the AH structure and modifies it to address the relevant functional requirements identified in Section 3.1 and to support dynamic simulation.

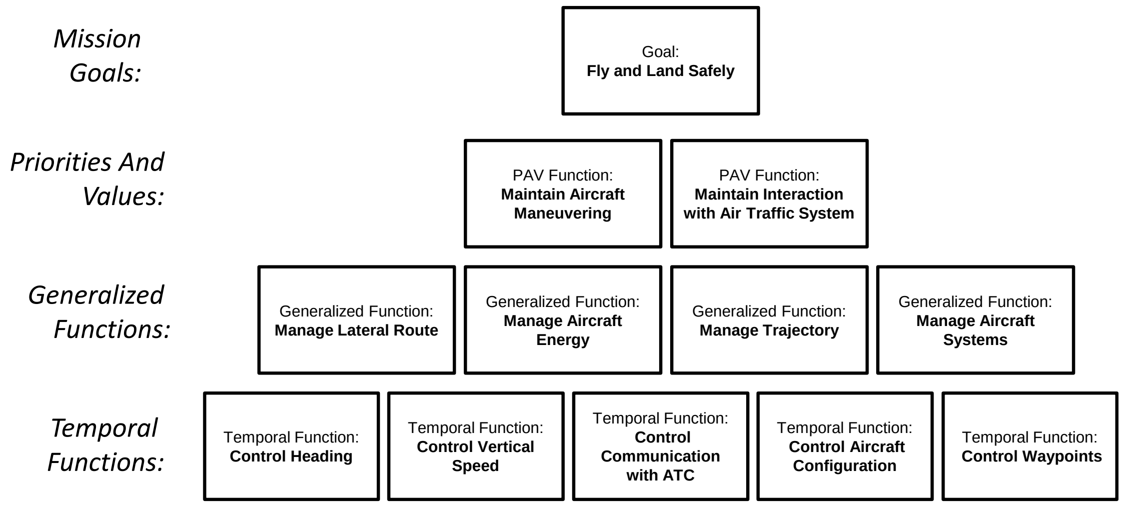

Figure 13. An example of a multi-level work model

Figure 13 illustrates an example of a multi-level work model. The Temporal Function level aggregates (temporal) actions according to inherently-coupled dynamics and purposes. These temporal functions are grouped into a generalized function at the Generalized Function level. At this level, the work is described as functions that must be performed to achieve the mission goals. These generalized functions are grouped into Priorities and Values functions that describe priorities and values that this work must promote or preserve. Lastly, these priorities and values are grouped into Mission Goals function, representing the ultimate objectives of the work.

3.2.4 Modeling Work in Context

Changes in context affect how work is done by driving the selection of appropriate strategies, which motivates three more constructs: *strategy*, *configuration* *variable*, and *decision action*. In this framework, strategy specifically refers to a set of actions or functions to achieve a higher level function (adapted from Vicente, 1999). As noted in Section 3.1, multiple strategies may achieve the same goal and one of them should be selected at a time as appropriate to immediate context. Configuration variables are a special class of resources representing current context to facilitate strategy selections. For example, when function allocation changes, this transition in context can be explicitly expressed by configuration variables, which may be a broad classification of the environment such as phase of flight.

Lastly, decision actions are a special class of actions that select strategies based on contextual factors: environmental, function allocation and within-agent (i.e., cognitive controls modes). The selected strategies are implemented by the decision action setting configuration variables and activating/deactivating actions, assigning actions to agents, and identifying which resources in the environment an action gets and sets.

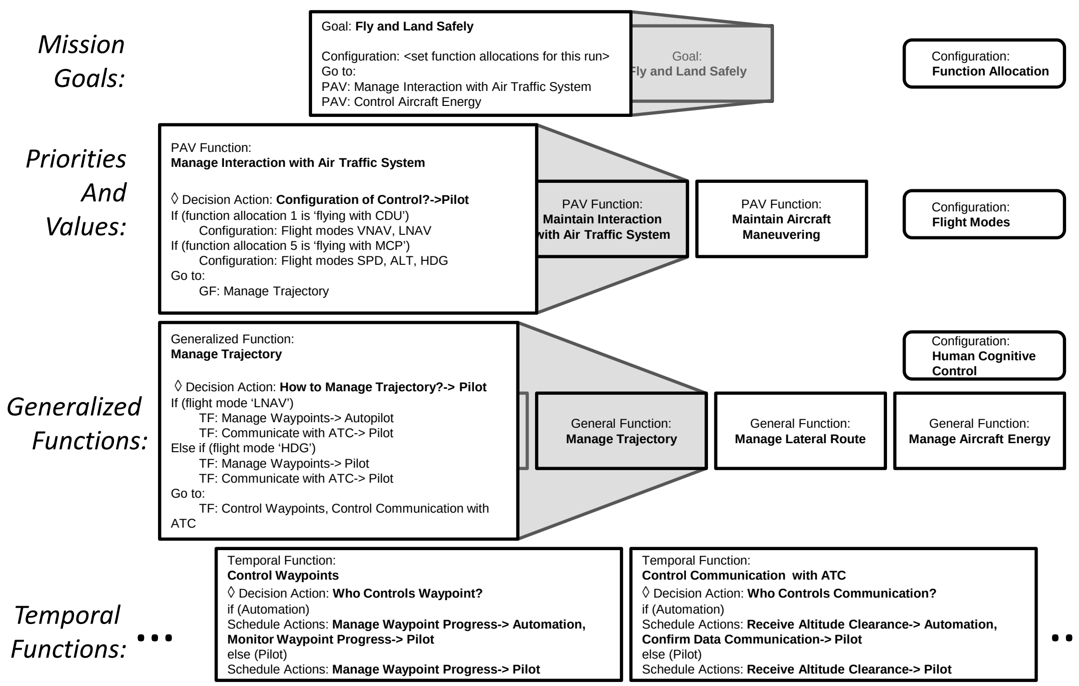

Figure 14. Strategy selection in the decision action based on the configuration variable

Figure 14 illustrates strategy selection by a decision action based on configuration variables. For example, controlling waypoints can be performed by the autopilot (strategy 1) or by the pilot (strategy 5). The decision action "How to control trajectory" selects the strategy based on the configuration variable "Flight modes" (depicted as a round-cornered box on the right).

Note that different strategies are best represented at different levels of abstraction within the work model. In Figure 14, for example, the Decision Action "Who Controls Waypoints" is a specific, detailed construct best described within a temporal function. On the other hand, the strategy for configuring the control of the automation (which depends on the chosen function allocation) is best described within a higher level of abstraction, priorities-and-values function using Flight Modes.

3.3 Work Model that Computes: Making It Computes

This section discusses how these work models are then simulated by the "Work Models that Compute" (WMC) simulation framework, indicating how the work is assigned to, and executed by, agent models. Section 3.2 specified the static aspects of actions and the resources that agents get and set. In addition, each action and resource needs to specify additional constraints that specify their timing in a dynamic simulation and that link actions to agents and to resources. (The simulation engine also constructs the reverse links of agents to actions and resources to actions.) These additional attributes for actions and for resources are listed in Table 5 and Table 6, respectively.

Table 5. Attributes of an action required for dynamic simulation

Attribute	Description
Next update time	Each action is required to reflect when it will next be updated to correctly model its processes.
Executing agent	The agent who executes this action.
Responsible agent	Annotation of the agent who is responsible for the outcome of this action.
Action priority	The priority of this action compared to other actions. (This can be used in a task management component within the human agent model.)
Action duration	The required duration for which the agent model is occupied with this action.
Resources that action gets	Resources that this action needs to get.
Resources that action sets	Resources that this action needs to set.

Table 6. Attributes of a resource required for dynamic simulation

Attribute	Description
Actions linked to this resource	A list of actions linked to (that can *get*) this resource.
Actions that can set	A list of actions that can set this resource.
Last update time	The time when a new value was set within the resource.

With these constructs, a work model can be simulated by the WMC framework. Specifically, the simulation engine scans through the work model and loads the actions into the WMC simulation engine's action list. As shown in Figure 15, the simulation engine, then, orders them by their next update time, that is, the action declaring the soonest next update time is placed at the top of the list and executed next. After the action at the top of the list is executed, that executed action declares a new next update time and is sorted back into the action list accordingly. Whichever action is now at the top of the action list is executed next, and so forth for the duration of the simulation.

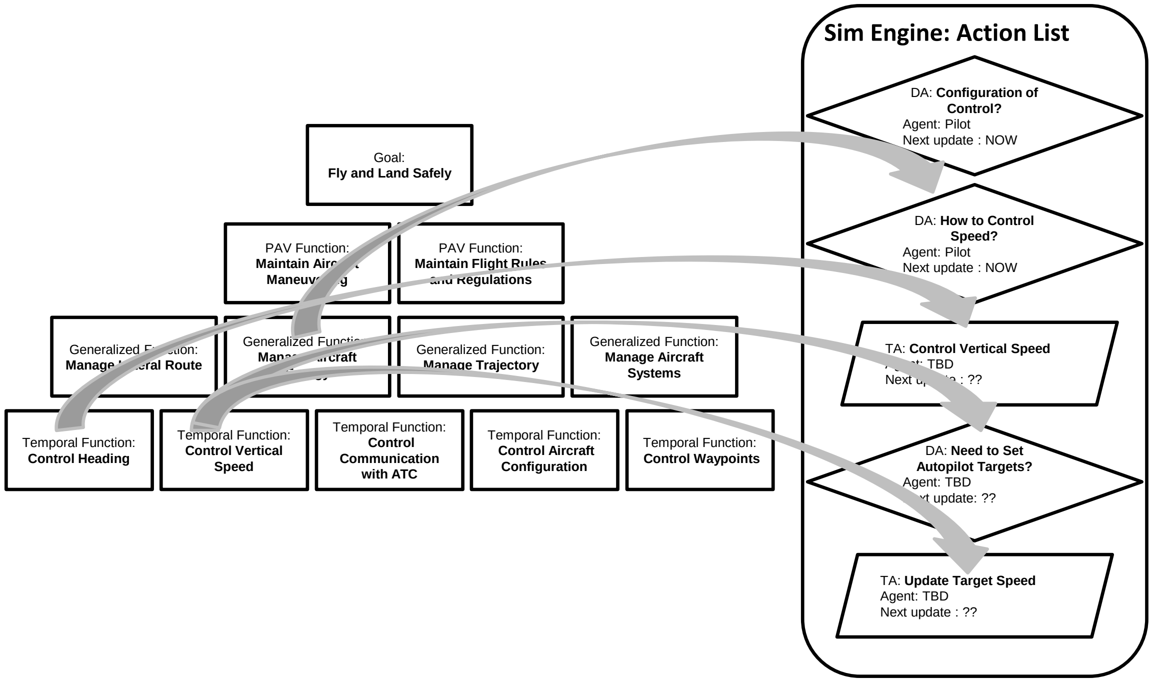

Figure 15. Composing an action list at time = 0 from the static work model in simulation engine

3.3.1 Agent Models in WMC Simulations

The agent model in the WMC framework is inspired by agent-based modeling and by Laughery and Corker's (1997) concept of "first-principle modeling of human-performance" in which the same aspects of human performance are applied to all of an agent's tasks, as previously applied in Corker's Air-MIDAS model.

Thus, the agent models in the WMC framework have a unique relationship with work models. Rather than conflating models of agent behavior with the work to be performed, the framework uses agents as a means to further allocate and regulate the decision and temporal actions described in the work model. This approach has many capabilities including tracking workload (or taskload), modeling how an agent might manage multiple actions demanding their attention, and assessing whether an agent, when an action is required, has the correct information and other environment resources available to perform it. Therefore, this approach is not intended to predict or describe individual elements of human cognitive behavior within isolated tasks, but instead to add to a description of work a further view of how agents - particularly human agents - may manage the actions they have to execute in response to the demands of the work environment.

The basic agent model executes actions whenever the simulation engine requests. This basic agent model executes all actions without any limitation intrinsic to characteristics and capabilities of humans such as delay in executing actions, workload saturation, interior dynamics (e.g., information processing) or maintaining any internal representation of context and task. The basic agent model can be of use for modeling automation and for examining whether an operating procedure, if perfectly executed, will have sufficient performance.

In addition, some aspects of human performance are included in a more elaborate model, as shown in Figure 16. This agent model can mimic multiple aspects of human performance. First, the actions currently active within the agent can be tallied, representing (or indicating) workload/taskload. Likewise, when an incoming action tips the agent's active actions beyond some limits of capability, the agent interrupts or delays lower priority actions, placing sufficient lower-priority actions in queues of interrupted and delayed actions to keep the list of active actions within the bounds of their capacity. Whenever the capacity becomes available with the completion of an active action, the next-highest-priority interrupted or delayed action will be executed. Finally, the agent can routinely poll the action list for its own upcoming actions as an indication of which actions the agent "expects."

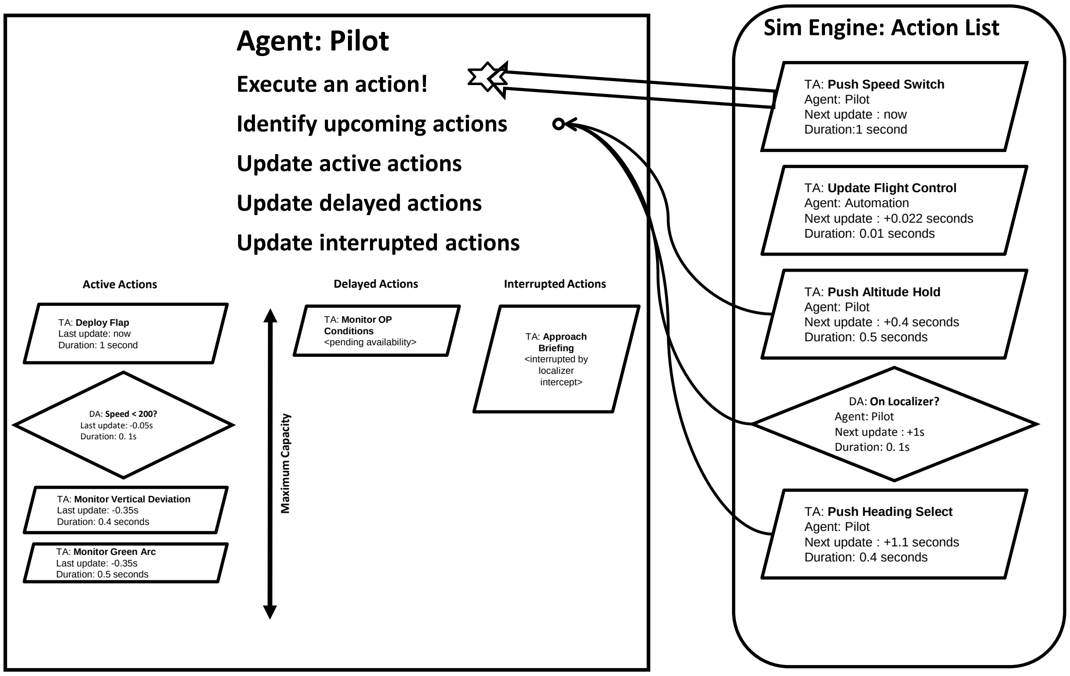

Figure 16. Agent model structure with characteristics of human performance

3.4 Summary

This chapter described a modeling and simulation framework that can capture and predict issues with human-automation function allocation. To be able to capture subtle-yet-important aspects from which issues with human-automation function allocation would arise, a model of human-automation function should represent that work is a purposeful activity on the environment and that work is situated and embodied in the environment, including the selection of strategies in response to context. In addition, a model should capture the taskwork as well as the teamwork. Lastly, a model should capture the way that agents abstract work, and its complexity should be manageable by the modeler.

The effects of function allocation are described in this model in several ways. First, the function allocation will be implemented as selection criteria for high-level strategies. Then, further lower-level function-allocation-specific strategies will be progressively selected from the static work model and will result in agents each being assigned actions and given access to resources in the environment.

Then, the dynamic simulation will demonstrate function allocation as all its actions are executed throughout the simulation. Given a set of relationships among actions, resources, and agents, the simulation reveals potential conflicts in using resources, potential workload spike or saturations of human agents, and, further, potential situations where human agents and automated agents attempt to achieve conflicting goals. Therefore, the issues with function allocation are identified statically and dynamically.

Therefore, with the modeling framework described in this chapter, a function allocation can be assessed in terms of the issues summarized in Chapter 2. To measure these issues systematically, the following chapter will introduce eight metrics of function allocation and how to assess them.

CHAPTER 4

ASSESSING THE FUNCTION ALLOCATION METRICS

This chapter introduces quantifiable metrics of the function allocation issues identified in Chapter 2. Throughout, this chapter details how they can be assessed from both the models provided in Chapter 3 and, where applicable, observations of real operations or human-in-the-loop experiments.

Chapter 2 identified the eight issues with function allocation. Based on these eight issues, eight types of metrics are established here. The purpose of these metrics is to assess the extent to which each of issues exist with a given function allocation. Specifically, the eight types of metrics assess workload, incoherency in a function allocation, mismatches between responsibility and authority, interruptive automation, automation boundary conditions, human adaptation to context, stability of the human's work environment, and mission performance.

4.1 Workload

At its most precise, workload can be defined as an intervening variable that indicates the relationship between the demands of the environment and the capability of the operator (Kantowitz, 1988 as cited in Kantowitz, 2000). In work models and their dynamic simulation, each action can be annotated by the workload it imposes. Workload can be incorporated as a uni- or multi-variate construct, depending on the type of agent model desired. A univariate representation of workload requirements, while perhaps simplistic, can potentially be parameterized from observational or human-in-the-loop experimental data. A multi-variate representation can employ a multiple-resource model of workload that provides more detail, potentially at the expense of difficulty in validating each action's workload requirements. When knowledge of workload per action is not available, the simpler construct of "taskload" may instead be modeled where each action is described as imposing a workload of “1” and, thus, the sum of workload required at any time represents the number of tasks assigned.

These assessments of workload or taskload, because they originate from detailed work models, can provide a quick baseline inclusive of the *non-linear* additions of load resulting from different function allocations. For example, assigning an additional function to an agent may not linearly increase load by the sum of the function’s component actions if some of those actions are already performed by the agent for other functions. Conversely, removing one function from a human agent may not linearly decrease by the sum of the function’s component actions because those component actions may also be required by (or contribute to) other functions assigned to that agent. As a result the agent may need to perform other actions to collect necessary information that is no longer a byproduct of the function that is removed.

A static descriptor of potential workload or taskload just sums the number of tasks potentially required of each agent in a given function allocation. This provides a quick assessment to identify gross concerns with a function allocation. A dynamic estimate of workload or taskload can be recorded throughout simulations. A useful analysis of this dynamic workload is to identify periods in which estimated workload or taskload exceeds the human’s estimated capacity limits as a maximum human taskload, which can then be examined to see if they represent brief workload spikes or longer-duration periods of workload saturation.

In real operations, a number of methods may be used to assess the workload associated with a given function allocation. These include subjective ratings in multiple dimensions such as measuring via psychophysical scaling (see Dixon, Wickens & Chang, 2005; Gopher & Braune, 1984), multi-dimensional rating systems (see Hart & Staveland, 1988; Potter, 1989), and a combination of each to compensate and, sometimes, to strengthen effectiveness of the resulting measurements (see Cegarra & Chevalier, 2008).

4.2 Coherency of a Function Allocation

The coherency of a given function allocation can be quantified by measuring how many *levels* up the abstraction hierarchy model one can traverse while fully describing the functions assigned to one agent. For example, in Figure 17, the pilot and automation are each assigned a smattering of functions across the work domain and, thus, the coherency for either agent can only be expressed as *level 2*. In contrast, in Figure 18, the pilot is given the entire set of functions for the priorities and values function “Maintain Flight Rules and Regulations” and “Maintain Interaction with Air Traffic Systems” and, thus, the pilot’s coherency can be expressed as *level 3*.

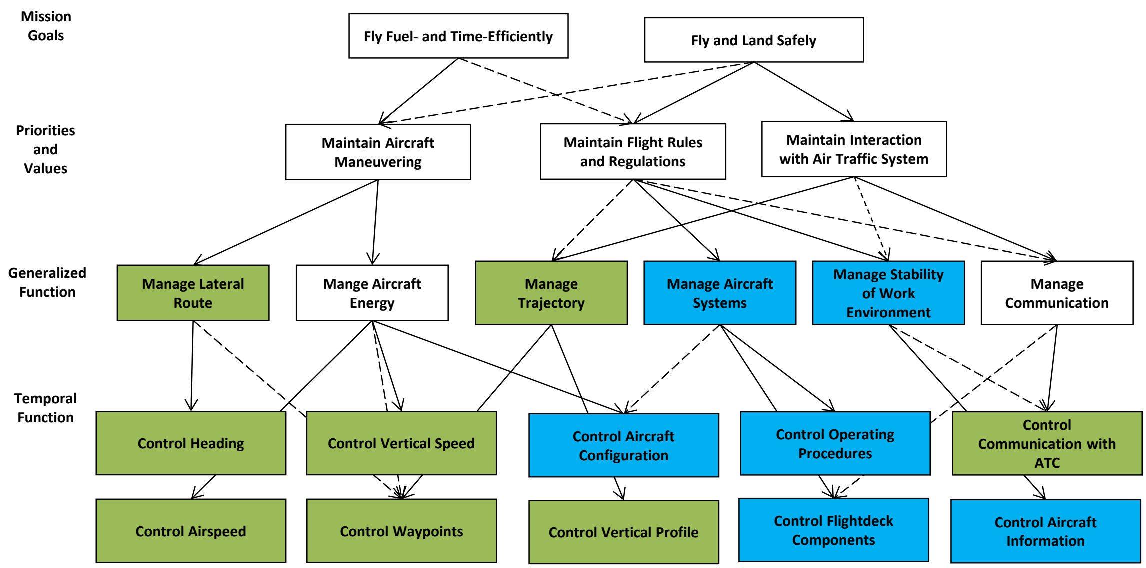

Figure 17. Assessing coherency: low level of coherency (functions assigned to automation are green-coded while functions assigned to the pilot are blue-coded)

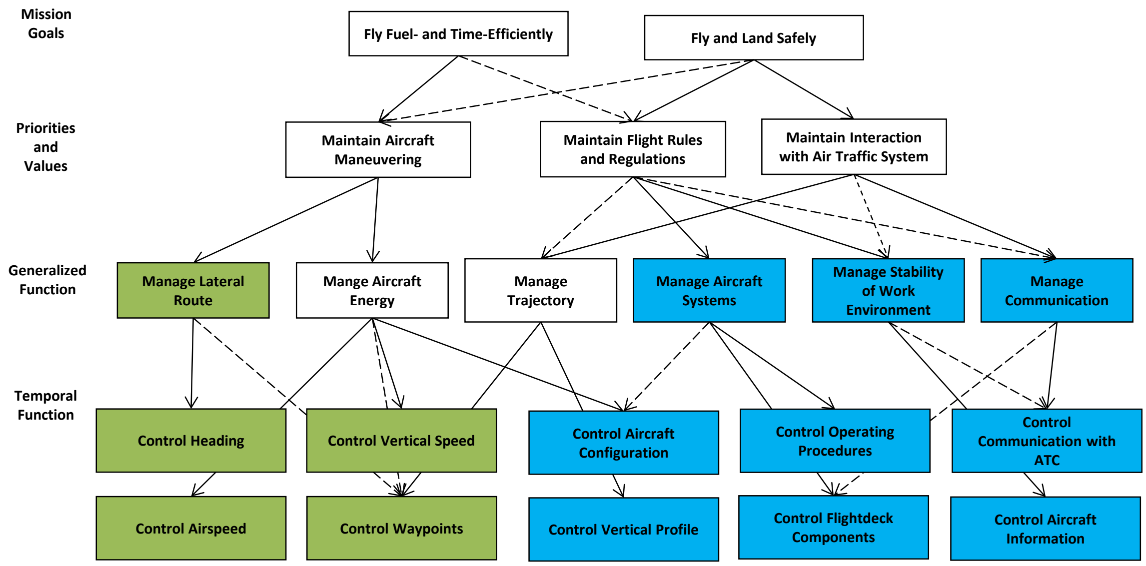

Figure 18. Assessing coherency: high level of coherency (functions assigned to automation are green-coded while functions assigned to the pilot are blue-coded)

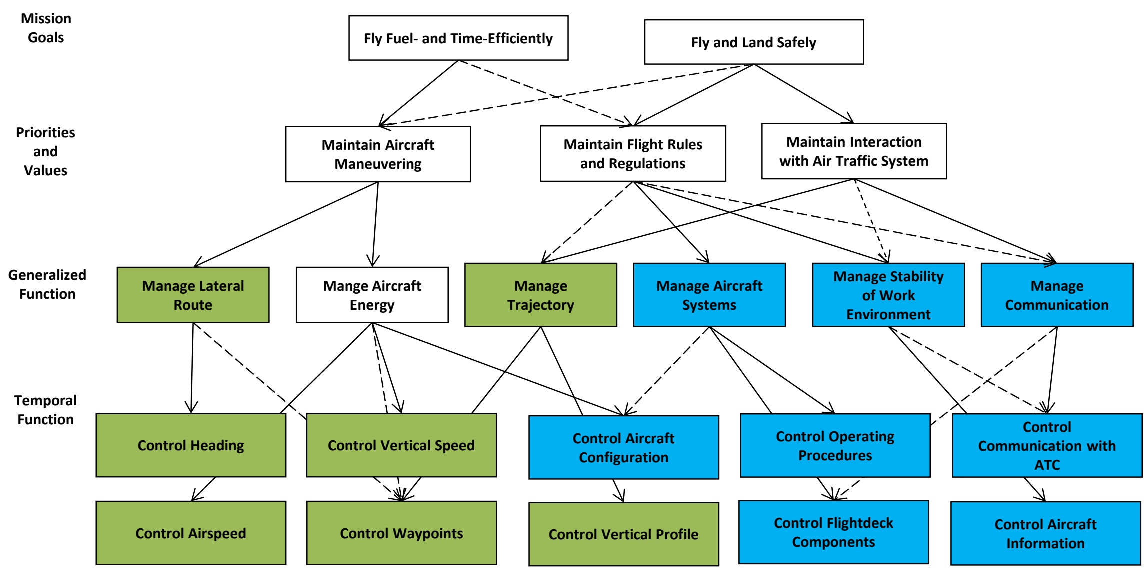

Figure 19. A different function allocation resulting in the same coherency level as the function allocation shown in Figure 17.

Although this measure provides a general indication of coherency, it is a fairly coarse measure. For example, Figure 17 and Figure 19 are different function allocations that are both measured as *level 2*. However, Figure 17 has fewer functions assigned entirely to the pilot or the automation. Therefore, an additional measure can also be assessed: the coherency percentage. The coherency percentage is measured as the percentage of the functions that are each 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. For example, the function allocation shown in Figure 17 is measured as having a coherency percentage of 67% because 14 functions are assigned to one agent compared to 21 functions which are required to describe the work of the entire team (the pilot and flight deck automation). In a same manner, the function allocation shown in Figure 19 has a coherency percentage of 71% (15 functions are assigned to one agent out of 21). Therefore, coherency percentage can further distinguish differences between function allocations.

Of course, the particular values resulting from this metric will depend on the structure of the abstraction hierarchy 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 enable comparison between function allocations for obvious effects that break-up an agent's work in a manner that cannot be sensibly abstracted.

The coherency of function allocation can also be assessed dynamically during simulations by counting "resource conflicts" as indications where two agents may act upon the same values in the environment (specifically, where their actions attempt to set the same variable) or other inter-dependencies between agents. Such situations highlight where agents' function allocation may overlap, or require detailed coordination.

Finally, the coherency of a function allocation can also be measured during the observations of the operations. The static measure can be identified qualitatively through interviews and surveys while the dynamic measure can be obtained through observations of the operations while the experimenter (or modeler) records the instances in which agents have conflicting outputs.

4.3 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. When the agent who is authorized to perform the function is also responsible for the outcome, there is no need for separate supervising or monitoring. However, when the agent who is responsible for the outcome of the function and the agent who is authorized to perform the function are not the same, then additional functions to supervise and monitor are induced. For example, if the agent responsible for controlling airspeed is not the agent authorized to control the airspeed, the agent with the responsibility would have to monitor and ensure that the airspeed is controlled properly. Therefore, the metric of the mismatch between responsibility and authority can be quantified by the number of functions with mismatches between responsibility and authority (static measure) and by the number of the teamwork actions induced by these mismatches (static and dynamic measures).

4.4 Interruptive Automation

The frequency of a human's actions that are interrupted by the automation can be recorded. For example, interruptions to pilots while conducting checklists and the approach briefing can be recorded during simulation or observed in actual operations. Also, to assess if the automation interrupts unnecessarily or unduly, the impact of the interruption can be assessed qualitatively or by quantitatively measuring its impact on performance.

4.5 Automation Boundary Conditions

The appropriateness of the allocation of functions to the automation in context can be measured by instances (and/or durations) in which the automation is placed outside of its boundary conditions. In some cases, these boundary conditions may be explicitly acknowledged and any exceedance can be measured. In addition, “being placed outside of its boundary conditions” can be inferred when automation cannot achieve its targets while operating according to its specification. For example, an autoflight system is normally given a target altitude and airspeed to achieve by a particular location (waypoint). However, there are environmental conditions (e.g., tailwind, late descent instructions from air traffic controllers) in which the autoflight system cannot achieve these targets. In these situations, this metric records the duration of any excessive deviation in the target profile. Likewise, this metric can be observed from real operations by identifying conditions in which the automation is placed outside of its operating conditions or cannot meet its targets. The inability to meet its targets may also be considered an issue with mission performance in some cases.

4.6 Human Adaptation to Context

The appropriateness in context of the functions allocated to the human can likewise be measured. As a qualitative measurement, the modeler can identify when specific cognitive control modes are not supported by the actions required of them by a given function allocation. Of note, human-automation interfaces commonly assume a fairly strategic behavior from the human, including extensive button pushes before an action will be executed. These function allocations may be optimal in other contexts but, when the human cannot provide the patterns of behavior these function allocations demand, they become suboptimal or, worse, unsafe. For example, in case of a pilot and flight deck automation, when an air traffic controller provides different air traffic instructions than the ones normally given by standard arrival and approach procedures, the pilot is required to respond within a limited amount of time which may not be enough for function allocations requiring extensive reprogramming via the CDU (e.g., highly- or mostly automated function allocations described in Chapter 2).

Further, should dynamic simulations use a human performance model capable of switching between cognitive control modes, this metric can be assessed by examining the proportion of time spent in each cognitive control mode and the frequency with which the modes change. These results can then be examined for conditions in which, for example, pilots are likely to be in cognitive control modes in which they may shed monitoring tasks, or situations where a function allocation may drive overly-frequent changes in cognitive control modes.

These cognitive control modes can also be estimated during real operations or human-in-the-loop experiments using surveys and interviews (Feigh, 2008; Feigh, K. M., 2010; Stanton, Ashleigh, Roberts & Xu, 2001).

4.7 Stability of the Human's Work Environment

The stability of each human's work environment can be measured by counts of the actions predicted by the humans. If actions are planned by the pilots, then they can "predict" when and which action will be demanded. In contrast, the pilots may be asked to perform actions that they predict neither when nor which (Type 1 Unpredictability). For example, during the arrival and approach phases of flight, pilots normally expect air traffic instructions close to the printed standard arrival and approach procedures. However, the pilots may be given entirely different instructions by the air traffic controller. As an intermediate level of predictability, the pilots may recognize the potential for some actions to be required, but not be able to predict exactly when these actions will be demanded (Type 2 Unpredictability).

Therefore, metrics of the (in)stability of the work environment are the number of both type 1 actions and type 2 actions. This "instability level" can be given as a percentage of the number of each type of unpredicted actions with respect to the total number of actions executed by the agent.

4.8 Mission Performance

Finally, the collective team's mission performance can be measured via simulations or assessed in actual operations. The definition of mission performance is dependent on the domain and the team's objectives but should reflect the mission goals given in the work model. In some cases, performance includes measures of the safety and robustness in off-nominal scenarios. With simulation using simple agent models, this metric will capture the extent to which established work practices can meet mission goals; this can then be contrasted with the performance predicted using more intricate agent models and performance observed in human-in-the-loop experiments and actual operations.

CHAPTER 5

CASE STUDY: ARRIVAL-APPROACH MODEL

Chapter 3 established the WMC framework and Chapter 4 proposed function allocation metrics. This chapter demonstrates the framework and the metrics in the case of a pilot flying an aircraft with flight deck automation during the arrival and approach phases of flight. A model of arrival and approach is established with each of the four different function allocations available in the flight deck described in Chapter 2. This model is then simulated within the WMC framework and used to assess the static and dynamic measures of the each function allocation.

This chapter starts by specifying a nominal arrival and approach scenario following a continuous descent arrival procedure. Next, the chapter describes how the arrival-approach model is formed within the WMC framework: a work model is described in detail, including: the configuration variables describing function allocations and high-level strategies; representations of the four different function allocations and of a range of pilot cognitive control modes; and simulation of the functions and actions comprising the taskwork and teamwork described in the work model. The chapter then describes an experiment design in which the function allocation metrics are assessed from the work model and its dynamic simulation in the nominal scenario and off-nominal scenarios. The chapter ends with a discussion of the efficacy of the metrics in identifying issues with function allocation.

5.1 Describing the Arrival-Approach Model

This section describes the work model of a specific arrival and approach scenario. The section illustrates how different function allocations are represented in the work model in terms of the teamwork and taskwork actions included in each function allocation, as well as how different cognitive modes are represented. Lastly, the section details how the WMC framework uses the work model for dynamic simulation.

The case study examines a scenario spanning the arrival and approach phases of flight described in Chapter 2: the aircraft flies the RIIVR TWO Arrival starting approximately 30 nm from the T/D point at 31000ft and then follows the ILS or LOC RWY 25L approach to 150ft (MSL). For each waypoint in the flight route, Table 7 lists its time of arrival, altitude requirement, airspeed requirement, and distance to runway threshold. Note that the time to arrive is an approximate time that assumes a nominal speed profile.

Table 7. List of waypoints and altitude and speed profile of the flight route (note that time to arrive is an approximate time)

Name	Time to Arrive (sec)	Altitude (ft.)	Speed (kts)	Distance to Runway Threshold (nm)
TOD	300	31000	350	-94.96
GRAMM	650	19000	325	-64.27
RUSTT	730	16000	300	-57.22
HABSO	800	15000	280	-52.68
RIIVR	850	12100	280	-46.40
DECEL	950	10600	265	-42.70
LUVYN	960	10330	250	-38.93
KRAIN	1040	9700	240	-34.62
FUELR	1170	6500	240	-24.35
GAATE	1300	4400	220	-17.46
HUNDA	1405	3500	185	-10.42
LIMMA	1540	1890	163	-5.39
RWY 25L	1660	150	150	0.00

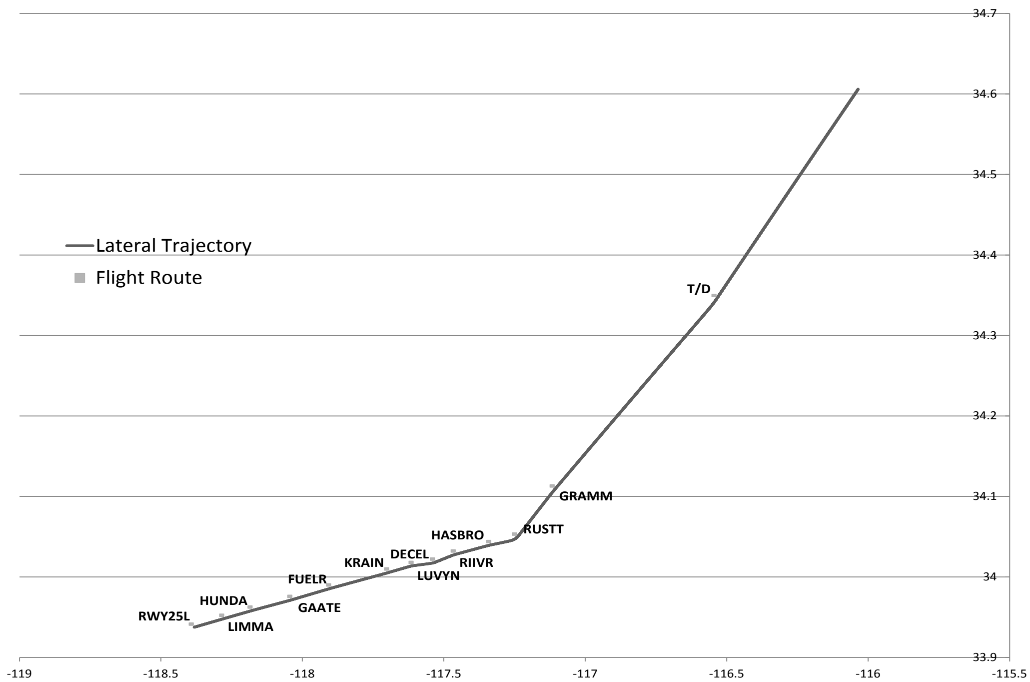

Figure 20. Lateral profile of nominal (continuous descent) arrival and approach scenario

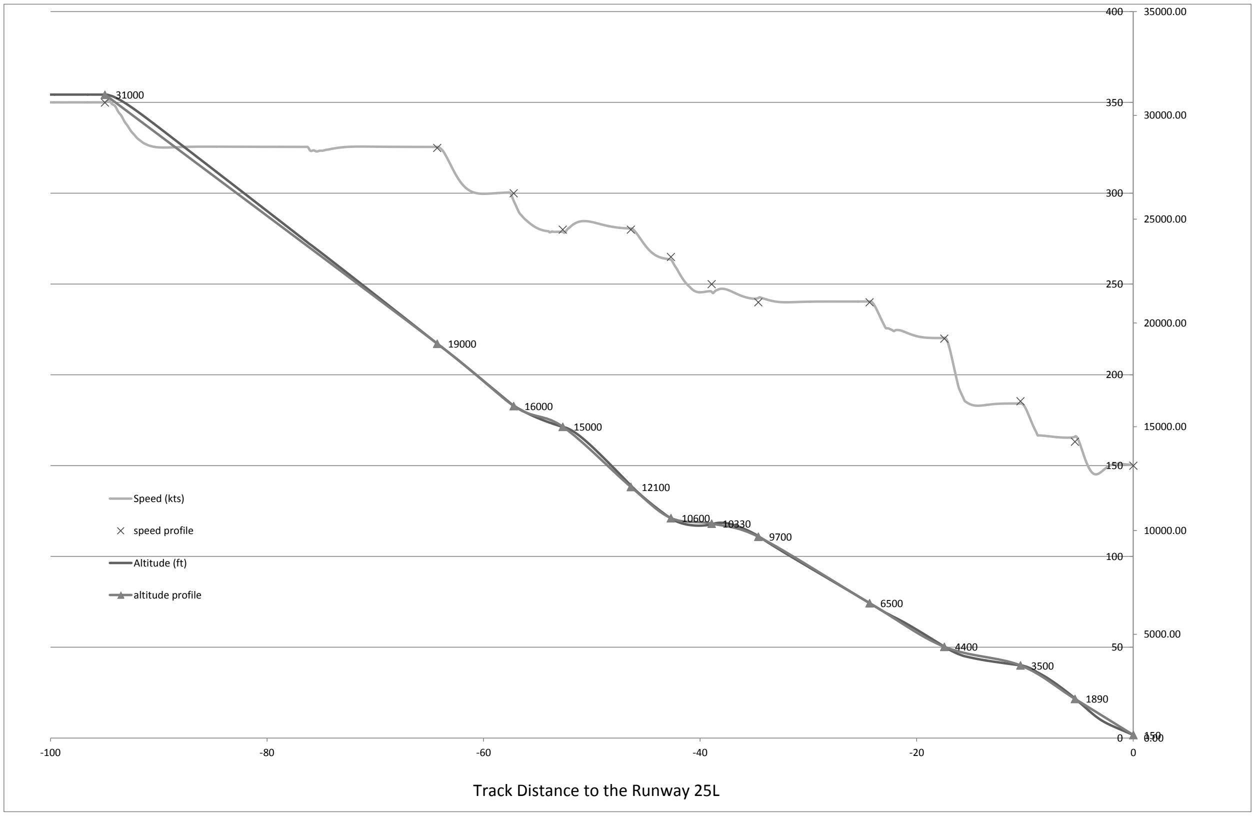

Figure 21. Vertical profile of a nominal (continuous descent) arrival and approach with associated altitude and airspeed restrictions of the STAR and the approach procedure

Figure 20 illustrates the lateral profile of a nominal (continuous descent) arrival and approach, and Figure 21 illustrates its altitude-speed profile. Throughout, the aircraft may receive many different instructions from air traffic controllers. A controller may clear the aircraft for the entire arrival and approach at once, or (more likely) clear the aircraft down to successively lower altitudes as the aircraft flies through the airspace down to the runway. Note that, although the aircraft can be cleared down to any waypoint, altitude, or the runway at once, if there are intermediate waypoints and corresponding altitude and/or airspeed restrictions, then the aircraft should maintain all altitude- and airspeed- restrictions marked with *Xs* and *triangles*.

5.1.1 Describing the Arrival-approach Model at Multiple Levels of Abstraction

As described in Chapter 3, the description of work during the arrival and approach phases is represented using multiple levels of abstraction. This work model, the arrival-approach model, is illustrated in Figure 22. The identified functions and the relationships between each level shown in Figure 22 are carefully modeled so that different function allocations can be represented.

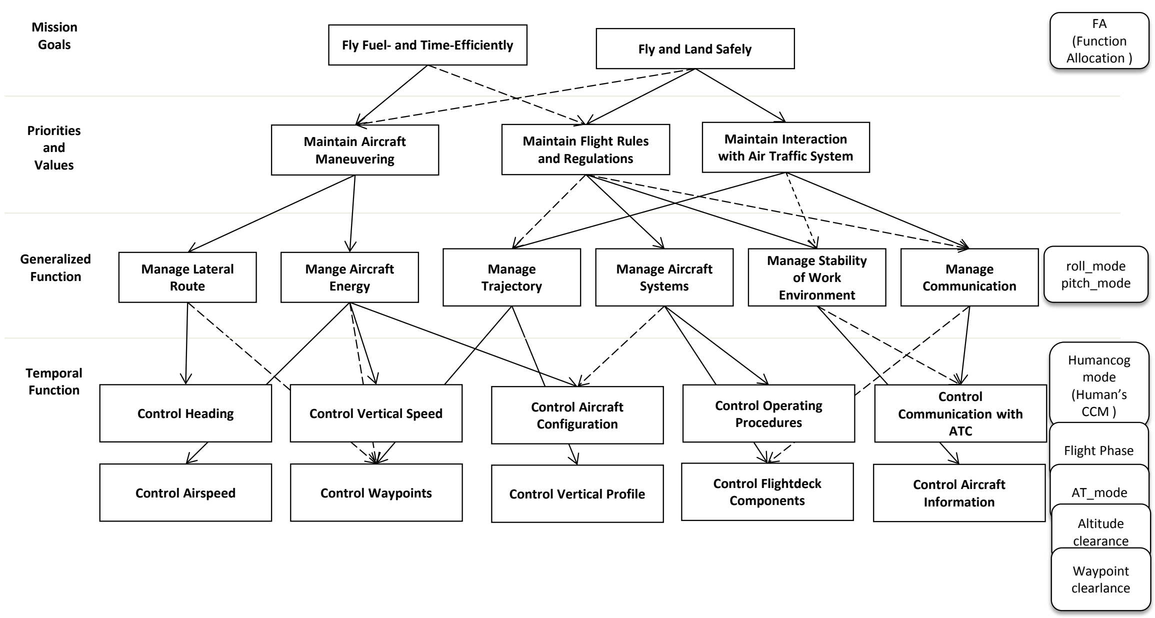

Figure 22. The arrival-approach model (note that round-cornered boxes indicate configuration variables)

The goals of the arrival-approach model are "Fly fuel- and time- efficiently" and "Fly and land safely." To achieve these goals, the pilot and the flight deck automation are required to fly the aircraft within the air traffic system, while maintaining flight safety as well as maximizing energy efficiency. The three priorities-and-values (PAV) functions are: "Maintain aircraft maneuvering," "Maintain flight rules and regulations," and

“Maintain interaction with air traffic system,” reflecting “how” the pilots and the flight deck automation achieves the mission goals.

Then, the work model expands these functions into lower-level, generalized functions: “Manage lateral route,” “Manage aircraft energy,” “Manage trajectory,” “Manage aircraft systems,” “Manage stability of work environment,” and “Manage Communication.” These functions describe how the PAV functions are achieved. Finally, these generalized functions are decomposed into temporal functions, maintaining means-end relationships with the generalized functions.

The temporal function level describes work in detail. The “Manage lateral route” generalized function is achieved by “Control heading” based on the given (planned) lateral path. The “Manage aircraft energy” function is achieved by “Control vertical speed” and “Control airspeed” functions that are critical to stabilized approach and fuel consumption. The “Manage aircraft systems” generalized function is achieved by “Control operation procedures” such as operating procedures and other checklists. The “Manage stability of work environment” generalized function is achieved by “Control aircraft information,” anticipating the future states and preparing for any abnormality during the flight. The “Manage trajectory” generalized function is achieved by “Control waypoints” and “Control vertical profile” that comprise planning, reviewing, and modifying trajectories as needed. In summary, the temporal functions are: “Control heading,” “Control vertical speed,” “Control airspeed,” “Control aircraft configuration,” “Control waypoints,” “Control vertical profile,” “Control operating procedures,” “Control communication with ATC” and “Control aircraft information.”

The temporal functions are interconnected to each other where they describe coupled dynamics. This coupling is often reflected by actions grouped in one temporal function using resources updated by actions in other temporal functions. For example, the resource “Airspeed,” set by the temporal function “Control Airspeed,” is then referenced in almost all the other temporal functions.

Configuration variables are used to represent contextual factors such as flight modes, the pilot’s cognitive control modes and function allocation. These configuration variables are placed graphically in Figure 22 at the level of abstraction at which they support strategy selection and are summarized in Table 8. These configuration variables are used in three ways: first, to select strategies corresponding to different cognitive control modes; second, to select strategies to distribute specific actions and resources to each agent, and invoke appropriate strategies, according to a given function allocation; and, third, to select strategies in response to environmental factors.

Table 8. Configuration variables used in the arrival-approach model

Variables	Comments
FA	Indicates the function allocation
humancogmode	Indicates the humans’ cognitive control modes
roll_mode	Indicates the mode of autopilot for lateral navigation
pitch_mode	Indicates the mode of autopilot for vertical navigation
AT_mode	Indicates the mode of autothrottle
flight_phase	Indicates the phase of the flight (e.g., cruise, approach, etc.)
waypoint_clearance	Indicates the next waypoint in the assigned flight path
altitude_clearance	Indicates the altitude clearance given by ATC

5.1.2 Modeling Different Function Allocations

Of special interest here is the representation of function allocations. Taskwork actions are assigned to different agents with each function allocation, and each function allocation also adds its own set of teamwork actions. Thus, while the overall structure of the taskwork, and the higher-level functions are the same throughout different function allocations, the decisions made within them, and the assignment of teamwork and taskwork actions assigned to each agent, varies between function allocations.

The available function allocations in the flight deck were described in detail in Chapter 2. Table 9 briefly summarizes these function allocations.

Table 9. Function allocations modeled in the arrival-approach model

Function Allocation		Description
FA1	Highly-automated function allocation	Pilot using LNAV/VNAV with air traffic instructions directly processed by the flight deck automation.
FA2	Mostly-automated function allocation	Pilot using LNAV/VNAV with pilot receiving air traffic instructions and programming the autoflight system.
`FA3	Mixed-automated function allocation	Pilot selecting the vertical autoflight targets and receiving air traffic instructions, and the FMS commanding the lateral autoflight targets.
FA4	Mostly-manual function allocation	Pilot selecting the autoflight targets and receiving air traffic instructions.

Consider the function allocation in which the pilot uses LNAV/VNAV with air traffic instructions directly processed by the flight deck automation (FA1). This function allocation is the most highly automated; thus, most of the taskwork actions are assigned to the flight deck automation. Table 10 lists the temporal functions for this function allocation in the first column, and actions within each temporal function are assigned to each agent, pilot and flight deck automation, as listed in the next two columns.

Table 10. Function allocation 1: "Highly-automated" function allocation (teamwork actions in bold).

Temporal Function	Pilot	Automation
Control Vertical Profile	**Modify CDU Pages** **Reduce Airspeed for Late Descent** **Confirm Target Altitude** **Confirm Target Speed**	Manage Waypoint Progress
Control Waypoints	**Modify CDU Pages** **Monitor Waypoint Progress** **Confirm Active Waypoint** **Monitor Dist Active Waypoint**	Calculate Dist Current Waypoint Evaluate Flight Phase Manage Waypoint Progress Direct To Waypoint
Control Communication With ATC	Respond Handoff **Confirm Data Communication**	Receive Altitude Clearance Receive ILS Clearance Receive Waypoint Clearance
Control Heading	**Monitor Heading Trends**	Update Lateral Control
Control Vertical Speed	**Monitor Altitude** **Monitor Vertical Deviation**	Adjust Speed Control Update Pitch Control Evaluate Vertical Mode Evaluate VNAV Mode Transition Evaluate Alt Restriction Mode Altitude Reminder
Control Airspeed	**Monitor Descent Airspeed**	Update Thrust Control Calculate Speed Deviation
Control Aircraft Configuration	Deploy Flap Deploy Gear Deploy Speed Brake Retract Speed Brake **Confirm Configuration Change**	
Control Aircraft Information	**Verify TOD Location** **Verify Crossing Restriction**	
Control Operating Procedures	Perform Approach Briefing Perform Approach Checklist Perform Landing Checklist	
Control Flight Deck Components	**Turn off Altitude Alert** **Respond to Drag Required**	

On the other hand, Table 11 describes actions required for the mostly-manual function allocation in which the pilot is flying the aircraft by selecting the heading, airspeed, and altitude targets and autopilot modes using the MCP (FA4). Compared to FA1 shown in Table 10, the mostly-manual function allocation (FA4) does not require the human to perform teamwork actions such as "monitor waypoint progress." Instead, many taskwork actions such as "manage waypoint progress" are shifted from the automation to the human. Also different teamwork actions such as dialing the altitude selector have been added to the list due to the use of a different interface (i.e., MCP) in the flight deck.

Table 11. Function allocation 4: "Mostly-manual" function allocation (teamwork actions in bold).

Temporal Function	Pilot	Automation
Control Vertical Profile	**Monitor Altitude** **Reduce Airspeed for Late Descent**	
Control Waypoints	Manage Waypoint Progress Direct To Waypoint	Calculate Dist Current Waypoint Evaluate Flight Phase
Control Communication With ATC	Receive Altitude Clearance Receive ILS Clearance Receive Waypoint Clearance Respond Handoff Request Clearance	
Control Heading	**Dial Heading Selector** **Push Heading Selector** **Monitor Heading Trends**	Update Lateral Control
Control Vertical Speed	**Dial Altitude Selector** **Dial VS Selector** **Push Alt Hold Switch** **Push FLCH Switch** **Push Vertical NAV Switch** **Push Vertical Speed Switch** **Monitor Green Arc**	Update Pitch Control Evaluate Vertical Mode Evaluate Alt Restriction Mode Altitude Reminder
Control Airspeed	**Dial Speed Selector** **Push Speed Switch** **Monitor Descent Airspeed**	Update Thrust Control Calculate Speed Deviation
Control Aircraft Configuration	Deploy Flap Deploy Gear Deploy Speed Brake Retract Speed Brake **Confirm Configuration Change**	
Control Aircraft Information	**Verify TOD Location** **Verify Crossing Restriction**	
Control Operating Procedures	Perform Approach Briefing Perform Approach Checklist Perform Landing Checklist	
Control Flight deck Components	**Turn off Altitude Alert** **Respond to Drag Required**	

The mostly-automated function allocation (FA2) is shown in Table 12. This function allocation is similar to the highly-automated function allocation (FA1) except that communicating with ATC is assigned to the pilot. Therefore, temporal actions such as “receive altitude clearance” and “receive waypoint clearance” are allocated to the pilot. Also, because the pilot executes these actions directly, the teamwork action “confirm data communication” is no longer needed.

Table 12. Function allocation 2: "Mostly-automated" function allocation (teamwork actions in bold).

Temporal Function	Pilot	Automation
Control Vertical Profile	**Modify CDU Pages** **Reduce Airspeed for Late Descent** **Confirm Target Altitude** **Confirm Target Speed**	Manage Waypoint Progress
Control Waypoints	**Modify CDU Pages** **Monitor Waypoint Progress** **Confirm Active Waypoint** **Monitor Dist Active Waypoint**	Calculate Dist Current Waypoint Evaluate Flight Phase Manage Waypoint Progress Direct To Waypoint
Control Communication With ATC	Receive Altitude Clearance Receive ILS Clearance Receive Waypoint Clearance Respond Handoff Request Clearance	
Control Heading	**Monitor Heading Trends**	Update Lateral Control
Control Vertical Speed	**Monitor Altitude** **Monitor Vertical Deviation**	Adjust Speed Control Update Pitch Control Evaluate Vertical Mode Evaluate VNAV Mode Transition Evaluate Alt Restriction Mode Altitude Reminder
Control Airspeed	**Monitor Descent Airspeed**	Update Thrust Control Calculate Speed Deviation
Control Aircraft Configuration	Deploy Flap Deploy Gear Deploy Speed Brake Retract Speed Brake **Confirm Configuration Change**	
Control Aircraft Information	**Verify TOD Location** **Verify Crossing Restriction**	
Control Operating Procedures	Perform Approach Briefing Perform Approach Checklist Perform Landing Checklist	
Control Flight Deck Components	**Turn off Altitude Alert** **Respond to Drag Required**	

Lastly, the "mixed" function allocation (FA3), shown in Table 13, describes the function allocation in which the pilot executes actions relevant to managing the vertical profile (e.g., by setting altitude and speed targets in the MCP and commanding the appropriate autoflight modes to achieve them) while the flight deck automation executes actions relevant to managing the lateral profile (i.e., lateral control via LNAV to follow the route programmed into the FMS using the CDU). Therefore, this function allocation is “mixed” between FA2 and FA4.

Table 13. Function allocation 3: “Mixed (using CDU and MCP)” function allocation (teamwork actions in bold).

Temporal Function	Pilot	Automation
Control Vertical Profile	**Monitor Altitude** **Reduce Airspeed for Late Descent**	
Control Waypoints	Manage Waypoint Progress **Monitor Waypoint Progress** **Confirm Waypoint Target** **Monitor Dist Active Waypoint**	Calculate Dist Current Waypoint Evaluate Flight Phase Direct To Waypoint
Control Communication With ATC	Receive Altitude Clearance Receive ILS Clearance Receive Waypoint Clearance Respond Handoff Request Clearance	
Control Heading	**Monitor Heading Trends**	Update Lateral Control
Control Vertical Speed	**Dial Altitude Selector** **Dial VS Selector** **Push Alt Hold Switch** **Push FLCH Switch** **Push Vertical NAV Switch** **Push Vertical Speed Switch** **Monitor Green Arc**	Update Pitch Control Evaluate Vertical Mode Evaluate Alt Restriction Mode Altitude Reminder
Control Airspeed	**Dial Speed Selector** **Push Speed Switch** **Monitor Descent Airspeed**	Update Thrust Control Calculate Speed Deviation
Control Aircraft Configuration	Deploy Flap Deploy Gear Deploy Speed Brake Retract Speed Brake **Confirm Configuration Change**	
Control Aircraft Information	**Verify TOD Location** **Verify Crossing Restriction**	
Control Operating Procedures	Perform Approach Briefing Perform Approach Checklist Perform Landing Checklist	
Control Flight Deck Components	**Turn off Altitude Alert** **Respond to Drag Required**	

5.1.3 Representing Pilot Cognitive Control Modes

In the arrival-approach model, three cognitive control modes are used to represent three patterns of behaviors: opportunistic, in which the pilot only responds to immediate needs in context, thus attempting only to "finish the job;" tactical, in which the pilot conducts monitoring and information seeking efforts as a part of procedures; and strategic, in which the pilot conducts monitoring and information seeking actions to anticipate upcoming needs.

Thus, in this arrival-approach model, these cognitive control modes determine how a pilot monitors the state of the aircraft and the environment, and how he/she prepares for the future taskwork actions as anticipated by some of the monitoring actions, as shown in Table 14. In the opportunistic mode, the pilot only executes the most essential monitoring actions such as "Monitor Altitude" and "Monitor Descent Airspeed." These monitoring actions are essential in that the outcomes of these actions initiate necessary taskwork actions such as deploying flaps or executing checklists. In the tactical mode, the pilot executes most of the monitoring actions including confirming the behavior of the automation as changes are entered into the MCP and CDU. In the strategic mode, the pilot executes all actions listed in Table 14. These include certain monitoring actions that attempt to respond to anticipated future states and, thus, to ameliorate impacts from the off-nominal events (e.g., if the descent clearance appears to be past due, reduce airspeed within the allowed margin of 0.02 Mach or request a lower airspeed).

Table 14. Monitoring actions included within each cognitive control mode and their timing

States Relevant to the Action	Actions of the Pilots	Cognitive Control Mode		
		Opportunistic	Tactical	Strategic
States of Aircraft				
Configuration	Confirm Configuration Change		Periodically	Anticipated
Position	Monitor Altitude	As required	Periodically	Anticipated
	Monitor Vertical Deviation		Periodically	Anticipated
	Monitor Distance to Waypoint		Periodically	Anticipated
	Verify TOD Location			Anticipated
	Verify Crossing Restriction			Anticipated
	Monitor Green Arc		Periodically	Anticipated
	Confirm Target Altitude		Periodically	Anticipated
	Confirm Target Airspeed		Periodically	Anticipated
Direction	Monitor Heading Trends		Periodically	Anticipated
	Monitor Waypoint Progress		Periodically	Anticipated
	Confirm Active Waypoint		Periodically	Anticipated
Speed	Monitor Descent Airspeed	As required	Periodically	Anticipated
	Reduce Airspeed for Late Descent			Anticipated
States of Environment				
Communication	Confirm Data Communication		Periodically	Anticipated
	Request Clearance			Anticipated

Pilot cognitive control modes are further differentiated by how the pilot determines when to perform the actions. Actions are "anticipated" and thus scheduled more frequently (or targeted to future times of likely interest) when the pilot is in the strategic mode seeking to "notice" any changes in the states of aircraft and environment. In contrast, those actions are scheduled "periodically" when the pilot is in the tactical mode, as if the pilot is executing a routine scan pattern.

5.1.4 Dynamic Aspects of the Model

An aspect of dynamics is captured in the work model by having each action define its next update time. This can represent a variable time step or a fixed time step. For example, the action that models strategic pilot monitoring of flight path progress, "Monitor Waypoint Progress," anticipates its next update time by the following calculation:

Next_update_time = distance_to_next_waypoint / maximum_airspeed.

Although this is not an exact calculation, it provides a conservative estimate of when waypoint progress should be next monitored. Therefore, the next update time will get closer (and the pilot will monitor more frequently) as the aircraft gets closer to the waypoint.

Another example considers the autoflight system in the VNAV control mode with a target altitude indicated in the MCP altitude window. When the aircraft reaches that target altitude, the autoflight system transitions to a different flight mode that captures and holds that altitude. For this action, the next update time calculation is:

Next_update_time = (MCP_altitude – current_altitude) / maximum_vertical_speed.

Note that, as the difference in altitude gets smaller, the next update time gets closer. Therefore, a minimum is specified so that the next update time does not become unreasonably small, corresponding to the time step of the fastest component contributing to the triggering condition (in this case, the aircraft dynamics).

Two temporal actions are used to simulate the aircraft: aircraft dynamics (`calculate_guidance`) and guidance (`flyaircraft`). These two actions are executed with a time step of 0.05 sec, emulating the autoflight system and aircraft dynamics with a full 6 DOF dynamics model of a Boeing 747-400, with a model of autoflight behavior used (and validated) in prior human-in-the-loop studies (e.g., Kalambi, Pritchett, Bruneau, Endsley & Kaber, 2007).

Temporal actions also represent specific taskwork and teamwork processes. For example, monitoring the distance to the next waypoint is provided by calculateDistCurrentWaypoint. When the air traffic controller instructs a lower altitude, for example 19000ft at GRAMM, the pilot responds and updates the target altitude in the MCP altitude window by receive_altitude_clearance and dialAltitudeSelector.

Once the aircraft reaches the T/D point, several actions are executed. First, the action evaluateFlightPhase updates the flight phase from CRZ to DES (cruise to descent, i.e., arrival), updates the configuration variables pitch_mode and AT_mode, and schedules the temporal actions updatePitchControl and updateThrustControl. The decision actions evaluateVNAVModeTransition and/or evaluateAltRestrictionMode are scheduled depending on the configuration variables pitch_mode and AT_mode.

5.2 Experiment Design

This experiment was designed to validate the proposed model framework and function allocation metrics' ability to predict and capture the issues with function allocation identified in Chapter 2. This experiment includes four independent variables and eight classes of dependent variables. The four independent variables are scenario, function allocation, cognitive control mode, and maximum human taskload, as shown in Table 15. The eight classes of dependent variables span the proposed function allocation metrics described in Chapter 4, as shown in Table 16. The following sections detail these variables and the scenarios used in the experiment.

Table 15. Independent variables and their levels

Independent Variable	Level	Description
Scenario	SC0	Nominal arrival and approach (achieving continuous descent arrival)
	SC1	Air traffic controller instructing clearance to descend after T/D point (late descent)
	SC2	Air traffic controller instructing unexpected re-route
	SC3	Unexpected tailwind
Function Allocation	FA1	Pilot using LNAV/VNAV with air traffic instructions directly processed by the flight deck automation
	FA2	Pilot using LNAV/VNAV with pilot receiving air traffic instructions and programming the autoflight system
	FA3	Pilot updating the vertical autoflight targets and receiving air traffic instructions, and the FMS commanding the lateral autoflight targets
	FA4	Pilot programming the autoflight targets and receiving air traffic instructions
Cognitive Control Mode	CCM1	Opportunistic
	CCM2	Tactical
	CCM3	Strategic
Maximum Human Taskload	MHT1	Tight (3)
	MHT2	Moderate (7)
	MHT3	Unlimited (50)

Table 16. Dependent variables and their measurements

Dependent Variable	Measurement
Workload	Total number of actions executed
	Combined duration of actions executed
	Total number of taskwork actions executed
	Combined duration of taskwork actions executed
	Total number of interactions (with the flight deck automation) executed
	Combined duration of interactions (with the flight deck automation) executed
	Total number of monitoring actions executed
	Combined duration of monitoring actions executed
	Total number of workload spikes
	Duration of workload saturation
Coherency of a Function Allocation	Coherency level of pilot's function allocation
	Coherency level of automation's function allocation
	Coherency percentage of a function allocation
Mismatches between Responsibility and Authority	Total number of mismatched temporal functions
	Total number of actions executed as induced by a mismatch
	Combined duration of actions executed as induced by a mismatch
Interruptive Automation	Total number of actions in which the pilot is interrupted by the automation
Automation Boundary Conditions	Duration of vertical deviation higher/lower than ±400ft
	Duration of airspeed deviation higher/lower than 10/15knots
	Duration of required vertical speed higher than maximum vertical speed of the aircraft or the descent rate programmed in the FMS
Human Adaptation to Context	Discussion of the impact of CCM1, CCM2, and CCM3 on the pilot's work
Stability of the Human's Work Environment	Total number of actions not predicted by the pilot when and what will be required (type1)
	Total number of actions not predicted by the pilot when they will be required (type2)
Mission Performance	Average thrust used per second
	Time to land
	Number of violations of crossing restrictions
	Average vertical deviation from the nominal profile
	Average speed deviation from the commanded speed

5.2.1 Scenario Descriptions

In this experiment, the aircraft flies a STAR, *RIIVR TWO ARRIVAL*, and then a standard approach procedure, *ILS or LOC RWY 25L*, from the northeast into LAX (previously shown in Chapter 2). The simulation starts with the aircraft flying at flight level 310 and approximately 30nm from the T/D point. The simulation ends when the aircraft reaches 150ft (MSL) on final approach. Each scenario is designed to exercise certain function allocation metrics.

The nominal scenario (SC0) provides a baseline for each of the dynamic measures and follows the vertical and lateral profile shown earlier in Figure 20 and Figure 21. Thus, the scenario represents the ideal case of the arrival and approach phases executed according to the printed arrival and approach procedures. The air traffic controller clears the aircraft at appropriate times to lower altitudes (flight level 190, 12100ft, 6500ft, 3500ft, and 1890ft) as indicated in the arrival and approach charts, as summarized in Table 17. There is no wind and, thus, no deviation from vertical profile. The pilot is responsible for following the route including meeting the altitude and airspeed restrictions indicated in the STAR chart as well as the altitude clearance provided by the air traffic controller, as listed earlier in Table 7.

Table 17. ATC script with time and altitude cleared for the nominal (continuous descent) arrival and approach scenario

Time at Which an Instruction is Given	Altitude Cleared
100sec	Flight Level 190
450sec	12100ft
570sec	9700ft
700sec	4400ft
1100sec	150ft (Runway)

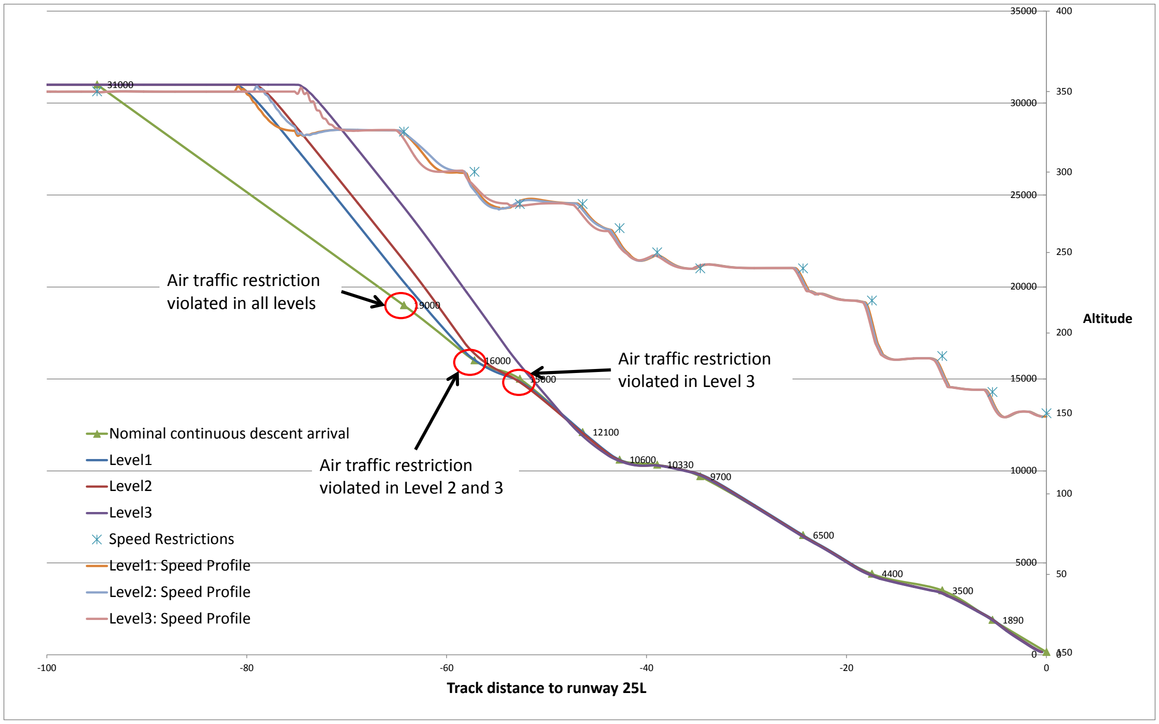

Figure 23. Vertical profile of three levels of the late descent scenario (SC1) with violated air traffic restrictions at each level highlighted

Figure 23 depicts the vertical profile of the "late descent" scenario (SC1). In this scenario, the air traffic controller is delayed in initiating the descent, and provides an altitude clearance to a lower altitude, 12100ft, "a certain amount of time" after the aircraft passes the T/D point. Note that the pilot and the flight deck automation are still required to meet the air traffic restrictions at 19000ft, 16000ft, and 15000ft. The time of the controller's delayed descent instruction is varied by three levels, probing the capability of the pilot and the flight automation to respond to a progressively "more-abnormal" situation. The times at which the initial descent instruction is given are shown in Table 18. Figure 23 highlights the air traffic restrictions violated at each of the levels.

Table 18. ATC script with time and altitude cleared for the late descent scenario (SC1)

Time at which descent clearance is given			Altitude Cleared
Level 1	Level 2	Level 3	
430sec	450sec	500sec	12100ft

This scenario challenges the pilot's ability to meet air traffic restrictions. With FA3 and FA4, in which the pilot is assigned to managing the vertical profile, vertical speed is limited only by what the aircraft physically achieve (while maintaining flight safety). In contrast, with FA1 and FA2, in which the flight deck automation is assigned to managing the vertical profile, the rate of descent that the FMS can command is limited to a maximum vertical speed, the default value for which the pilot can override in the FMS using the CDU. If the descent clearance is given "too late," the vertical speed required to meet the restriction cannot be achieved. Thus, the limit on the vertical speed applied in FA1 and FA2 is usually lower than the actual maximum vertical speed that the aircraft can achieve. Therefore, for FA1 and FA2, when the air traffic descent clearance is given, the pilot does not notice the limit on the vertical speed programmed in the FMS, limiting the capability of the automation to meet the air traffic restrictions. This assumption models the difficulty in understanding the flight deck automation used in FA1 and FA2. With FA3 and FA4, when the air traffic descent clearance is given, the pilot has more direct control over the target of the autoflight system.

The pilot's cognitive control mode is also assumed to impact behavior in this scenario. To ameliorate the risk of violating air traffic restrictions, the pilot in the strategic mode will implement a potential risk-mitigating action. As discussed in Section 5.1.3, the pilot in the strategic mode is modeled as reducing airspeed by the "allowed" margin (0.2 Mach) as seen as he/she realizes that the descent clearance is late, as well as anticipating and monitoring for deviations as appropriate.

Therefore, in this scenario, the automation boundary condition metric is expected to capture situations in which the automation is placed outside of its boundary condition. In addition, the mission performance metric is expected to capture the violation of air traffic restrictions, demonstrating how robust each function allocation is in terms of their collective capability to meet air traffic restrictions. Further, the different cognitive control modes are expected to result in different performance.

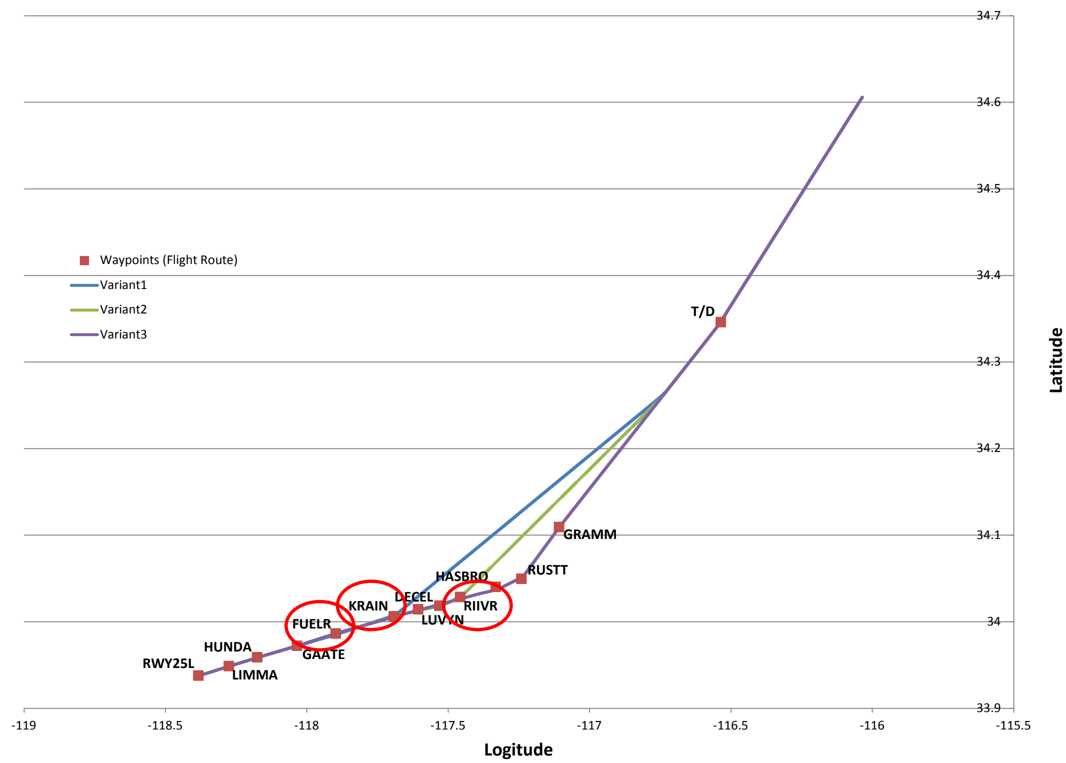

Figure 24. Lateral profile of three variants of the unpredicted re-routing scenario (SC2), re-routed waypoints for each variant highlighted

Figure 24 depicts the lateral profile of the "unstable work environment" scenario (SC2) in which air traffic instructions are not what the pilot would expect from the printed arrival and approach procedures. As described in the nominal continuous descent scenario (SC0), the simulation starts with the aircraft flying at flight level 310 (approximately 31,000ft MSL) and approximately 30nm from the T/D point. The air traffic controller clears the aircraft to descend to 19000ft at “GRAMM” at an appropriate time. However, the next clearance requires a direct routing to a different (unpredicted) waypoint either before the aircraft reaches the 19000ft at GRAMM (e.g., variant 1 and variant 2) or at some time after passing GRAMM (variant 3). Note that the clearance is a direct routing that negates air traffic restrictions at intermediate waypoints. Table 18 describes these three variants of the unstable work environment scenario. In Figure 24, the waypoints defining the re-routes are highlighted.

Table 19. ATC script with time and altitude cleared for the three variants of the unstable work environment scenario (SC2)

1		2		3	
Time at which an instruction is given	Altitude or Waypoint Cleared	Time at which an instruction is given	Altitude or Waypoint Cleared	Time at which an instruction is given	Altitude or Waypoint Cleared
100sec	19000ft	100sec	19000ft	100sec	19000ft
390sec	KRAIN	390sec	RIIVR	800sec	FUELR
900sec	4400ft	820sec	4400ft	1000sec	4400ft
1100sec	150ft (Runway)	1100sec	150ft (Runway)	1100sec	150ft (Runway)

In this scenario, two aspects of different function allocations are modeled: 1) the pilot receives an unpredicted instruction, requiring him or her to perform unpredicted actions, and 2) the required action for the direct routing requires significant pilot interaction to program the flight deck automation in FA1, FA2, and FA3 where the automation is managing the lateral profile. Thus, the direct routing task is expected to have lower pilot workload in FA4.

Therefore, the work environment stability metric is expected to flag a greater percentage of actions as “unpredicted.” In addition, the workload metric is expected to capture workload spikes or saturation in FA1, FA2, and FA3 as the pilot reprograms the new route information into the FMS.

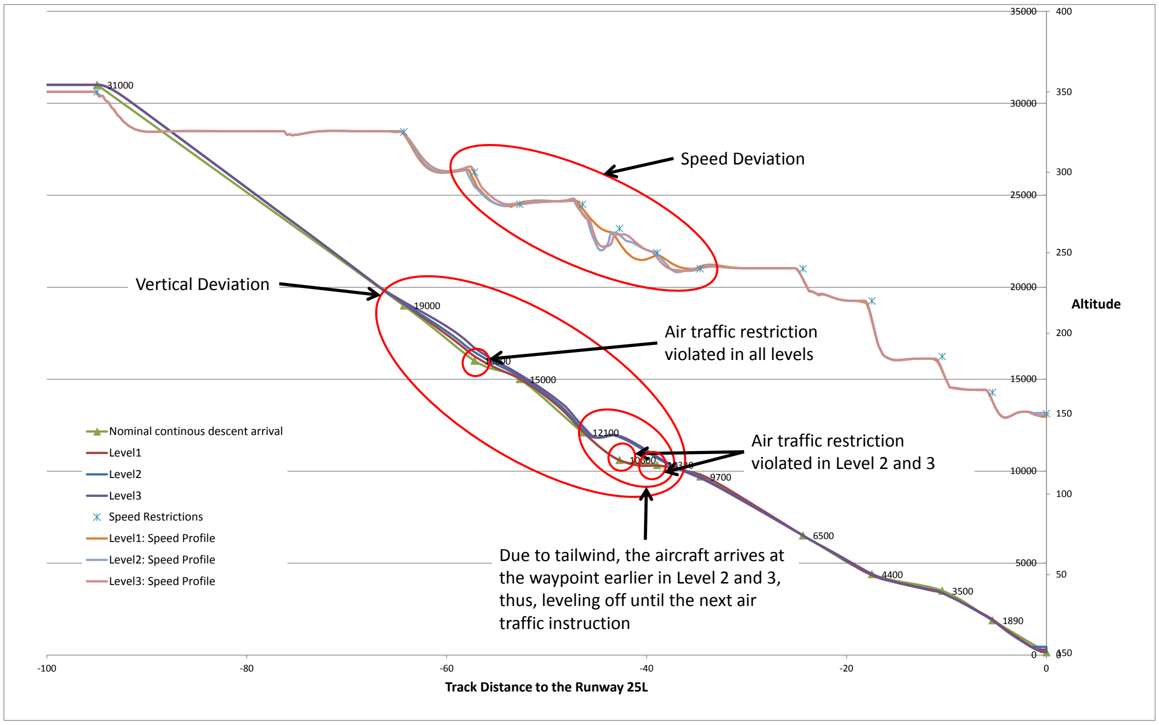

Figure 25. Vertical profile of three levels of unexpected tailwind scenario (SC3)

Figure 25 illustrates the vertical profile of the “tail wind” scenario (SC3). As with the nominal continuous descent scenario (SC0), the simulation starts with the aircraft flying at flight level 310 (approximately 31,000ft MSL) and approximately 30nm from the T/D point, and the air traffic controller provides the same instructions. However, the simulation generates an unexpected tailwind while the aircraft is between altitude 20000ft and 12000ft. Thus, the pilot and flight deck automation need to correct the vertical speed and airspeed from drifting above the planned profile as well as adjusting the heading of the aircraft to prevent drift laterally. To adjust these profiles with FA1 and FA2, the flight deck automation constantly updates the autoflight targets. In contrast, with FA3 and FA4 the pilot estimates and updates the targets via MCP. If this adjustment is performed poorly, then the aircraft drifts above the planned profile and laterally along the wrong track, resulting in vertical, speed, and track deviations. Table 20 describes the three levels of the tailwind scenario, and Figure 25 highlights their corresponding deviations.

Table 20. ATC script with time, altitude cleared, and tailwind for the unexpected tailwind scenario (SC3)

Time at which an instruction is given	Altitude Cleared	Tailwind		
		Level 1	Level 2	Level 3
100sec	19000ft	30knots (2000ft to 12000ft)	50knots (2000ft to 12000ft)	80knots (2000ft to 12000ft)
450sec	12100ft			
570sec	9700ft			
700sec	4400ft			
1100sec	150ft (Runway)			

This scenario captures specific issues with function allocation. With FA1 and FA2, the FMS is constantly updating the autoflight targets, adjusting the targets where the tailwind impacts on the trajectory. However, with FA3 and FA4, the pilot recalculates the autoflight targets periodically at a longer interval compared to FA1 and FA2; the specific interval used by the pilot varies with the pilot's cognitive control modes. The pilot in the strategic mode would estimate and adjust the targets more frequently compared to the tactical mode; he/she monitors every 15sec, but if the deviation gets larger than 50ft then he/she monitors every 2sec. On the other hand, the pilot in the tactical mode monitors simply every 60sec. The pilot in the opportunistic mode only focuses on adjusting the lateral profile, thus showing the poorest performance at managing the vertical profile.

Therefore, in this scenario, the automation boundary condition metric is expected to reflect durations in which the automation is placed outside of its boundary condition.

In addition, the pilot's cognitive control mode is expected to impact mission performance as the pilot monitors and acts upon more frequently in the strategic and the tactical modes than in the opportunistic mode. In addition, the work environment stability metric is expected to capture the impact of the unpredicted tailwind, which will require actions whose exact timing was not predicted by the pilot but must be responded to immediately.

5.2.2 Dependent Variables

The dependent variables assess the function allocation metrics described conceptually in Chapter 4. This section describes in detail how these metrics are assessed in this case study via the dependent variables. First, six aspects of workload are assessed using the number of actions demanded (i.e., taskload): The first four are (1) total taskload, (2) taskload due to taskwork, (3) taskload due to interaction with the flight deck automation, (4) taskload due to monitoring demands, and each of these four aspects is measured by both the number of actions demanded and their combined duration. The last two aspects consider extreme taskload in terms of (5) workload spikes and (6) workload saturation. Taskwork includes performing operating procedures, managing aircraft configuration, communicating with air traffic controllers, and any other actions where the pilot is directly operating on the work environment rather than interacting with automation. Interaction with the flight deck automation includes engaging autoflight modes and modifying CDU pages, i.e., actions used by the pilot to change the functioning of the automation. Monitoring includes monitoring information relevant to the flight deck automation's behavior and monitoring the states of the aircraft and the environment.

Expected findings with respect to workload are: 1) the highly- and mostly-automated function allocations require more monitoring actions than taskwork or interaction actions, 2) the mostly-manual function allocation demands more taskwork and interaction actions compared to the highly- and mostly-automated function allocations, and 3) the total workload will not decrease in highly- and mostly- automated function allocations compared to the mostly-manual one due to higher monitoring demands.

The coherency of a function allocation metric is measured on the work model in two ways: a coherency level and a coherency percentage. As described in Chapter 4, the metric is assessed as the level in the work model for which all functions allocated to an agent can be fully described. Therefore, a higher level indicates more coherent function allocation. This coherency level is measured for both the pilot and the automation. The coherency percentage assesses the work model and is measured as the percentage of functions entirely assigned to any one agent compared to the total number of functions required to completely describe the entire team's work.

An expected finding is the coherency level and the coherency percentage will be higher in the highly-automated and mostly-manual function allocations compared to the mostly-automated and mixed function allocations in which the functions assigned to each agent are scattered throughout the work domain.

A static measure of mismatches between responsibility and authority is measured on the work model, and a dynamic measure is recorded during simulations. The static measure counts the functions at the temporal function level for which the pilot is responsible for their outcome even as their execution is allocated to the flight deck automation. These mismatches induce teamwork actions from the pilot in the form of monitoring the flight deck automation's behavior (which will also be reflected in the measures of workload). The dynamic measure counts the total number of monitoring actions executed during the simulation and their combined duration. Expected findings are: 1) the static measure will be highest with the highly-automated function allocation and decrease as the function allocation becomes more "manual," and 2) the dynamic measure will show a similar trend to the static measure.

The interruptive automation metric is measured as the number of times the pilot is interrupted by the automation while he/she is performing procedures such as checklists.

The automation boundary conditions metric is measured as, first, the combined duration of vertical deviations from the proper flight profile, second, the combined duration of speed deviations between the actual airspeed and the commanded speed, and lastly, the duration of the required vertical speed being higher than the maximum vertical speed programmed in the FMS (with FA1 and FA2) or the maximum vertical speed that the aircraft can physically achieve (with FA3 and FA4). Specifically, a vertical deviation is identified when the aircraft is more than 400ft below/above the nominal profile (Casner, 2001; Stimpson, 2010) and a speed deviation is identified when the actual airspeed is higher than 10knots above and 15knots below the commanded airspeed (Stimpson, 2010).

The human adaptation in context metric is measured qualitatively by the modeler identifying inappropriate assumptions in any of the function allocations about pilot activity relative to any of the cognitive control modes. Expected findings are: (1) the highly- and mostly-automated function allocations (FA1 and FA2) will be a match with the pilot in strategic mode, (2) the mostly-manual function allocation (FA4) will be a match with the pilot in tactical mode, and (3) the opportunistic mode may not fully mirror the pilot behavior expected by any function allocation but its impact will be the least in the mostly-manual function allocation (FA4).

The stability of the human's work environment metric is measured as the total number of pilot actions that are not "predicted" by the pilot and their combined duration. Note that the notion of "predictability" in this case study has two levels: first, cases the pilot does not know that action would be demanded at all (type 1); and second, cases where the pilot knows that the action will be demanded, but does not know exactly when (type 2). Thus, the predictability in this case study is a comparison of the percentage of actions falling into each of these two categories of predictability.

The mission performance metric is measured via three different aspects of the mission goals: the average thrust used per second during a simulated flight (as a predictor of fuel burn and efficiency), the time that the aircraft takes to land (as a predictor of efficiency), and the number of violated air traffic restrictions (as a predictor of error exceedance and safety). Note that the average thrust used per second measure and the time to land measure of the nominal continuous descent arrival scenario (SC0) serves as baselines for measures in other scenarios. These measures are interpreted as follows: (1) a smaller measure of average thrust used per second is better, (2) a shorter time to land is better, and (3) fewer air traffic restrictions are better.

5.2.3 Experiment Design

Table 21 delineates the full factorial design of the experiment. Each combination of the independent variable is tested in a simulation run. The numbers in the table indicates the number of replications for each combination of function allocation (FA), human cognitive control mode (CCM), and maximum human taskload (MHT) due to the levels or variants within each scenario type: as mentioned previously, each scenario (except the nominal scenario) has three levels or variants.

Table 21. Full-factorial design with function allocation (4 levels), cognitive mode (3 levels), scenario (4 levels), and maximum human taskload (3 levels)

Max Human Taskload	Tight (3)												Moderate (7)												Unlimited (50)											
Cog Mode	CCM1				CCM2				CCM3				CCM1				CCM2				CCM3				CCM1				CCM2				CCM3			
SC/FA	FA1	FA2	FA3	FA4	FA1	FA2	FA3	FA4	FA1	FA2	FA3	FA4	FA1	FA2	FA3	FA4	FA1	FA2	FA3	FA4	FA1	FA2	FA3	FA4	FA1	FA2	FA3	FA4	FA1	FA2	FA3	FA4	FA1	FA2	FA3	FA4
SC0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
SC1	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
SC2	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
SC3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3

5.3 Results

The focus of this data analysis is on validating the function allocation metrics. Thus, the results will be discussed in terms of their ability to identify issues with function allocation, both as individual metrics and in combination.

5.3.1 Taskload (as a Predictor of Workload)

The taskload metric is measured in terms of several components: taskwork, teamwork due to interaction with the flight deck automation, and teamwork due to monitoring demands. Total taskload, i.e., all the actions demanded during the simulation, is the sum of these components. Each component and total taskload were measured in terms of the number of actions executed during each simulated flight, as well as their combined duration. Also, workload spikes were recorded as the number of instances where the total number of actions demanded of the pilot at one time was higher than the maximum human taskload, and workload saturation was assessed as the integration of required taskload and duration in which the total number of actions requested at one time was higher than the maximum human taskload.

To examine how much taskload was expected of the pilot, Figure 26 and Figure 27 illustrate the taskload and their combined durations as a function of function allocation and cognitive control mode averaged across all scenarios for those cases where the maximum human taskload was "Unlimited."

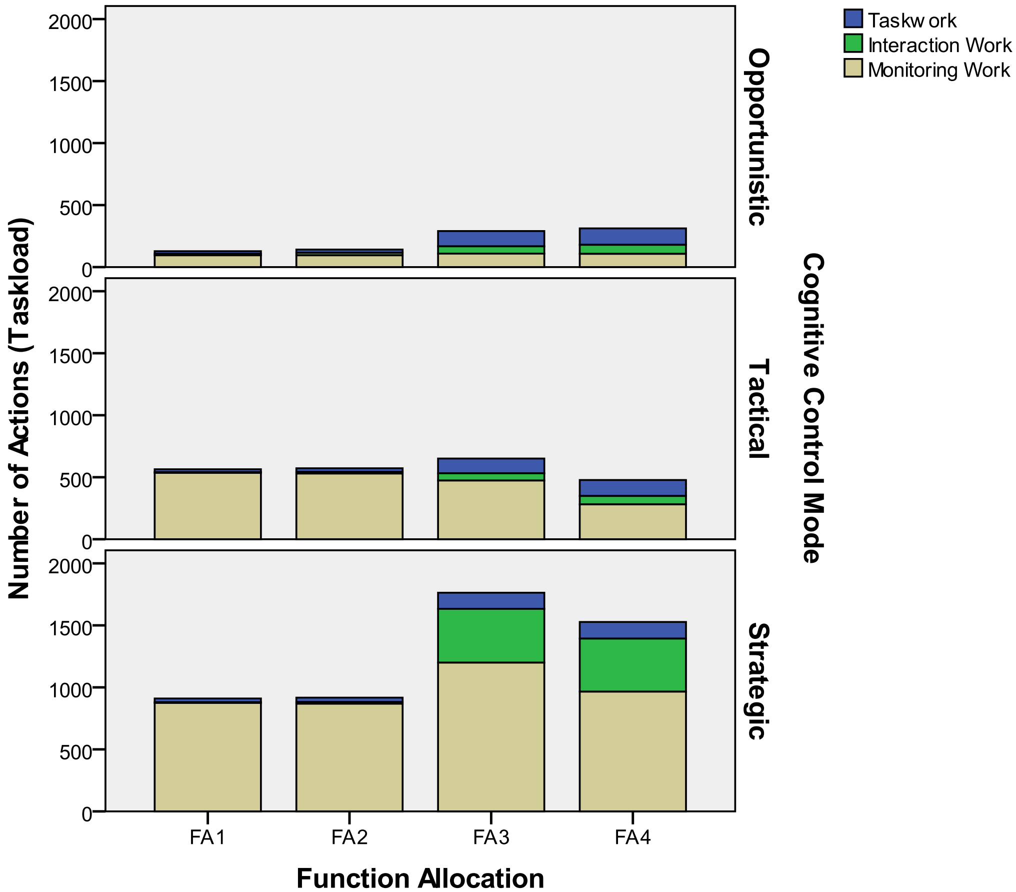

Figure 26. Number of actions (taskload) per simulated flight by function allocation and cognitive control mode averaged across all scenarios in cases with "Unlimited" maximum human taskload

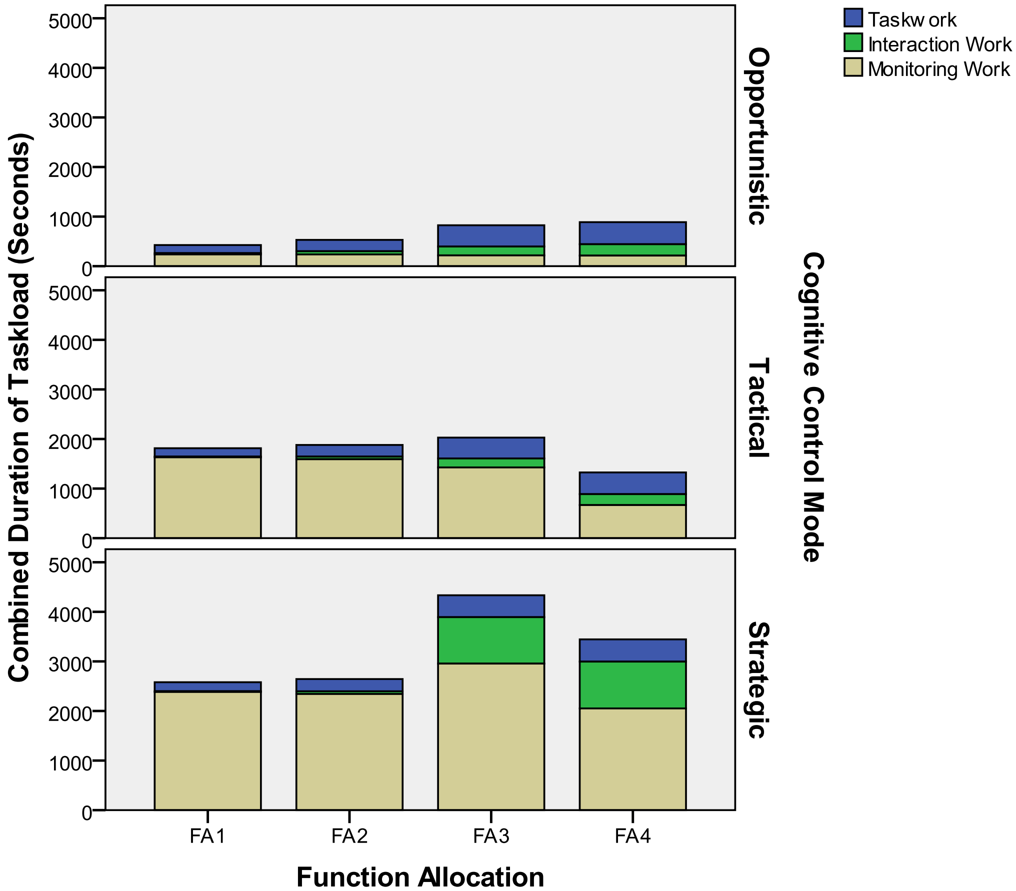

Figure 27. Combined duration of taskload per simulated flight by function allocation and cognitive control mode averaged across all scenarios in cases with "Unlimited" maximum human taskload

These figures clearly show that FA1 and FA2 are dominated by the monitoring actions whereas all three components of taskload appear in FA3 and FA4 in significant amounts. Consider FA3, the mixed function allocation, in which the pilot manages the vertical profile while the flight deck automation manages the lateral profile. This function allocation has the interaction work of FA4 and the monitoring work of FA2. Therefore, the pilot experienced the highest taskload in FA3. A one-way ANOVA found that the number of actions (taskload) varied significantly with function allocation (p=0.010) and the combined duration of taskload also significantly varied with function allocation (p=0.042). Post-hoc comparisons using the Tukey HSD test indicated that the mean number of pilot actions with FA1 (M=543.30, SD=327.12) and FA2 (M=543.60, SD=324.42) were significantly lower than with FA3 (M=901.70, SD=660.71), but they did not differ significantly from FA4 (M=772.50, SD=577.33). Note that assumptions of homogeneity of variance are violated; however, post-hoc robust tests of equality of means (Welch test [p=0.018] and Brown-Forsythe test [p=0.011]) identified significant differences in the means of each group. Thus, these results show that *automating one function did not consistently decrease the human taskload*, capturing some of the workload issues noted with function allocation in Chapter 2.

Examining Figure 26, the number of actions demanded in FA1 and FA2 appears to be less than in FA3 and FA4. However, the combined duration, shown in Figure 27, reveals that the type of actions demanded in FA1 and FA2 are different than in FA3 and FA4 and may occupy the pilot longer. Thus, post-hoc comparison using the Tukey HSD test of the combined duration of taskload found fewer differences between function allocations. While the mean score for FA1 (M=1606.63, SD=910.39) was significantly lower than FA3 (M=2396.00, SD=1521.09), none of other function allocations differed from each other. Of interest, the number of actions (taskload) required with most automated function allocation (FA1) did not differ from the least automated function allocation (FA4). Thus, these results show that *total workload was not reduced with the introduction of automation, but instead changed its nature*, another issue with workload noted in Chapter 2.

Examining the effects of cognitive control mode, as expected the strategic mode required the highest taskload among three modes, examining both the number of actions shown in Figure 26 and their combined duration shown in Figure 27. This increase is dominated by the interaction and monitoring components of taskload.

The interaction of function allocation and cognitive control modes on the taskload have similar trends between the number of pilot actions shown in Figure 26 and their combined duration as shown in Figure 27. The opportunistic cognitive control mode shows similar monitoring demands regardless of function allocation, although the taskwork and interaction demands are higher in the more manual function allocations (FA3 and FA4) compared to the more highly-automated function allocations (FA1 and FA2). Tactical mode shows increased monitoring demands in all four function allocations (with a greater increase in FA1 and FA2) while taskwork and interaction demands are similar compared to the opportunistic mode. Strategic mode shows the highest monitoring demands and interaction demands across all function allocations.

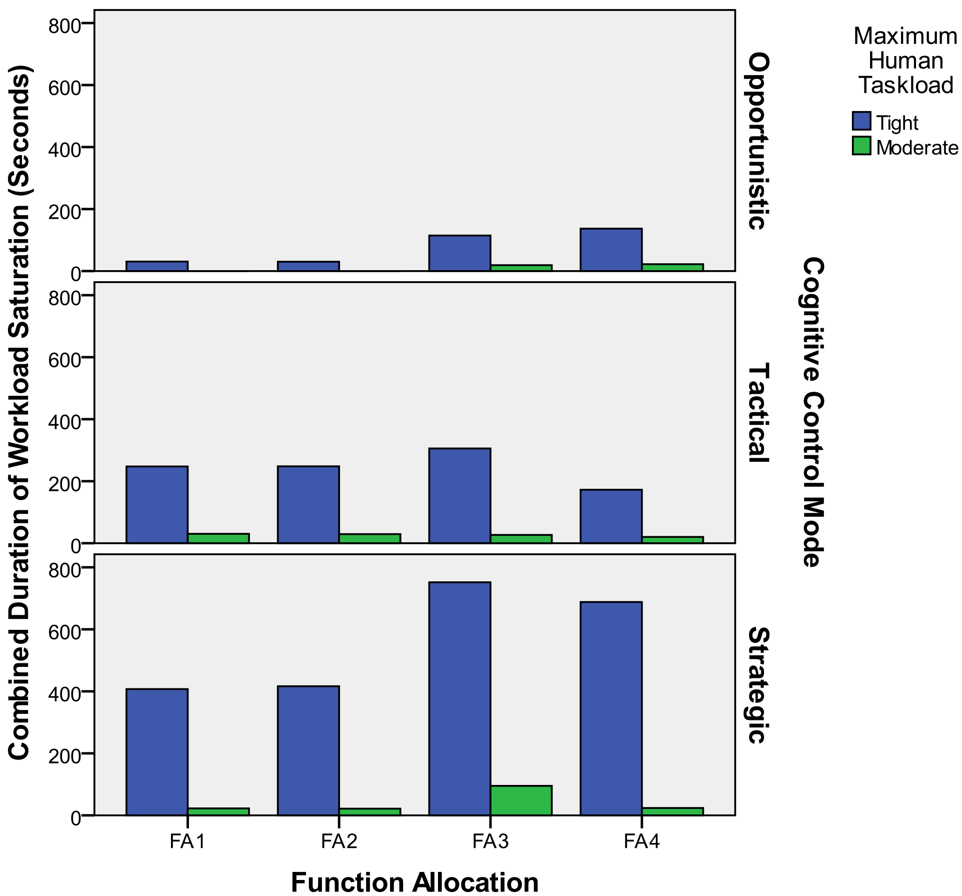

Figure 28. Combined duration of workload saturation per simulated flight by function allocation, cognitive control mode, and maximum human taskload averaged across all scenarios

Figure 28 illustrates the average integrated duration of workload saturation that the pilot experienced throughout each flight. As expected, the conditions with the highest taskload in Figure 26 and Figure 27 (i.e., FA3 with the strategic mode) show also the most workload saturation. Of the maximum human taskload limits tested here, the "moderate" limit created significantly less workload saturation than the "tight" limit. (Note the "unlimited" limit did not cause any workload saturation by design.)

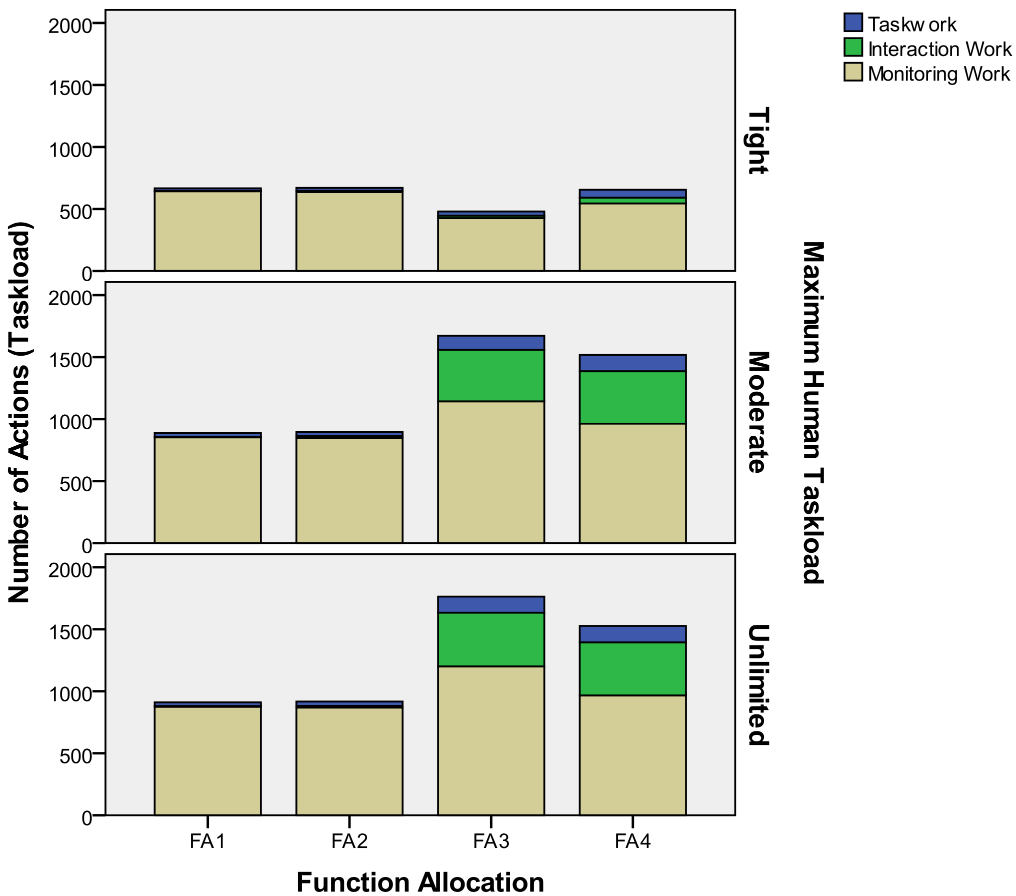

Figure 29. Number of actions (taskload) per simulated flight by function allocation and maximum human taskload averaged across all scenarios with the pilot in the strategic cognitive control mode

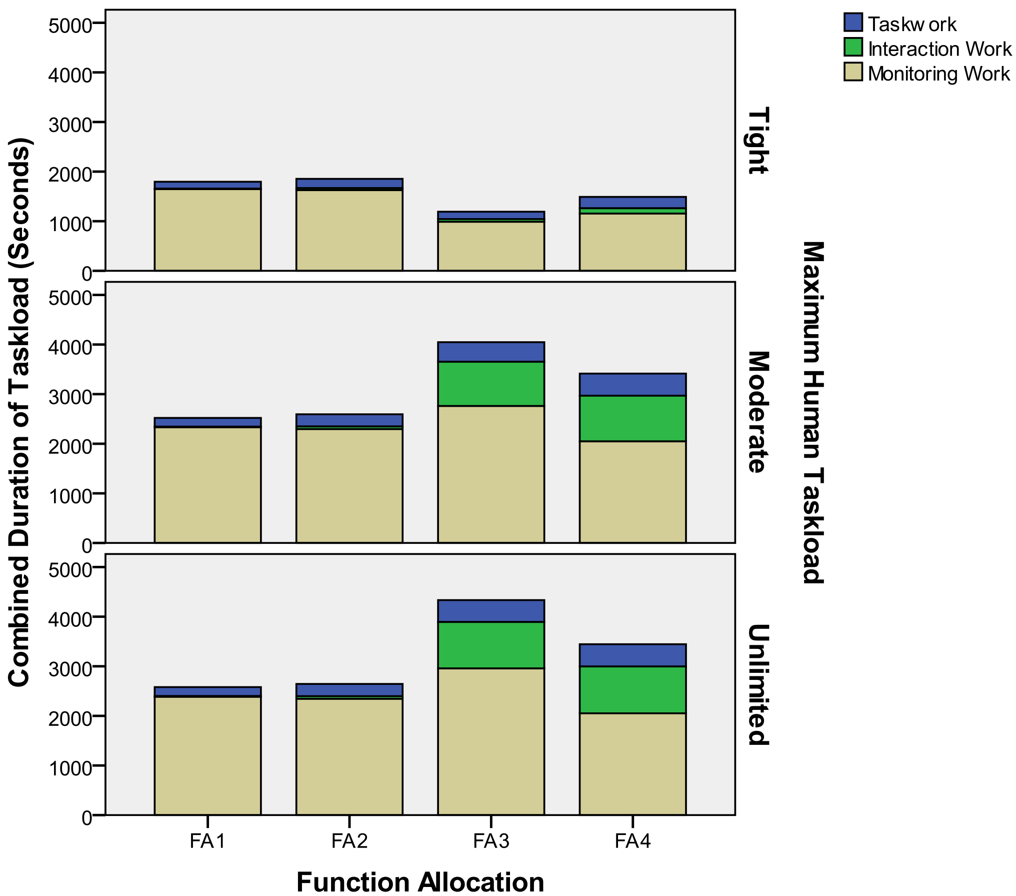

Figure 30. Combined duration of taskload per simulated flight by function allocation and maximum human taskload averaged across all scenarios with the pilot in the strategic cognitive control mode

Figure 29 and Figure 30 illustrate the effect of limiting the maximum human taskload capacity. In this model, actions were prioritized such that, when the maximum human taskload limit was reached, lower priority actions were delayed or interrupted. In most cases monitoring actions were given a lower priority than interaction actions and taskwork actions. Thus, the maximum human taskload capacity limits, once reached, reduced monitoring in all function allocations, especially with the highly-automated function allocations. With the mixed function allocation (FA3) and the mostly-manual function allocation (FA4), the "tight" maximum human taskload capacity limit also impacted some of the interaction and taskwork actions.

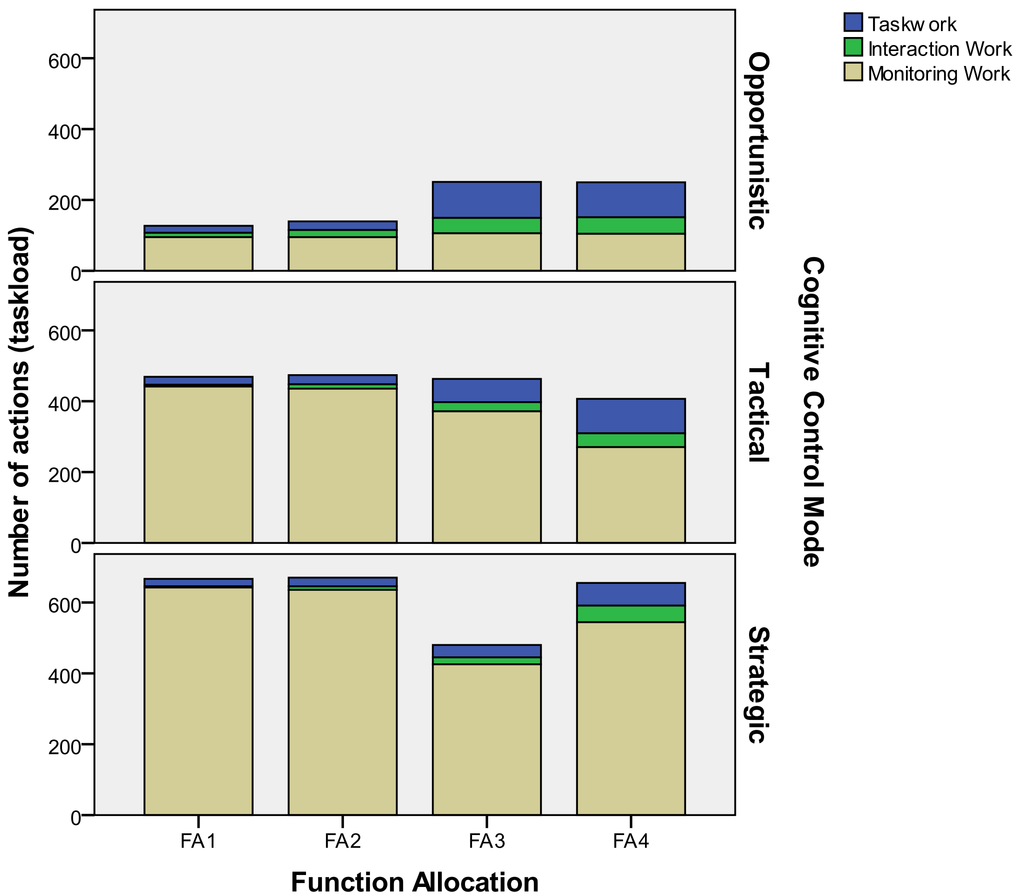

Figure 31. Number of actions (taskload) per simulated flight by function allocation and cognitive control mode with the "Tight" maximum human taskload

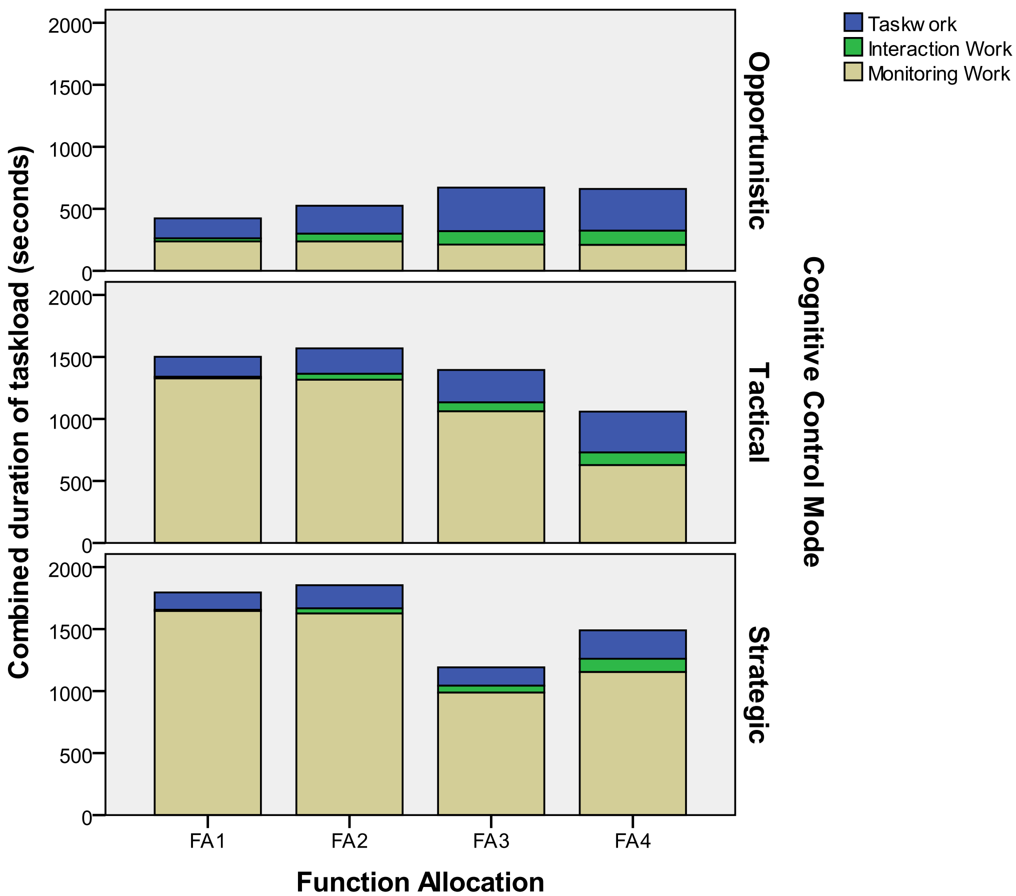

Figure 32. Combined duration of taskload per simulated flight by function allocation and cognitive control mode with the "Tight" level of maximum human taskload

Figure 31 and Figure 32 also show the impact of maximum human taskload limit, here focusing on results with the "tight" limit as a function of cognitive control mode and function allocation. A notable characteristic of the representation of cognitive control modes in this work model is that each spans the same taskwork and interactions but varies the monitoring actions the pilot will execute (and their frequency). Thus, the monitoring actions that are assumed to characterize strategic behavior are the first to be delayed or interrupted. Interestingly, with the tactical cognitive control mode, the total taskload decreased as the function allocation became more manual (FA3 and FA4), indicating that the pilot dropped more monitoring tasks than in the more highly-automated function allocations (FA1 and FA2).

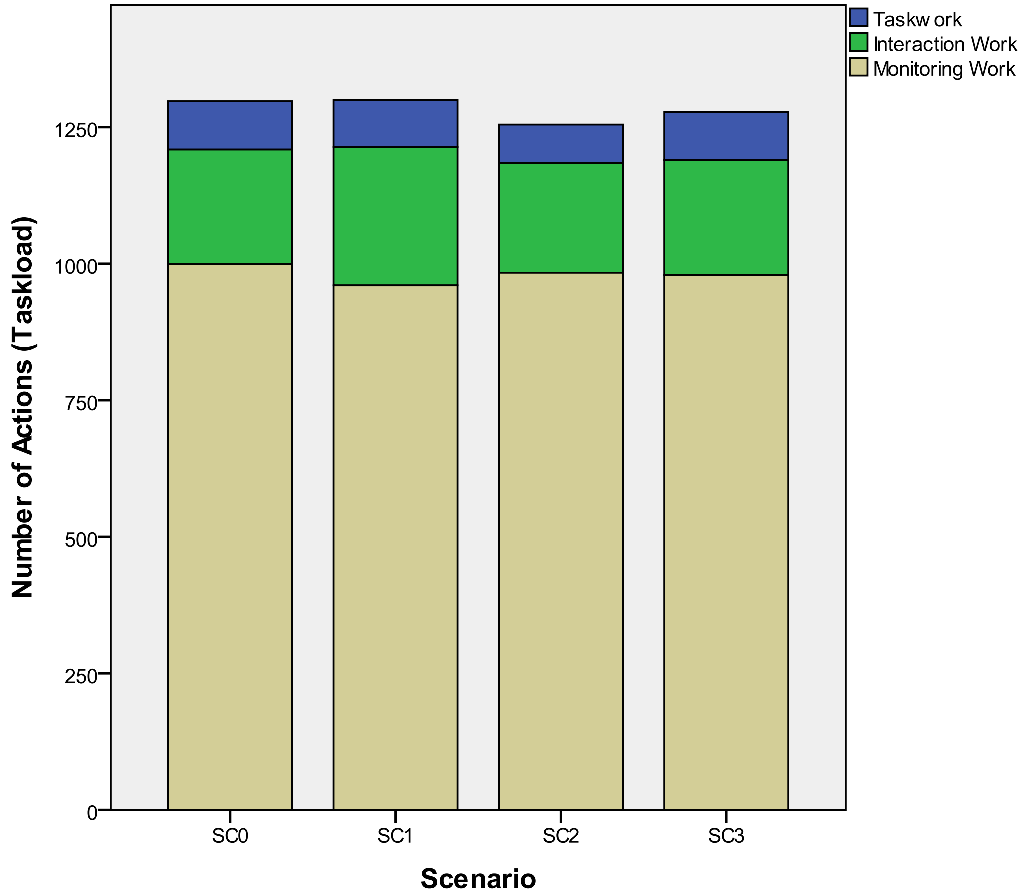

Figure 33. Number of actions (taskload) per simulated flight by scenario averaged across all function allocations, cognitive control modes, and levels of maximum human taskload

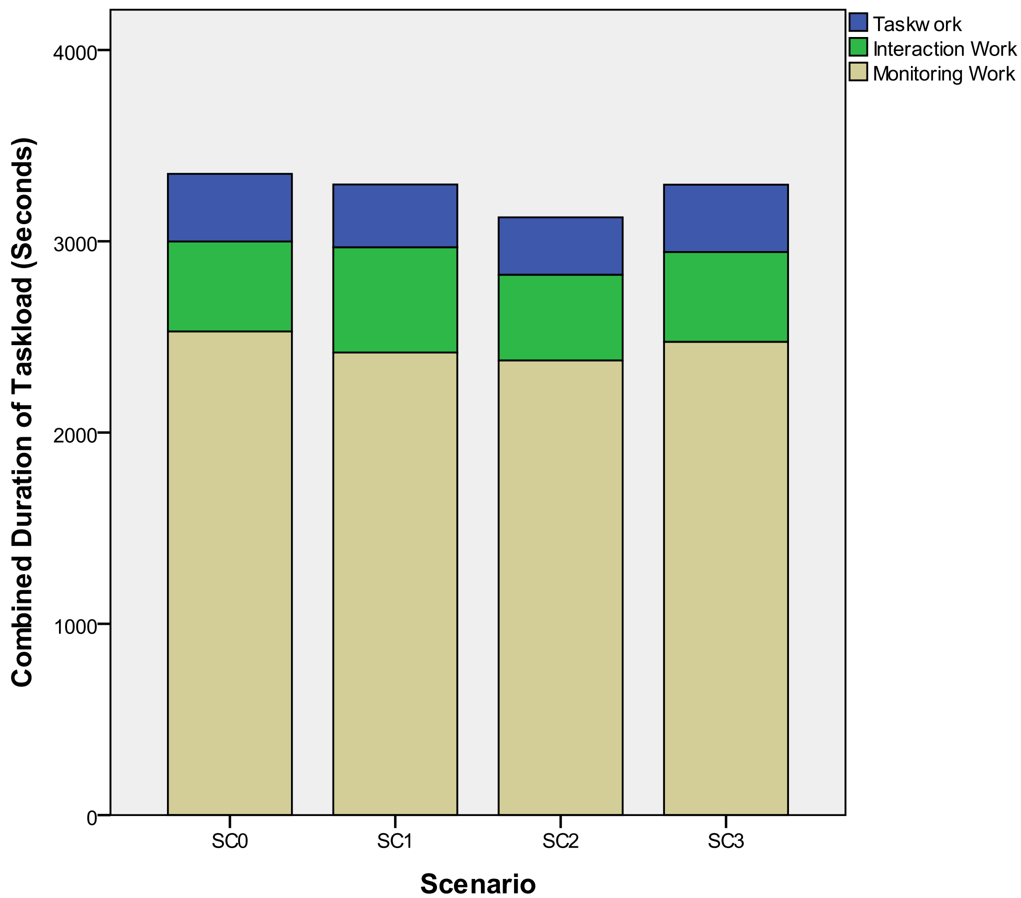

Figure 34. Combined duration of taskload per simulated flight by scenario averaged across all function allocations, cognitive control modes, and levels of maximum human taskload

Finally, Figure 33 and Figure 34 displays the taskload experienced within each scenario. Of note, the unstable work environment scenario (SC2) required additional reprogramming of the FMS with FA1, FA2, and FA3, but the corresponding increase in the interaction component of taskload is small and is offset by this scenario flying through fewer waypoints and, thus, having fewer taskwork and monitoring actions associated with responding to waypoint passage. Overall, the off-nominal scenarios did not cause higher taskload; note, however, all possible pilot responses to their off-nominal events were not extensively described in the work model.

5.3.2 Coherency of a Function Allocation

The coherency of a function allocation is measured as a level for each agent and a percentage within the work model. Specifically, the coherency level for each agent is measured as the level from the bottom of the static work model at which all the functions allocated to one agent can be described. Second, the coherency percentage is measured as the percentage of functions in the static work model entirely assigned to any agent with respect to the total number of functions required to describe the team's work.

As discussed in Section 4.2, one should note that these measures will depend on the structure of the abstraction hierarchy used in the work model. However, as long as the model is based on work-relevant means-end relationships, the relative values of this metric allows for comparison between function allocations for obvious effects that break-up an agent's work in a manner that cannot be sensibly abstracted.

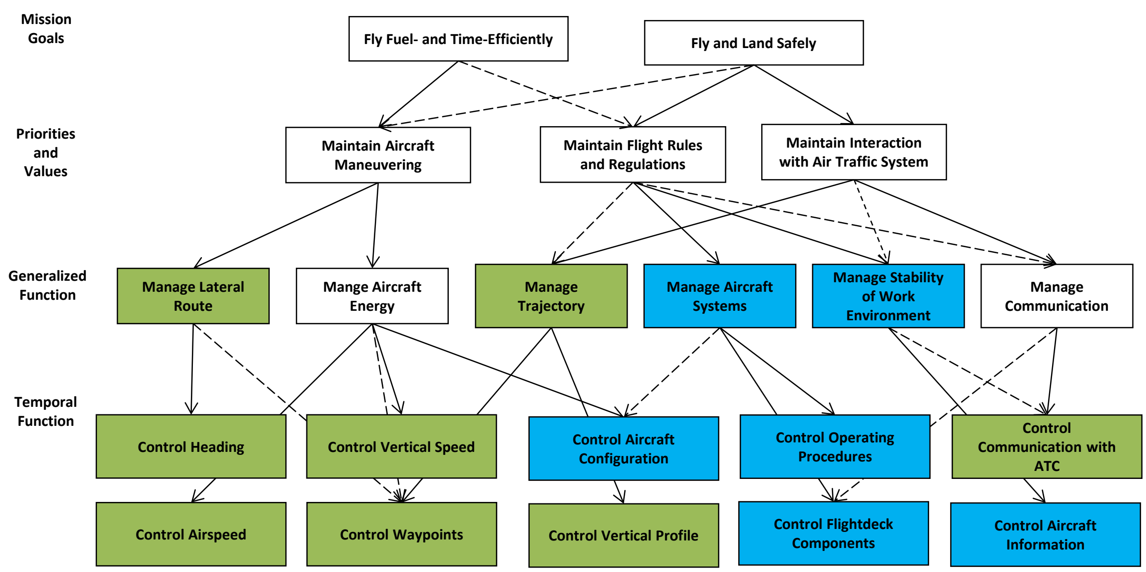

Figure 35. Highly-automated function allocation (FA1, functions entirely allocated to the automation are green-coded and to the pilot are blue-coded)

Figure 35 illustrates the coherency assessment of the highly-automated function allocation (FA1). In this function allocation, almost all flight path management functions are assigned to the flight deck automation. However, still the pilot is responsible for flight safety and required to monitor and supervise the automation. Because the vertical profile requires particular monitoring, the "Manage Aircraft Energy" generalized function cannot be described as being entirely assigned to the automation; likewise, the pilot is also expected to confirm air traffic instructions associated within the "Manage Communication" generalized function. All other generalized functions (and all temporal functions) are entirely assigned either to the pilot or to the flight deck automation. Therefore, from the bottom of the work model, only at the second (generalized function) level are any functions (and their lower-level component functions) assigned entirely to one (any) agent. Thus, in this function allocation, the coherency level is measured as level 2. The coherency percentage is measured as follows: 14 functions are entirely assigned to one agent (the pilot or automation) out of the 21 total functions required to describe the team's work; therefore, the coherency percentage is computed as 67% for this function allocation.

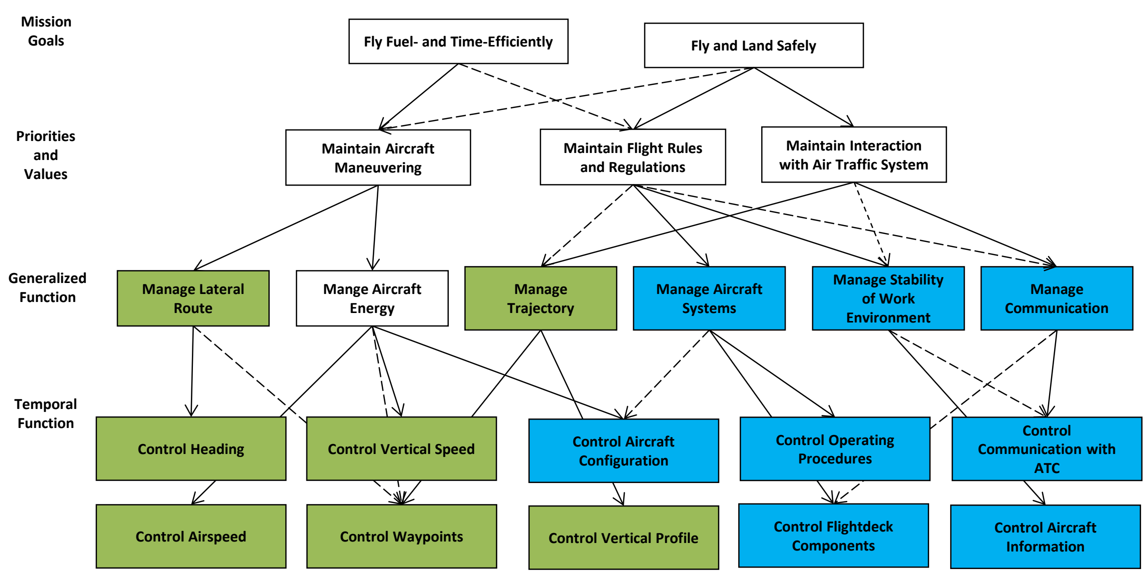

Figure 36. Mostly-automated function allocation (FA2, functions entirely allocated to the automation are green-coded and to the pilot are blue-coded)

Figure 36 illustrates the mostly-automated function allocation (FA2). In this function allocation, the pilot receives air traffic instructions, which corresponds to the generalized function "Manage Communication" and its constituent temporal function "Control Communication with ATC" now being assigned exclusively to the pilot. Therefore, the coherency is "increased" in a way that more generalized functions are entirely assigned to one agent. Thus, in this function allocation, the coherency level is measured as still level 2, but the coherency percentage is increased: 15 functions are entirely assigned to one agent (the pilot or automation) out of the 21 total functions; therefore, the coherency percentage is computed as 71% for this function allocation.

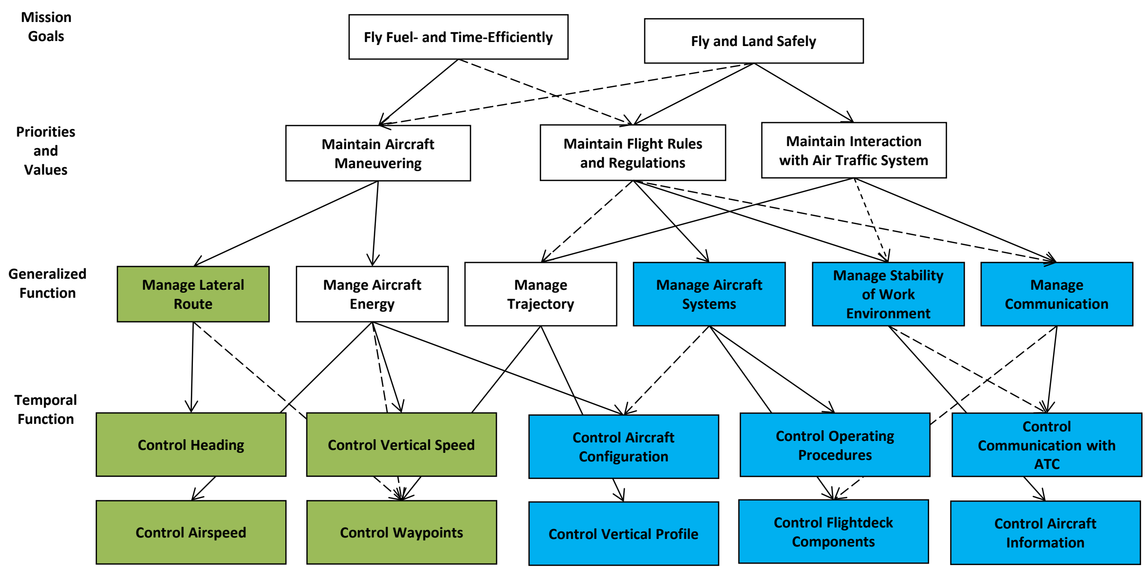

Figure 37. Mixed function allocation (FA3, functions entirely allocated to the automation are green-coded and to the pilot are blue-coded)

Figure 37 illustrates the mixed function allocation (FA3). With this function allocation, management of the flight path is distributed between the pilot and the flight deck automation. Therefore, the coherency is "decreased:" while the coherency level is still level 2, the coherency percentage is decreased back 67% given that 14 functions that are entirely assigned to one agent out of the 21 functions in the work model.

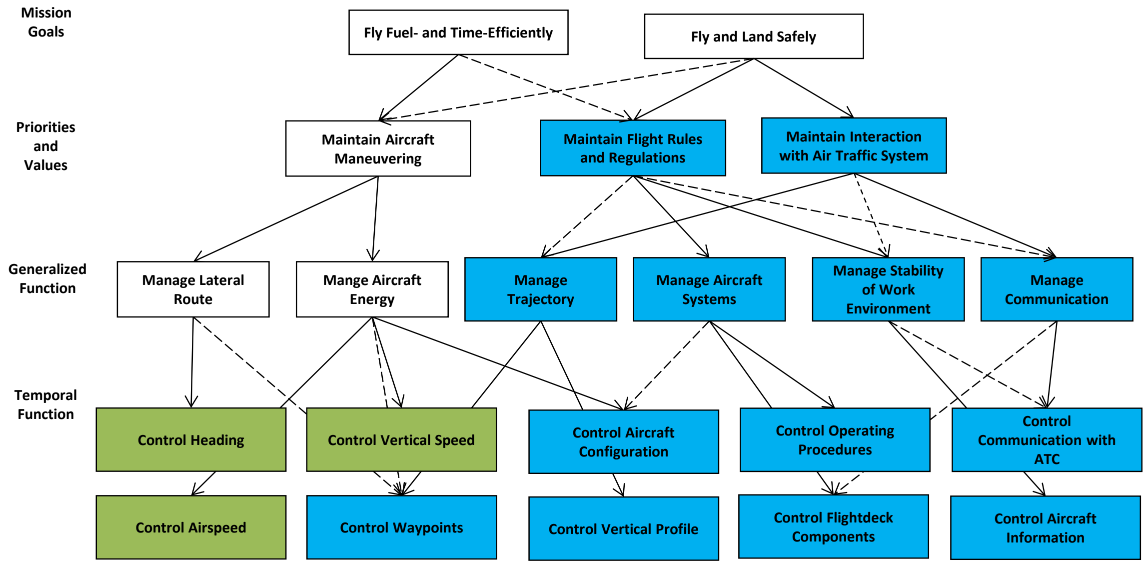

Figure 38. Mostly-manual function allocation (FA4, functions entirely allocated to the automation are green-coded and to the pilot are blue-coded)

Figure 38 illustrates the mostly-manual function allocation (FA4). In this function allocation, except for “Manage Lateral Route” and “Manage Aircraft Energy,” all generalized functions are assigned to the pilot. Thus, the pilot is exclusively assigned to the priorities and values function “Maintain Flight Rules and Regulations” and “Maintain Interaction with Air Traffic Systems” which are three levels from the bottom. Thus, the pilot’s coherency level is measured as level 3. The coherency percentage is increased as well: 16 functions are entirely assigned to one agent out of the total 21 functions, corresponding to a coherency percentage of 76%.

5.3.3 Mismatches between Responsibility and Authority

Mismatches between responsibility and authority are measured in two ways: static and dynamic. The static measure can be assessed from the work model as the number of temporal functions assigned to the automation for which the responsibility remains with the pilot. The dynamic measure can be assessed from simulated flights as the number of teamwork actions induced by a mismatch and their combined duration.

In general, as the function allocations become more “manual,” the number of mismatched functions decreases because more functions are assigned to the pilot. If we assume that the autoflight system is certified for the basic tasks of controlling heading, airspeed, and vertical speed, then it may be considered as having both responsibility and authority for the temporal functions “Control Heading,” “Control Airspeed,” and “Control Vertical Speed.” Because these are the only temporal functions assigned to the automation in FA4, its static measure of mismatch between responsibility and authority in its temporal functions is zero. However, as more temporal functions are allocated to the automation in the other function allocations, the mismatch measure grows: “1” in FA3, “2” in FA2, and “3” in FA1. Table 22 through Table 25 detail which temporal functions are mismatched in terms of responsibility and authority (the basis for the static measure) as well as teamwork actions induced by the mismatch (which are counted in simulations as the dynamic measure).

Table 22. Assignment of responsibility and authority within the highly-automated function allocation (FA1, red-coded functions and actions indicate mismatched functions and induced monitoring actions)

Temporal Function	Pilot	Automation
Automation Has Responsibility and Authority.		
Control Heading	Monitor Heading Trends	Update Lateral Control
Control Vertical Speed	Monitor Altitude Monitor Vertical Deviation	Adjust Speed Control Update Pitch Control Evaluate Vertical Mode Evaluate VNAV Mode Transition Evaluate Alt Restriction Mode Altitude Reminder
Control Airspeed	Monitor Descent Airspeed	Update Thrust Control Calculate Speed Deviation
Automation Has Authority. Human Has Responsibility.		
Control Vertical Profile	Modify CDU Pages Reduce Airspeed for Late Descent Confirm Target Altitude Confirm Target Speed	Manage Waypoint Progress
Control Waypoints	Modify CDU Pages Monitor Waypoint Progress Monitor Dist Active Waypoint Confirm Active Waypoint	Calculate Dist Current Waypoint Evaluate Flight Phase Manage Waypoint Progress Direct To Waypoint
Control Communication With ATC	Respond to Hand Off Confirm Data Communication	Receive Altitude Clearance Receive ILS Clearance Receive Waypoint Clearance
Human Has Responsibility and Authority.		
Control Aircraft Configuration	Deploy Flap Deploy Gear Deploy Speed Brake Retract Speed Brake Confirm Configuration Change	
Control Aircraft Information	Verify TOD Location Verify Crossing Restriction	
Control Operating Procedures	Perform Approach Briefing Perform Approach Checklist Perform Landing Checklist	
Control Flight Deck Components	Turn Off Altitude Alert Respond to Drag Required	

Table 23. Assignment of responsibility and authority within the mostly-automated function allocation (FA2, red-coded functions and actions indicate mismatched functions and induced monitoring actions)

Temporal Function	Pilot	Automation
Automation Has Responsibility and Authority.		
Control Heading	Monitor Heading Trends	Update Lateral Control
Control Vertical Speed	Monitor Altitude Monitor Vertical Deviation	Adjust Speed Control Update Pitch Control Evaluate Vertical Mode Evaluate VNAV Mode Transition Evaluate Alt Restriction Mode Altitude Reminder
Control Airspeed	Monitor Descent Airspeed	Update Thrust Control Calculate Speed Deviation
Automation Has Authority. Human Has Responsibility.		
Control Vertical Profile	Modify CDU Pages Reduce Airspeed for Late Descent Confirm Target Altitude Confirm Target Speed	Manage Waypoint Progress
Control Waypoints	Modify CDU Pages Monitor Waypoint Progress Monitor Dist Active Waypoint Confirm Active Waypoint	Calculate Dist Current Waypoint Evaluate Flight Phase Manage Waypoint Progress Direct To Waypoint
Human Has Responsibility and Authority.		
Control Communication With ATC	Receive Altitude Clearance Receive ILS Clearance Receive Waypoint Clearance Respond to HandOff Request Clearance	
Control Aircraft Information	Verify TOD Location Verify Crossing Restriction	
Control Operating Procedures	Perform Approach Briefing Perform Approach Checklist Perform Landing Checklist	
Control Flight Deck Components	Turn Off Altitude Alert Respond to Drag Required	

Table 24. Assignment of responsibility and authority within the mixed function allocation (FA3, red-coded functions and actions indicate mismatched functions and induced monitoring actions)

Temporal Function	Pilot	Automation
Automation Has Responsibility and Authority.		
Control Heading	Monitor Heading Trends	Update Lateral Control
Control Vertical Speed	Dial Altitude Selector Dial VS Selector Push Alt Hold Switch Push FLCH Switch Push Vertical NAV Switch Push Vertical Speed Switch Monitor Green Arc	Update Pitch Control Evaluate Vertical Mode Evaluate Alt Restriction Mode Altitude Reminder
Control Airspeed	Dial Speed Selector Push Speed Switch Monitor Descent Airspeed	Update Thrust Control Calculate Speed Deviation
Automation Has Authority. Human Has Responsibility.		
Control Waypoints	Manage Waypoint Progress Monitor Waypoint Progress Monitor Dist Active Waypoint Confirm Waypoint Target	Calculate Dist Current Waypoint Evaluate Flight Phase Direct To Waypoint
Human Has Responsibility and Authority.		
Control Vertical Profile	Monitor Altitude Reduce Airspeed for Late Descent	
Control Communication With ATC	Receive Altitude Clearance Receive ILS Clearance Receive Waypoint Clearance Respond to Handoff Request Clearance	
Control Aircraft Configuration	Deploy Flap Deploy Gear Deploy Speed Brake Retract Speed Brake Confirm Configuration Change	
Control Aircraft Information	Verify TOD Location Verify Crossing Restriction	
Control Operating Procedures	Perform Approach Briefing Perform Approach Checklist Perform Landing Checklist	
Control Flight Deck Components	Turn Off Altitude Alert Respond to Drag Required	

Table 25. Assignment of responsibility and authority within the mostly-manual function allocation (FA4, red-coded functions and actions indicate mismatched functions and induced monitoring actions)

Temporal Function	Pilot	Automation
Automation Has Responsibility and Authority.		
Control Heading	Dial Heading Selector Push Heading Selector Monitor Heading Trends	Update Lateral Control
Control Vertical Speed	Dial Altitude Selector Dial VS Selector Push Alt Hold Switch Push FLCH Switch Push Vertical NAV Switch Push Vertical Speed Switch Monitor Green Arc	Update Pitch Control Evaluate Vertical Mode Evaluate Alt Restriction Mode Altitude Reminder
Control Airspeed	Dial Speed Selector Push Speed Switch Monitor Descent Airspeed	Update Thrust Control Calculate Speed Deviation
Human Has Responsibility and Authority.		
Control Vertical Profile	Monitor Altitude Reduce Airspeed for Late Descent	
Control Waypoints	Manage Waypoint Progress Direct To Waypoint	Calculate Dist Current Waypoint Evaluate Flight Phase
Control Communication With ATC	Receive Altitude Clearance Receive ILS Clearance Receive Waypoint Clearance Respond Handoff Request Clearance	
Control Aircraft Configuration	Deploy Flap Deploy Gear Deploy Speed Brake Retract Speed Brake Confirm Configuration Change	
Control Aircraft Information	Verify TOD Location Verify Crossing Restriction	
Control Operating Procedures	Perform Approach Briefing Perform Approach Checklist Perform Landing Checklist	
Control Flight Deck Components	Turn Off Altitude Alert Respond to Drag Required	

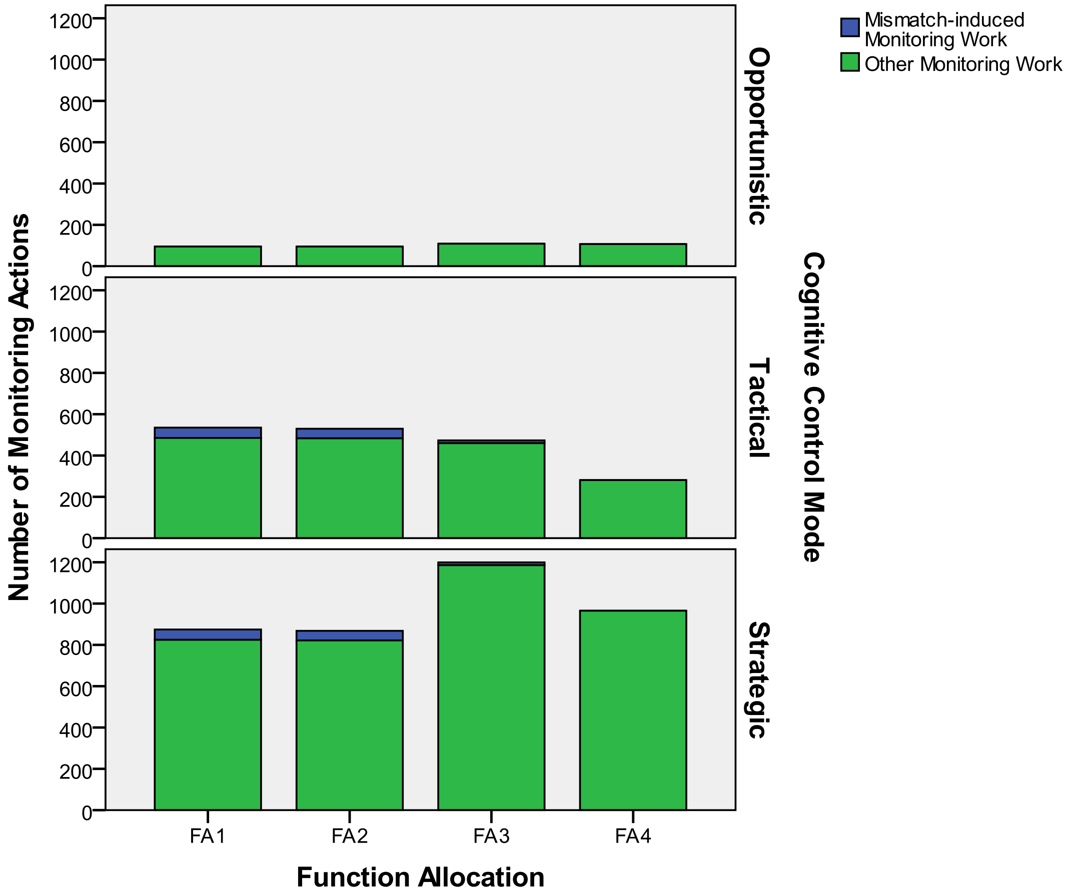

Figure 39. Number of monitoring actions per simulated flight, distinguishing between mismatch-induced monitoring work and other monitoring work, by function allocation and cognitive control mode across all scenarios with the unlimited maximum human taskload

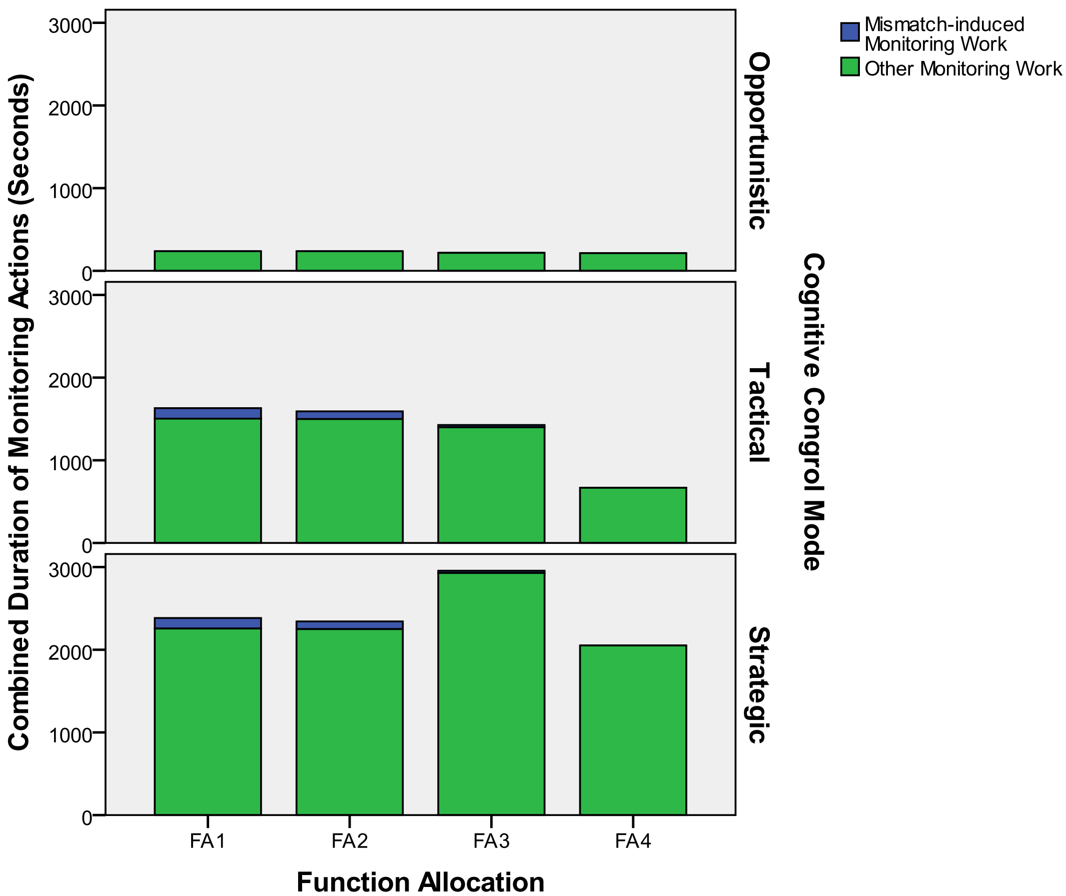

Figure 40. Combined duration of monitoring actions per simulated flight, distinguishing between mismatch-induced monitoring work and other monitoring work, by function allocation and cognitive control mode across all scenarios with the unlimited maximum human taskload

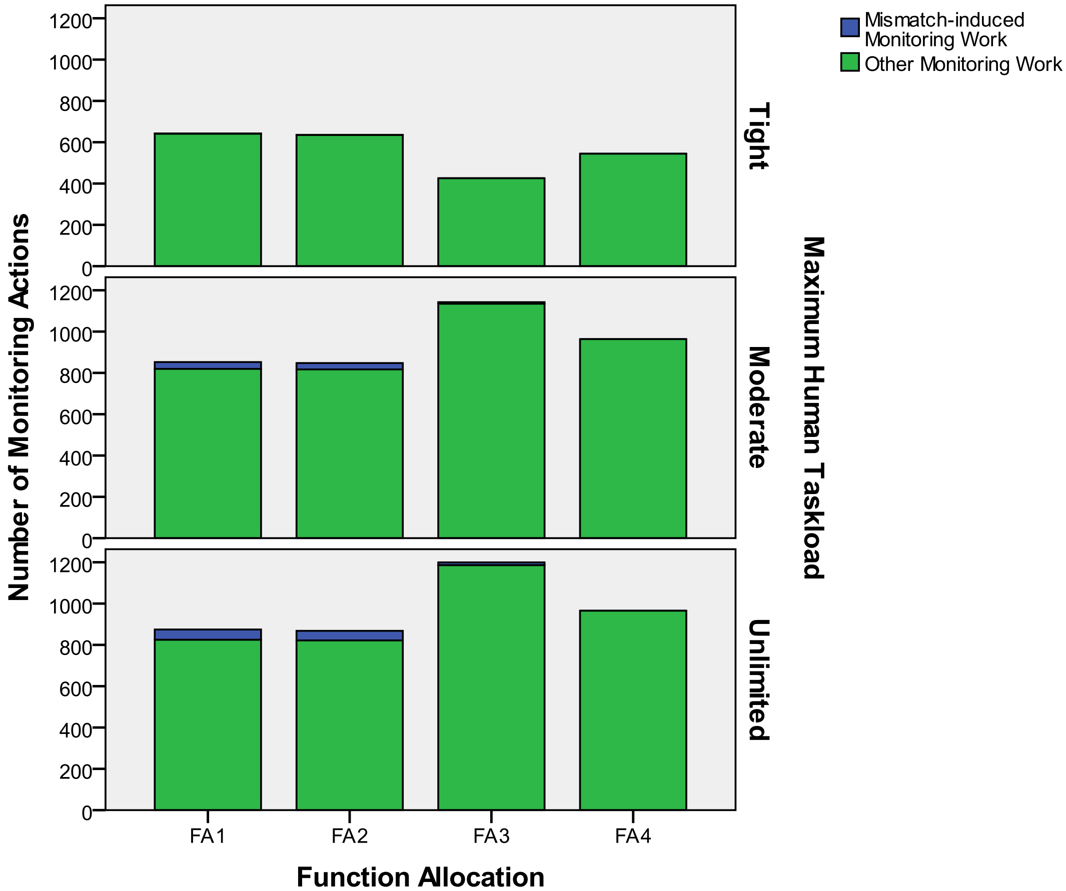

Figure 41. Number of monitoring actions per simulated flight, distinguishing between mismatch-induced monitoring work and other monitoring work, by function allocation and maximum human taskload across all scenarios with the "Strategic" cognitive control mode

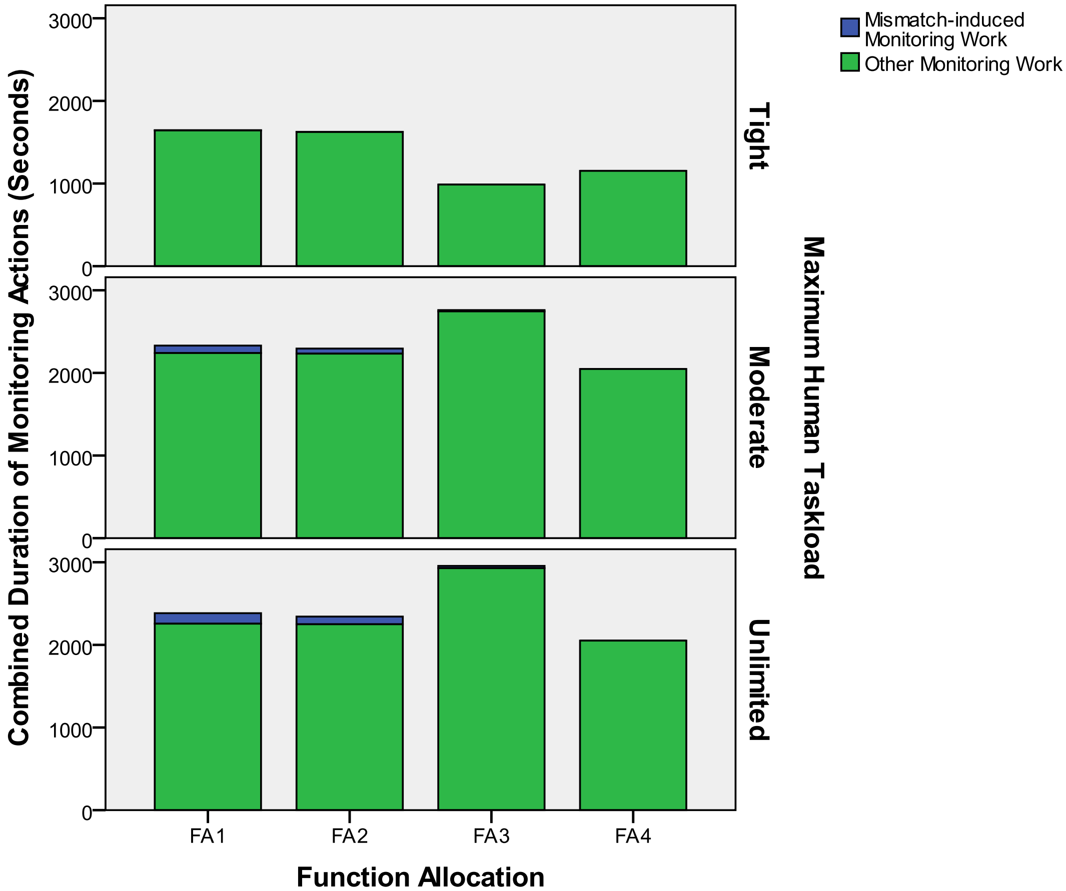

Figure 42. Combined duration of monitoring actions per simulated flight, distinguishing between mismatch-induced monitoring work and other monitoring work, by function allocation and maximum human taskload across all scenarios with the "Strategic" cognitive control mode

The simulation counted number of monitoring actions demanded of the pilot in general, due to mismatches between responsibility and authority in particular. Figure 39 and Figure 40 illustrate the number of monitoring actions demanded of the pilot by the work environment (i.e., with the "Unlimited" maximum human taskload). More mismatch-induced monitoring actions were demanded by the highly-automated function allocations (FA1 and FA2), which also have the higher static measure of mismatch. The monitoring actions demanded by the mismatch were the same in the tactical and strategic cognitive control modes, but were dropped in the opportunistic cognitive control mode. Figure 41 and Figure 42 show that these actions were also dropped when maximum human taskload limits were reached.

5.3.4 Interruptive Automation

Interruptive automation is measured dynamically as the average number of instances per simulated flight where the automation interrupts the pilot while he/she performs operating procedures. This model allowed for three types of interruptions: first, when the MCP altitude target is not lower than the cruise altitude after reaching 10nm before the T/D point, the automation displays "RESET MCP ALTITUDE" (this interruption was only triggered in the late descent scenario, SC1, and only with the more highly-automated function allocations, FA1 and FA2); second, the altitude alert which sounds off when the aircraft reaches within 1000ft of the MCP altitude target (this interruption is therefore given once per every entry of a new MCP altitude target and thus is reflects how often the scenario requires altitude changes); and, third, when the airspeed is 10knots higher than the planned descent airspeed, the automation displays "DRAG REQUIRED" to the pilot (this interruption is therefore a reflection of the speed tracking established by the function allocation).

The results are shown in Table 26, Figure 43, Figure 44, and Figure 45. Function allocation impacts this measure: the pilot is interrupted by automation during roughly one operation procedure per simulated flight in the more manual function allocations (FA3 and FA4), but only roughly one operating procedure is interrupted per five simulated flights with the more automated function allocations (FA1 and FA2). Outside of the first and second types of interruptions generated by the scenarios, the difference in interruptions between function allocations appears to be caused by the third interruption type noted above, which reflects the speed tracking used with each function allocation.

Table 26. Mean and standard deviation of instances of interruption by function allocation averaged across all scenarios, cognitive control modes, and levels of maximum human taskload

Function Allocation	Number of Cases	Mean of Instances of Interruption	Standard Deviation of Instances of Interruption
FA1	90	0.18	0.04
FA2	90	0.22	0.04
FA3	90	0.99	0.09
FA4	90	1.03	0.09

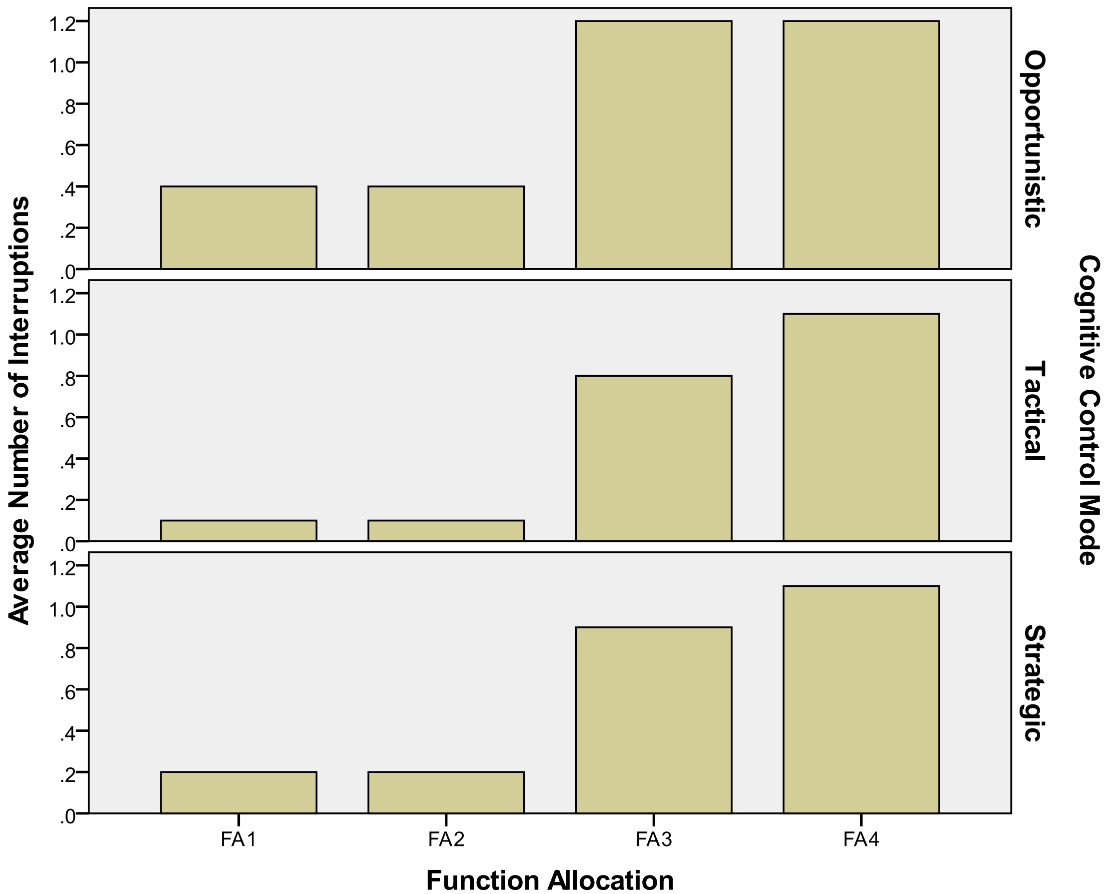

Figure 43. Average number of interruptions by the flight deck automation per simulated flight by function allocation and cognitive control mode across all scenarios with the "Unlimited" maximum human taskload

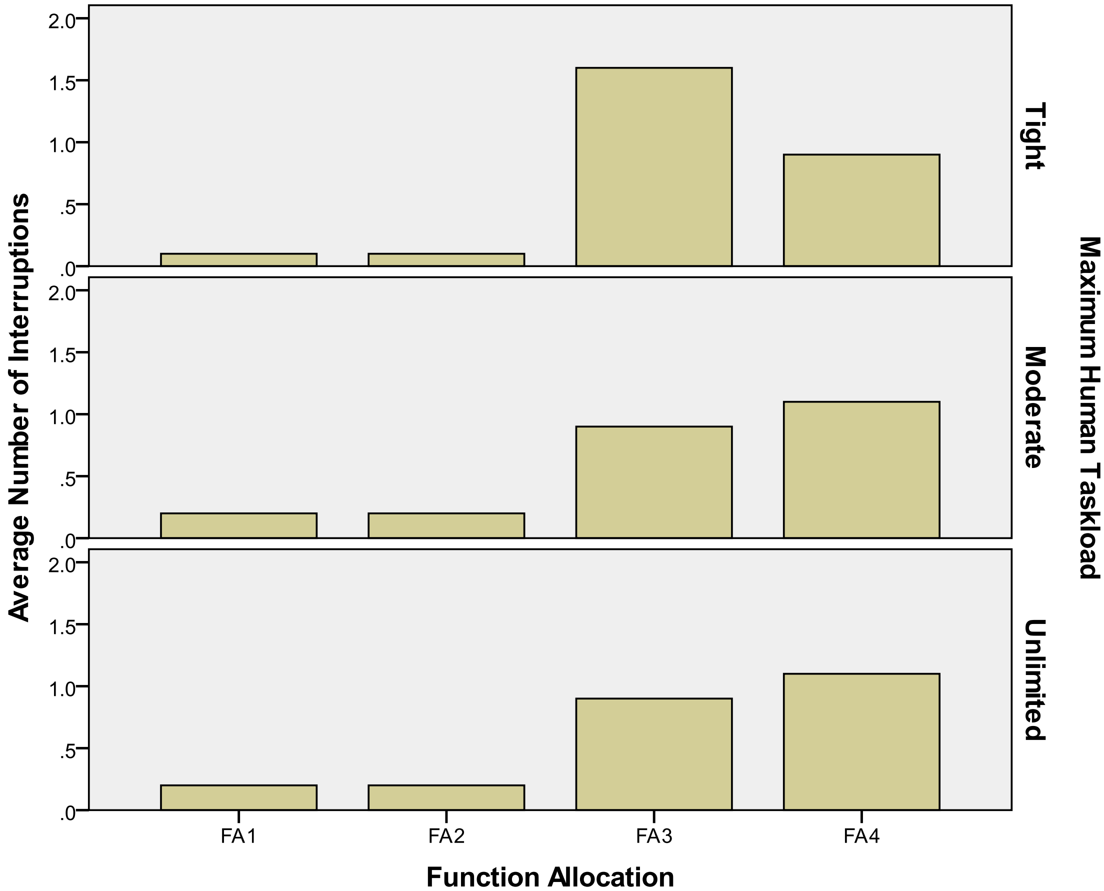

Figure 44. Average number of interruptions by the flight deck automation per simulated flight by function allocation and maximum human taskload across all scenarios with the "Strategic" cognitive control mode

Figure 44 shows slight decrease as the maximum human taskload limits become greater. This is because pilot actions for performing operation procedures have lower priority than other taskwork, including responding to the interruptions from the automation. Thus, with limited maximum human taskload, the operating procedures were often halted and resumed later (i.e., the operating procedures were often "dragged out"), increasing the likelihood that the pilot was performing the operating procedures when the interruptions occurred.

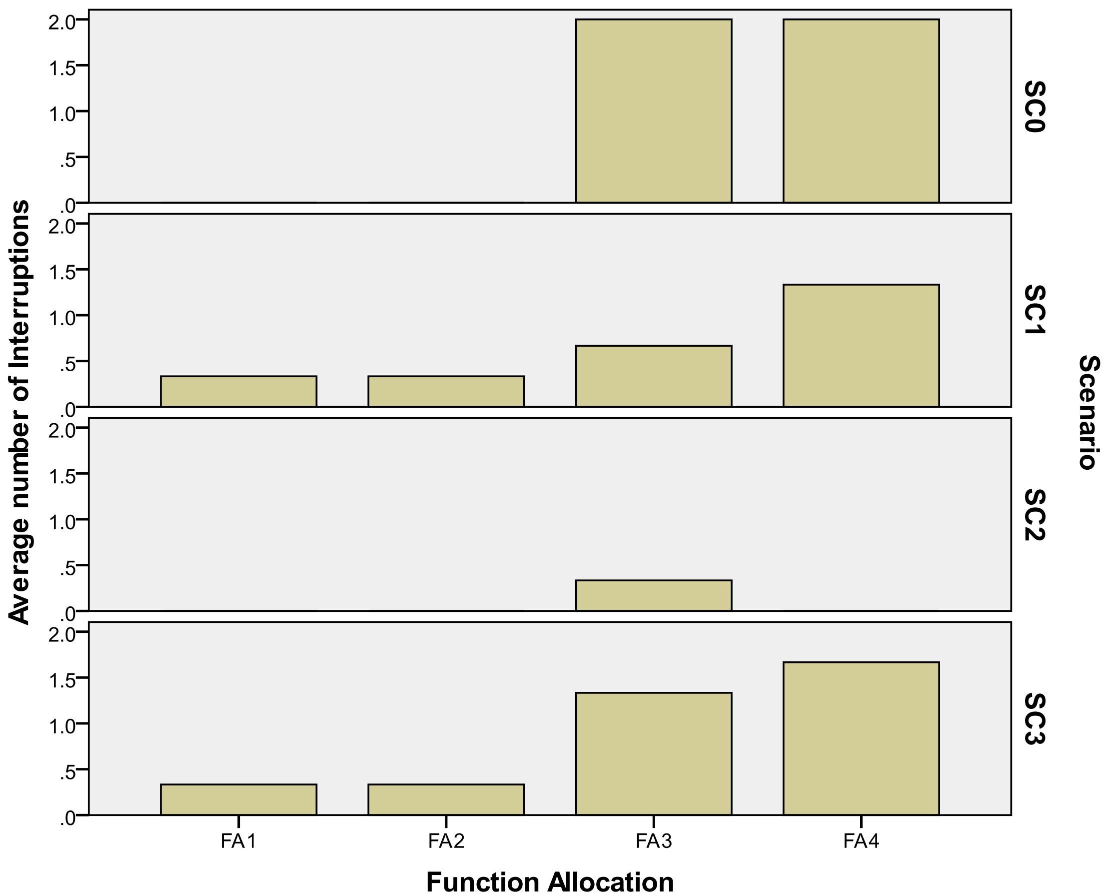

Figure 45. Average number of interruptions by the flight deck automation per simulated flight by function allocation and scenario with the "Strategic" cognitive control mode and the "Unlimited" maximum human taskload

The function allocation effects are consistent across the different cognitive control modes. Note that the model implemented the operating procedures to be performed with the same timings and duration across the cognitive control modes. Also, SC0, SC1, and SC3 show more interruptions than SC2. This is due to the nature of the interruptions due to the altitude alert. Thus, the scenarios requiring passage through more waypoints, each with associated changes in altitude, tended to create more triggers for pilot interruptions by the automation.

5.3.5 Automation Boundary Conditions

Three aspects of automation boundary conditions were measured here: duration of actual speed deviations 15knots higher/10knots lower than the commanded speed, duration of vertical deviations from the planned vertical profile greater than 400ft above/below, and duration of the vertical speed required to meet an air traffic restriction being higher than the maximum vertical speed that the aircraft can physically achieve (with FA3 and FA4) or than the maximum vertical speed programmed in the FMS (with FA1 and FA2).

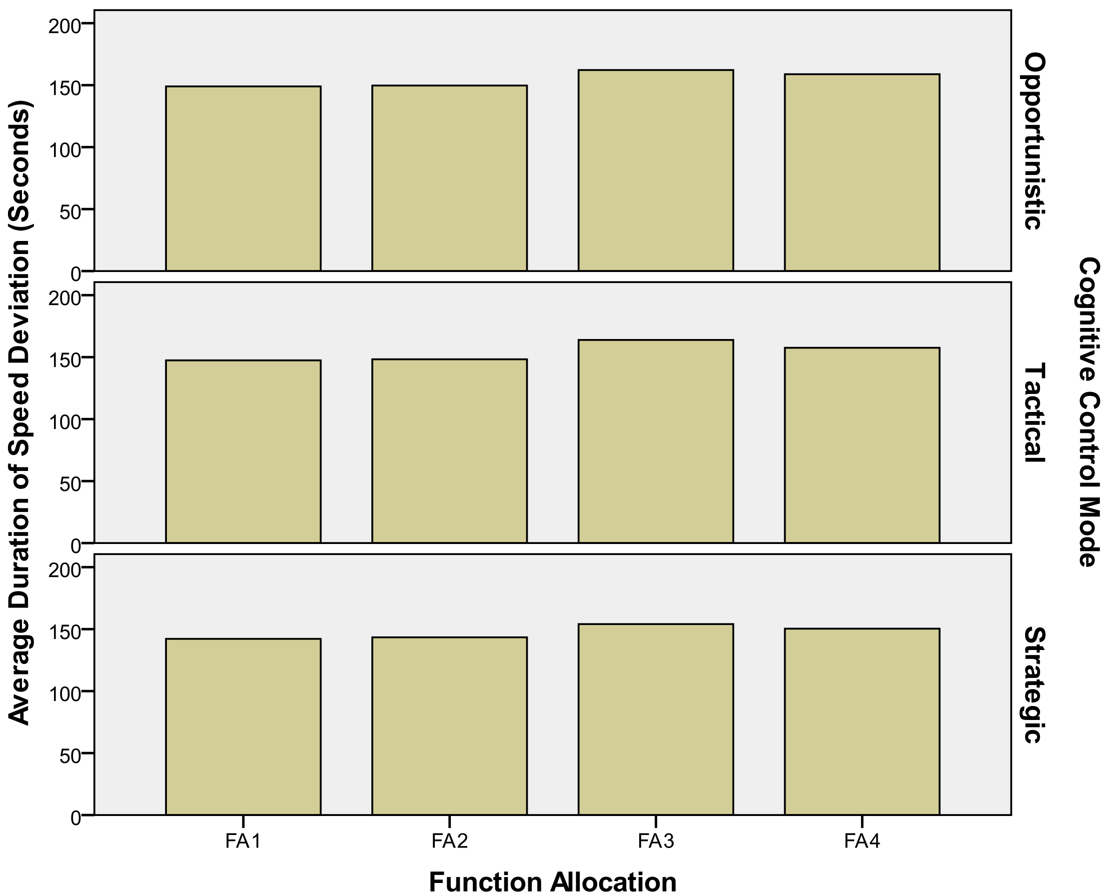

Figure 46. Average duration of speed deviation from the commanded speed by function allocation and cognitive control mode across all scenarios with the "Unlimited" maximum human taskload

Figure 46 illustrates the average duration of speed deviations per simulated flight by function allocation and cognitive control mode. A one-way ANOVA found that the measure varies significantly across function allocation ($p<0.0005$). (The homogeneity of variance assumptions was violated, but the robust test of equality means showed that there were significant differences between means across function allocation.) Post-hoc comparisons using the Tukey HSD test indicated that the average integrated duration of speed deviation for the more highly-automated function allocations, FA1 (M=146.37, SD=6.88) and FA2 (M=147.30, SD=7.24) was significantly lower than for the more manual function allocations, FA3 (M=161.07, SD=12.26) and FA4 (M=157.23, SD=13.32).

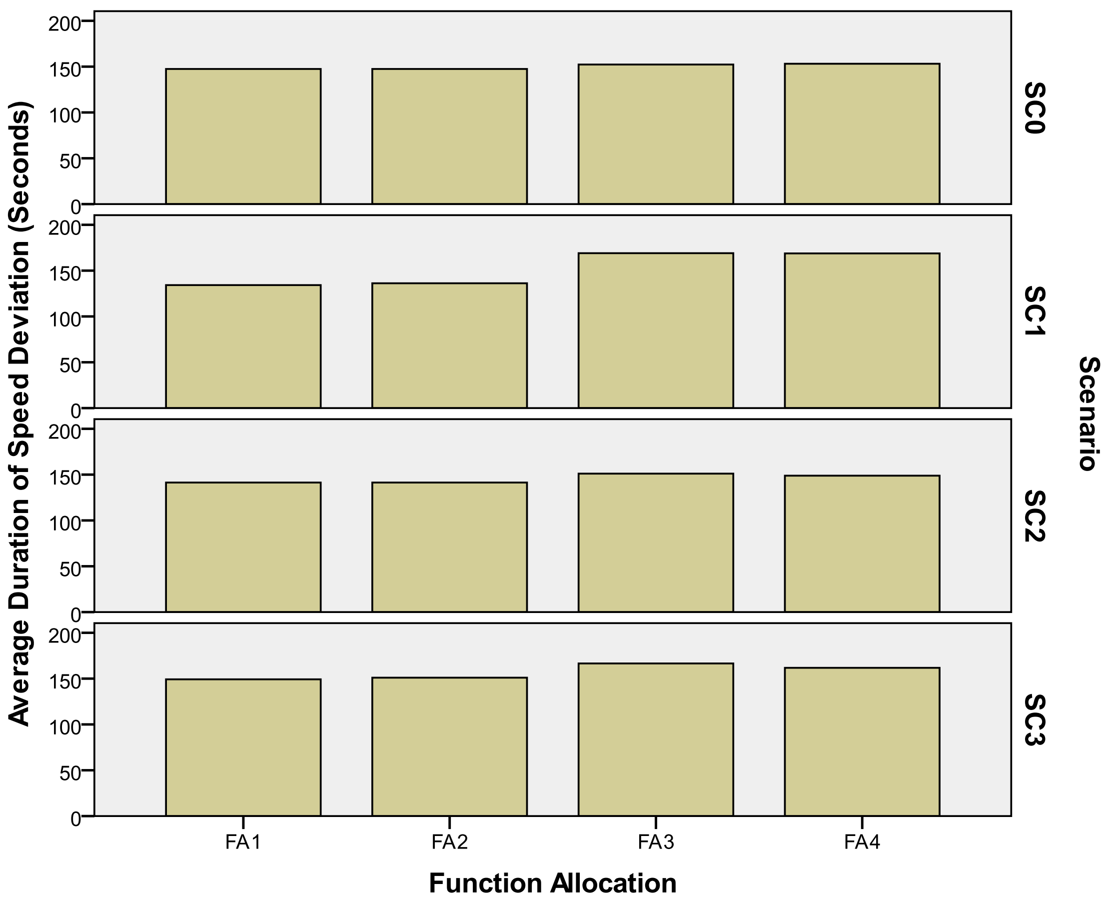

Figure 47. Average duration of speed deviation from the commanded speed by scenario and function allocation in the “Strategic” cognitive control mode and the “Unlimited” maximum human taskload

Figure 47 illustrates the duration where the actual speed deviated 10knots higher/15knots lower from the commanded speed per simulation flight by scenario across all other independent variables. Note that this deviation was recorded not only when the speed drifted from the actual value, but also when it started correctly tracking a newly-entered speed target, and thus was non-zero even in the nominal scenario SC0. The pattern shows that SC0 and SC2 experienced shorter durations than SC1 and SC3. A one-way ANOVA found that the measure varies significantly across scenarios ($p<0.0005$). (The homogeneity of variance assumption is violated, but the robust test of equality means showed that there are significant differences between means across scenario.) Post-hoc comparisons using the Tukey HSD test indicated that the average duration of speed deviations in SC2 (M=146.16, SD=9.18) was significantly lower than in SC1 (M=158.47, SD=14.34) and in SC3 (M=156.28, SD=8.26).

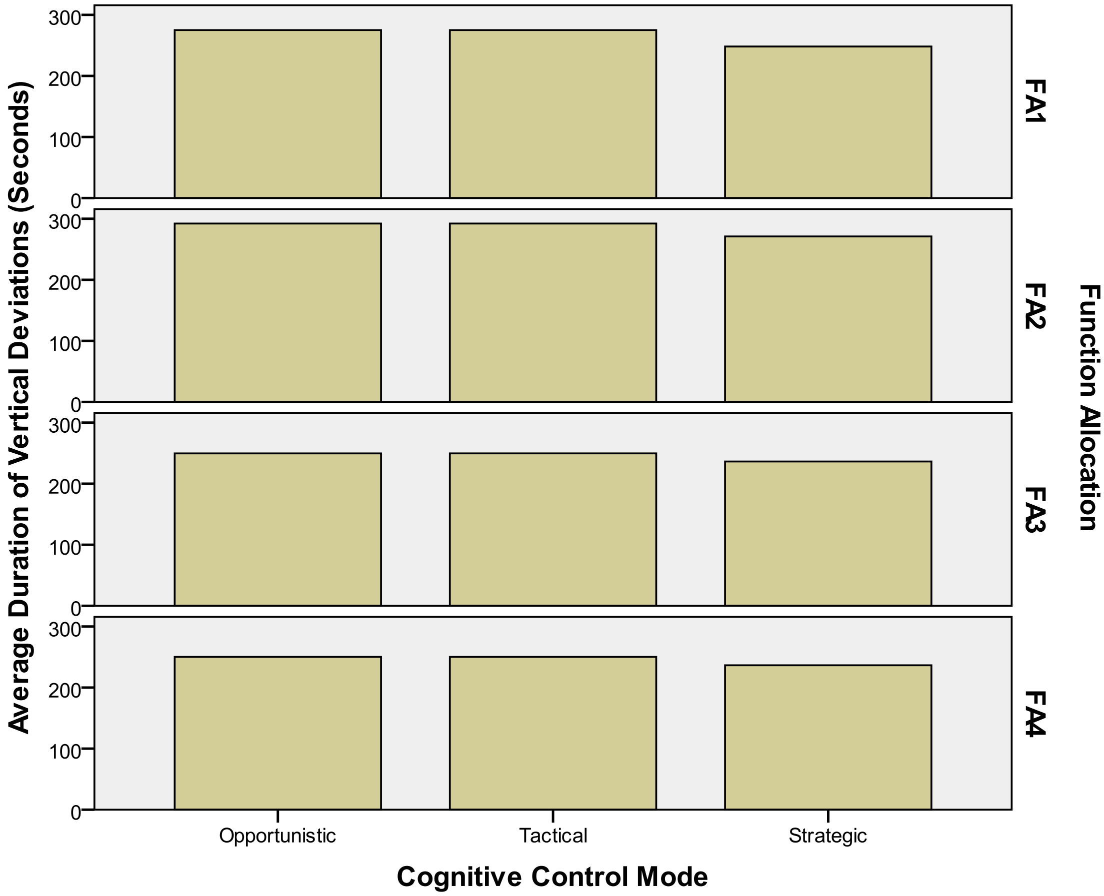

Figure 48. Average duration of deviations from the vertical profile more than 400 ft. by function allocation and cognitive control mode across three scenarios with "Unlimited" maximum human taskload (SC2 cases excluded because its re-route nullified the optimal planned vertical profile)

Figure 48 illustrates the duration of vertical deviations. Statistical analysis found that this measure varied with neither function allocation nor cognitive control mode (Kruskal-Wallis Test and p=0.730 and One-way ANOVA, p=0.093, respectively used).

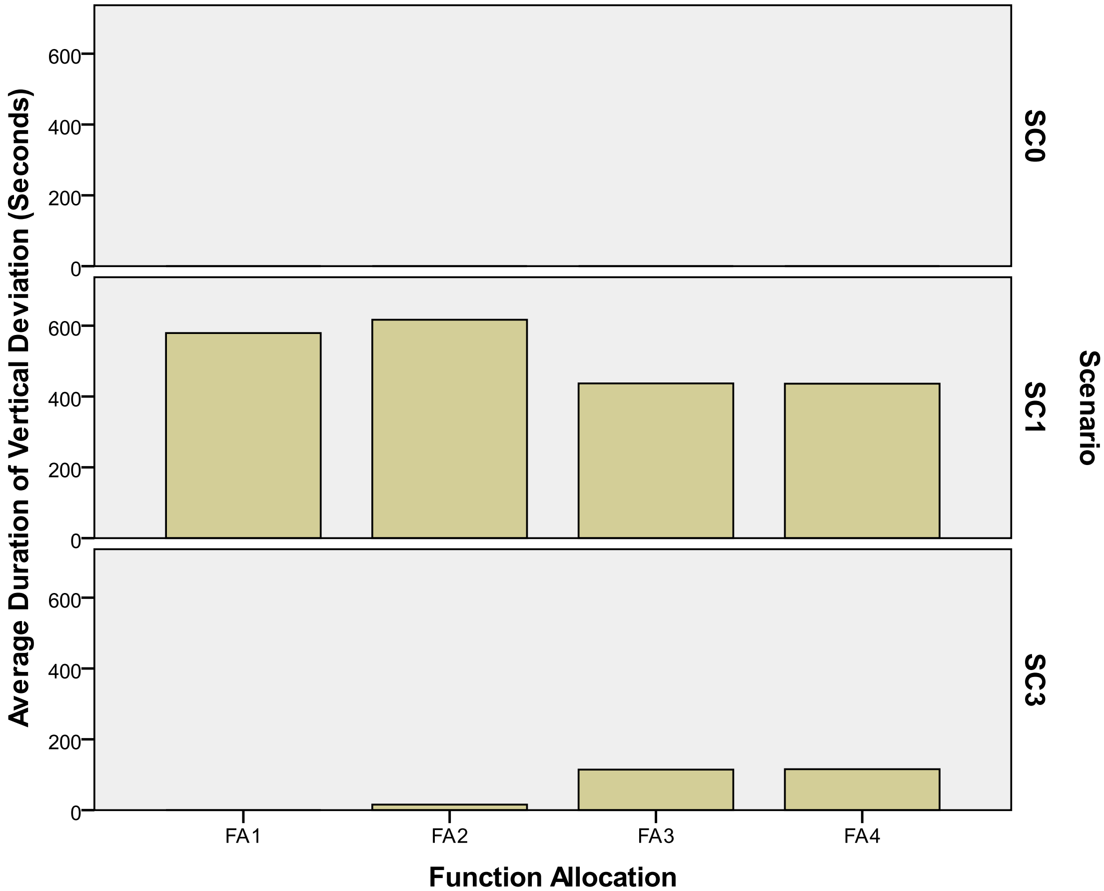

Figure 49. Average duration of deviations from the vertical profile more than 400 ft. by function allocation and scenario with the "Strategic" cognitive control mode and the "Unlimited" maximum human taskload (SC2 cases excluded from the analysis because the air traffic instruction to direct to another waypoint nullify the optimal planned vertical profile)

Figure 49 illustrates the duration of vertical deviations by function allocation and scenario. The nominal scenario (SC0) did not show any vertical deviation, mirroring its circumstances as the nominal scenario without any disturbances in the environment. The unexpected re-route scenario (SC2), as mentioned before, was omitted from analysis of this measure. The effects of the other two scenarios interact with function allocation. In the late descent scenario (SC1), the more highly automated function allocations (FA1 and FA2) appear to have longer duration of vertical deviations than FA3 and FA4. A one-way ANOVA found that the measure varies significantly across function allocations ($p<0.0005$). Post-hoc comparisons using the Tukey HSD test indicated that the average integrated duration of vertical deviation for FA1 (M=620.87, SD=67.38) and FA2 (M=649.61, SD=58.70) were significantly higher than FA3 (M=435.32, SD=57.85) and FA4 (M=435.14, SD=58.61).

On the other hand, in the tailwind scenario (SC3), the more manual function allocations (FA3 and FA4) had longer durations of vertical deviations than the more highly automated function allocations (FA1 and FA2). Because the homogeneity of variance is violated and robust tests of equality means failed, a non-parametric test as used. The Kruskal-Wallis test indicates that the measure varies significantly across the function allocations ($p<0.0005$).

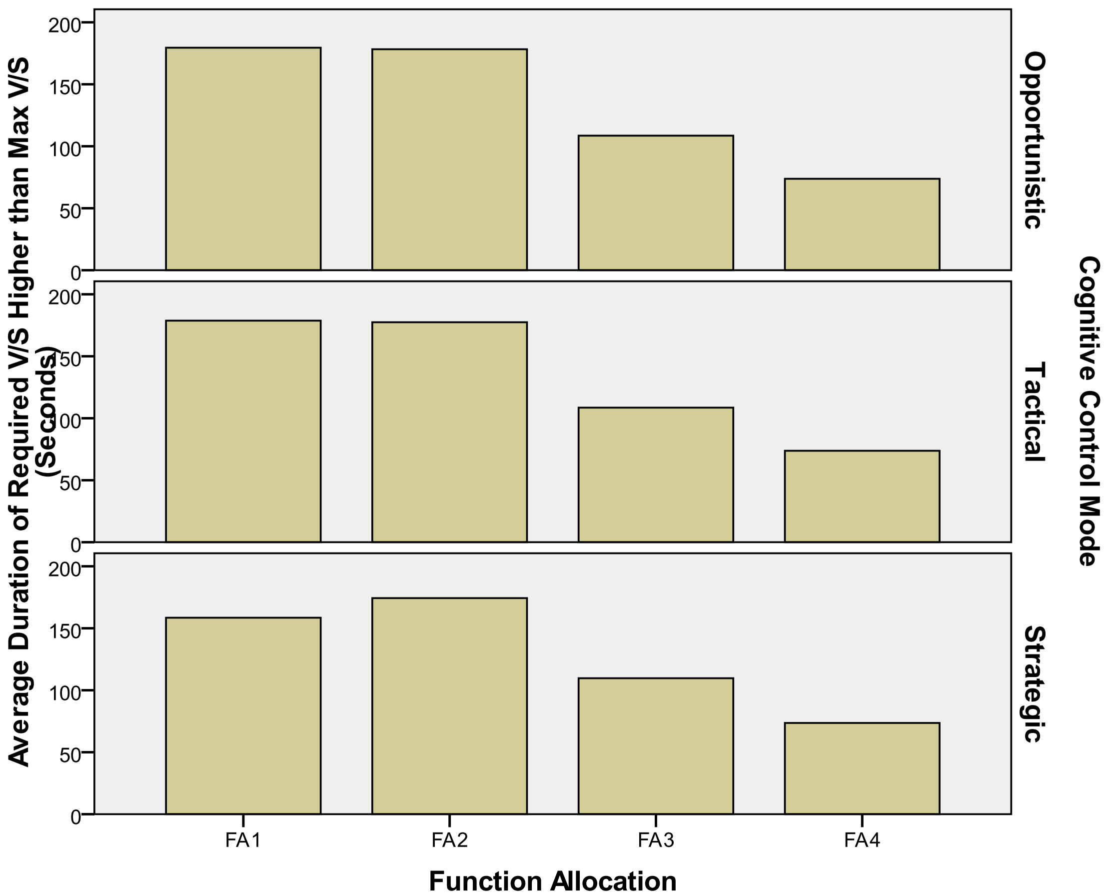

Figure 50. Average duration of required vertical speed higher than the maximum vertical speed of the aircraft or the descent rate preprogrammed in the FMS per simulated flight by function allocation and cognitive control modes averaged across all scenarios with "Unlimited" maximum human taskload

Figure 50 illustrates the average duration during which the required vertical speed was higher than the maximum vertical speed of the aircraft (for FA3 and FA4) or the one preprogrammed in the FMS (for FA1 and FA2). A one-way ANOVA found that the measure varies significantly across function allocation ($p<0.0005$). Post-hoc comparisons using the Tukey HSD test indicated that the average integrated duration of required vertical speed higher than the maximum vertical speed for the mostly-manual function allocation FA4 ($M=73.75$, $SD=105.32$) was significantly lower than the highly-automated function allocation FA1 ($M=172.14$, $SD=199.88$) and FA2 ($M=176.58$, $SD=100.01$), and that FA2 was significantly greater than FA3 ($M=108.80$, $SD=133.43$). One-way ANOVA did not find any significant variation between cognitive control modes ($P=0.814$).

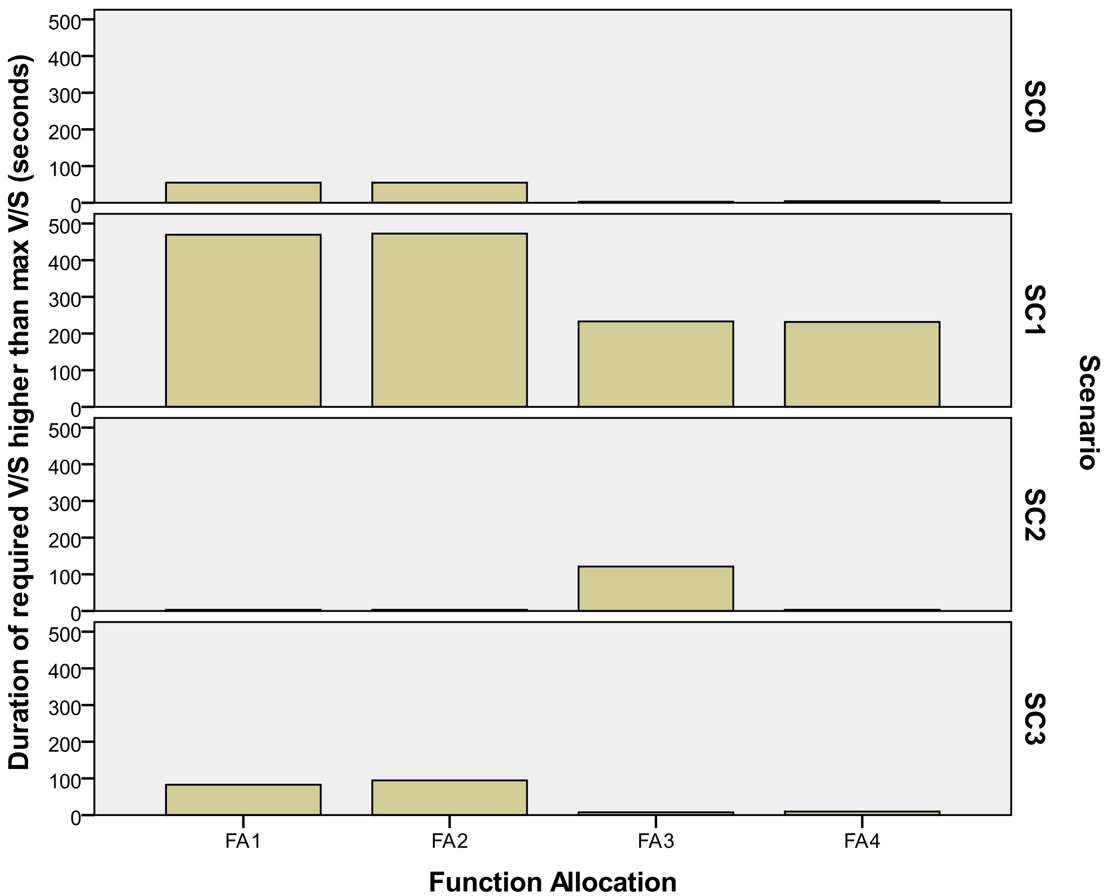

Figure 51. Average integrated duration of required vertical speed higher than the maximum vertical speed of the aircraft or the descent rate preprogrammed in the FMS per simulated flight by function allocation and scenario averaged across all cognitive control modes and maximum human taskload

Figure 51 illustrates the duration during which the required vertical speed was higher than the maximum vertical speed of the aircraft or the descent rate preprogrammed in the FMS per simulated flight by function allocation and scenario. Even the required vertical speed in the nominal scenario (SC0) was higher than the limited programmed in the FMS, which was a factor in FA1 and FA2. Also, the late descent scenario (SC1) requires the harshest performance from the automation, as expected. Thus, in SC1 the more manual function allocations (FA3 and FA4) shows less use of the automation outside its boundary conditions compared to the more highly-automated function allocation (FA1 and FA2). On the other hand, the tailwind scenario (SC3) shows the reverse pattern that FA1 and FA2 required the automation to be outside of its boundary conditions less. These observed patterns across SC1 and SC3 will also be discussed in detail in Section 5.3.7 as they resulted in effects in the performance measures of violations of air traffic restrictions.

5.3.6 Stability of the Human's Work Environment

Stability of the human's work environment is inferred from measures of the total number of actions demanded that are "not predicted" by the pilot. As discussed in Chapter 4 and Section 5.2.2, an action may be unpredictable in two ways. First, the most unpredictable (type1) action was not predicted at all (e.g., an unexpected re-route issued by air traffic controllers). Second, for some actions (type2), the pilot may have predicted they *could* or *might* occur, but he/she did not know exactly when; instead, the action is initiated by dynamics in the environment or actions by other agents' actions. Because each simulated flight required a different number of total actions (and thus a different number of unpredicted actions), the "unpredictability level" represents the percentage of actions that the pilot did not predict.

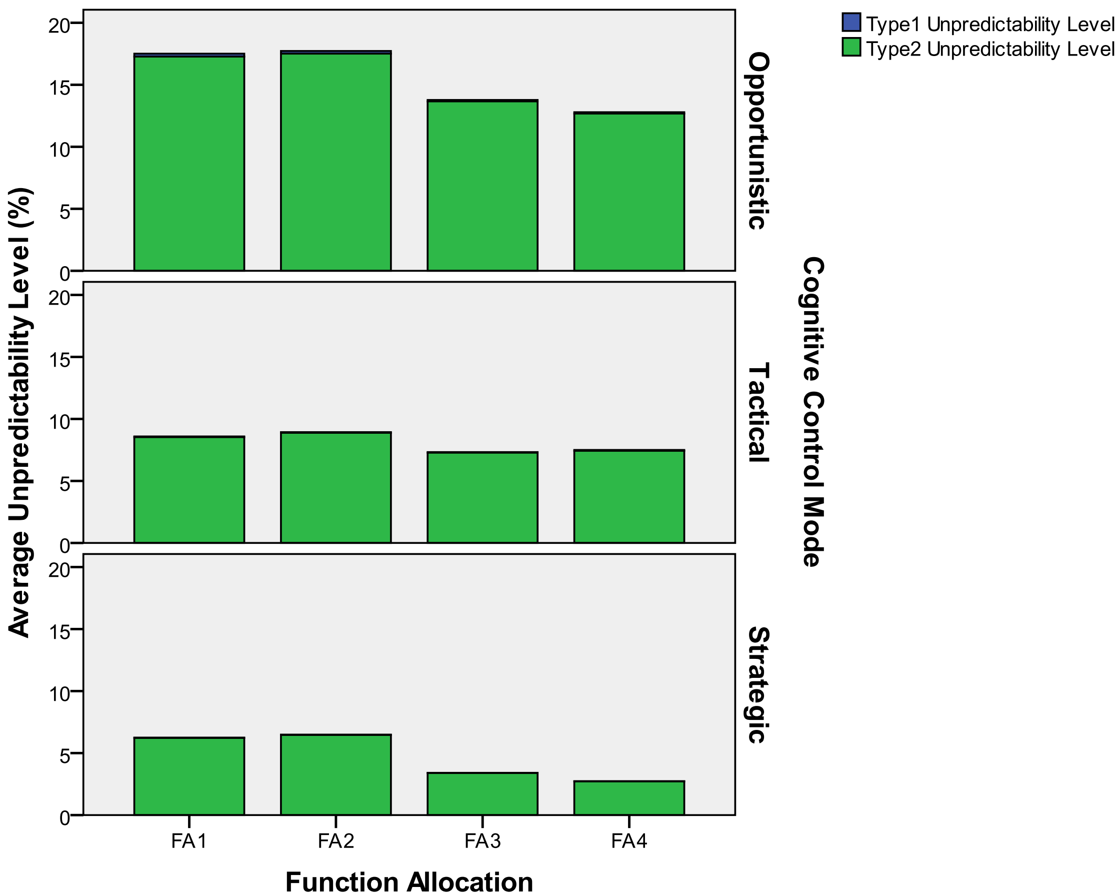

Figure 52. Average unpredictability level per simulated flight by function allocation and cognitive control mode across all scenarios with the "Unlimited" maximum human taskload

Figure 52 illustrates the unpredictability level, i.e., the count of unpredicted actions normalized by the total number of actions in its simulated flight. Examining the impact of function allocation, the more manual function allocations (FA3 and FA4) provide lower unpredictability levels compared to the more highly-automated function allocations (FA1 and FA2). In the study by Miller and Parasuraman (2007), the human's predictability of the environment increased with more functions allocated to them which is shown in this metric as the unpredictability level decreasing with the more manual function allocations (FA3 and FA4).

Another finding is that the pilot operating in the strategic cognitive control mode experienced a lower unpredictability level than the pilot operating in the tactical cognitive control mode, and the tactical cognitive control mode showed a lower unpredictability level than the opportunistic cognitive control mode. A one-way ANOVA found that the unpredictability level varied significantly with cognitive control mode ($p<0.0005$). Post-hoc comparisons using the Tukey HSD test indicated that the average unpredictability level for all three modes is significantly different from each other. This difference across the cognitive control modes may be because the unpredicted actions could be mitigated by the better management of flight route used in the tactical and, especially, strategic cognitive control mode. As seen in Section 5.3.1, as the taskwork was anticipated and thus performed at the best times, it did not need to be refined and re-executed later in response to events by the automation or in response to the environment.

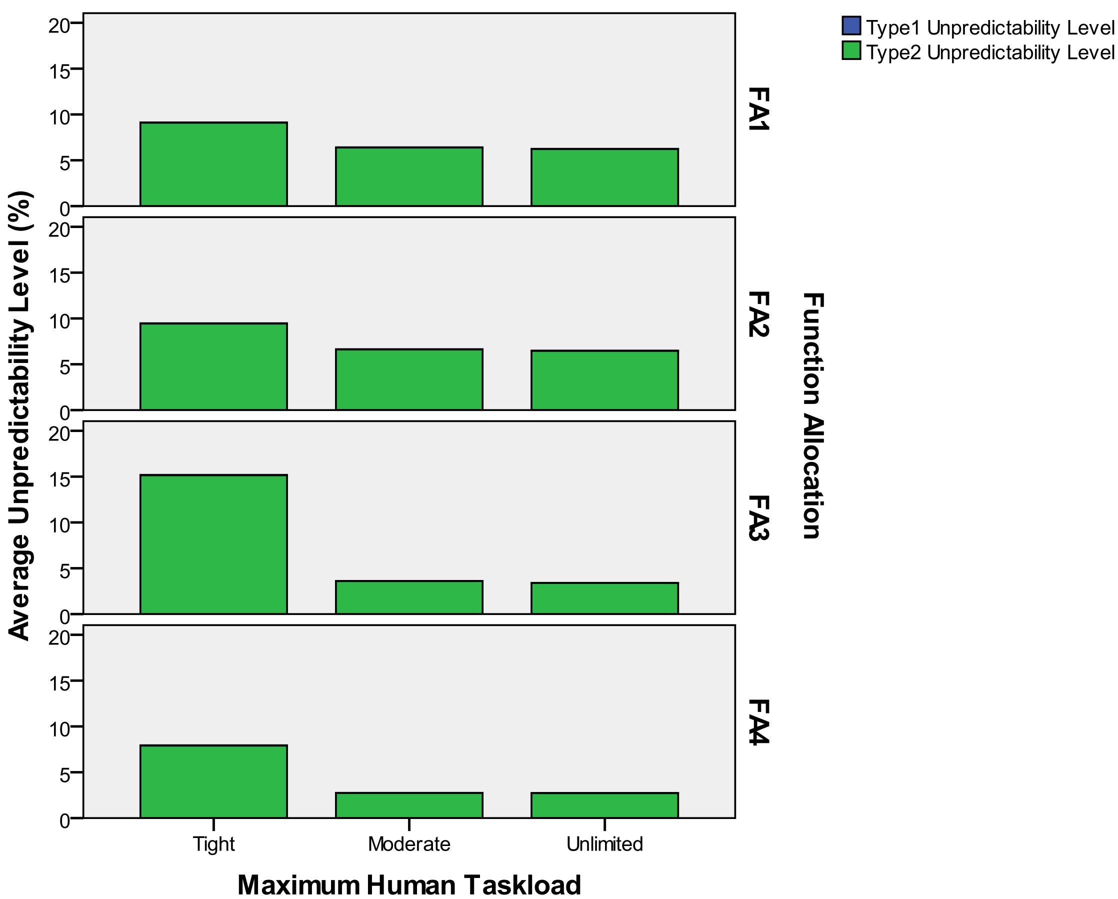

Figure 53. Average unpredictability level per simulated flight by function allocation and maximum human taskload across all scenarios with the "Strategic" cognitive control mode

Figure 53 illustrates the unpredictability level in the strategic cognitive control mode as a function of maximum human taskload and function allocation. A distinctive pattern is that, when the pilot is limited to fewer tasks, the unpredictability level increases. A one-way ANOVA found that the unpredictability level varies significantly with maximum human taskload ($p<0.0005$). Post-hoc comparisons using the Tukey HSD test indicated that the average unpredictability level for the "Tight" cases across all function allocations are significantly higher than the "Moderate" and "Unlimited" cases.

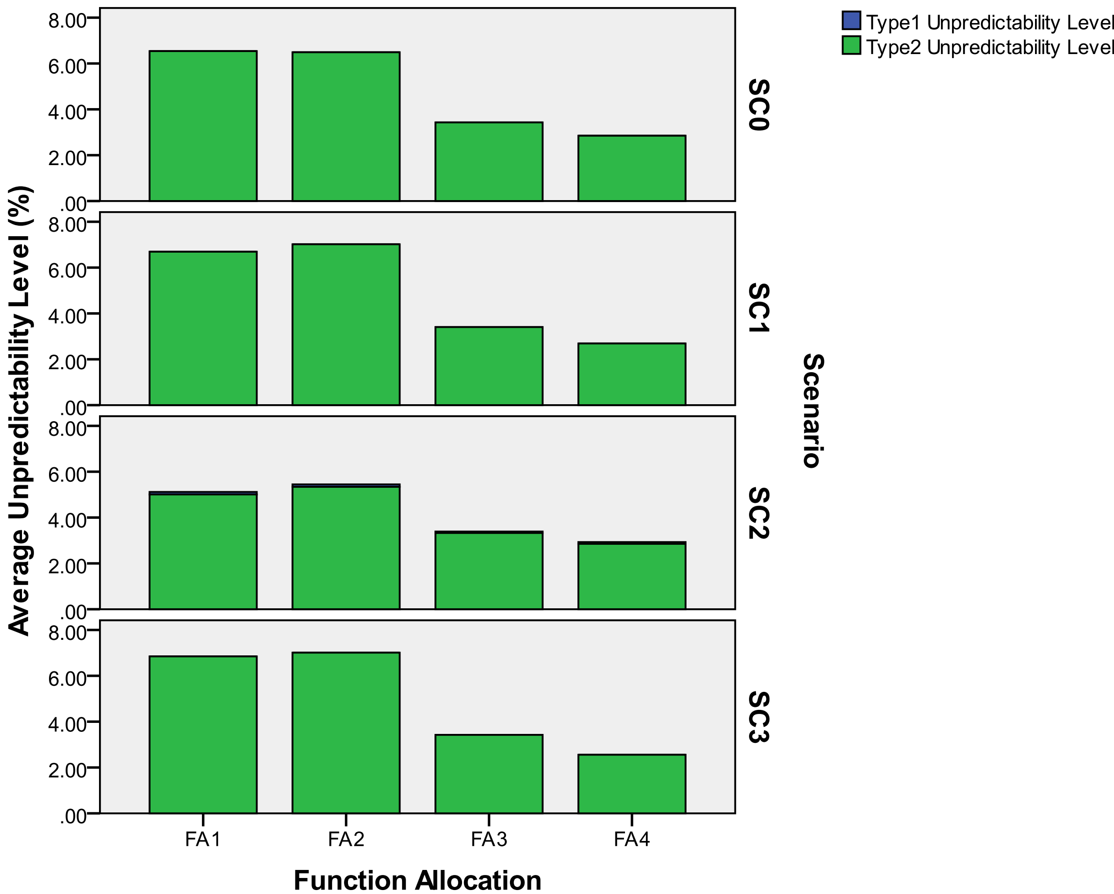

Figure 54. Average unpredictability level per simulated flight by function allocation and scenario with the "Strategic" cognitive control mode and the "Unlimited" maximum human taskload

Figure 54 illustrates the unpredictability level by function allocation and scenario. A distinctive pattern here is that the type1 unpredictability is concentrated in SC2. This scenario simulates the "unpredicted" re-route; thus, the pilot did not predict that the direct-routing instruction would be given at all. Even in SC2, the type1 unpredictability is small compared to type2 unpredictability.

5.3.7 Mission Performance

The mission performance metric has three aspects: average thrust used per second, time to land, and the average number of violated air traffic restrictions.

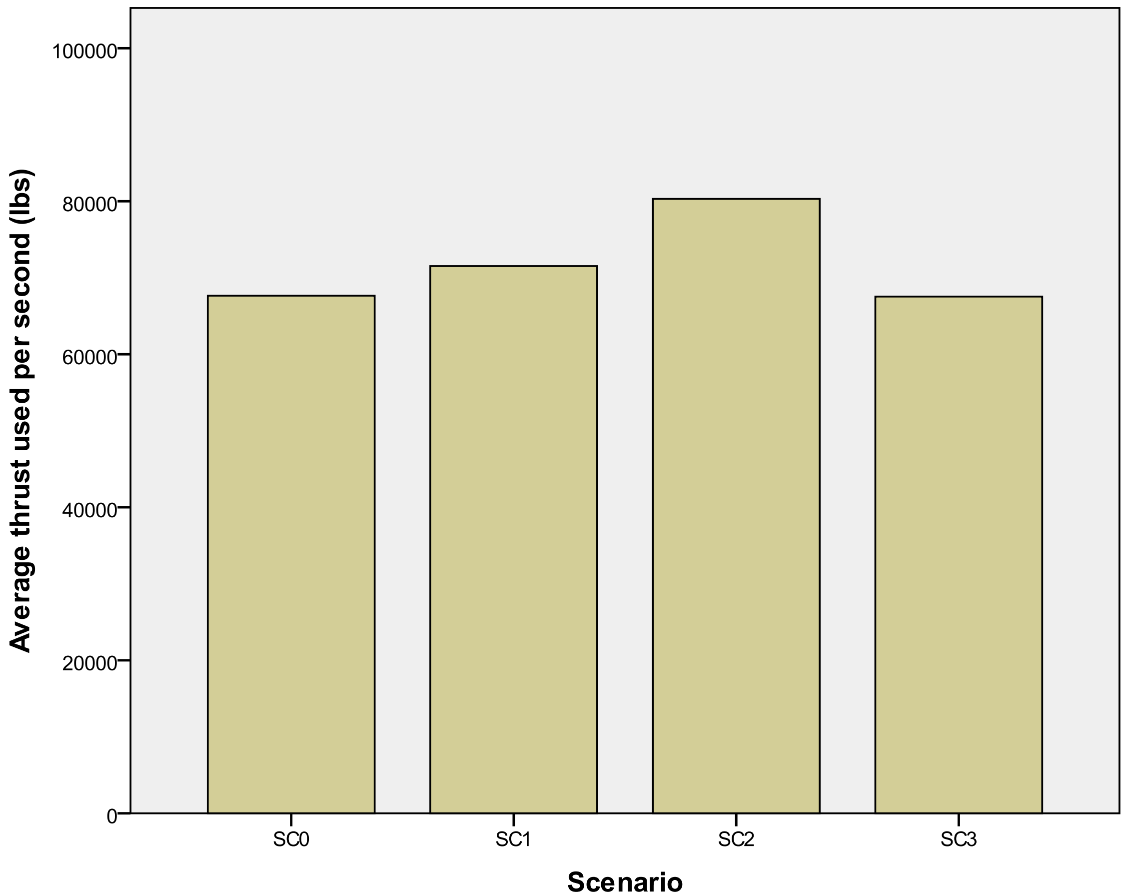

Figure 55. Average thrust used per second by scenario across all function allocations, cognitive control modes, and maximum human taskload limits

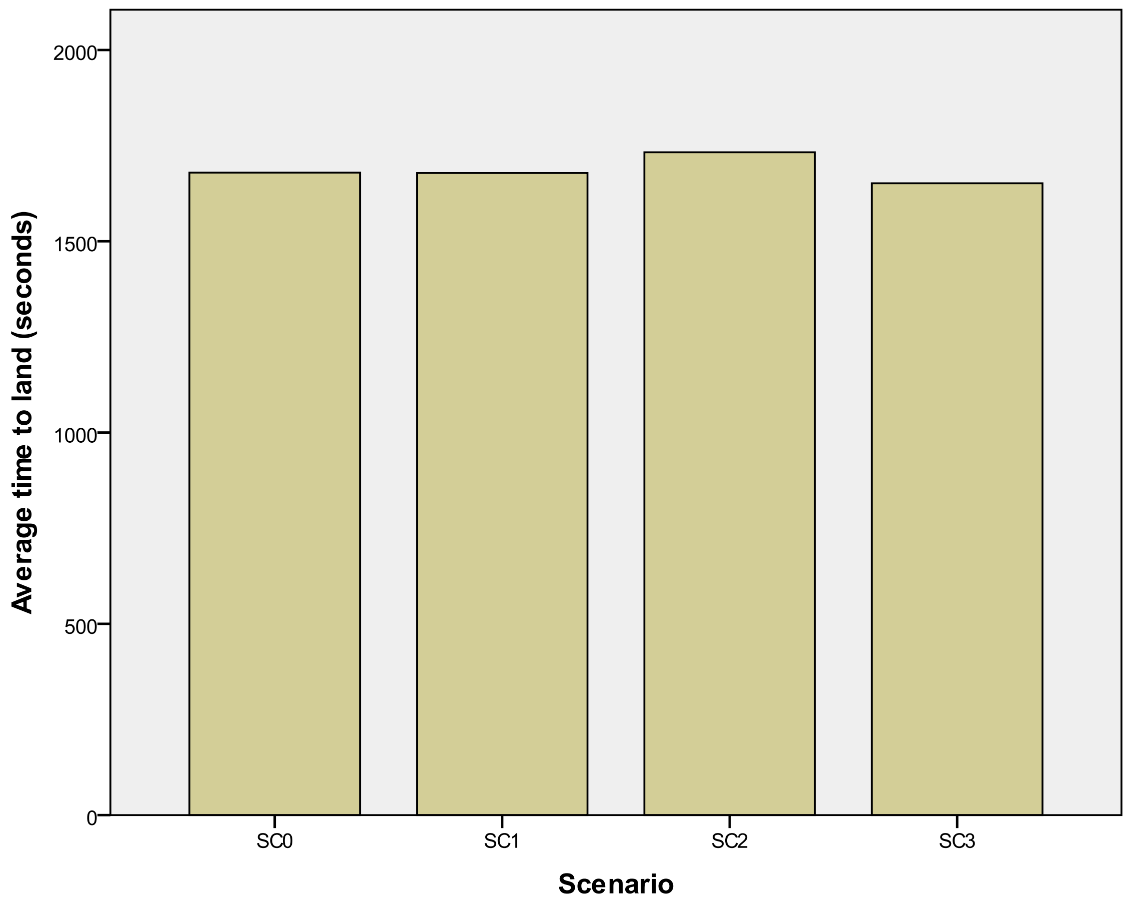

Figure 56. Average time to land per simulated flight by scenario across all function allocations cognitive control modes, and maximum human taskload limits

Both the thrust used and time to land measures did not show any distinctive differences as a function of function allocations, cognitive control modes, or maximum human taskload limits. Both measures varied between scenarios, as shown in Figure 55 and Figure 56.

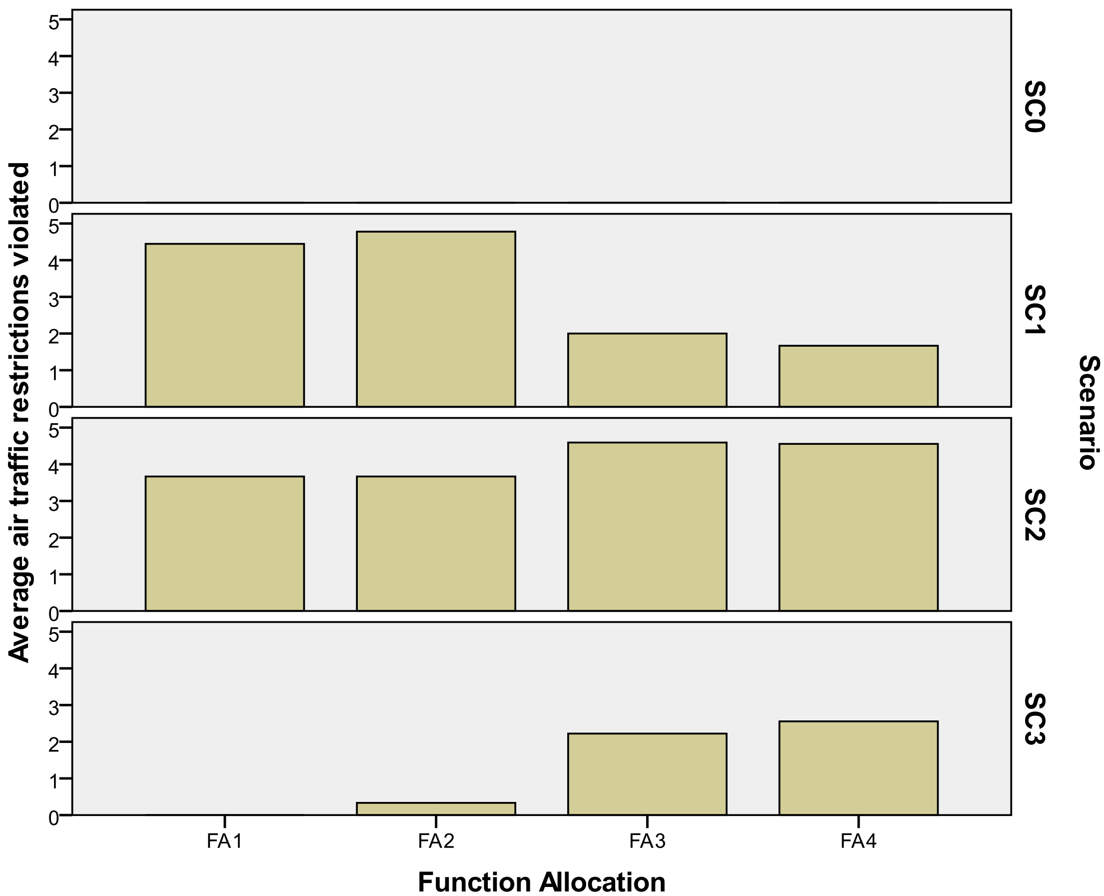

Figure 57. Average number of air traffic restrictions violated per simulated flight by function allocation and scenario across all cognitive control modes and maximum human taskload

The number of violated air traffic restrictions did not show any pattern across function allocations. The effect of function allocation and scenario interact for this measure, as shown in Figure 57. A distinctive reverse pattern is observed between the late descent scenario (SC1) and the tailwind scenario (SC3). The more highly-automated function allocation (FA1 and FA2) perform better in SC3 whereas the more manual function allocation (FA3 and FA4) perform better in SC1. This is due to the behavior demanded by these scenarios. SC1, the late descent scenario, requires a one-time vertical speed calculation to the next waypoint with a steep descent rate. In this model, without the pilot reprogramming the FMS, the descent rate with FA1 and FA2 is limited; thus, in these function allocations, it is more likely that the aircraft would not meet the air traffic restriction. On the other hand, the more manual function allocations, FA3 and FA4, allow for direct programming of the vertical speed by the pilot; thus, the pilot quickly estimates and commands the required vertical speed to meet the restriction.

The tailwind scenario, SC3, requires constant adjustment in the target heading and vertical speed in the autoflight system. Therefore, compared to the pilot regularly updating the targets in the autoflight system, the FMS constantly adjusting the targets based on the wind data is more efficient and effective.

As described in Section 5.2.1, in certain scenarios, the mission performance measures were expected to show certain relationship with the cognitive control modes. First, in SC1, the strategic mode is implemented in that if the pilot expects the late descent, he/she would decrease the speed by 0.2 Mach to manage the aircraft energy easily once the initial descent instruction would be given.

Figure 58 illustrates average number of air traffic restrictions violated by function allocation and cognitive control mode. The non-parametric test, Kruskal-Wallis test found no significant difference across the cognitive control modes ($p=0.086$) considering all function allocations. Clearly, the strategic mode shows better performance (fewer violated air traffic restrictions) with FA1 and FA2. A one-way ANOVA test of the total number of violated air traffic restrictions only considering FA1 and FA2 found that the cognitive control modes varied significantly ($p<0.0005$). Post-hoc comparisons using the Tukey HSD test of this measure indicated that the number of violated air traffic restrictions in the strategic cognitive control mode was significantly lower than ones in the opportunistic and tactical cognitive control mode.

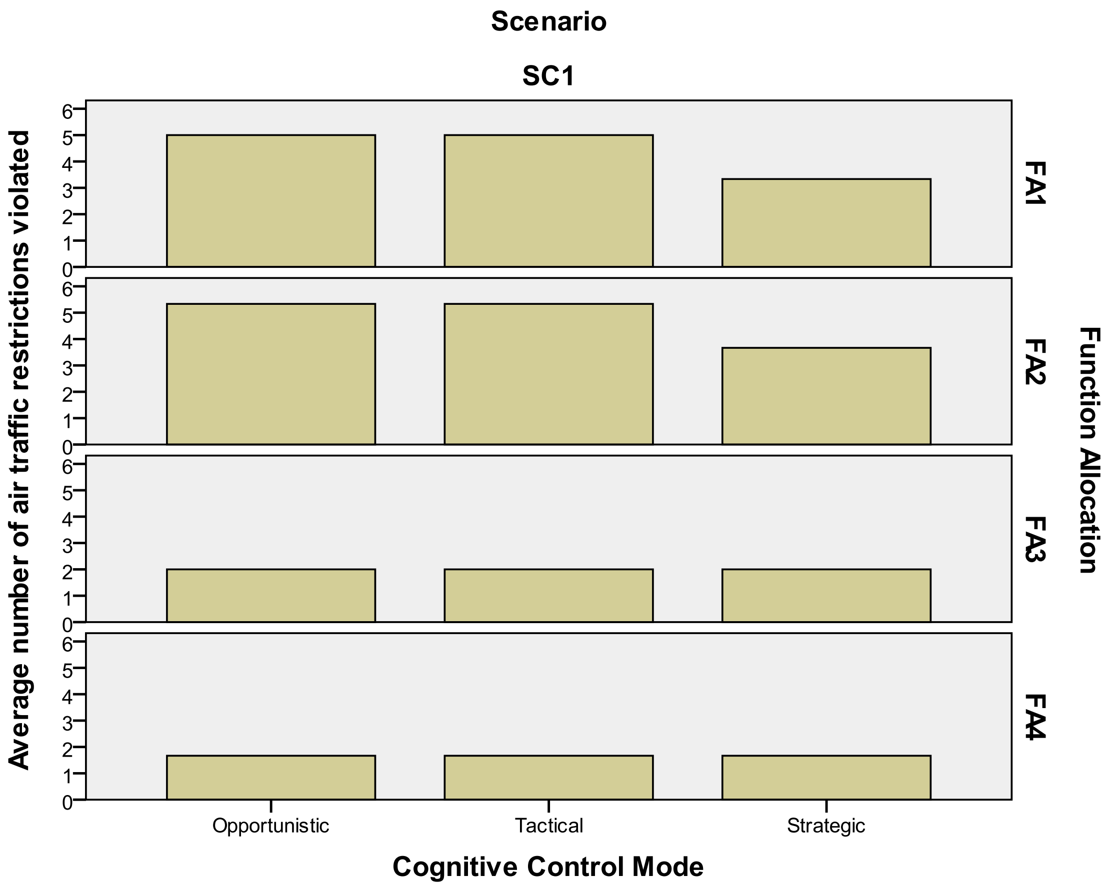

Figure 58. Average number of air traffic instruction violated per simulated flight by function allocation and cognitive control modes in SC1, the late descent scenario across all maximum human taskload

In the tailwind scenario (SC3), different pilot behaviors are expected between cognitive control mode in the more-manual function allocations (FA3 and FA4); the pilot in the opportunistic mode would only focus on adjusting the lateral profile but not the vertical profile, resulting in the poorest performance; the pilot in the tactical mode would adjust the lateral and vertical profiles regularly at large time intervals; and the pilot in the strategic mode would adjust the lateral and vertical profiles regularly at smaller time intervals. Thus, as shown in Figure 59, the more highly-automated function allocations (FA1 and FA2) show better performance than the more manual function allocations (FA3 and FA4). FA1 and FA2 were not affected by cognitive control mode whereas FA3 and FA3 were. FA3 shows the expected pattern, but FA4 shows the reverse pattern which the strategic cognitive control mode recorded the worse performance. This may be because as the tailwind pushes the aircraft to arrive earlier than expected, the air traffic instruction to descent to next altitude was not provided before the aircraft leveled off; therefore, by the time the air traffic instruction to the next altitude was given, the aircraft position was too close to the next waypoint to meet the air traffic restrictions. These circumstances are only observed with FA3 and FA4, thus resulting in more violated air traffic restrictions with those function allocations. This situation was illustrated in Figure 25 and discussed in Section 5.2.1 Scenario Descriptions.

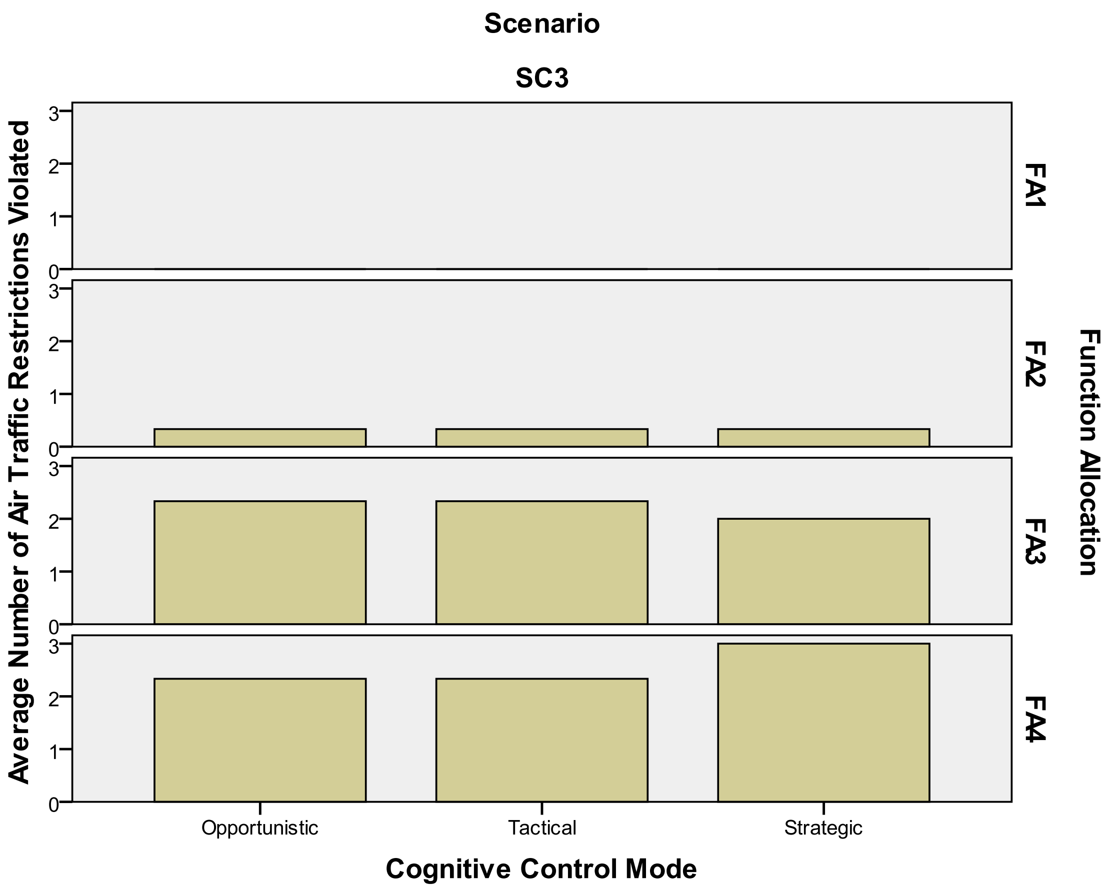

Figure 59. Average number of air traffic restrictions violated per simulated flight by two function allocations (FA3 and FA4) and all cognitive control modes only concerning the tailwind scenario (SC3) across all maximum human taskload

5.3.8 Human Adaptation to Context

This experiment used cognitive control mode as an independent variable and its impact on the other metrics of function allocation have been noted throughout the preceding sections. In addition, the ability of a function allocation to support human adaptation to context can be examined in its own right as a detailed, qualitative assessment. Specifically, during the model development, each cognitive control mode was carefully studied. The pilot behavior expected with each cognitive control mode was carefully considered relative to the function allocation and the two primary interfaces by which the pilot could interact with the various automated functions.

The CDU is the primary interface for more highly-automated function allocations (FA1 and FA2). Because the CDU is designed for future planning and information gathering efforts for predicting future states and environmental factors such as wind and waypoints in the flight route and refining the flight route, the CDU assumes a pattern of behavior fitting the strategic cognitive control mode better than other two modes. On the other hand, MCP is designed for setting the current targets of the autoflight system using an established set of autoflight modes. Thus, the MCP is found to support the tactical cognitive control mode better than other two modes.

For the purpose of flight path management, no interfaces or displays explicitly supports the opportunistic cognitive control mode; the information that could support this mode should be salient and clear, but is distributed across several displays (e.g., PFD, ND, MCP, and CDU) in the current flight deck. Therefore, the pilot who operates in the opportunistic cognitive control mode would experience difficulty identifying the most appropriate information to act upon.

Finally, examining the cognitive control mode relative to all other metrics discussed in the preceding sections, the workload measures highlight the significant amount of monitoring work required in the strategic cognitive control mode. Thus, one can infer that the human with a “Tight” maximum human taskload limit may not be able to meet the demands of the “Strategic” cognitive control model. The coherency measure showed which function allocations were considered to be incoherent. The “Opportunistic” cognitive control mode may not be well-suited for incoherent function allocations which require systematic (procedural) or strategic approaches to a piecemeal function allocation; by responding to a salient aspects of the environment, the “Opportunistic” cognitive control mode may miss disparate aspects of the work environment unrelated to their current action. The mismatch measures showed that, if the human is in the “Opportunistic” cognitive control mode, he/she will drop all the monitoring actions induced by the mismatches between responsibility and authority: only the “Strategic” cognitive control mode appeared to handle all mismatch-induced actions. The interruptive automation measures did not show any distinctive trends over the cognitive control modes. Measures of the automation boundary conditions showed that the “Strategic” cognitive control mode placed the automation out of its boundary conditions less. Unpredictability level showed a distinctive decrease as the cognitive control mode transitioned from “Opportunistic” to “Strategic,” demonstrating that the “Strategic” cognitive control mode mitigates the unpredictability of the work environment better. Finally, the cognitive control modes impacted mission performance.

5.4 Validation of Function Allocation Metrics

This section reviews the findings from Section 5.3 and discusses to what extent the metrics can be considered to be validated in terms of their ability to identify and describe issues with function allocation.

5.4.1 Workload

The “workload” issue with function allocation discussed in Chapter 2 includes four aspects: 1) total workload may not decrease with introduction of highly-automated systems (from the human-centered perspective); 2) automating one function does not necessarily decrease the workload as much as the workload that the function would require previously, but instead may change its nature from “manual” taskwork to more “cognitive” activities (from the human-centered perspective); 3) workload spikes and saturation often occur due to clumsy automation (from the human-centered perspective); and 4) lastly, teamwork actions, including both interaction and monitoring actions, can be created by function allocation and contribute significantly to workload (from the team-oriented perspective).

The first aspect was captured in the comparison of total taskload between function allocations. While the total number of pilot actions was slightly lower with the most automated function allocation, the combined duration of the taskload shows that the highly-automated function allocation (FA1) and the mostly-manual function allocation (FA4) may not differ significantly. The second and fourth aspects were also captured in these results, as highly-automated function allocations replaced taskwork and interaction actions with monitoring actions.

The third aspect, workload saturation, was also examined here using notional constructs: a range of potential maximum human taskload limits and an assumption that monitoring actions would be of lower priority should taskload limits be reached. Within these constructs, periods of workload saturation were identified, especially with the “Tight” maximum human taskload limit. The result of this workload saturation manifested as a significant reduction in the monitoring actions that are normally required in highly-automated function allocation and that correspond to the strategic behavior normally assumed of the pilot.

5.4.2 Coherency of a Function Allocation

The “incoherency in function allocations” issue with function allocation discussed in Chapter 2 includes three aspects: 1) function allocations designed based only on the machine's capabilities (from the technology-centered perspective); 2) function allocations with ambiguous team structure between the human and the automation (from the team-oriented perspective); and 3) function allocations creating heavily-interdependent and coupled work (from the work-oriented perspective).

As expected, the mixed function allocation (FA3) recorded low measures in both the coherency level (level 2) and percentage (67%). This is because only the "manage lateral route" generalized function is allocated to the automation, leaving the "manage vertical profile" generalized function to the pilot. Thus, the low level and percentage in coherency with FA3 captured the third aspects: managing the lateral route is heavily-coupled with managing vertical profile. Thus, distributing two functions that are *heavily inter-dependent with each other* to different agents should be done carefully.

Interestingly, the highly-automated function allocation (FA1) also recorded the same low measures in both the coherency level and percentage. Following the first aspect noted above, with this function allocation, almost all functions are allocated to one agent, the automation. However, *assigning all the functions possible to the machine based on its capability* did not consider the impact the "manage communication" generalized function not being entirely assigned to the flight deck automation. That is, in this notional function allocation, the automation receives the air traffic instructions, but managing flight deck components (such as change communication frequencies) is still be assigned to the pilot. This division of functions also corresponds to the second aspect, resulting in *ambiguous team structure*.

5.4.3 Mismatches between Responsibility and Authority

The issue with mismatches between responsibility and authority discussed in Chapter 2 describes situations where the human remains responsible for functions that are allocated to the automation (from the technology-centered and team-oriented perspectives). These mismatches are observed to create additional teamwork actions to monitor and supervise the automation. Static and dynamic measures show that there are more mismatches between responsibility and authority as the automation is allocated more functions and, thus, more mismatch-induced actions were required of the pilot during the simulated flight. This describes the issue with mismatches between responsibility and authority arising when *the functions are allocated to the automation while the responsibility for flight safety still remains with the pilot*.

5.4.4 Interruptive Automation

The issue with interruptive automation discussed in Chapter 2 arises from the automation interrupting the human unduly (from the team-oriented perspective) and the automation interrupting the workflow of the human (from the work-oriented perspective). The interruptive automation metric was measured as the number of instances of the automation interrupting the pilot while he/she performed operating procedures including the approach briefing, approaching checklist, and landing checklist. Given that the various scenarios invoked interruptions at different times during the arrival and approach, the interruptions were observed in all function allocations, capturing the issue *automation interrupting the pilot's workflow unduly*. In addition, two function allocations generated roughly five times the interruptions caused by the other two function allocations, showing that the decision to implement a particular function allocation can drive the frequency of interruptions.

5.4.5 Automation Boundary Conditions

The issue with automation boundary conditions discussed in Chapter 2 recognized the fixed set of boundary conditions in which the automation is operable (from the technology-centered perspective) and as a limitation to resilience of a system (from the work-oriented perspective). In this model, the automation boundary conditions metric is measured by three aspects: duration of actual speed deviation from the commanded speed, duration of vertical deviation from the planned vertical profile, and duration of the required vertical speed being higher than the maximum vertical speed that the aircraft can physically achieve (with FA3 and FA4) or than the maximum vertical speed programmed in the FMS (with FA1 and FA2). These findings captured the issue with *automation boundary conditions* in that each measure showed which function allocation and which type of operations (represented by scenarios) cause the flight deck automation to exceed these indications of its boundary conditions.

5.4.6 Stability of the Human's Work Environment

The issue with stability of the human's work environment discussed in Chapter 2 includes two aspects: 1) building on the cognitive control mode construct, function allocations may support or impede the human in maintaining a stable work environment (from the human-centered perspective), and 2) function allocations may destabilize the work environment (from team-oriented perspective). The stability of the human's work environment is a general construct that is inferred here by assessing the unpredictability. This metric captured the first aspect of this issue: the relationship to the cognitive control modes where more actions were predicted *in the strategic cognitive control mode than in the opportunistic mode*. The second aspect is captured as shown in Figure 52 in that the more "manual" function allocation recorded the lowest unpredictability level, indicating *a function allocation can change the predictability of the work environment*.

5.4.7 Mission Performance

The issue with mission performance discussed in Chapter 2 includes whether the work performed by the human and automation agents indeed meets its mission goals. Thus, in this model, the mission performance metric is measured relative to the mission goals of the arrival and approach phases of flight. Some measures of the mission performance metric showed interesting trends. For scenarios requiring rapid descent, the function allocations supporting the pilot's direct management of flight path perform better whereas for operations requiring constant adjustments to the flight path, the best performance was found with the function allocation assigning flight path management to the automation (and, thus, explicitly allocating or implicitly inducing monitoring by the pilot).

5.4.8 Human Adaptation to Context

The issue with human adaptation to context discussed in Chapter 2 is typified here as conflicts between their required actions and their cognitive control modes (from the human-centered perspective) and as limits on their strategy selection (from the work-oriented perspective). Beyond the ability of cognitive control mode as an independent variable to predict effects in other measures, human adaptation to context was also discussed qualitatively based on the insights raised during the detailed implementation of cognitive control modes in the work model. Of note, the interfaces provided in the flight deck assume one particular pattern of cognitive activity: the CDU assumes the more "Strategic" behavior of the pilot whereas the MCP assumes the more "Tactical" behavior of the pilot. The "Opportunistic" cognitive control mode is found to be not supported in any interfaces provided in the current flight deck (for the purpose of flight path management).

In addition, the discussion of the cognitive control modes on other measures of function allocation showed that 1) cognitive control modes could predict and explain other measures and 2) certain cognitive control modes are not appropriate for certain operating conditions. For example, workload and unpredictability levels showed the most distinctive correlations with the cognitive control modes. Comparing with other independent variables such as maximum human taskload, certain cognitive control modes are not appropriate or may not be feasible in some situations. These insights can be further applied to identify problematic requirements with given function allocations by examining their assumptions about human behavior in detail relative to the structures of cognitive control and strategy selection.

CHAPTER 6

CONCLUSION

6.1 Summary of Project Work

Various fields of research have studied human-automation function allocation implicitly or explicitly. Engineering and computer science have developed many new automation technologies, “pushing” the boundaries of what automation can do, described here as the technology-centered perspective on function allocation. This perspective highlighted the following issues with function allocation: 1) *incoherency in function allocations* in which the human “picks up” any functions beyond the automation’s capabilities, 2) *mismatches between responsibility and authority* due to function allocations only considering the capabilities of automation, and 3) function allocation creating the requirement for the human to monitor for *automation boundary conditions*.

As an opposite approach, human factors focuses on “How can technology best support human performance?” representing the human-centered perspective. This perspective highlighted the following issues with function allocation: 1) *workload* that is not decreased or is increased by the function allocation, workload spikes and saturation, clumsy automation, and changes in the nature of the workload; 2) function allocation preventing *human adaptation to context* such as conflicts between their required actions and their cognitive control modes; and 3) *function allocation destabilizing the human’s work environment* by reducing predictability.

Also, studies of team human factors, team and organization design, and management provide many useful insights for teams of humans and automation. The team-oriented perspective highlighted following the issues with function allocation: 1) *mismatches between responsibility and authority* where a function allocation delegates authority without delegating responsibility; 2) *incoherency in function allocations* compared to a clearly defined team structure; 3) *interruptive automation* compared to human-to-human communication; 4) *workload* through induced teamwork; and 5) *function allocation destabilizing the human's work environment* through poor adaptation of, or rigidity in, coordination strategies.

Finally, cognitive systems engineering turned the question to "How can the human-automated team improve work performance?" representing the work-oriented perspective. Studies in this field examine together the structure of work, its environment, and the agents. The work-oriented perspective highlighted the following issues with function allocation: 1) *mission performance*; 2) *interruptive automation* relative to the established workflow; 3) *automation boundary conditions* as a limit to resilience; 4) function allocation preventing *human adaptation to context* by limiting strategy selection; and 5) *incoherency in function allocations* both in terms of clear role distribution and in terms of inter-dependencies where the action of one agent may drive the actions of the other.

Examining these disparate perspectives, eight common issues with function allocation were identified across the perspectives. This effort included developing the WMC framework that first establishes a detailed static work model and then dynamically simulates it. Based on this work model and the inputs dynamic simulation can bring, the eight categories of issues can be assessed by the eight types of function allocation metrics.

Finally, the project demonstrated the metrics and the work model in an experiment, with the following insights:

- The workload metric captured all aspects of the workload issue identified across the human-centered and team-oriented perspectives: 1) total workload may not decrease with introduction of highly-automated systems; 2) automating one function does not necessarily decrease the workload from that which the function would required previously, but instead may change its nature from “manual” taskwork to more “cognitive” activities; 3) workload spikes and saturation may occur due to clumsy automation; and 4) lastly, additional teamwork actions, including both interaction and monitoring actions, can be created by function allocation and contribute significantly to workload.

- The coherency of a function allocation metric captured all aspects of the coherency issue identified from the technology-centered, team-oriented, and work-oriented perspectives: 1) function allocations designed based solely on the machine’s capabilities; 2) function allocations with ambiguous team structure between the human and the automation; and 3) function allocations performed in heavily-interdependent and coupled work.
- The mismatches between responsibility and authority metric captured mismatch issues where the human remains responsible for functions that are allocated to the automation.
- The interruptive automation metric captured the issue of automation interrupting the pilot’s workflow unduly, specifically interruptions of operating procedures, as identified from the team- and work-oriented perspectives.
- The automation boundary conditions metric captured the issue by showing that within different operations, function allocations can increase the probability that the automation will be forced outside its boundary conditions.
- The human adaptation to context metric captured the issue with function allocation by using the construct of cognitive control modes to systematically discuss where certain cognitive control modes conflict with a function allocation’s expectation for pilot behavior and with flight deck automation. Cognitive control modes were also found to be an explanation and predictor of several other metrics of function allocation.

- The metric for stability of the human's work environment captured these aspects of this issue with function allocation: 1) function allocations supporting or impeding the human in maintaining a stable work environment were shown to have a relationship with cognitive control mode; and 2) function allocations destabilizing the work environment as identified from the human-centered and team-oriented perspectives were suggested by increased unpredictability levels in specific function allocations.
- Lastly, the mission performance metric showed how mission goals were achieved with, and impacted by, a set of representative function allocations.

6.2 Contributions

6.2.1 Metrics from Multiple Perspectives

Issues with human-automation function allocation have been studied from various fields: engineering, human factors, team and organization design, and cognitive systems engineering. However, these independent studies have addressed only relevant aspects, which cannot fully address issues with function allocation on their own. Thus, issues with human-automation function allocation should be considered from multiple perspectives.

This comprehensive review of function allocation from different perspectives resulted in the eight types of function allocation metrics that can predict and capture issues with function allocation. These metrics can identify detailed problematic aspects of allocating functions between human and automated agents in complex work environments such as aviation, with particular regard to concerns with safety. Thus, these metrics, and the models used to assess them, provide valuable insights for designers of function allocations.

In addition, this review across multiple domains can provide insights into broader aspects of function allocation that serve as common underlying constructs of concern, and highlight considerations that have not been comprehensively covered in individual domains.

6.2.2 Work Model that Computes

The WMC framework is itself a contribution whose components were created around the need to assess the function allocation metrics. Unlike other modeling frameworks proposed to investigate specific aspects of function allocation, WMC mirrors many aspects of agent behavior and of the work environment, and allows for a broad view of the myriad tasks required: physical taskwork that mirrors the work that needs to be done; induced teamwork that represents human-automation interactions; and strategies that reflect the context of the situation including delegation within the human-automation team.

The WMC framework aids in function allocation by providing a systematic means to explore the important characteristics of a work environment relative to the design task of allocating functions. Specifically, the modeler is required to investigate all possible and potential taskwork, teamwork, and required resources, including the likely cognitive control modes of the humans, required monitoring behaviors due to concerns with responsibility and authority, and dynamic considerations of when actions will be performed. Thus, the development of the work model is itself a formative process that provides insights into (and static measures of) function allocation.

A common criticism of established work domain analysis and its resulting work models (i.e., the abstraction hierarchy) is that they are static and extremely difficult to validate. Unlike other work models, the WMC framework developed here can "compute," that is to say, be dynamically simulated. This implies that the work model can be validated in terms of assessing whether the model provides a full and accurate representation of the dynamics of a complex work environment by comparing the behaviors found in computational simulations to any data available about real operations or gathered in high-fidelity human-in-the-loop simulator tests. Once validated in terms of reflecting reasonable system behaviors, the simulations then enable assessment of dynamic measures of the function allocation metrics.

6.3 Recommendation for Future Efforts

6.3.1 Reinforcement of Metric Validation

As described in Chapter 4, the proposed eight metrics of function allocation can be measured in human-in-the-loop simulations or in real operations. Comparing the results from the computational experiment conducted here with results from human-in-the-loop simulations (and/or results from the real operations) can provide more rigorous validation and also interesting insights that may still be hidden within the limitations of the work modeling framework used here, especially with regards to representing human cognitive control modes and human behavior. For example, modeling scenarios that present humans with "entirely" unpredicted circumstances were difficult here because the pilot's responses must be described in the model, and therefore are unpredicted to the modeler.

Note also the workload can be assessed instead of taskload. In this project, taskload was measured (by counting the number of actions required of the pilot and their combined durations) due to a lack of workload data to better parameterize the workload inherent in each of the pilot's actions. With such workload data, the WMC framework (a work model and a dynamic simulation) can provide more accurate measures of the workload metric.

6.3.2 Intricate Human Agent Model

Among the conceptual discussions in assessing the function allocation metrics in Chapter 4, certain aspects of human performance were not included in this model:

transition between cognitive control modes and "forgetting" a task. An agent model was used that can maintain only one cognitive control mode during a simulated flight. As a next step, the transition capability in response to context could be implemented. This capability will describe how, based on the dynamics of transitions between cognitive control modes, a given function allocation shapes the pilot's activities and thus also impacts other metrics of function allocation, including mission performance.

In addition, an agent model was used with a basic task management capability that does not include subtle and more intricate behaviors of the humans such as, forgetting a delayed or interrupted task. Here, when the interruptions occurred, the pilot interrupted the current task but later resumed it correctly. This particular aspect of human performance is of particular interest to the interruptive automation metric, but can also impact the mission performance metrics, particularly when the system is not resilient to forgotten pilot tasks.

6.3.3 Modeling Dynamic Function Allocation

The work model and function allocation metrics are applicable to any type of function allocation. The current effort demonstrated the assessment of static function allocations. However, humans may switch between function allocations (in realistic operations, for example, pilots switching one function allocation using LNAV/VNAV to another using MCP) as a form of "adaptable" dynamic function allocation. More elaborate function allocation selection mechanisms can also be envisioned and analyzed by these metrics and work model. For example, a dynamic function allocation using the "playbook" delegation scheme can be tested via the WMC framework and the function allocation metrics. Going further, "adaptive automation" concepts in which the automation selects function allocations based on the situation and some awareness of the pilot's state can also be examined. Thus, the function allocation metrics capable of assessing dynamic function allocations can be used to assess and compare the cost and benefit between the current function allocations vs. potential dynamic function allocations.

Going further, many of these function allocation metrics can be identified "in real-time, during real-operations." Thus, the metrics themselves may be part of an on-board mechanism that detects when a new function allocation would score higher on the metrics and identifies which function allocation would be the most appropriate to the situation.

6.3.4 Other Applications

Other domains can utilize the WMC framework (including the function allocation metrics) as well. For example, human-robot interaction application can benefit from this framework and the function allocation metrics. More broadly, the WMC framework is intended to address the needs of designing a wide range of complex, dynamic, safety-critical work environments.

REFERENCES

Abbott, K., Slotte, S. M., & Stimson, D. K. (1996). *The interface between flightcrews and modern flight deck systems*. Washington D.C.: Federal Aviation Administration.

Arthur, W., Edwards, B. D., Bell, S. T., Villado, A. J., & Bennett, W. (2005). Team task analysis: Identifying tasks and jobs that are team based. *Human Factors: The Journal of the Human Factors and Ergonomics Society, 47*(3), 654.

Bainbridge, L. (1983). Ironies of automation. *Automatica, 19*(6), 775-779.

Bass, E., & Pritchett, A. (2008). Human-automated judge learning: A methodology for examining human interaction with information analysis automation. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on, 38*(4), 759-776.

Beyer, H., & Holtzblatt, K. (1998). *Contextual design: defining customer-centered systems*: Morgan Kaufmann Pub.

Billings, C. E. (1997). *Aviation automation: The search for a human-centered approach*: Lawrence Erlbaum Associates Publishers.

Bowers, C. A., Oser, R. L., Salas, E., & Cannon-Bowers, J. A. (1996). Team performance in automated systems *Automation and human performance: Theory and applications* (pp. 243–263). Mahwa, NJ: Lawrence Erlbaum.

Byrne, M. D., Kirlik, A., & Fleetwood, M. D. (2007). An ACT-R Approach to Closing the Loop on Computational Cognitive Modeling: Describing Dynamics of Interactive Decision Making and Attention Allocation. In D. C. Foyle & B. L. Hooey (Eds.), *Human Performance Modeling in Aviation*. Boca Raton, FL: CRC Press.

Carley, K. M., & Kamneva, N. Y. (2004). *A network optimization approach for improving organizational design* (No. CMU-ISRI-04-102): Carnegie Mellon University, School of Computer Science, Institute for Software Research International.

Casner, S. M. (2001). *The pilot's guide to the modern airline cockpit*. Ames, IW: Blackwell Publishing.

Cegarra, J., & Chevalier, A. (2008). The use of Tholos software for combining measures of mental workload: Toward theoretical and methodological improvements. *Behavior Research Methods, 40*(4), 988.

Chapanis, A., Frick, F. C., Garner, W. R., Gebhard, J. W., Grether, W. F., Henneman, R. H., et al. (1951). *Human Engineering for an effective air navigation and traffic control system*. Washington D. C.: National Research Council.

Chialastri, A., & Pozzi, S. (2008). Resilience in the Aviation System. *Lecture Notes in Computer Science, 5219/2008*, 86-98.

Christoffersen, K., & Woods, D. D. (2002). How to make automated systems team players. *Advances in human performance and cognitive engineering research, 2*, 1-12.

Corker, K. M., Muraoka, K., Verma, S., & Jadhav, A. (2007). Air MIDAS: A Closed-Loop Model Framework. In D. C. Foyle & B. L. Hooey (Eds.), *Human Performance Modeling in Aviation* Boca Raton, FL CRC Press.

Damos, D. L., & Tabachnick, B. G. (2001). *The Effect of Interruptions on Flight Crew Performance: ASRS Reports*. Los Angeles, CA: Damos Research Associates.

Degani, A., & Wiener, e. (1990). *Human Factors of Flight-Deck Checklists: The Normal Checklist* (No. 177549). Moffett Field, California: NASA Ames Research Center.

Degani, A., & Wiener, E. L. (1993). Cockpit checklists: Concepts, design, and use. *Human Factors: The Journal of the Human Factors and Ergonomics Society, 35*(2), 345-359.

Dekker, S., & Woods, D. (2002). Maba-maba or abracadabra? Progress on human–automation co-ordination. *Cognition, Technology & Work, 4*(4), 240-244.

Deutsch, S. E., & Pew, R. W. (2007). D-OMAR: An Architecture for Modeling Multitask Behaviors. In D. C. Foyle & B. L. Hooey (Eds.), *Human Performance Modeling in Aviation*. Boca Raton, FL: CRC Press.

Dixon, S. R., Wickens, C. D., & Chang, D. (2005). Mission control of multiple unmanned aerial vehicles: A workload analysis. *Human Factors: The Journal of the Human Factors and Ergonomics Society, 47*(3), 479.

Entin, E. E., & Entin, E. B. (2001). *Measures for evaluation of team processes and performance in experiments and exercises*. Paper presented at the 6th International Command and Control Research and Technology Symposium.

Entin, E. E., & Serfaty, D. (1999). Adaptive team coordination. *Human Factors: The Journal of the Human Factors and Ergonomics Society, 41*(2), 312.

Feigh, K. M. (2008). *Design of Cognitive Work Support Systems for Airline Operations.* Georgia Institute of Technology, Atlanta, GA.

Feigh, K. M. (2010). Incorporating Multiple Patterns of Activity into the Design of Cognitive Work Support Systems. *Cognition, Technology & Work*.

Feigh, K. M. (2010). Incorporating multiple patterns of activity into the design of cognitive work support systems. *Cognition, Technology & Work*, 1-21.

Funk, K., & Lyall, B. (1998). *Human factors issues of flight deck automation.* Paper presented at the Proceedings of Digital Avionics Systems Conference, 17th DASC.

Funk, K., & Lyall, B. (2000). *A comparative analysis of flightdecks with varying levels of automation (Federal Aviation Administration, Grant 93-G-039, Phase 1 final report).* Washington D. C.: Federal Aviation Administration.

Gopher, D., & Braune, R. (1984). On the psychophysics of workload: Why bother with subjective measures? *Human Factors: The Journal of the Human Factors and Ergonomics Society, 26*(5), 519-532.

Hart, S., & Staveland, L. (1988). Development of NASA-TLX (Task Load Index): Results of empirical and theoretical research. *Human mental workload, 1*, 139–183.

Hollenbeck, J. R., Ilgen, D. R., Sego, D. J., Hedlund, J., Major, D. A., & Phillips, J. (1995). Mutlilevel theory of team decision making: Decision performance in teams incorporating distributed expertise. *Journal of Applied Psychology, 80*(2), 292.

Hollnagel, E. (1993). *Human reliability analysis: Context and control*. London, UK: Academic Press

Hollnagel, E. (2002). Time and time again. *Theoretical Issues in Ergonomics Science, 3*(2), 143-158.

Hollnagel, E. (2004). *Barriers and accident prevention*: Ashgate Pub Ltd.

Hollnagel, E., Woods, D. D., & Leveson, N. (2006). *Resilience engineering: Concepts and precepts*: Ashgate Pub Co.

Hutchins, E. (1995). *Cognition in the Wild*. Cambridge, MA: MIT Press.

Javaux, D. (1998). *Explaining Sarter and Woods' classical results.* Paper presented at the HESSD '98 100 Error in Aviation Decision Making: A Factor in Accidents and Incident.

Javaux, D. (2002). A method for predicting errors when interacting with finite state systems. How implicit learning shapes the user's knowledge of a system. *Reliability Engineering & System Safety, 75*(2), 147-165.

Jett, Q. R., & George, J. M. (2003). Work interrupted: A closer look at the role of interruptions in organizational life. *Academy of Management Review*, *28*(3), 494-509.

Johnson, E. N., Calise, A. J., & de Blauwe, H. (2008). *In Flight Validation of Adaptive Flight Control Methods*. Paper presented at the AIAA Guidance, Navigation and Control Conference and Exhibit.

Kalambi, V. V., Pritchett, A. R., Bruneau, D. P. J., Endsley, M. R., & Kaber, D. B. (2007). *In-Flight Planning and Intelligent Pilot Aids for Emergencies and Non-Nominal Flight Conditions Using Automatically Generated Flight Plans*.

Kantowitz, B. H. (1988). Defining and Measuring Pilot Mental Workload. In J. R. Comstock (Ed.), *Mental State Estimation 1987* (pp. 179-188). Hampton, VA: NASA, Scientific and Technical Information Division.

Kantowitz, B. H. (2000). *Attention and Mental Workload*. Paper presented at the IEA 2000/HFES 2000 Congress.

LaJoie, A. S. (1999). *A Review and Annotated Bibliography of the Literature Pertaining to Team and Small Group Performance (1989 to 1999)*: ARMY RESEARCH INST FOR THE BEHAVIORAL AND SOCIAL SCIENCES FORT KNOX NY ARMORED FORCES RESEARCH UNIT.

Laughery Jr., K., & Corker, K. M. (1997). *Computer modeling and simulation of human/system performance*. New York, NY: Wiley.

Lee, J., & Moray, N. (1992). Trust, control strategies and allocation of function in human-machine systems. *Ergonomics, 35*(10), 1243-1270.

Lee, J. D., & See, K. A. (2004). Trust in automation: Designing for appropriate reliance. *Human Factors: The Journal of the Human Factors and Ergonomics Society, 46*(1), 50.

Leiden, K., Keller, J., & French, J. (2002). *Information to Support the Human Performance Modeling of a B757 Flight Crew during Approach and Landing*. Moffett Field, CA: NASA Ames Research Center.

MacMillan, J., Entin, E. E., & Serfaty, D. (2004). Communication overhead: The hidden cost of team cognition. *Team cognition: Understanding the factors that drive process and performance*, 61-82.

Meriweather, J. from http://www.meriweather.com/747/fd-747.html

Miller, C. A., & Parasuraman, R. (2007). Designing for flexible interaction between humans and automation: Delegation interfaces for supervisory control. *Human Factors: The Journal of the Human Factors and Ergonomics Society, 49*(1), 57.

Molloy, R., & Parasuraman, R. (1996). Monitoring an automated system for a single failure: Vigilance and task complexity effects. *Human Factors: The Journal of the Human Factors and Ergonomics Society, 38*(2), 311-322.

Mosier, K., Palmer, E., & Degani, A. (1992). *Electronic checklists: Implications for decision making*.

Muir, B. M. (1994). Trust in automation: Part I. Theoretical issues in the study of trust and human intervention in automated systems. *Ergonomics, 37*(11), 1905-1922.

Muir, B. M., & Moray, N. (1996). Trust in automation. Part II. Experimental studies of trust and human intervention in a process control simulation. *Ergonomics, 39*(3), 429-460.

Naikar, N. (1999). Work domain analysis for training-system definition and acquisition. *The International Journal of Aviation Psychology, 9*(3), 271-290.

Naikar, N., Pearce, B., Drumm, D., & Sanderson, P. M. (2003). Designing teams for first-of-a-kind, complex systems using the initial phases of cognitive work analysis: Case study. *Human Factors: The Journal of the Human Factors and Ergonomics Society, 45*(2), 202.

Norman, D. A. (1990). The 'problem' with automation: inappropriate feedback and interaction, not 'over-automation'. *Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences, 327*(1241), 585-593.

Ockerman, J., & Pritchett, A. (2000). A review and reappraisal of task guidance: Aiding workers in procedure following. *International Journal of Cognitive Ergonomics, 4*(3), 191-212.

Palmer, E., & Degani, A. (1991). *Electronic checklists: Evaluation of two levels of automation*.

Palmer, M. T., Rogers, W. H., Press, H. N., Latorella, K. A., & Abbott, T. S. (1995). *A crew-centered flight deck design philosophy for high-speed civil transport (HSCT) aircraft* (No. TM 109171). Hampton, VA: NASA Langley Research Center.

Parasuraman, R., & Riley, V. (1997). Humans and automation: Use, misuse, disuse, abuse. *Human Factors: The Journal of the Human Factors and Ergonomics Society, 39*(2), 230-253.

Parasuraman, R., Sheridan, T. B., & Wickens, C. D. (2000). A model for types and levels of human interaction with automation. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on, 30*(3), 286-297.

Paris, C. R., Salas, E., & Cannon-Bowers, J. A. (2000). Teamwork in multi-person systems: a review and analysis. *Ergonomics, 43*(8), 1052-1075.

Pew, R. W., & Mavor, A. S. (1998). *Modeling human and organizational behavior: Application to military simulations*: National Academies Press.

Pitt, J. (2004). Digital blush: towards shame and embarrassment in multi-agent information trading applications. *Cognition, Technology & Work, 6*(1), 23-36.

Potter, S. S. (1989). *Subjective Workload Assessment Technique (SWAT): A User's Guide* (No. ADA215405): DTIC.

Pritchett, A. R. (2001). Reviewing the role of cockpit alerting systems. *Human Factors and Aerospace Safety, 1*(1).

Pritchett, A. R. (2010). The System Safety Perspective. *Human factors in aviation*, 65.

Pritchett, A. R., Kim, S. Y., Kannan, S. K., & Feigh, K. M. (2010). *Simulating situated work*. Paper presented at the Cognitive Methods in Situation Awareness and Decision Support (CogSIMA), 2011 IEEE First International Multi-Disciplinary Conference on

Rasmussen, J. (1985). The role of hierarchical knowledge representation in decisionmaking and system management. *IEEE Transactions on Systems, Man, & Cybernetics, 15*(2), 234-243.

Rasmussen, J., Pejtersen, A. M., & Goodstein, L. P. (1994). *Cognitive systems engineering*. New York, NY: John Wiley & Sons, Inc.

Rasmussen, J., Pejtersen, A. M., Schmidt, K., & Risø, F. (1990). *Taxonomy for cognitive work analysis* (No. Risø-M-2871): Risø National Laboratory.

Roth, E. M., & Bisantz, A. M. (in press). Cognitive work analysis. In A. Kirlik & J. Lee (Eds.), *Handbook of Cognitive Engineering*: Oxford Press.

Salas, E., Dickinson, T. L., Converse, S. A., & Tannenbaum, S. I. (1992). Toward an understanding of team performance and training. In R. W. Swezey & E. Salas (Eds.), *Teams: Their training and performance*. Norwood, NJ: Ablex.

Sarter, N., Woods, D., & Billings, C. (1997). Automation surprises. *Handbook of human factors and ergonomics, 2*, 1926-1943.

Sarter, N. B., & Woods, D. D. (1992). Pilot interaction with cockpit automation: Operational experiences with the flight management system. *The International Journal of Aviation Psychology, 2*(4), 303-321.

Sarter, N. B., & Woods, D. D. (1994). Pilot interaction with cockpit automation II: An experimental study of pilots' model and awareness of the flight management system. *The International Journal of Aviation Psychology, 4*(1), 1-28.

Sarter, N. B., & Woods, D. D. (1995). How in the world did we ever get into that mode? Mode error and awareness in supervisory control. *Human Factors: The Journal of the Human Factors and Ergonomics Society, 37*(1), 5-19.

Sarter, N. B., & Woods, D. D. (2000). Team play with a powerful and independent agent: a full-mission simulation study. *Human Factors: The Journal of the Human Factors and Ergonomics Society, 42*(3), 390.

Schraagen, J. M., & Rasker, P. (2003). Team design. In E. Hollnagel (Ed.), *Handbook of Cognitive Task Design* (pp. 753-786). Mahwah, NJ: Lawrence Erlbaum.

Schutte, P. C. (1999). Complemation: An alternative to automation. *Journal of Information Technology Impact, 1*(3), 113-118.

Schutte, P. C., Goodrich, K. H., Cox, D. E., Jackson, E. B., Palmer, M. T., Pope, A. T., et al. (2007). *The Naturalistic Flight Deck System: An Integrated System Concept for Improved Single-Pilot Operations*. Hampton, VA, NASA Langley Research Center.

Sheridan, T. B. (1998). Allocating functions rationally between humans and machines. *Ergonomics in Design: The Quarterly of Human Factors Applications, 6*(3), 20-25.

Sheridan, T. B. (2008). Risk, human error, and system resilience: Fundamental ideas. *Human Factors: The Journal of the Human Factors and Ergonomics Society, 50*(3), 418.

Sheridan, T. B., & Verplank, W. (1978). *Human and computer control of undersea teleoperators* Cambridge: MIT.

Sherry, L., Feary, M., Polson, P., Mumaw, R., & Palmer, E. (2001). *A cognitive engineering analysis of the vertical navigation (VNAV) function*. Moffet Field, CA: NASA-ARC.

Sherry, R. R., & Ritter, F. E. (2002). *Dynamic task allocation: issues for implementing adaptive intelligent automation*: School of Information Sciences and Technology, The Pennsylvania State University.

Speier, C., Valacich, J., & Vessey, I. (1999). The influence of task interruption on individual decision making: An information overload perspective. *Decision Sciences, 30*(2), 337-360.

Stanton, N. A., Ashleigh, M. J., Roberts, A. D., & Xu, F. (2001). Testing Hollnagel's Contextual Control Model: Assessing team behavior in a human supervisory control task. *International Journal of Cognitive Ergonomics, 5*(2), 111-123.

Stimpson, R. D. (2010). *UNDERSTANDING VNAV PATH DESCENTS*. Atlanta, GA: Delta.

Stout, R., & Salas, E. (1993). *The role of planning in coordinated team decision making: Implications for training*. Paper presented at the Human Factors and Ergonomics Society Annual Meeting

Szilagyi Jr., A. D., & Wallace Jr., M. J. (1980). *Organizational behavior and performance*. Santa Monica, CA: Goodyear Pub. Co.

Totah, J., Krishnakumar, K., & Vikien, S. (2007). *Integrated Resilient Aircraft Control-Stability, Maneuverability, and Safe Landing in the Presence of Adverse Conditions*.

Vicente, K. J. (1999). *Cognitive work analysis: Toward safe, productive, and healthy computer-based work*: Lawrence Erlbaum.

Vicente, K. J., & Rasmussen, J. (1992). Ecological interface design: Theoretical foundations. *IEEE Transactions on Systems, Man and Cybernetics, 22*(4), 589-606.

Weiner, E. L., & Curry, R. E. (1980). Flight-deck automation: Promises and problems. *Ergonomics, 23*(10), 995-1011.

Whitelaw, J. from www.airliners.net

Wickens, C. D., McCarley, J. S., Alexander, A. L., Thomas, L. C., Ambinder, M., & Zheng, S. (2007). Attention-Situation Awareness (A-SA) Model of Pilot Error. In D. C. Foyle & B. L. Hooey (Eds.), *Human Performance Modeling in Aviation*. Boca Raton, FL: CRC Press.

Wiener, E. (1989a). *Human factors of advanced technology (glass cockpit) transport aircraft* (No. 177528). Moffett Field, CA: NASA Ames Research Center.

Wiener, e. (1989b). Human factors of advanced technology (glass cockpit) transport aircraft.

Wiener, E. L. (1985). *Human factors in cockpit automation: A field study of flight crew transition* (No. 177333). Moffett Field, CA: NASA Ames Research Center.

Wiener, E. L., & Curry, R. E. (1980). Flight-deck automation: Promises and problems. *Ergonomics, 23*(10), 995-1011.

Woods, D. D. (1985). Cognitive technologies: The design of joint human-machine cognitive systems. *AI Magazine, 6*(4), 86.

Woods, D. D., & Hollnagel, E. (2006). *Joint cognitive systems: Patterns in cognitive systems engineering*: CRC.

Zellmer-Bruhn, M. E. (2003). Interruptive events and team knowledge acquisition. *Management Science*, 514-528.