ANALYSIS AND SYNTHESIS OF ALLOCATIONS OF AUTHORITY AND RESPONSIBILITY IN NOVEL AIR TRAFFIC CONCEPTS OF OPERATION

A Thesis
Presented to
The Academic Faculty

by

Raunak Pushpak Bhattacharyya

In Partial Fulfillment
of the Requirements for the Degree
Master of Science in the
School of Aerospace Engineering

Georgia Institute of Technology
August 2016

ANALYSIS AND SYNTHESIS OF ALLOCATIONS OF AUTHORITY AND RESPONSIBILITY IN NOVEL AIR TRAFFIC CONCEPTS OF OPERATION

Approved by:

Dr. Amy Pritchett, Advisor
School of Aerospace Engineering
Georgia Institute of Technology

Dr. Magnus Egerstedt
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. John-Paul Clarke
School of Aerospace Engineering
Georgia Institute of Technology

To my mother and father

ACKNOWLEDGEMENTS

First, I would like to thank my advisor, Dr. Amy Pritchett, for giving me the opportunity to join the Cognitive Engineering Center. Her guidance, expertise and knowledge have been an inspiration to me. I would like to thank her for challenging me to be independent in my research and learn to be the master of my own path.

Next, I would like to thank my committee members. I would like to thank Dr. Magnus Egerstedt for introducing me to the wonderful world of networks and Dr. John-Paul Clarke for helping me come up new ideas through his insightful feedback.

I would like to thank Dr. Karen Feigh and all my CEC labmates for their constant support and team spirit. I would like to thank Martijn for his brilliant ideas. A special thanks also to Sebastien for all the insightful discussions. Thanks to Anil for being my go-to person in the lab. A big thank you to Lee for his constant encouragement and cheering up. I would also thank the ISYE IT department for software support and maintenance.

My friends here and back home have been crucial in helping me stay positive. I would like to thank my room mates Mayank, Neil and Achyut for helping me to keep going. A special thanks to Tejas for being an inspirational motivator and to Pushkar for all the discussions on Graph Theory. Thanks to Kaustubh and Adittya back home for all the skype pep talks. A big thank you to my Tabla guruji for keeping my musical side alive.

Finally, whatever I am today is because of my mother and father. They are my foundation and pillars of support. I would like to thank my mother for her contagious positivity under every circumstance and my father for all his priceless advice, for being my life guru, and for always urging me to aim higher and strive for excellence.

This work was supported by the NASA Aviation Safety Program with Dr. Guillaume Brat serving as Technical Monitor under grant number NNX13AB71A S04.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

The Next Generation Air Transportation System (NextGen) in the US and the Single European Sky Air Traffic Management (ATM) Research (SESAR) program in Europe are redefining ATM, allowing for transformative new concepts of operation that may radically re-allocate authority and responsibility between air and ground. There is a need for methods that can systematically incorporate innovative allocations of authority and responsibility in the design of novel concepts of operations to enable them to meet their specified performance and safety goals.

This need translates to two objectives: 1) Create the methodology and tools for analysis of allocation of authority and responsibility in novel air traffic concepts of operation, and 2) Create the methodology and tools for synthesis of allocation of authority and responsibility in novel air traffic concepts of operation.

This thesis first establishes concrete definitions of capability, authority and responsibility in the context of function allocations in the design of concepts of operations. Then, it addresses the first objective by proposing a computational modeling and simulation methodology to assess allocations of authority and responsibility with respect to the performance and safety goals of the concept of operations. Subsequently, it addresses the second objective by proposing a methodology based on network modeling and optimization to systematically synthesize allocations of authority under specified allocations of responsibility to meet performance and safety goals. The proposed methodologies are demonstrated on a case study designing of allocations of authority and responsibility in aircraft merging and spacing operations during arrival.

The methodologies described and demonstrated in this thesis can be used by designers of concept of operations to both analyze and synthesize allocations of authority and responsibility. Further, the results of the case study can inform the design of similar concepts of operations.

CHAPTER 1 - INTRODUCTION

1.1 Motivation

The Next Generation Air Transportation System (NextGen) in the US and the Single European Sky Air Traffic Management (ATM) Research (SESAR) program in Europe are redefining ATM. Increasing reliance on software, the use of new technologies, and higher inter-connectivity and increased coupling among airborne, ground, and satellite systems will allow for transformative new concepts of operation that may radically re-allocate authority and responsibility between air and ground.

Historically, technological constraints drove the design of air traffic concepts of operation, including the allocation of authority and responsibility between agents. For example, the attributes of ground-based radar and voice radio communications strongly defined the role of the air traffic controller, what clearances or instructions s/he had the authority to issue, and when and to what purpose they should be given. Now, innovative concepts of operation are possible that may be based on constructs that are substantially different from current-day airways, flows and sectors. For example, the concepts of free flight or distributed/decentralized decision making envision aircraft being allowed to determine their own trajectory, typically with commensurate authority for separation assurance and/or spacing within the traffic flow [1]. Other concepts envision a strong centralized role in determining trajectories, such as providing scheduled or required times of arrival at specific points or specifying an entire trajectory that is planned to be conflict free and to maximize throughput [2].

With technology no longer the constraining factor, the design of these innovative concepts, and specifically innovative allocations of authority and responsibility, should be driven by system-wide safety and performance requirements, and by agent-level concerns such as taskload and requirements for information transfer. These transformative concepts of operation can have significant impact on the agents themselves (e.g. their taskload), on requirements for information transfer between agents (by whatever communication mechanisms), and potentially on system performance. For instance, new concepts of operation that allocate in-trail spacing functions to the flight deck may significantly reduce the task load of the air traffic controller (which may or may not be beneficial), while increasing both the task load assigned to aircraft agents (notably, the flight crew) and the requirement for information transfer between the controller and the aircraft about spacing. Ultimately, these measures tie into system-wide metrics such as the distribution of taskload and information transfer between agents, among others.

However, current methods to design new concepts of operation rely heavily on subject matter experts (SMEs) who, in turn, rely on heuristics, experience, or rules of thumb. This process has several shortcomings. First, if the concept of operations is truly innovative and transformative, the expertise of SMEs will not extrapolate well to its new characteristics and concerns. Second, air traffic systems are complex by any measure, and thus need design processes with evaluation methods that scale to their scope and detail. Third, many of the interesting effects – and metrics – are *emergent*, i.e., they arise at a different level of abstraction than the behavior that creates them. Finally, testing methods typically rely on Human-In-The-Loop (HITL) simulations which, while being the appropriate final test before implementation, occur too late in the design cycle for easy testing of key issues that may require significant changes to the entire concept of operation or to supporting technologies.

Thus, there is a need for methods that can systematically incorporate innovative allocations of authority and responsibility in the design of novel concepts of operations to enable the concepts of operations to meet their specified performance and safety goals. This thesis proposes methodologies comprising of computational modeling and simulation, and network science and optimization, to systematically analyze and synthesize the allocation of authority and responsibility in the design of novel concepts of operation.

1.2 Definitions

This thesis defines *concept of operation* as the specification of how an operation is to be carried out, framed from the viewpoint of the actors who will execute it. Within a concept of operations, this thesis defines an *action* as an activity that is executed at a single point(s) in time by a single agent. An air traffic concept of operation, by these definitions, identifies what actions will be carried out to create a desired outcome, and by whom; it is sufficiently specific to locate the conditions where each action needs to be carried out, and criteria for successful completion of each action.

The design decision that determines which agent performs what action is called *allocation*. When allocating actions to agents, a concept of operation should unambiguously specify which agents have capability, authority and responsibility for which action. The *capability* of an agent to perform a particular action is defined here as the ability of that particular agent to perform that particular action. For example, "agent A has capability for action B" is interpreted as "agent A can perform action B". In the more general sense, capability can be viewed as a knob instead of a switch, with different agents having varying degrees of capability for performing the same action, or the same agent having varying degrees of capability for different actions, or the same agent having varying degrees of capability for the same action with varying contexts (e.g. lack of information may render an agent incapable of performing an action).

In this thesis, the agent assigned to perform a certain action is said to have been allocated *authority* for that particular action. This has been recognized in basic requirements for allocations of authority, notably that: (1) the allocation of authority be within each agent's capability, i.e., each agent should be capable of each action it is given authority for; and (2) each agent should also be capable of the collective set of actions it is allocated, i.e., the set should reflect a reasonable task load at all times [3]. Particularly when examining allocations to human agents such as air traffic controllers and pilots, this taskload needs to fit within both upper limits representing saturation, and lower limits representing boredom and lack of task engagement; further, the task load needs to fit within these limits not only in the aggregate, but also at all points in time to prevent concerns such as workload spikes. Some machine agents may also have constraints, particularly when they are sensitive to real-time computing limits.

This thesis defines *responsibility* as a designation that is typically a legal or policy matter that assigns accountability to an agent for an action's outcome. An *authority-responsibility mismatch* is created when the agent with responsibility for an action's outcome does not have authority to (nominally) execute the action. Such a mismatch may be purposefully created to enhance safety: it represents a redundancy that provides monitoring, oversight and error checking within the operation. However, whether purposefully created or not, such mismatches also require more work by the agents with responsibility. This added work is generally referred to as "monitoring work" throughout this thesis, although it may also demand intervening and (re-)doing the actions of others. This monitoring has several relevant attributes. First, while such monitoring is targeted at rare situations where another agent fails, the time of these situations is not known *a priori* and thus the monitoring work needs to occur continuously (at least at some level). Second, to be effective, the monitoring needs to have access to the same, or at least equivalent, information, often requiring significant information transfer between agents. Third, such passive monitoring is not generally considered to be a strength of the human [3], and the decision to intervene may be difficult and often time-critical. Thus, the monitoring work implicit in the relative assignments of authority and responsibility should be explicitly identified early in design.

Capability, authority and responsibility (AA&R) can be viewed as a mapping from the set of agents to the set of actions. Figure 1.1 gives an example of allocating AA&R in a scenario involving 3 actions and 2 agents. Agent and action boxes having the same outline style (dashed or solid) represent capability. Thus, in the figure, agent 1 has capability for action 1 and agent 2 has capability for action 3. Action 2, on the other hand, has both dashed and solid outlines to show that that both agents 1 and 2 have capability for action 2. Thus, in a potential concept of operations involving these actions and agents, agent 1 can be assigned authority for action 1 and agent 2 can be assigned authority for action 3. Further, either agent 1 or agent 2 could be assigned authority for action 2.

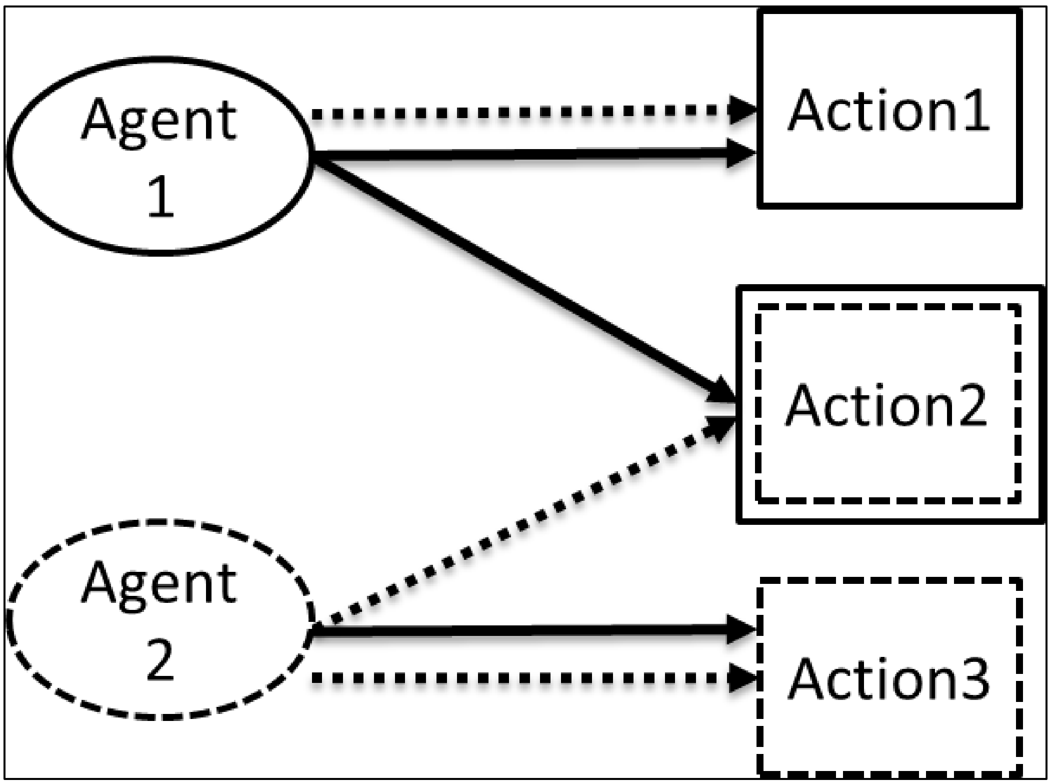

Figure 1.1: Allocation of capability, authority and responsibility. The solid arrows show the allocation of authority and dotted arrows show the allocation of responsibility

The allocation of authority is represented by the solid arrows in the figure. Thus, agent 1 has authority for actions 1 and 2, which means that agent 1 will perform these actions in the concept of operations. Similarly, agent 2 has authority for action 3 which means that action 3 will be performed by agent 2 in the concept of operations.

The allocation of responsibility is represented by the dotted arrows in the figure. Thus, agent 1 has responsibility for action 1 and agent 2 has responsibility for actions 2 and 3. Note that an authority-responsibility mismatch is created in the case of action 2 as agent 1 has authority for performing it but agent 2 has responsibility for its outcome. Thus, the legal implication of the outcome of action 2 falls on agent 2 which creates a requirement for agent 2 to monitor the performance of action 2.

Finally, it should be noted that the allocation of authority and responsibility may be fixed or dynamic. Key examples of current-day dynamic allocation of actions to agents include the changes in pilots' authority and responsibility inherent with the triggering of a Traffic Collision Avoidance System (TCAS) Resolution Advisory, and the changes in both pilots' authority and responsibility when they accept a clearance for a visual approach. Future aviation concepts of operations may dramatically increase the number of potential function allocations of authority and responsibility, and triggers for dynamic re-allocation.

1.3 Objectives

The goal of this thesis is to address the need for methods that can systematically incorporate innovative allocations of authority and responsibility in the design of novel concepts of operations to enable the concepts of operations to meet their specified performance and safety goals. This goal translates to the following two objectives:

1. Create the methodology and tools for *analysis* of allocation of authority and responsibility in the design of novel air traffic concepts of operation; and
2. Create the methodology and tools for *synthesis* of allocation of authority and responsibility in the design of novel air traffic concepts of operation.

1.4 Overview of Thesis

This thesis proposes methodologies comprising computational modeling and simulation, and network science, to systematically analyze and synthesize the allocation of authority and responsibility in the design of novel concepts of operation. Whereas computational modeling and simulation quantifies emergent metrics of the concept of operation, thereby providing the capability to *analyze* the impact of allocation, the representation of the concept of operation as a network allows the application of network theoretic techniques to *synthesize* allocations to meet the desired performance and safety goals of the concept of operations.

Chapter 2 provides a background on function allocation from two perspectives: 1) air-ground function allocation and 2) human-automation function allocation. A review of the requirements that any function allocation should satisfy, and the metrics for quantitatively evaluating function allocations, is also provided in this chapter.

Chapter 3 and 4 describe methodologies for analysis and synthesis of allocations to meet the performance and safety goals of the concept of operations. Chapter 3 describes a methodology for analyzing function allocations using computational modeling and simulation to capture and quantify the emergent metrics of the concept of operations. Chapter 4 describes a methodology to synthesize function allocations by network modeling and subsequently network optimization to find clusters of authority and responsibility. While computer simulation identifies and quantifies the emergent properties of a concept of operation, the network representation provides a structure to identify and exploit inherent properties in the work underlying the concept. Taken together these two tools can work in tandem for the allocation of authority and responsibility with the computer simulation parametrizing the network representation and the network representation abstracting the emergence obtained from computer simulation.

Finally, Chapter 5 summarizes the thesis, notes its contributions to allocation design for air traffic concepts of operations, and discusses potential future research and further application of the proposed methodologies to the design of allocations.

CHAPTER 2 - BACKGROUND

Function allocation is the design decision that defines which agents – human or automated and air or ground – have authority and responsibility to perform the actions required for the system to operate, and is a crucial factor in determining concepts of operation. This chapter reviews the function allocation literature from both the air-ground and the human-automation perspective.

2.1 Air-Ground Function Allocation

Separation assurance and spacing are key functions within ATM in which the problem of air-ground allocation has been examined. The primary separation assurance and spacing functions are: predict/detect the conflict, determine a trajectory modification to safely and efficiently resolve the conflict, and execute the trajectory modification [4]. To safely accommodate significantly higher traffic densities in the future, substantial changes are being proposed in the allocation of functions to air and ground. There is a large body of literature reporting investigations of such future concepts using computer based simulations, HITL simulations, and operational evaluations.

The literature reporting concepts of operations for separation and spacing can be broadly categorized into the following categories: Air-Ground Separation Assurance and Spacing with Limited Delegation, Air-Ground Separation Assurance and Spacing with Mixed Air-Ground Separation Responsibility, Ground-Based Separation Assurance and Spacing, and Airborne Separation Assurance and Spacing. The following sub-sections summarize the research from these categories relevant to methods for function allocation.

2.1.1 Separation Assurance and Spacing with Limited Delegation

Among future configurations of air-ground function allocation, the most well studied has been shared air-ground control where the Air Navigation Service Provider (ANSP) delegates specific separation or spacing functions to the cockpit crew on a limited (per-event) basis. The flight deck equipage generally consists of a Cockpit Display of Traffic Information (CDTI) with an ADS-B data feed of local traffic. An HITL simulation investigated the concept of limited delegation of separation assurance to the aircraft for crossing and passing in en-route airspace, and sequencing in terminal areas [5]. Qualitative evaluations gathered from pilots and controllers indicated the feedback for the proposed method to be "promising with great potential". However, when applied under certain conditions, the method resulted in increased workload and communication requirements for both pilots and controllers.

Other HITL simulations have studied airborne spacing for arrival traffic [6]. Results showed a significant reduction in the variance of inter-arrival spacing at the metering fix. The overall feedback from controllers was positive with eye-fixations and analysis showing positive impact on controller activity in terms of relief from late vectoring. Other HITL studies have evaluated the CDTI assisted visual separation (CAVS) concept where the CDTI is used to maintain situational awareness when out-the-window visual contact with a target aircraft is temporarily lost. The 'traffic to follow' is designated by the ANSP and the cockpit crew use the CDTI and their own judgements to achieve self-determined spacing [7], [8]. The concept was found to be technically and operationally feasible based on the feedback from pilots and controllers in the HITL studies. However, to conduct these operations, means would have to be found to enable pilots to fly their wake mitigation techniques of choice while still complying with the airline's requirement to fly stabilized approaches.

Further, time based airborne merging/spacing operations on FMS arrival routes have also been evaluated using HITL simulations [9] which indicate that airborne spacing improves accuracy and is feasible for FMS operations and mixed spacing equipage. Several studies have evaluated the Flight-Deck based en-route Merging and Spacing (FDMS) concept using HITL simulations [10][11][12]. Results indicate that FDMS reduces the number of controller-issued maneuvers, communications and workload. A limited implementation of FDMS has been certified and is currently in use for United Parcel Service (UPS) revenue flights. An HITL study at Eurocontrol examined controller activity with and without the allocation to the flight deck of merging and in-trail spacing functions [6]. This study found that the allocation of these functions to the flight deck not only reduced the number of communications that the controllers had to initiate to the aircraft – it also changed when these communications were made. When juxtaposed next to limitations of communications systems such as latency and drop-outs, these communication requirements may exceed what is feasible.

Likewise, other HITL studies compared air-ground allocation of the separation assurance and spacing tasks. For example, two HITL studies at NASA, one controller-focused and one flight deck-focused, compared the effects of mixed-equipage in delegating separation functions to some aircraft [13]. Similarly, under the Advanced Air Transportation Technologies (AATT) program, a study at NASA Ames investigated the performance of Distributed Air Ground Traffic Management (DAG-TM) [14]. Two competing en-route concept elements examined delegating the separation assurance task to the flight deck or leaving it on the ground with trajectory based operations. Initial results showed a benefit of moving some functions to the air in terms of flight efficiency, notably flight time. Similar results were found by NLR in an HITL study, part of the European INTENT project [15]. This project investigated the level of intent information requirements due to a different allocation but also found differences in terms of airspace capacity. These studies represented substantial research efforts that simultaneously examined both different allocations and how automation and algorithm design (e.g., the conflict detection look-ahead) might vary with different allocations. A flight test of an A340 in Reykjavic ATC south sector investigated the in-trail procedure through an operational evaluation [16]. The results show both operational and technical feasibility and show benefits in terms of reduced fuel burn and emissions.

Concepts involving automation of spacing functions have also been investigated. HITL simulations of the airborne self-spacing concept have been studied with positive evaluations from pilots and controllers [14] [17]. Results indicate that it is possible to conduct continuous descent arrivals in high-density airspace. Operational evaluations have also been performed to investigate highly automated spacing tools [18][19]. The results were positive with pass-behind and merge-behind maneuvers being successfully repeated.

2.1.2 Separation Assurance and Spacing with Mixed Air-Ground Responsibility

In contrast to air-ground control with the ANSP delegating functions to the flight crew on a limited number of cases, there have been studies on concepts where some aircraft are under ground-based control whereas others have limited or continuous delegation for separation assurance and/or spacing. HITL simulations have been performed to investigate the concept feasibility of shared air-ground separation responsibility [20][21][22]. Results indicate that, while safety was not compromised, pilots and controllers tended to have differing opinions about the operational suitability of the proposed concept. While the pilots found the concept generally favorable, controllers reported higher workload and expressed safety concerns, which was demonstrated by their cancellations of the delegation of authority to the aircraft. Controllers also preferred to resolve conflicts earlier and tended to cancel delegation of authority when they perceived pilots were delaying the conflict resolution. The authors recommended more research to define optimal roles for flight crews and controllers in separation assurance.

HITL simulations of an integrated air/ground system in arrival sectors and terminal airspace have also been conducted [23][24][25]. While the pilots deemed the concept to be acceptable, controllers raised several safety concerns which mostly pertained to the occurrence of near-term conflicts between autonomous and managed aircraft. These results showed that mixed operations might be feasible in the same airspace if the unequipped aircraft count is held to a workable level; this level will decrease with increasing complexity of the airspace, implying that an integrated airspace configuration is feasible up to a limit.

2.1.3 Ground-Based Separation Assurance and Spacing

In ground-based control, the ANSP predicts conflicts and sends resolution clearances to cockpit crew for execution. A key feature of future ground-based separation assurance concepts is trajectory exchange via datalink. Clearances are delivered using the Controller-Pilot Data Link Communications (CPDLC). Procedures for sending automation-generated trajectories to cockpit FMS for execution have been demonstrated on a Boeing 737 Level-D flight simulator hardware [26]. HITL and computer based evaluations have shown that the ground-based automation can accommodate high-altitude en route traffic with a very high rate of conflict resolution [27][28].

Concepts that automate more functions have been described in the Automated Airspace Concept (AAC) [29] where a ground based component, the Advanced Airspace Computer System (AACS), generates trajectories and directly sends them to the aircraft via datalink. Another ground-based component, the Tactical Separation Assisted Flight Environment (TSAFE), would provide safety in the event of failures in the AACS or on-board systems. Computer based analyses have been conducted to evaluate the AACS [30][31]. Results state that, if the automation were to fail, the collision rate during the subsequent airspace evacuation would be under 1 per million hours.

Many issues were identified with off-nominal short-term conflict detection and resolution. An operational evaluation of Boeing 777 demonstrated datalink-enabled conflict free continuous descent approaches into San Francisco International Airport [32]. The automation was able to predict meter-fix arrival times to within a mean accuracy of 3 seconds over a 25-minute prediction horizon. While these studies establish the performance of automation in providing separation assurance, there is a need for evaluations under high-density traffic in arrival sectors with and without metering constraints.

2.1.4 Airborne Separation Assurance and Spacing

Several HITL studies have been performed to investigate the impact of the Airborne Separation Assurance System (ASAS) [11][33]. Results showed that pilot workload remained well within tolerable limits and pilots positively accepted ASAS. However, an operational evaluation in the vicinity of the Frankfurt International airport concluded that, for flight under Visual Flight Rules (VFR), it is not possible to replace the visual scan with the use of a traffic presentation onboard as it introduces new risks [34].

More complex airborne automation has been tested for situations involving short-notice conflicts, highly constrained operations due to Special Use Airspace (SUA) or weather, transitions to/from free-flight airspace, lack of intent information, uncertainties/delays in data, and pilot delays in interaction with the ASAS. Several HITL studies have been performed with evaluation goals including intent information [35], near-term conflicts [36], strategic vs tactical resolutions provided by automation [37], priority flight rules to maintain safety in potentially hazardous situations [38] and transitions between Managed and Free Flight Airspace [39]. In general, the results were positive and appeared to be well accepted by the participating pilots.

More recently, computer-based Monte-Carlo simulation experiments have been performed to assess and quantify the safety behavior of airborne separation [40][41][42]. The consensus reached by these studies seems to be that ASAS can be very effective in low density en route airspace. However, off-nominal scenarios, especially in high density airspace, had substantial safety concerns. The results indicated a potential for clogging of conflicts i.e. simultaneous conflict situations occur and then such a cluster of conflicts tends to grow faster in size than the conflict resolution can handle. Further, the clogging of conflicts is an emergent behavior that is difficult to observe and analyze with established approaches.

The literature review found two HITL simulations specifically designed to formally compare traffic operations with ground-based and airborne separation assurance. [43]

conducted a pair of coordinated simulations to gain insight on the comparability of the two function allocations for separation assurance. No substantial differences in performance or operator acceptability were found but certain conflict situations were detected too late to be resolved in a timely manner. [44] performed HITL simulations to evaluate pilot performance, workload and situational awareness under concepts of operation that differed with the allocation of separation responsibility across human pilots to ATC. The findings suggested that keeping pilots engaged in separation assurance tasks may be preferable to having them rely on automation alone.

2.1.5 Summary of Air-Ground Allocations to Date

It has been recommended in the literature [4] that more studies objectively evaluate the relative strengths and weaknesses of proposed operational concepts with varying air-ground function allocations. Some studies have analyzed the allocation of interval management between air and ground, particularly in HITL simulations [45]. A computational study of tactical conflict resolution in an en-route free flight environment similarly assessed the impact of varying the ‘locus of control’ in conflict detection and resolution functions from being solely allocated to the ground controller to an increasing proportion of the aircraft (up to 100%) [46]. While the results were specific to conflict detection and resolution, this study emphasized the benefits of computational modeling as a cost and time-effective form of analysis, and modeled the actions as being completed the same by all the agents so that any observed effects could be isolated to the function allocation without confounds. Similar computational studies have demonstrated the impact of allocating functions between air and ground in arrival merging and spacing operations in terms of taskload, information transfer requirements and monitoring [47][48][49][50]. Thus, for the purpose of analyzing air-ground allocations, computational simulation appears to have promise.

2.2 Human-Automation Function Allocation

The literature reflects a wide range of opinions on human-automation function allocation. In fact, whether human-automation function allocation should be considered science, engineering or a black art has been debated [51][52]. There are three main perspectives for the human-automation function allocation problem: 1) the man vs. machine capabilities perspective, 2) the team perspective, and 3) the work perspective. The following sections first describe these three perspectives and subsequently describe the overarching requirements and metrics for modeling and measuring human-automation function allocation.

2.2.1 The Man vs. Machine Capabilities Perspective

The most common perspective on allocating functions to humans and automation recommends the comparison strategy that compares each function to the individual capabilities of all the agents and assigns it to the most capable [53]. Such a strategy is fostered by the Fitts list, which proposes capabilities that men are good at versus those that machines are good at [54].

Implicit to function allocation focused on the machine's capabilities is the leftover allocation strategy [53]: Automate as many functions as technology will permit, and let the human pick up the rest. This allocation strategy has given rise to categorizing allocation using levels of automation [55][56]. However, with the focus on automation, the allocation of functions to the humans has often been haphazard and ill-suited to the capabilities of the human, leading to situations where, at one extreme, the human has consistently overridden automation and, at the other extreme, has been required to act as an unthinking actuator for an automation that alerts or suggests decisions [57][58]. Likewise, this strategy makes for brittle automation that can fail in an unexpected manner or provides little support in off-nominal conditions, which is generally when automated support for the human operator is needed the most [59].

To avoid situations where automation is put outside its boundary conditions, the role of the human in the leftover allocation strategy often involves a significant amount monitoring in addition to performing the left over functions: either monitoring the automation or monitoring the conditions or both [60][61]. However, consistent findings have shown that humans are not effective at being monitors [62][63]. This further creates concerns with low task engagement leading to vigilance and boredom issues that ultimately can culminate in the human being 'out of the loop' [60][64].

Literature supporting the leftover allocation strategy has noted the capability of automation to reduce average workload of human operators, often by reducing the manual control or execution functions that the humans must perform. This has been touted as a contribution of automation and, in aviation, has led to reduced staffing requirements including reduction of the number of flight crew members in the flight deck from three to two. However, there have been many incidents where automation has created workload spikes for a human team member [55][60][65]. For example, highly automated autoflight systems can suddenly require significant programming from the human flight crew in response to commanded reroutes by ATC, while the flight crew must simultaneously respond to the controller and in addition, potentially execute other tasks such as finding charts for the new route.

2.2.2 The Team Perspective

The allocation of taskwork functions within any team creates the need for additional teamwork functions that go toward coordination of the taskwork. These include the coordination of taskwork timing so that the team activities can be synchronized after a period of taskwork activities in parallel. In the context of human-automation function allocation, a team perspective includes considering the automation as a team member [66][67][68][69][70]. When automation is viewed as a team member, human-automation interaction becomes crucial. However, too often automation is clumsy: unduly interrupting its human team members because unlike humans, automation cannot implicitly sense information about whether other team members would benefit from an interruption [71]. Therefore, predefined sets of function allocations may serve as more explicit coordination strategies in the case of human-automated teams, such as the function allocations that the pilot of a current day aircraft may invoke, or the playbook metaphor proposed by [72].

Authority and responsibility do not always need to be aligned. Authority-responsibility mismatches, as first identified by Woods, occur whenever one agent is authorized to execute a task, but a different agent is responsible for the outcome [69]. As a result of the mismatch, the responsible agent needs to get information about the task outcome (and perhaps performance), monitor the authorized agent, and perhaps intervene. Thus, when the allocation generates authority-responsibility mismatches, it also implicitly creates additional information transfer and monitoring-taskload beyond that visible when only the authority allocation is examined.

2.2.3 The Work Perspective

Work is an ongoing response to, and action upon, the work environment. The physical environment has dynamics that drive the taskwork and at the same time, teamwork modifies the environment of agents, creating a dynamic interplay. For example, when the pilot is flying via the control column while the auto throttle is on, the pilot controls the elevator and the auto throttle controls the throttle setting. However, pitch and speed are intrinsically coupled leading to, for example, the pilot having to compensate for a change in pitch due to a change in speed made by the auto throttle. Thus, dynamic analysis is required to identify situations where interleaving of functions results in significant coordination requirements between agents, idling as one agent waits on another, or when one agent may be interrupting another. In aviation, function execution is dictated by established procedures which serve as an intrinsic dynamic within the work environment and mirror their structures, thereby guaranteeing consistency and safety [73]. Therefore, function allocation should support how the dynamics of the work environment are managed by the agents.

Further, in a *coherent* allocation, many functions naturally go together in terms of the information they act upon and the actions they take. Thus, they can benefit from the same information sources and, when conducted together by the same agent, can be timed and executed synergistically: this would form a coherent allocation. Conversely, incoherent function allocations can require different agents to interleave their activities, each waiting upon the other, to perform related actions.

2.3 Requirements and Metrics for Human-Automation Allocations

The construct of function allocation is not simple and the problem of designing teams remains an unsolved challenge. Further, effective allocations cannot be simply created according to one design principle or objective. Instead, each allocation must simultaneously meet the following requirements [74]:

Requirement 1: Each agent must be allocated functions that it is capable of performing.

Requirement 2: Each agent must be capable of performing its collective set of functions.

Requirement 3: The function allocation must be realizable with reasonable teamwork.

Requirement 4: The function allocation must support the dynamics of the work.

Requirement 5: The function allocation should be the result of deliberate design decisions.

[75] and [76] identified several metrics to evaluate function allocations relative to these requirements. These metrics may be identified from detailed models of work, from computational fast-time simulations, and from assessments in simulated or actual operations.

The collective *taskwork* of a team spans the actions required to achieve work goals and the allocation of taskwork actions then creates the need for additional *teamwork* actions to coordinate between agents, the subject of Requirement 3. The metric for assessing the workload imposed upon the agent due to the taskwork and teamwork actions is taskload.

Further, a metric for assessing the coordination requirement imposed is information transfer.

The collective set of taskwork and teamwork must be examined together within the model to evaluate Requirements 1 and 2. This includes assessing authority-responsibility mismatches as another possible metric for function allocations for requirement 3. Likewise, to address requirement 4, function allocations can be assessed using the metric of coherence.

Building on the aforementioned metrics, this thesis applies the metrics of taskload and information transfer, and examines a qualitative estimate of coherence. The total taskload demanded of an agent given a specific allocation is the sum total of taskwork contributing to mission performance, teamwork due to interaction with other agents, and teamwork due to monitoring demands. The resulting taskload on every agent due to performing the actions is important so that workload spikes or longer durations of workload saturation can be identified. Workload is a variable that indicates the relationship between the crude measure of taskload and the capability of the agent executing the actions [77]. Moreover, concerns with vigilance, complacency, and low engagement can be assessed using the metric of taskload. Note that the taskload can be an emergent construct created by the evolving demands on the team, and thus its extent can only be predicted through some form of simulation.

Information transfer captures the teamwork aspect of the work required to be performed to meet the goals of the concept of operation. Different allocations will also establish different requirements for information transfer within a concept of operations, particularly in systems such as airspace that involve mixed-initiative systems and substantial contributions by human operators. At one extreme, if only one agent can perform all the work, then s/he/it never needs to transfer information to, and receive information from, others because it 'owns' all the information in the system. An allocation divides up the taskwork between agents, and the resulting need to coordinate activities creates the need for information transfer. Whereas concerns with efficient communication require information transfer to be as low as possible, safety demands that more information be exchanged to foster redundancy and error checking. When information transfer will be required, and how often, is also an emergent construct in that it arises in response to the evolving dynamics of the operation to a degree that cannot be predicted with confidence a priori.

2.4 Summary

In the light of increasing advances in the novelty and the complexity of aviation concepts of operations, the air-ground allocation literature has called for more studies to systematically evaluate the relative strengths and weaknesses of proposed operational concepts with varying air-ground function allocation. Thus, there is a need to develop methodologies that can analyze the impact of allocating authority and responsibility for functions in aviation concepts of operation.

Further, as there is little to no literature dealing with the synthesis of air-ground function allocation, the design of these allocations still largely remains a subjective process devoid of systematic mapping of allocations to the performance and safety goals of concepts of operations. Therefore, methodologies that can capture these system level goals and use them to synthesize allocations of authority and responsibility are needed by the air-ground function allocation community.

The requirements and metrics identified from the literature can inform the methodologies for the analysis and synthesis of air-ground allocations. These methodologies could the allocation of authority and responsibility in two different ways for analysis and synthesis, and subsequently use metrics to assess the impact of a given allocation, or guide the synthesis process.

CHAPTER 3 - ANALYZING ALLOCATIONS OF AUTHORITY AND RESPONSIBILITY

This chapter proposes a methodology for the quantitative analysis of allocations of authority and responsibility in novel concepts of operation using computational modeling and simulation. Such simulation has the advantage over current-day subjective analysis by subject matter experts (SMEs) in that it can look for effects that are hard to predict for several reasons. First, if the concept of operations is truly innovative and transformative, the expertise of SMEs will not extrapolate well to its new characteristics and concerns. Second, air traffic systems are complex by any measure, and thus benefit from computational methods that scale to their scope and detail. Third, many of the interesting effects – and metrics – are *emergent*, i.e., they arise at a different level of abstraction than the behavior that creates them.

The analysis of the impact of allocations of authority consists of two stages, documented next. First, the work of the concept of operations is modeled separately from the agent models. Second, the authority and responsibility for the work is allocated to the agent models during the run time of the simulation to assess metrics of the allocation.

3.1 General Methodology for Analyzing Allocations of Authority

This general methodology assesses metrics of a proposed allocation of authority and responsibility relative to the performance and safety goals of the concept of operations. Its goal is to guide designers of concepts of operations in their decision making concerning the allocation of authority and responsibility. Toward this goal, the assessment of metrics is intended to be a fast process such that the analysis time between proposal and assessment of novel concepts of operation is as short as possible.

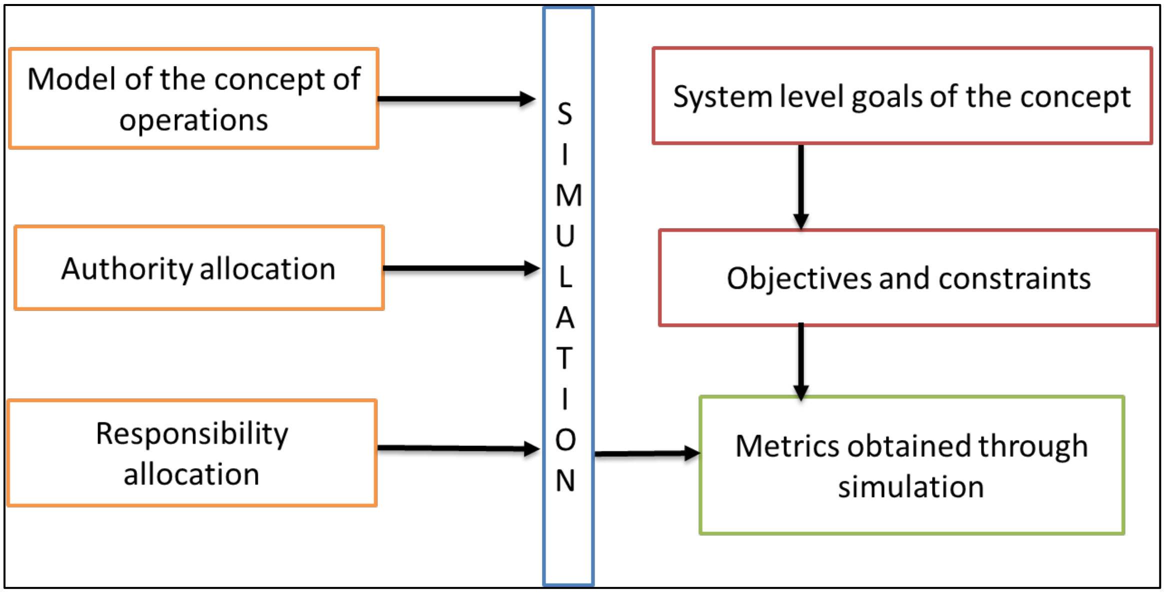

Figure 3.1: General methodology for analysis of allocations

Figure 3.1 gives an overview of the methodology in terms of its inputs and outputs. The inputs are the prototype allocations of authority and responsibility to be analyzed, and a model of the concept of operations. This model can represent the concept of operations at any level of detail as appropriate to the stage of design. For example, as detailed in the next section, this thesis demonstrates early-in-design analysis where broad function allocation decisions are being made. Therefore, this thesis models the concept of operations as a mid-level model of the actions required of the concept of operations. The action models are deemed sufficient if they represent their successful completion, rather than emulating in detail how a specific agent or technology would perform them. However, were this methodology to be applied in later phases of the concept of operations design process, more detailed representations of the actions can be included in the model. In this way, the assessment of the allocation of authority and responsibility can be refined as the specific phase of design warrants.

Similarly, for the early-in-design analysis examined in this thesis, the agents are modeled using base agent models that need only be capable of receiving an action to execute and of calling the method modeling its activity. In early analysis of concepts of operation, use of the base agent model has a significant benefit: because each action is executed the same way regardless of the agent performing it, any differences in resulting system-level metrics are due to the allocation itself. Subsequent design phases could apply progressively detailed performance models for the agents.

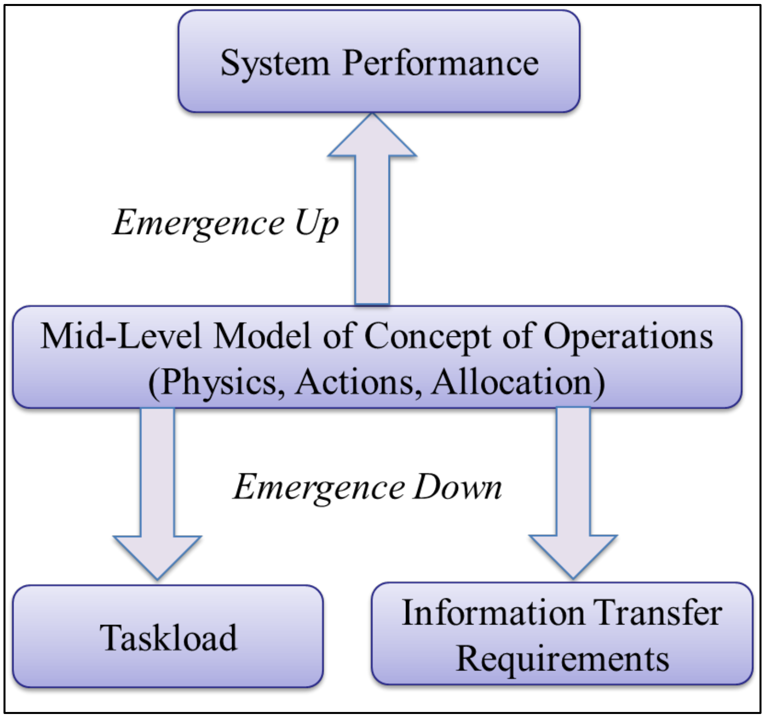

Figure 3.2: Mid-level model and emergence

These inputs are passed into a simulation engine that then outputs metrics. Within the simulation runs, some aspects of system performance emerge "up", such as traffic flows that can be assessed for their stability and consistent, high-throughput spacing. Other metrics emerge "down" when they examine the specific demands on, and needs of, each agent, as shown in Figure 3.2. For example, the designer of a concept of operations may know a priori that the concept of operations assigns each pilot the action of calculating and commanding an airspeed that maintains a given spacing within an arrival stream; it takes simulation of several interacting aircraft, however, to systematically identify when (and how often) each pilot will need to adjust airspeed, with associated metrics of the task load placed on each pilot and what information the pilots will need (and when).

Out of the many possible metrics that can be captured, the system level goals of the concept of operations should govern the specific metrics that should be obtained from the simulation to assess the impact of allocating authority and responsibility. The case study later in this chapter uses the metrics of taskload, and information transfer requirements.

Off-nominal events should start being examined early as well, particularly for foreseeable events that the concept of operations should be capable of handling. This implies thinking through, and capturing in a model, the actions expected in responding to these events. These actions may be added progressively such that, as soon as the first analysis verifies that the nominal operations do indeed proceed as planned, then an expanded range of off-nominal events (and increasingly detailed representations of their actions) can be examined.

The analysis methodology described in this section can serve two purposes. First, it informs the designer about emergent features of the concept of operation and the behaviors that arise within it, informing further refinements to the design. Second, to the degree that the metrics predict the performance of the concept of operations, they allow comparisons between various proposed allocations of authority and responsibility.

3.2 Modeling and Simulating Concepts of Operation

This section details the two stages of analyzing allocations: 1) modeling the concept of operations, and 2) simulating the model to extract metrics to assess the allocation.

3.2.1 Mid-Level Model of the Concept of Operations

This thesis demonstrates the utility of a mid-level model of concepts of operation, particularly for early-in-design analysis. A mid-level model emulates the performance specification for each action without attempting to describe the specific mechanisms any one particular agent would employ or more detailed factors affecting how it achieves the performance specification. For example, a mid-level model would consider the action "maintain commanded spacing" according to an idealized 2nd order model with a given damping and natural frequency. The input to this black box would be the state of the leader aircraft and the output would be a value for the airspeed that the trailing aircraft should set. This type of mid-level model is thus not specific to any particular agent and thus the metrics isolate design issues with the concept of operation itself without confounds due to possible issues in the performance of the actions.

In contrast, a detailed-level model would use an actual algorithm that takes into account factors such as the noise in the state of the leader aircraft and the latency in communication between the aircraft, and would also need to model those factors. Such detailed modeling would typically be agent specific and thus would be more appropriate once the allocations are fairly well established.

The mid-level model builds on what is known (or determined) early in the design of concepts of operation: the physics of flight, and the specifications for the actions that will be carried out. The physics of flight may be considered here as a constraint on what a concept of operations may expect of agents. Fortunately, the basic physical principles are fairly well known early in design, and a range of aircraft dynamic models is available (from simple point-mass models to full non-linear six degree of freedom). Performance differences between aircraft have also been extensively modeled such that aircraft dynamic models can be adjusted to represent a mix of aircraft types [78].

Reasonable emulations of autoflight systems also exist [79]. Where new control behaviors might be needed by the concept of operations, during the early phases of design and analysis they are often better emulated by mid-level models that represent their desired inputs and outputs. For example, in the case study presented later in this chapter, the calculation of airspeed to maintain an interval in a flow of aircraft does not need to account noisy, unreliable information and is emulated with a simple Proportional-Derivative (PD) regulator; while this emulation may be considered simplistic relative to a final design, at early stages of design it allows for comparison of different function allocations using an algorithm common to all agents.

Next, the design of a concept of operation specifies its actions. Here, an *action* is defined as an activity that is executed at a single point(s) in time by a single agent. Models of these actions identify their impact on the state of the air traffic system and/or changes they make in the timing of subsequent actions, and identify the conditions where each action needs to be carried out. At the most basic level, some actions are needed to define the trajectory of the aircraft for the autopilot/autoflight to track by defining waypoints and required/scheduled times of arrival and/or by defining more immediate steering commands for speed, heading, vertical speed and altitude, etc. Some of these actions also relate trajectory to the aircraft configuration, e.g. extend flaps as speed is reduced during the arrival.

The state of the world is captured by models of *resources*, with the specific values cast as appropriate to any of a wide range of data types. Actions are explicitly linked to resources via *get* and *set* relationships. With this representation, computational models of *actions* are specifically defined as containing a method defining which resources the action gets, the calculations it makes based on knowledge of the resource values, the resulting output that it sets in resources to update some aspect of the state of the world, and any other actions that they request be scheduled immediately. For example, in the case study described later in this chapter, the action "Command OPD Speed" gets the resources "Altitude" and "Distance to runway" and sets the resource "Commanded Airspeed".

Models of actions must also provide a method by which they declare the time they should next be scheduled to execute. This structure allows for simultaneous execution of heterogeneous model types with significantly varying bandwidth. For example, the actions updating the aircraft dynamic state have a sufficiently-small timestep that they approximate continuous dynamics (e.g., 0.01 seconds), the action "Calculate IM Airspeed" may need to iterate every second, and the "Start Descent" action need only be called once for each aircraft. This method of timing – only executing each action model when necessary – has two significant benefits: first, it is computationally efficient, and second, it ensures that the modeler of each action can define update rates representative of the action's dynamics without being constrained by mechanisms such as simulation-wide synchronous updates at small time steps [80].

The actions and resources inherent to a concept of operation collectively comprise *work models* that are instantiated outside of the agent models: during run-time, the authority to execute an action, and the responsibility for that action, can be fluidly assigned to any agent at that instant, such that a range of function allocations can quickly be tested by varying the script or algorithm driving the allocation. Multiple copies of work models can be instantiated at the same time; for example, in the case study that follows later in the chapter, three copies of the work models create a set of identical actions and resources for each aircraft that are then initialized and parameterized according to that aircraft's particular configuration.

Within this framework, then, the base agent models need only be capable of receiving an action to execute and of calling the method modeling its activity. More advanced agent models are available that can add some aspects of agent dynamics to the simulation, such as delaying or interrupting actions according to a model of task management under workload constraints [81]. However, in early analysis of concepts of operation, such as the comparisons of multiple different allocations of responsibility and authority being examined in this thesis, the base agent model has a significant benefit: because each action is executed the same way regardless of the agent performing it, any differences in resulting system-level metrics are due to the allocation itself in a fundamental way.

3.2.2 The Work Models that Compute Simulation Framework

Work Models That Compute (WMC) is a fast-time computational simulation tool written in C++ that can dynamically simulate the model of concepts of operation detailed in the previous section [82]. A simulation run is conducted by starting the simulation with reference to a script that indicates which work models to instantiate, and their initial conditions, to represent a desired concept of operations. This script can also create exogenous events to be triggered at any point during the simulation to initiate off-nominal events or reconfigure some aspect of the concept of operation.

Then, the simulation progresses by maintaining a list of the actions, sorted according to their desired next update time. The action at the top of the list is examined and the global simulation clock set to its update time. This action is then passed to the agent allocated with authority for its execution, who commands its method representing its activity. Once this is completed, the action declares its next update time and is sorted back into the list accordingly. The simulation then checks whether any termination criteria have been reached and, if not, proceeds to execute the next action.

To specifically allow for quick evaluation of different function allocations, some further extensions were added in this thesis to the simulation framework. First, during run-time, when an action is given to an agent to execute based on its authority allocation, its responsibility allocation is also checked. If a different agent has responsibility for the action, a *monitoring action* is automatically created and scheduled immediately after the current action. Authority and responsibility for this monitoring action are given to the agent responsible for the current action. This mechanism negates the need to explicitly develop and include all monitoring actions inherent to all function allocations to be analyzed: instead, the monitoring requirements emerge as the simulation progresses, and are automatically logged.

Likewise, a metric of emergent behavior of interest is the extent to which a function allocation will require information transfer between agents. For the purposes of large numbers of computational simulations comparing multiple allocations of authority and responsibility, this is assessed automatically: when one action tries to get a resource value, the simulation compares this action with the action that set that resource value. If these two actions are different, then it is logged as an instance where information would need to be transferred between the two actions. Further, if the agents that have been allocated authority for executing these actions are different, then this represents an instance where the activity of one agent results in information that would need to be transferred to another agent.

Finally, another metric assessed automatically by the computational simulation is the workload or taskload imposed upon each agent. Whenever an agent executes an action, the workload imposed by that action is added to the running tally of that particular agent's workload. The case study described next assumes that, at the early stage of design, the workload imposed by each action is not yet quantified: instead, every action imposes a unit taskload on the agent executing it and so these simulations assess the simpler metric of taskload, i.e. the number of action instances they are called to execute. More detailed design stages might choose to model actions as each imposing differing amounts of workload depending on their complexity and the capabilities of the agent executing them.

3.3 Case Study

This section demonstrates the methodology described in the previous section in a case study of merging and spacing operations in aircraft arrival. This case study examines three aircraft on arrival into runway 18R at Schiphol airport via two merging arrival routes. Figure 3.3 shows the arrival route for Runway 18R into Schiphol airport. Aircraft 1 arrives from the west on RIVER 2B transition and aircraft 2 and 3 arrive from the east on ARTIP 2C transition with aircraft 2 leading aircraft 3.

The aircraft enter the airspace along trajectories created for Optimal Profile Descent (OPD) but, as necessary to space the aircraft such that they ultimately cross the runway threshold with the proper wake vortex spacing, they are sequenced onto merging arrival paths and interval management (IM) is applied when a “following” aircraft must adjust its speed behind its “lead” aircraft for proper spacing. The aircraft configuration (in terms of flaps, speedbrake, and landing gear) is also managed to assist with the energy management of the aircraft and to ensure the aircraft was properly configured for management.

In the nominal scenario, aircraft 1 (arriving from the west) performs OPD until EH 608. It then merges into the final approach fix between aircraft 2 and 3 and performs IM in-trail behind aircraft 2. Aircraft 2 performs OPD throughout the approach. Aircraft 3 performs IM in-trail behind aircraft 2 until aircraft 1 merges into the final approach fix after which aircraft 3 switches to IM with aircraft 1 as its leader aircraft. The altitude and airspeed profiles of the three aircraft are presented in Figure 3.4.

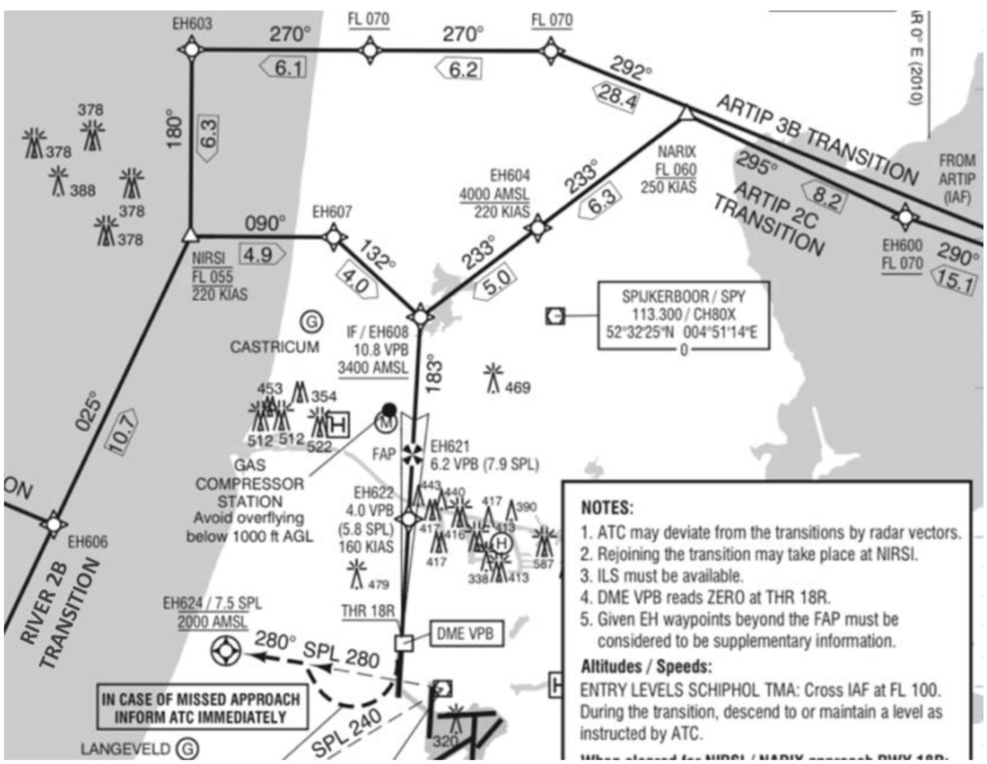

Figure 3.3: Arrival routes into Schiphol airport RWY 18R (*from Jeppesen EHAM charts)*

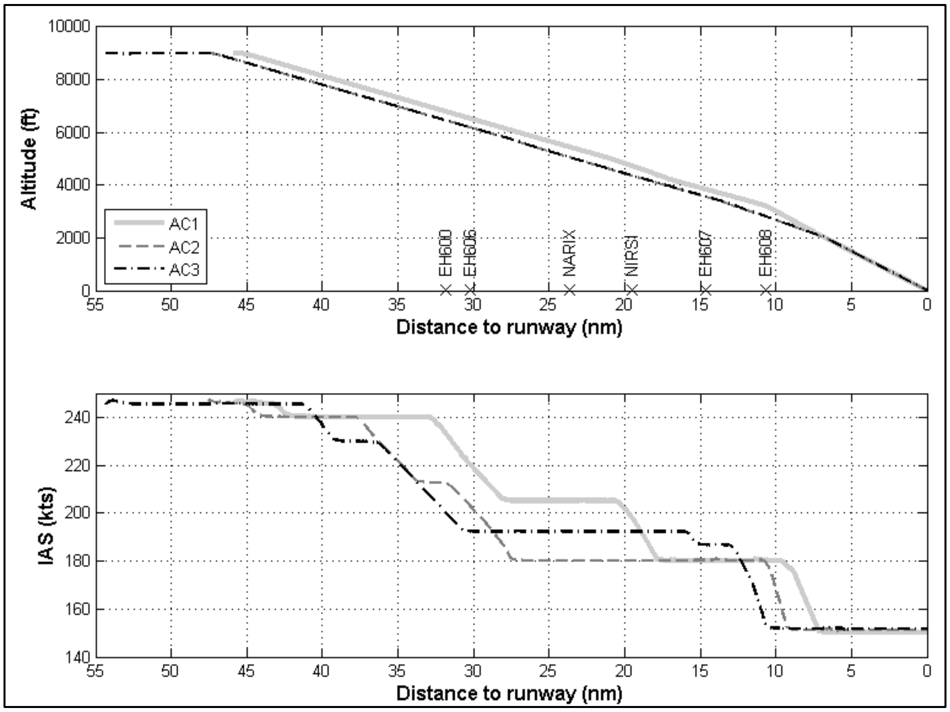

Figure 3.4: Altitude and airspeed profiles for the three aircraft arriving into Schiphol airport RWY 18R

This case study also investigates an off-nominal event. The off-nominal event examines the situation where one aircraft needs priority, and thus the traffic sequence may need to be re-adjusted, including stretching others' paths to let the priority aircraft go first. The off-nominal operations are illustrated in Figure 3.5 where aircraft 1 (arriving from the west) requires priority to land first at RWY 18R and therefore aircraft 2 and 3 perform a fanning maneuver starting at the NARIX waypoint and rejoin the stipulated trajectory at EH 608 so that they can slot behind aircraft 1 in the arrival queue, contrasted with the initial plan for aircraft 1 to merge into the final approach fix between aircraft 2 and 3.

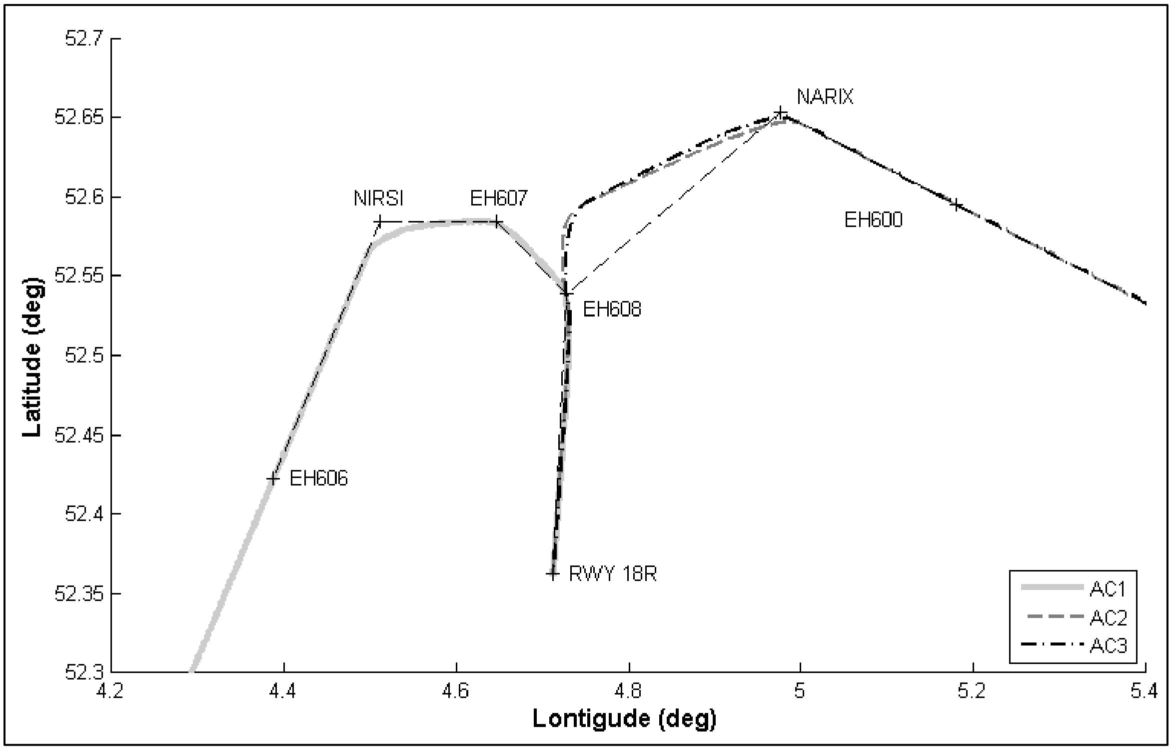

Figure 3.5: Arrival routes for the three aircraft in the off-nominal scenario. Aircraft 1 requests priority landing leading to aircraft 2 and 3 fanning off of the stipulated trajectory

3.3.1 Actions, Agents and Allocations

Table 3.1 summarizes the actions associated with the merging and spacing concept of operation in this case study. The actions are grouped into "functional blocks" where the actions in each block generally operate on the same information towards some shared objective or outcome. The designation of these blocks was based on the notions of control versus management, and the dimensions of lateral, vertical and speed, as well as configuring the aircraft and invoking actions suitable for managing off-nominal situations. Note, the aircraft and autoflight dynamics are also captured in an action "Fly Aircraft".

Table 3.1: Actions involved in the case study

Functional blocks	Action name	Action description
Vertical profile control	Start descent	When clearance is given, configure the autoflight system to establish a 2 degree descent. Triggered by "Clear for Descent"
	Intercept GS	Intercept glideslope signal and initiate 3 degree glideslope descent. Sets its next update time to when GS intercept is predicted.
	Land aircraft	Land the aircraft (removes aircraft from simulation). Sets its next update time to when the aircraft is predicted to land.
Vertical profile management	Clear for descent	Give clearance once aircraft reaches calculated top-of-descent location for OPD Sets its next update time to when it next predicts needing to check whether top of descent has been reached.
	Clear for final approach	Give clearance to start final approach and intercept ILS signal. Sets its next update time to when it next predicts needing to check whether aircraft is positioned for ILS intercept.
Lateral control	Direct to waypoint	Configure the autoflight system to fly a heading direct to the target waypoint. Triggered by "Manage waypoint.
Lateral profile management	Manage waypoint	Set the target waypoint. Sets its next update time to when it next predicts the target may be reached or need to be changed.
	Execute path stretch	Configure the autoflight system to execute a commanded path stretch. Triggered by "Command path stretch."
Speed control	Set airspeed	Enter commanded airspeed into the autoflight system (during both OPD and IM) Triggered by actions setting commanded speeds.
Speed management	Command OPD speed	Command an airspeed for OPD appropriate to altitude. Declares its next update time to when it next predicts will reach flight level corresponding to a lower OPD airspeed.
	Command IM airspeed	Calculate the required airspeed to maintain the interval. (Any IM algorithm can be applied) Sets its next update time to when it predicts needing to change the airspeed by more than 1 knot to maintain the commanded spacing for interval management.

Aircraft configuration management	Set flaps and speedbrakes	Command flaps and speedbrakes when needed. Sets its next update time to when it predicts needing to check that the aircraft has slowed down to an airspeed that corresponds to a different flap and speedbrake setting.
	Deploy gear	Deploy gear when below 2,000 ft. Sets its next update time to be when it predicts reaching 2,000 ft.
Off-nominal situation management	Command path stretch	Calculate and command a trombone or fanning maneuver to create path stretch sufficient to create desired spacing. Sets its next update time to when it predicts a path stretch may be required.
	Determine sequence at merge point	Determine the aircraft sequence at the merge point, including where IM will be required and whether a path stretch is required. For 3 aircraft case study, executed once according to script.
	Set lead aircraft	Set the lead aircraft for IM. Triggered by "Determine Sequence"
Aircraft dynamics (always executed by a separate flight agent)	Fly aircraft	Updates aircraft state using a non-linear 6 degree of freedom aircraft dynamic simulation and autoflight's determination of control surface deflections and throttle to track commanded trajectory. Declares its next update time to be every 0.01s.

Next, the relevant agents can be identified as appropriate for each stage of the analysis. Early in design, the agents may be defined using fairly coarse divisions. For example, this case study just looks at the Air ("A") and the Ground ("G") agents, where the "A" agent is an aggregation of the three flight crews corresponding to the three aircraft in the case study. Later, more-detailed analysis may further detail both agents, such that the single "A" agent is replaced by multiple agents representing the pilot flying, the pilot managing, the autoflight system, and any other relevant automated systems, and similarly the "G" is replaced by agents representing both the "R" and "D" side controllers and any decision aids or automation with which they interact. The "Fly Aircraft" action that represents the aircraft physics and calls to be updated at 100 Hz is allocated to a separate "Physics" agent.

The sheer scope of the design space, and the number of metrics that simulation can assess within each potential concept of operation, also demands some intermediate characterizations of likely good and bad designs. This case study characterizes potential allocations of authority and responsibility by their *coherence*. This qualitative construct examines whether the actions allocated to each agent contribute to the same intermediate objectives and operate on the same set of information. A coherent allocation, then, can be described in terms of the objectives that the set of actions allocated to any agent collectively achieve, where some of the actions create information that other actions then act on without needing to communicate between agents. In contrast, an incoherent allocation requires different agents to inter-leave their actions, requiring extensive coordination and information transfer between them. For example, when turning an aircraft on a 'fanning maneuver' extending its path so that it can be sequenced behind another aircraft, having one agent estimate and command an appropriate turn, and then another agent actually execute it, may leave the first agent needing to check the progression of the maneuver and to revise the commanded turn, and then the second agent performing the revised maneuver.

Once the agents are defined, then potential allocations can be designed for both authority and responsibility. Here, the construct of coherence can serve as an organizing principle. Using the aggregated functional blocks listed in Table 3.1 as inherently coherent groupings of actions, Table 3.2 and Table 3.3 describe the five coherent function allocations tested in this case study here for authority and responsibility, respectively.

Table 3.2: Authority allocations (A=Air, G=Ground)

Functional blocks	Authority allocations				
	1	2	3	4	5
Vertical profile control	G	A	A	A	A
Aircraft configuration management	G	A	A	A	A
Lateral control	G	A	A	A	A
Speed control	G	G	A	A	A
Lateral profile management	G	G	G	A	A
Vertical profile management	G	G	G	G	A
Speed management	G	G	G	G	A
Off-nominal situation management	G	G	G	G	A

Table 3.3: Responsibility allocations (A=Air, G=Ground)

Functional blocks	Responsibility allocations				
	1	2	3	4	5
Vertical profile control	G	A	A	A	A
Aircraft configuration management	G	A	A	A	A
Lateral control	G	A	A	A	A
Speed control	G	G	A	A	A
Lateral profile management	G	G	G	A	A
Vertical profile management	G	G	G	G	A
Speed management	G	G	G	G	A
Off-nominal situation management	G	G	G	G	A

In contrast, Table 3.4 illustrates an incoherent allocation. Within each of its functional blocks, authority for the actions is distributed so that the agents' actions are inter-leaved. For example, the Ground agent is allocated authority for "Calculate distance to runway"; the Air agent needs to wait for this information to be calculated and transmitted to it before it can execute "Start descent" and subsequently "Intercept GS". Similar divisions are made in the other functional blocks. Further, the allocation of authority and responsibility is set to maximum redundancy – an authority-responsibility mis-match is purposefully created for each action, whereby execution of any action by one agent must be monitored by the other.

Table 3.4: Incoherent allocations (A=Air, G=Ground)

Functional blocks	Actions	Authority	Responsibility
Vertical profile control	Calculate distance to runway	G	A
	Start descent	A	G
	Intercept GS	A	G
	Land aircraft	A	G
Aircraft configuration management	Set flaps and speedbrakes	A	G
	Deploy gear	A	G
Lateral control	Direct to waypoint	G	A
	Calculate distance to waypoint	A	G
Speed control	Set airspeed	G	A
Lateral profile management	Manage waypoint	A	G
	Execute path stretch	A	G
Vertical profile management	Clear for descent	A	G
	Clear for final approach	G	A
Speed management	Command OPD speed	A	G
	Calculate IM airspeed	A	G
Off-nominal situation management	Command path stretch	G	A
	Calculate distance to merge point	G	A
	Determine sequence at merge point	A	G
	Assign lead aircraft	G	A

3.3.2 Experiment Design

Altogether, 52 simulations were conducted. The first 50 simulations spanned all combinations of the five allocations of authority (Table 3.2), five allocations of responsibility (Table 3.3), and two different scenarios representing nominal and off-nominal conditions. The last two simulations examined the incoherent allocation (Table 3.4) in both nominal and off-nominal conditions. In the nominal case, the aircraft were sequenced from the merging arrival routes onto the final approach path according to a simple algorithm based on each aircraft's expect time of arrival over the point where the arrival routes merged, and then the traffic flow continued according to this sequence. In the off-nominal case, as noted before, this sequence was disrupted by one aircraft being given priority and placed first in the sequence, even as this required the aircraft behind to "fan" out their route to arrive later over the merge point.

Throughout, the simulation logged the metrics of taskload and information transfer. It also logged detailed measures of system performance, including the trajectories flown by each aircraft (defining their efficiency and delay), and measures of aircraft spacing in-trail in flight and arrival time at the runway. In this case study, all the actions were performed perfectly regardless of the allocation of authority and responsibility, and thus the system performance did not vary between allocations.

3.3.3 Results

The simulation logged detailed metrics of the emergent demands on the agents themselves in terms of the taskload imposed on them and their information requirements. Figure 3.6 and Figure 3.7 show the taskload, i.e., the number of actions the Air and Ground agents have to perform, respectively, given the off-nominal event. These actions are of two types: (1) the primary authority actions defining the taskwork inherent to the concept of operations, with their authority allocated directly to the agent; and (2) the monitoring actions created in response to an authority-responsibility mismatch.

Examining the impact of allocating primary authority for the actions inherent to the concept of operations, shown in grey in Figure 3.6 and Figure 3.7, the number of actions required across all the agents is a constant (note, some types of actions need to be executed multiple times, and each instance is counted here). Thus, the primary allocation of authority aspects of these two figures are duals of each other, representing a simple division of activity between two agents where the combined total set of action instances remains constant.

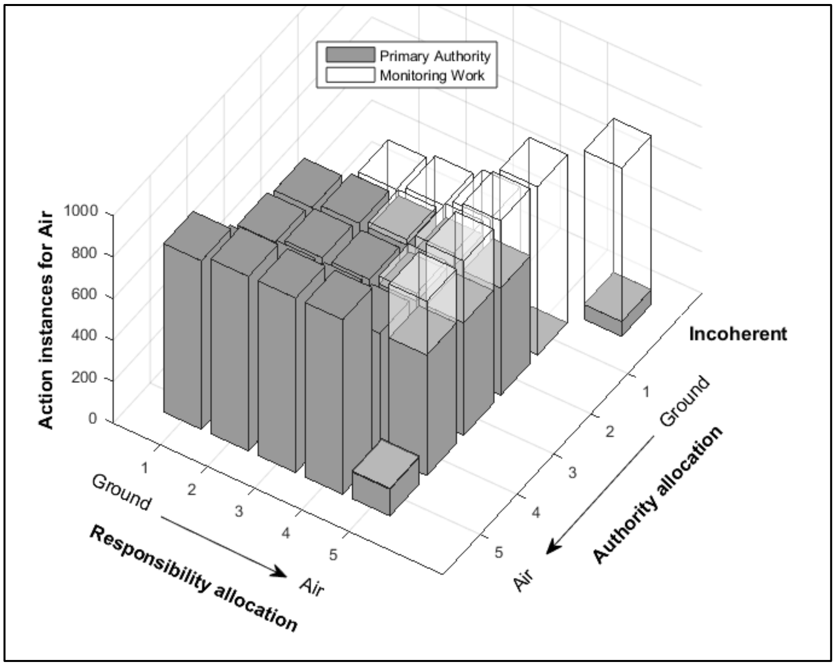

Figure 3.6: Total number of action instances performed by the three Air agents with varying authority and responsibility allocations

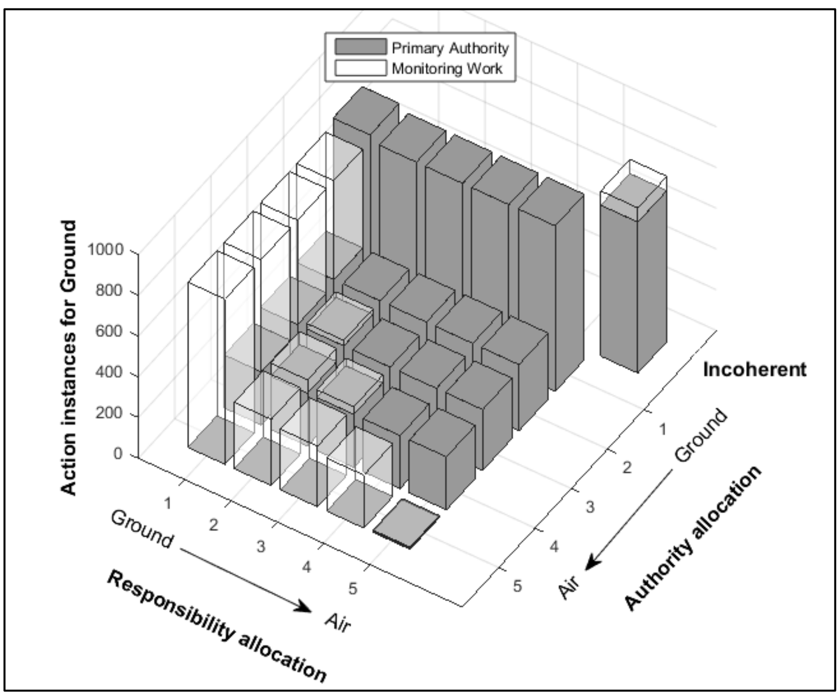

Figure 3.7: Total number of action instances performed by the Ground agent with varying authority and responsibility allocations

The number of actions required by the set of agents is instead only increased by the need for monitoring due to mismatches between authority and responsibility. Here, the extremal cases are when the aircraft are allocated authority for all actions, and the ground agent is allocated responsibility for all the actions' outcomes, or vice versa: assuming that the monitoring actions are conducted as frequently as the primary actions, the number of monitoring actions added by the allocation of authority and responsibility can equal the number of primary actions themselves.

Another emergent demand at the agent level is the information required by each agent when each action needs to be executed. As noted earlier, the simulator automatically flags any time an agent's action needs to get a resource value that is set by another agent's action, requiring some form of information transfer between agents. Examining the impact of coherently allocating authority for the actions inherent to the concept of operations, shown in grey in Figure 3.8 and Figure 3.9, the information transfer required for their execution is the greatest in the middle allocation of authority, i.e., when the work is spread equally over the agents.

Monitoring due to a mismatch between authority and responsibility also significantly increases the need for information transfer, as shown in the clear columns in Figure 3.8 and Figure 3.9. These results are more notional because they are predicated on how the monitoring is conducted: they assume one information transfer per monitoring action, and assume that the monitoring occurs with the same frequency as the primary action that is being monitored. With these assumptions, the information transfers required for monitoring can be significantly more than for the primary authority actions. For example, authority allocation 1 and responsibility allocation 5 in Figure 3.8 represents a case where the Air agent is responsible for all the actions but does not have authority for any of them. This, the Air agent needs to monitor the Ground agent. However, the Ground agent sets all the resources that it may need to get to perform its primary authority taskload whereas the Air agent does not set any of the resources that it needs to get to perform monitoring. From an information transfer perspective, this results in the Air agent requiring monitoring information transfer that is higher than the primary information transfer required by the Air agent in some of the other allocations.

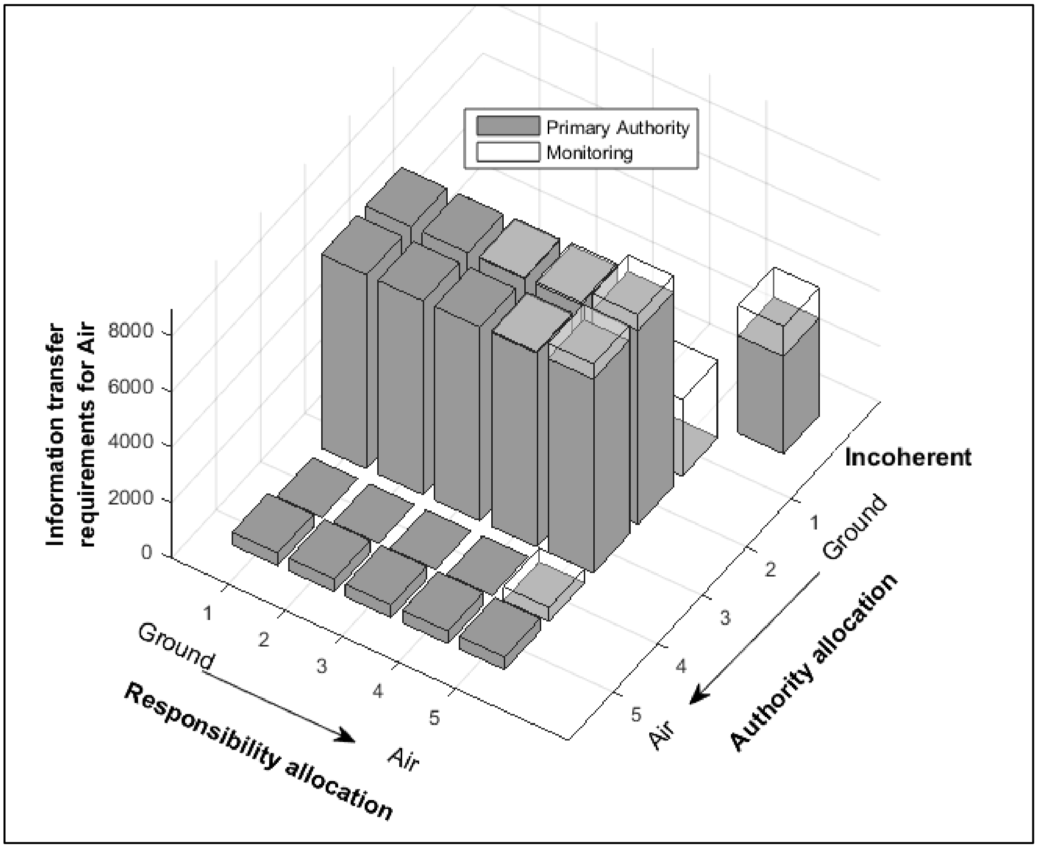

Figure 3.8: Information transfer requirements summed over the three Air agents with varying authority and responsibility allocations

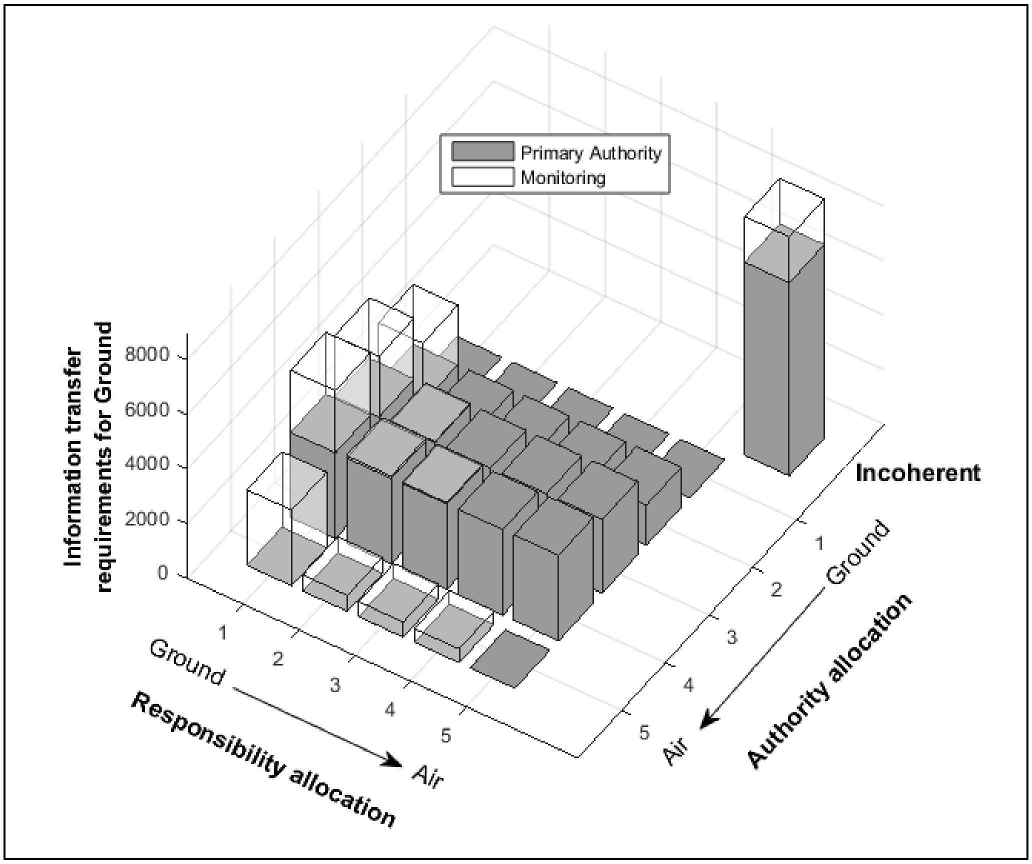

Figure 3.9: Information transfer requirements for the Ground agent with varying authority and responsibility allocations

The incoherent allocation also significantly increases the need for information transfers. As shown earlier in Table 3.4, this allocation requires information transfers both for the same coordination between functional blocks (as in the coherent allocations), and uniquely also between actions that are inter-leaved within the functional blocks. These results flag conditions that themselves would merit further investigation. For example, the inter-leaved actions (such as one agent calculating the distance to a waypoint as the basis for another to identify the trajectory to that waypoint) may need to be performed in quick succession to be accurate: this would make them particularly sensitive to latency or delay in their communication between agents. Thus, while this computational experiment assumed all information was transferred instantly, it also highlights cases where it may be particularly important to test this assumption.

Finally, while Figure 3.6, Figure 3.7, Figure 3.8 and Figure 3.9 provide aggregate assessments, their underlying data is also available for detailed analysis of each concept of operation. For example, Figure 3.11 and Figure 3.12 illustrate when each aircraft agent had to execute actions throughout a simulated flight with authority allocation 2 and responsibility allocation 4, with both the off-nominal event and only nominal conditions respectively (the detailed action trace further records which actions were performed at each time, and which triggered others). Many of these actions also required information transfer, which can also be itemized by time and event. Examining first the Air agents' actions with the off-nominal event as shown in Figure 3.11, different patterns of activity are visible between aircraft: aircraft 1 was given priority to land first and served as the lead aircraft which was able to follow an OPD trajectory down to final approach intercept and landing at around time 660s; the following aircraft, on the other hand, soon needed to re-sequence themselves and fan out their arrival route to delay their time of arrival over the merge point, which changed their distribution of actions. Likewise, the Air agents have significant monitoring with this allocation of responsibility: for aircraft 1 this monitoring is distributed through time, while for aircraft 2 and 3 this monitoring clusters around those periods in time where the IM requires the aircraft to reduce their speed.

Likewise, the impact of the off-nominal event can be observed by comparing Figure 3.11 and Figure 3.12. Most notably, the aircraft land in a different order. To do so in nominal conditions requires aircraft 1 to perform several actions as it establishes the correct interval behind aircraft 2 (mostly monitoring actions with this allocation of authority and responsibility). In contrast, in the off-nominal event, aircraft 2 instead needed to perform these actions; given their different relative positions these actions were required at different times.

Finally, the simulation also allows analysis of the ground agent's actions and their timing. With the allocation of authority and responsibility shown in Figure 3.10, the Ground agent is given mostly primary allocation of authority for actions inherent to the concept of operations, with the Air agents allocated the responsibility that generates monitoring. Thus, the Ground agent's taskload is primarily centered on speed control and lateral profile management tasks that contain some frequently occurring actions. These actions need to be completed for each of the three Air agents, which results in a large amount of actions for the Ground agent.

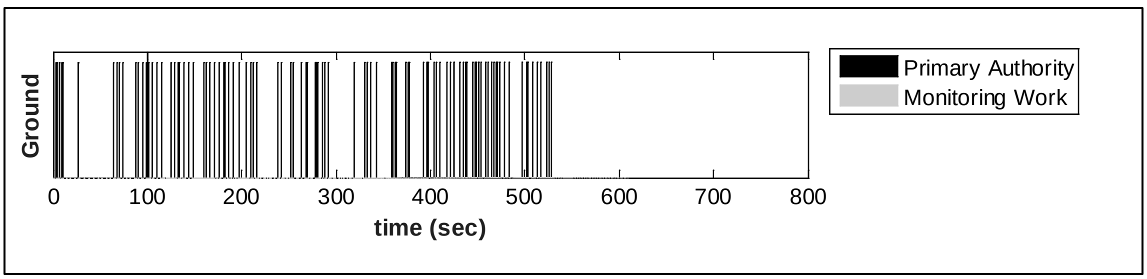

Figure 3.10: Trace of actions performed by the Ground agent

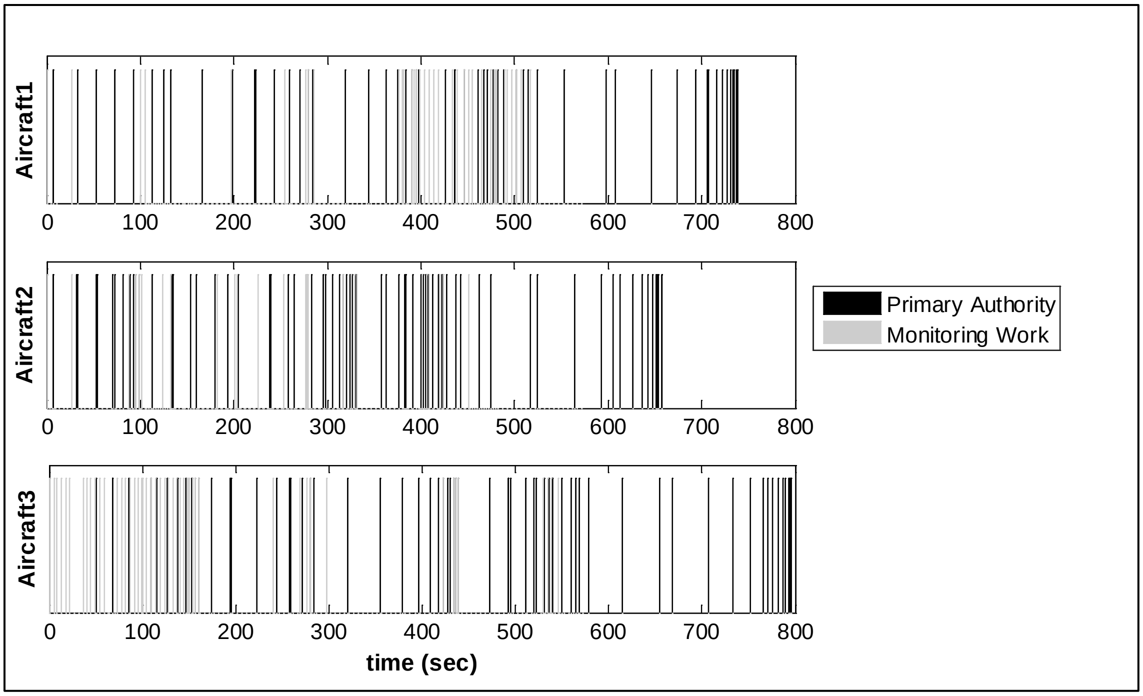

Figure 3.11: Trace of actions performed by the three Air agents in the nominal scenario

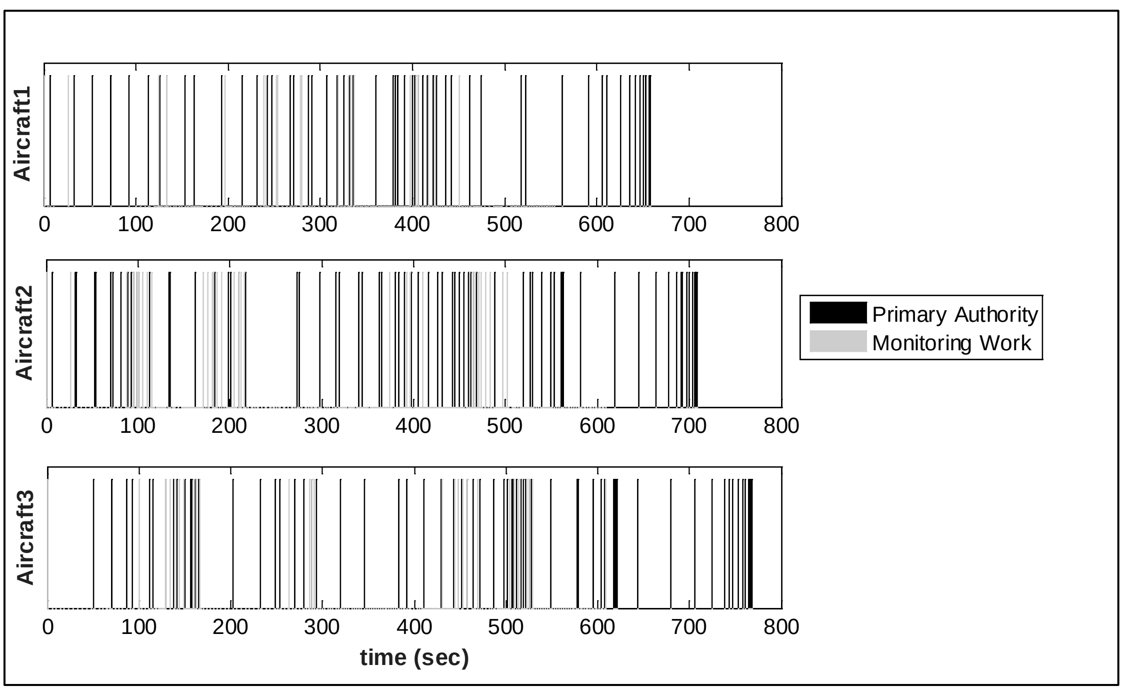

Figure 3.12: Trace of actions performed by the three Air agents in the off-nominal scenario

3.4 Summary

This chapter described a methodology to model concepts of operation according to the actions that define them and subsequently test allocations of authority and responsibility between agents in simulation. Using the methodology proposed in this chapter, analysts can predict and compare the system-level performance and agent-level requirements associated with different concepts. The methodology was demonstrated in a case study for the early-in-design allocation of authority and responsibility.

This chapter examined both the allocation of authority and of responsibility. The allocation of authority determines which agent will execute each action, and thus is the most directly observable; further, it generally drives what system-level behaviors and performance will emerge within the operation. This chapter further demonstrated the importance of also examining the allocation of responsibility for the outcome of each action. A full understanding of the agents' collective work needs to also account for the monitoring required when different agents have authority and responsibility for the same action; the case study here identified allocations where such monitoring can add significantly to both the taskload on specific agents, and their requirements for information transfer from other agents.

While the specific demands on the agents were predicted using simulation, the general trends in the results were predicted by the qualitative attribute of the coherence of the allocation. A coherent allocation assigns actions such that each agent is performing all the actions that contribute to the same general function using the same information, while an incoherent allocation breaks up these functions across agents such that they have to inter-leave their actions and to frequently transfer back-and-forth the intermediary information used within the functions. Thus, the coherence of allocations may serve as a quick qualitative guiding principle for defining potential allocations.

Once potential function allocations are defined qualitatively, this chapter demonstrated how they can be systematically and computationally analyzed early in design. This chapter represented concepts of operation building on what is known early in their design (the constraints due to the physics of flight and the mid-level models of the actions required), and on what can be specified early in their potential allocations of authority and responsibility. This representation allows for the simulation of concepts of operation before more detailed design aspects are known (such as specific algorithms underlying various technologies). Thus, the “work” of the concept of operations is not yet tied to any one agent, and the authority and responsibility for actions can be fluidly assigned – at any time during the simulation – to simulate a wide range of allocations.

However, while the methodology described in this chapter can be used for analyzing allocations, it cannot be used to synthesize allocations, especially as the design process progresses to more detailed stages. Thus, if the metrics obtained from the simulation indicate poor performance or safety (or both), under all the allocations that have been tested, the designer of the concept of operations needs a methodology to generate a new set of candidate allocations that can satisfy the performance and safety goals of the concept of operations. The current day processes of generating these candidate allocations are largely based on heuristics and trial and error. To formalize and streamline this process, the next chapter proposes a methodology for the systematic synthesis of allocations of authority and responsibility in novel concepts of operation.

CHAPTER 4 - SYNTHESIZING ALLOCATIONS OF AUTHORITY AND RESPONSIBILITY

This chapter proposes a methodology for the synthesis of allocations of authority in concepts of operations. The concepts of operations are assumed to be subject to externally specified allocations of responsibility, thereby reducing the scope of the methodology to generating only allocations of authority. This chapter proposes that properties inherent to the work of the concept of operations can be captured by modeling it as a network of actions.

The synthesis of allocations of authority consists of three stages. First, the concept of operations is modeled as a network of actions connected through getting and setting of resources. This network is enriched by the spawning of monitoring actions, reflecting the impact of authority-responsibility mismatches. Second, the synthesis of authority allocation is treated as an optimal network-partitioning problem where actions are clustered together in terms of authority allocation. The objective functions and constraints of this optimization problem are posed such that they reflect the goals of the concept of operation both from the standpoint of system performance and safety. Third, the allocation of authority of actions to agents is derived from the clusters obtained.

The following sections first describe a general methodology by which allocations of authority can be synthesized systematically, second review fundamental network theory to support the modeling the concept of operations as a network of actions, and finally demonstrate the methodology in a case study.

4.1 General Methodology for Synthesizing Allocations of Authority

This general methodology systematically synthesizes allocations of authority that achieve the performance and safety goals of the concept of operations. The synthesis of authority allocations is posed as a network optimization problem where objective functions and constraints reflect the performance and safety goals for the concepts of operations being designed.

Figure 4.1 gives an overview of the general methodology in terms of its inputs and outputs. The inputs are the objective functions and constraints that reflect the performance and safety goals of the concept of operations, the allocation of responsibility, and a network model of the concept of operations. As detailed in the subsequent section, this network model is framed in terms of the actions constituting the concept of operations where the actions are connected by getting and setting of resources. Further, the network is parametrized using the metrics obtained from the analysis methodology demonstrated in the previous chapter.

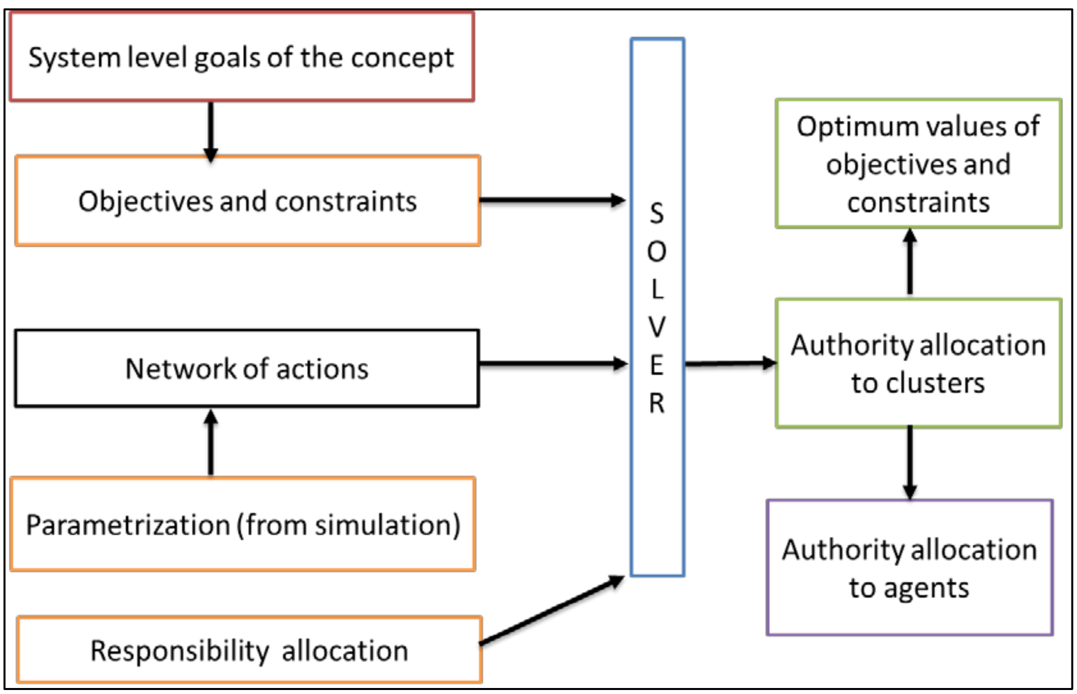

Figure 4.1: General methodology for synthesis of allocations

These inputs are passed into an optimization solver that identifies clusters in the network of actions that optimize attributes of the network model. Thus, the objective functions and constraints, while reflecting the performance and safety goals of the concept of operations, also need to be framed in terms of measurable attributes of the network model of the concept of operations. For example, the case study described later in this chapter achieves this using the metrics of taskload weighting the nodes of the network and information transfer weighting the edges of the network. The solver also reports the optimum values of the objective function and the associated values of the constraints.

The obtained clusters of actions can then guide the allocation of authority for clusters to agents. Depending on the scenario, agents can be allocated authority for multiple clusters of actions. Some agents may be allocated authority for more clusters as compared to other agents depending on the individual workload and capacities of the agents, or the information requirement of the agents. For example, the distribution of authority to control airspace sectors between just one air traffic controller versus two controllers is a function of how busy the traffic flows are on that particular day.

4.2 Network Representation of Concepts of Operation

In the context of designing allocations of authority for concepts of operation, the metrics that quantify the performance and safety of concepts of operations are generally emergent in that they are observed at a different level from that which is modeled. Since emergent features arise from inter-element interactions, a model of the concept of operation that can expose the structure and attributes of these interactions is the first step toward designing authority allocations. This section proposes a network representation of concept of operations that is used to first formulate, and second solve, the problem of allocating authority given external responsibility allocations. The subsequent sub-sections firstly define the terms from network theory and then model the concept of operations as a network of actions.

4.2.1 Network Theory: Mathematical Preliminaries

Graphs provide natural abstractions for how information is shared between agents in a network [83][84]. These abstractions contain virtually no information about what exactly is shared by the agents, through what protocols the exchange takes place, or what is subsequently done with the received information. Instead, the graph-based abstraction contains high-level descriptions of the network topology in terms of objects referred to as nodes and edges. Of particular importance is algebraic graph theory that provides the tools to analyze the combinatorial characteristics of networks (represented by graphs) within a dynamic system (an air traffic concepts of operations). This sub-section establishes the core definitions used in network theory that will be subsequently used in this chapter.

The concepts of the *vertex set* and *edges* define graphs. A graph is built upon a finite set referred to as the vertex set and denoted by *V*. Each element of *V* is a vertex (or node) of the graph. For a graph with *n* nodes, the vertex set is represented as:

$$V = \{v_1, v_2, v_3, \dots, v_n\}.$$

Edges are defined on the 2-element subsets of the vertex set, denoted as $[V]^2$. This set consists of elements of the form $\{v_i,v_j\}$ where $i,j = 1,2,\dots,n$ with $i \sim= j$. The set of *edges* of the graph is a particular subset of $[V]^2$. The graph *G* is formally defined as the pair:

$$G = (V, E).$$

where *V* is a finite set of vertices and *E* is a particular subset of $[V]^2$.

The graph is said to be *weighted* when the edges are assigned weights. The weights are defined by the function

$$w: E \to \mathbb{R}.$$

Graphs can be visually represented by dots (the vertices v_i) and lines between v_i and v_j when v_iv_j belongs to *E*. Figure 4.2 shows a dot-edge representation of a network with 8 nodes and 9 edges.

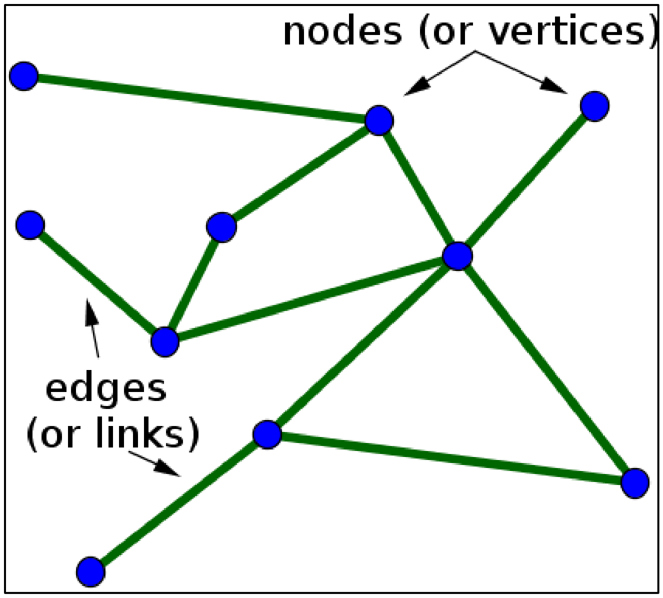

Figure 4.2: Graphical representation of networks

The *adjacency matrix A(G)* is the symmetric *nxn* encoding of the adjacency relationships in the graph G

$$[A(G)]_{i,j} = w_{ij} \; if \; v_i v_j \; belongs \; to \; E$$

A *cluster C* is a subset of the vertex set *V* = *[n]*. A *partition* of the graph is then a grouping of its vertex set into different clusters. Figure 4.3 shows a network partitioned into 3 clusters. Once a network is partitioned, the edges of the network can be distinguished as inter-cluster edges vs. intra-cluster edges.

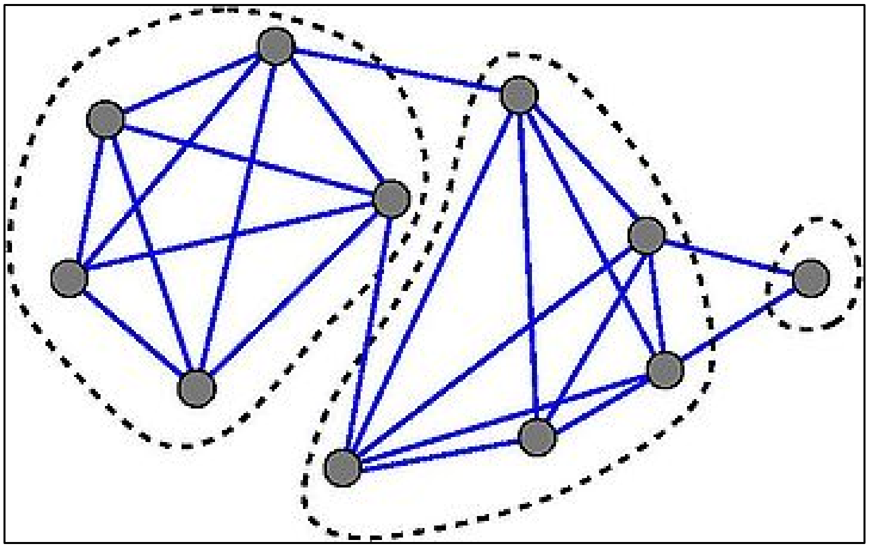

Figure 4.3: Graph partitioning to form clusters in a network

The problem of graph partitioning (GP) has been a subject of much study because of its applications in a wide variety of domains, such as the distribution of work to processors of a parallel machine [85]. GP is extensively used to ensure load balancing and to minimize communications across processors. Physical design of digital circuits for very large-scale integration (VLSI) heavily uses GP [86]. In computer vision, the goal of image segmentation is to partition the pixels of an image into groups that correspond to objects and GP has become one of the most important solution techniques for this problem [87].

Complex networks, i.e., weighted graphs created by real-life or modeling processes [88], have introduced numerous applications of GP to identify groups of similar nodes in the networks and also identify connected components. Minimizing the impact of cascading events in power grids [89], detection of biological processes by finding clusters of involved nodes in biological networks [90] and finding community structure in social networks [91] all heavily rely on the detection of partitions within the graph representation of the underlying system.

Many techniques and algorithms have been developed to solve the graph partitioning problem. Spectral partitioning uses spectral properties of the graph to find the optimal partition of the graph [92][93][94]. Many methods have been developed that rely on the branch-and-bound framework [95]. The max-flow min-cut theorem has also been used to separate node sets in a graph [96]. In recent years, evolutionary methods employing a number of metaheuristics have also been applied to solve the GP problem [97].

4.2.2 Forming a Network of Actions to Represent a Concept of Operations

In the models of concepts of operations described in Chapter 3, actions interact with the environment by getting and setting resources. Thus, actions interact with each other through resources that convey information about the state of the environment. This sub-section details how the interaction between actions can be represented as a network, which can then be used to guide the allocation of authority for actions.

When an action gets a resource that has been set by another action, an edge is formed between the action that set the resource and the action that got the resource. Throughout the application of simulations (as demonstrated in chapter 3), this edge can be weighted by the number of times this interaction happened, representing information transfer between the actions. For example, in Figure 4.4, w1 equals the total number of times that Action1 set Resource1 that was then gotten by Action2.

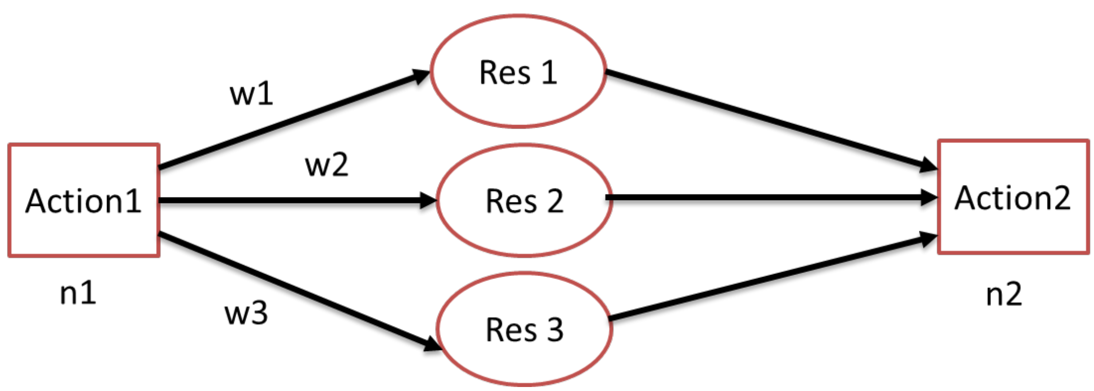

Figure 4.4: Actions interacting through resources

Further, note that the graph representing the connection between actions is a multigraph, i.e. there are multiple edges between two actions (because of there being multiple possible resources through which the actions interact). The multigraph nature of the action network can be further simplified by collapsing all the edges connecting two actions into a single edge that is weighted as the sum total of the weights of the original edges connecting the actions. Figure 4.5 shows the action pair Action1 and Action2 with a single edge that combines the edges connecting Action1 and Action2 as shown in Figure 4.4.

Further, Figure 4.5 also shows the weights on the actions themselves (n1 and n2). These node weights equal the number of actions instances, i.e. the number of times the action was executed during the course of the concept of operation. Thus, Action1 was executed n1 times and Action2 was executed n2 times. These weights on the nodes give an indication of the taskload imposed by the actions on the agent executing them. Thus, an action with a higher node weight represents a higher taskload on the executing agent as compared to an action with a lower node weight.

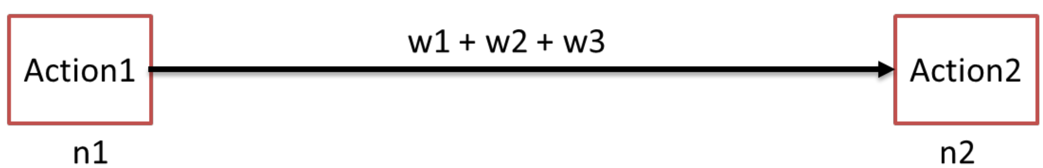

Figure 4.5: Forming a weighted edge between actions

The concept of operations in the case study described next involves the 25 actions listed earlier in Table 3.1. Figure 4.6 shows this network of actions with the edges weighted by the information transfer between actions as determined by the getting and setting of resources as measured in a single simulation in the nominal scenario.

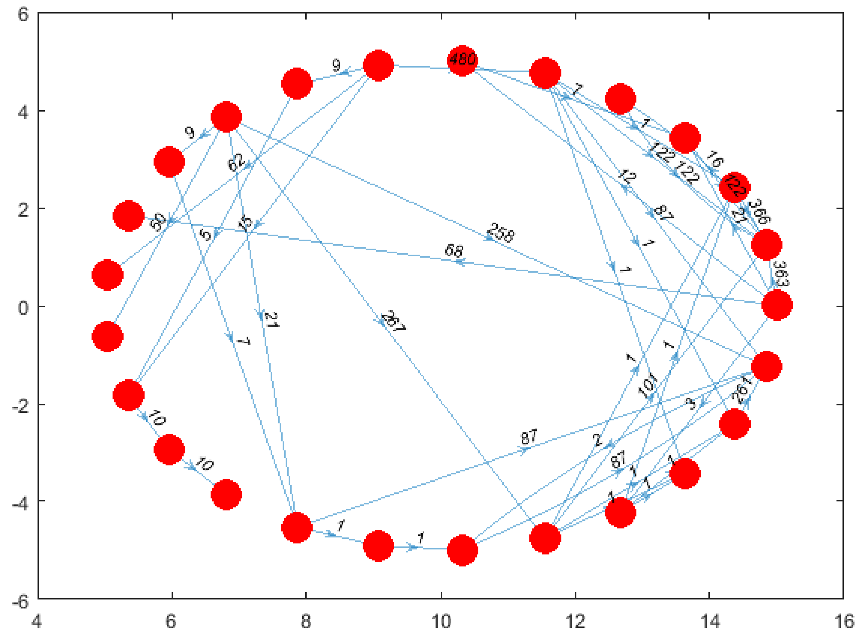

Figure 4.6: Network of primary authority actions

Authority-responsibility mismatches enrich the network by adding monitoring actions and their associated edges to the network. Thus, there can be two types of actions in the network: primary authority actions and monitoring actions. For the purpose of network analysis and partitioning, these two types of actions have an important difference: while primary authority actions always need to be captured as nodes in the network regardless of the allocation, the existence of monitoring actions in the vertex set depends on the allocation of authority compared to the specified responsibility allocation. Thus, monitoring actions add to the notion of emergence in the system as they are only added to the network when the clustering of the primary authority actions creates an authority-responsibility mismatch.

Corresponding to the formation of nodes in the network (primary authority action nodes and monitoring action nodes), edges can also be of two types. When both the end nodes for an edge are primary authority actions, the edge is said to be a primary authority edge. When either of the actions is a monitoring action, formed due to the mismatch in the authority and responsibility allocation for the corresponding primary authority action, the edge is said to be a monitoring edge. Figure 4.7 shows a situation where the authority allocation has led to a mismatch in the authority and the responsibility allocation for certain actions and thus has created monitoring actions. Circles represent primary authority actions and squares represent monitoring actions. Further, the monitoring edges are represented by dashed lines and the solid lines represent edges between two primary authority actions. Finally, the figure represents a case where the primary authority action network has been partitioned into two clusters shown by the red and green colors. The edges crossing the clusters are represented by heavier lines.

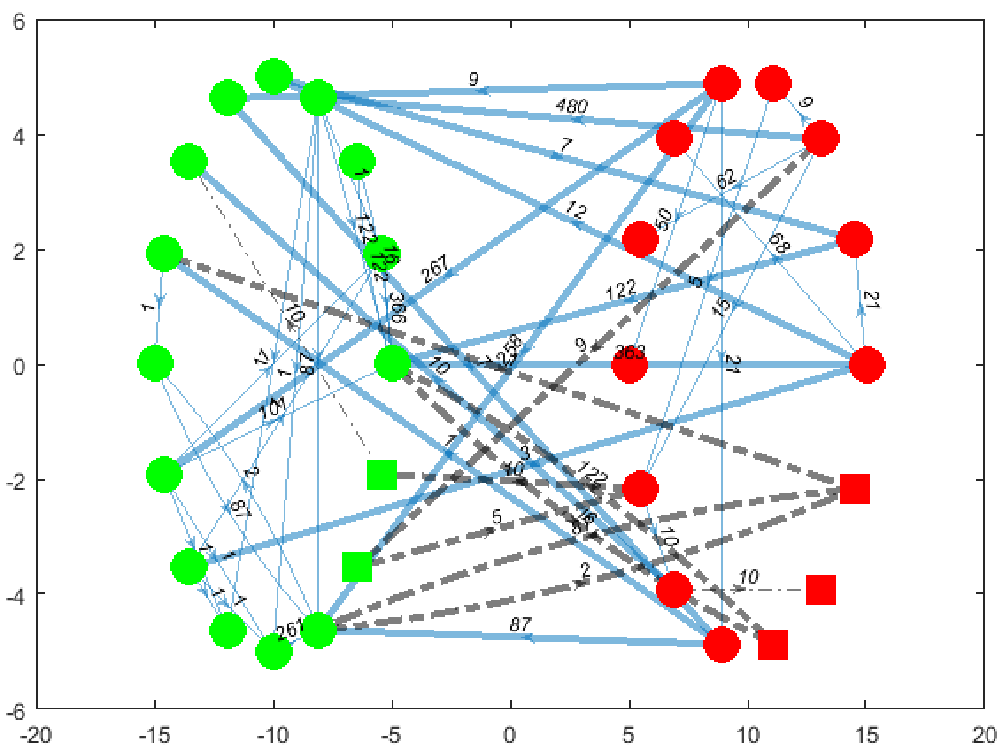

Figure 4.7: Network showing two clusters (red cluster on the right and green cluster on the left). Circles represent primary authority actions and squares represent monitoring actions. Solid lines represent primary authority edges and dashed lines show monitoring edges. The heavier lines indicate edges crossing clusters

While the unweighted network can be used for the preliminary design of authority allocations using the structure provided in terms of action linkages, weighting the network of actions, both in terms of the edge weights and the node weights, can better parametrize the network to represent scenarios that the concept of operations would be applied in. This parametrization is made possible through the computational simulation demonstrated in the previous chapter. Note that the simulation only needs to be run once to capture the edge weights (representing information transfer between actions) and node weights (representing taskload associated with an action). The objectives and constraints of the network optimization problem to find clusters of actions in the network are then framed in terms of these network attributes that represent action-level metrics.

Monitoring edges can be weighted in multiple ways depending on the type of monitoring being studied. In this thesis, the lower and upper bounds on monitoring activity are established by basic monitoring and complete monitoring, respectively. To represent basic monitoring, all the edges originating from, or culminating at, the monitoring action node are weighted as the number of times the monitoring action was executed in the simulation. This represents a basic monitoring strategy in that every time the primary authority action is executed, the monitoring action checks for a single indicator of the output's goodness [50]. Complete monitoring, on the other hand, is represented by weighting the monitoring edges with exactly the same weight as the edges of the corresponding primary authority action that is to be monitored. This weighting represents the scenario where the monitoring action gets and sets the same number of resources as the primary authority action.

4.3 Case Study

The merging and spacing concept of operation studied in the previous chapter is now studied from the perspective of synthesizing allocations of authority under the same five responsibility allocations. This case study considers authority allocation to two clusters: the Air cluster and the Ground cluster.

All possible combinations of two objective functions, six constraints and five responsibility allocations result in a total of 60 different syntheses of authority allocations. Specifically, the objective functions considered here are: 1) minimize total information transfer across clusters and 2) maximize total information transfer across clusters. These objective functions are representative of real operation objectives. Operations that want to reduce the risk of disruption due to cyber-threats like hacking would require minimized information transfer between agents such that each agent possesses localized information to perform its own operations and the coordination variables needing to be passed between agents are as few as possible. On the other hand, if redundancy is critical for the concept of operations, it makes sense to maximize the information transfer between agents so that more than one agent knows about every action being executed and can step in to rectify errors and mistakes.

The constraints are posed in terms of taskload associated with the clusters. When a cluster is allocated authority for an action, the taskload associated with that action gets imposed on that cluster. Thus, it is important to study the impact of authority on taskload imposed on clusters. The six constraints considered in this case study are as follows.

1. No constraints: The synthesis of authority allocations is posed as an unconstrained optimization problem. This allows full freedom to the optimization algorithm to allocate actions to either of the clusters without any constraints on the resulting taskload. Thus, certain allocations may allocate the entire set of actions to one cluster leaving the other cluster with no actions at all. These allocations may be useful to establish the boundaries in terms of how severely skewed the allocations may be in terms of taskload on clusters.
2. Balance primary authority taskload: This constraint requires the taskload due to primary authority actions on the Air cluster and the Ground cluster, to be equal.
3. Balance monitoring taskload: This constraint enforces the taskload due to monitoring actions on the Air cluster and the Ground cluster, to be equal.

4. Balance total taskload: This constraint enforces the total taskload, i.e., the sum of primary authority and monitoring taskload Air cluster and the Ground cluster to be equal.
5. Cap primary taskload on the Ground cluster: This constraint enforces an upper limit on the primary authority taskload that can be imposed on the Ground cluster, which is representative of scenarios where agents have a limit on their taskload capacity.
6. Cap monitoring taskload on the Ground cluster: This constraint enforces an upper limit on the monitoring taskload that can be imposed on the Ground cluster, which is representative of scenarios where agents have a limit on their capacity to monitor.

The external responsibility allocation is also specified for each action. The same five allocations of responsibility as studied in the previous chapter are presented and are numbered from 1 through 5 where 1 represents responsibility for the entire set of actions allocated to the Ground cluster and 5 represents the same for the Air cluster, as shown earlier in Table 3.3. Thus, authority-responsibility mismatches occur whenever the cluster with authority for that action and the cluster with responsibility for that action, respectively, are not equal. The optimization problem also has to account for such potential mismatches in authority and responsibility as it solves the authority allocation problem and has to determine which configuration of authority, with the associated spawning of monitoring actions according to the responsibility allocation, best achieves the goals of the concept of operations.

The authority allocation problem is now formulated as an optimization problem which, in this case study, was solved via a Genetic Algorithm. The design variables are the authority allocation for every action. The authority allocation problem represents the optimum grouping of actions into clusters such that the objective function is optimized while satisfying the constraints. The inter-cluster information transfer, the basis of the objective function of the optimization problem, is calculated using the weighted edges connecting actions that belong to different clusters. Similarly, the actions that belong to a cluster add their node weights to the taskload associated with the cluster due to an allocation of authority for primary authority actions, and due to monitoring actions reflecting authority-responsibility mismatches. These taskload values are used for checking whether the allocation is feasible relative to the taskload constraints.

4.3.1 Results

The optimization problem was studied with 60 combinations of objectives, constraints and responsibility allocations, resulting in 60 allocations of authority. For each authority allocation, the metrics of inter-cluster information transfer and taskload imposed on each cluster were captured. The following sub-sections split discussion of the results into two sections based on the two objective functions: minimizing versus maximizing inter-cluster information transfer.

4.3.1.1 Minimizing Inter-Cluster Information Transfer

Examining the authority allocations seeking to minimize inter-cluster information transfer, Figure 4.8 summarizes the information transfer between the Air and the Ground clusters for the 30 allocations generated by varying the 6 tasklaod constraints and the 5 responsibility allocations. The information transfer between the clusters is categorized into two types: primary authority information transfer (shown in grey in the figure) represents information transfer due to actions having to get resources that have been set by other actions, and monitoring information transfer (shown as transparent bars) represents the information transfer due to monitoring requirements generated by authority-responsibility mismatches.

Predictably, when no constraints are imposed on the possible taskload on the clusters, and responsibility for all actions is allocated entirely to either the Air cluster or the Ground cluster, authority is allocated to the same cluster as is given responsibility to negate the need for information transfer for monitoring. This unconstrained allocation thus results in the need to transfer information neither for primary authority actions, nor for monitoring actions.

Other taskload constraints result in allocations of authority requiring inter-cluster information transfer. For example, the constraint requiring primary authority taskload to be balanced across the air and ground clusters results in inter-cluster information transfer because, no matter what the responsibility allocation, both clusters have to transfer some information as each performs half the primary authority taskload. For this constraint, some allocations of responsibility can result in authority allocations requiring less information transfer where authority-responsibility mismatches, and commensurate monitoring, can be reduced.

The constraint requiring total taskload to be balanced across the air and ground clusters results in higher inter-cluster information transfer. To satisfy this constraint, the authority allocation has to be such that the sum total of primary authority and monitoring taskload is equal across the two clusters. When the responsibility for all actions is allocated entirely to one cluster, the resulting authority allocation creates monitoring in one cluster mirroring primary authority allocations to the other cluster. On the other hand, for responsibility allocations distributed across the two clusters, more authority-responsibility mismatches can be avoided, thereby reducing inter-cluster monitoring information transfer.

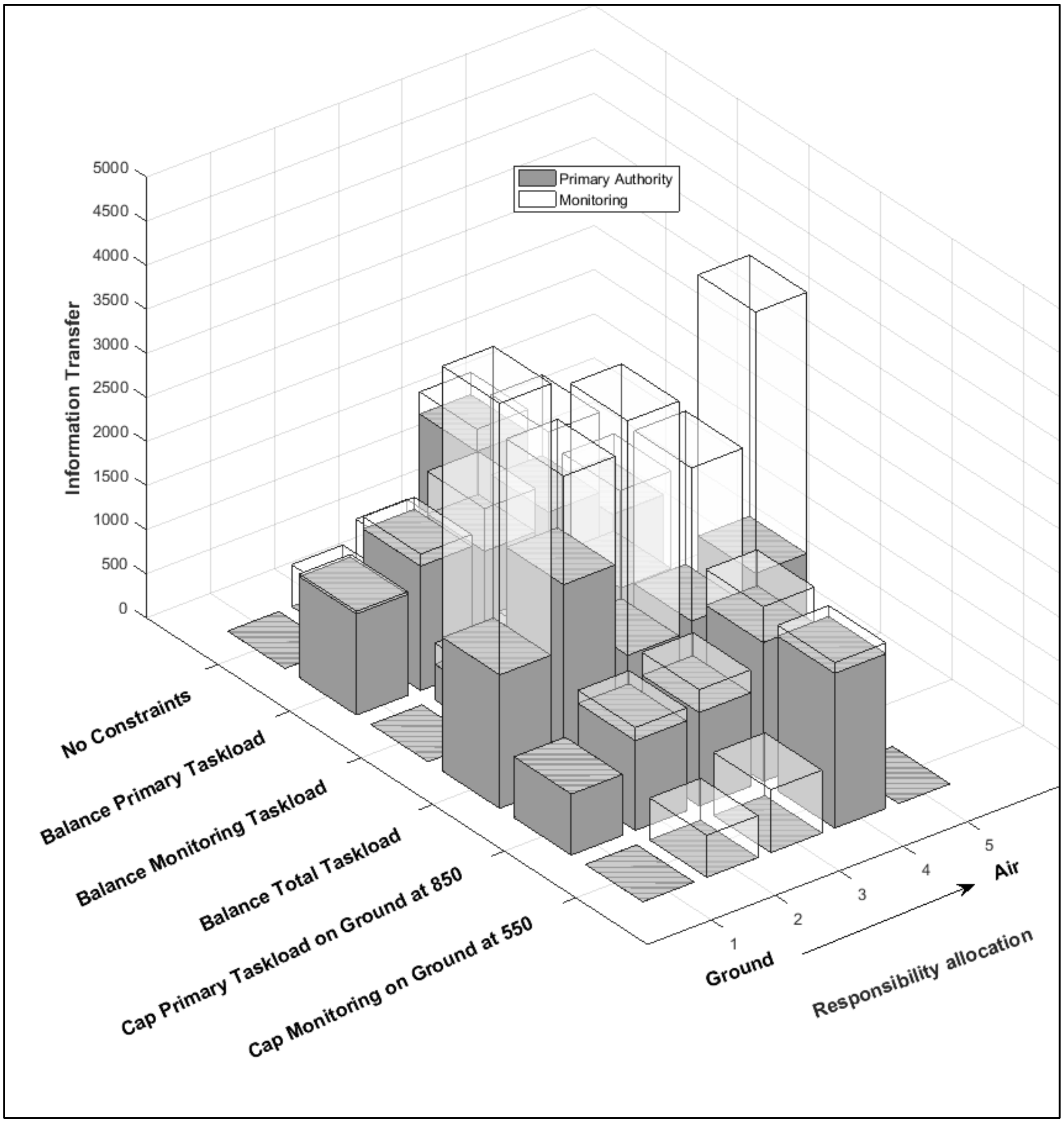

Figure 4.8: Information transfer values obtained by minimizing inter-cluster information transfer with varying responsibility allocation and constraints on taskload

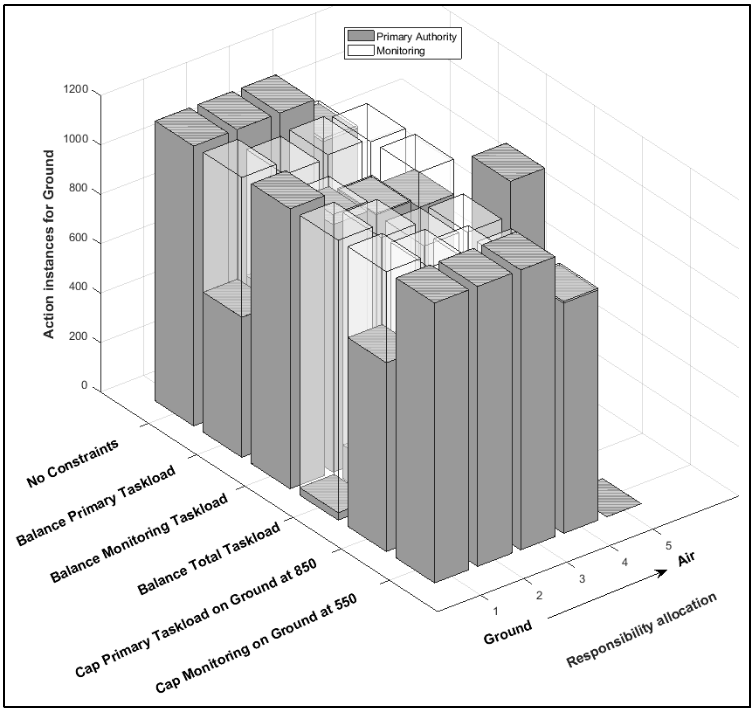

Figure 4.9: Taskload imposed upon Ground cluster for minimizing inter-cluster information transfer with varying responsibility allocation and constraints

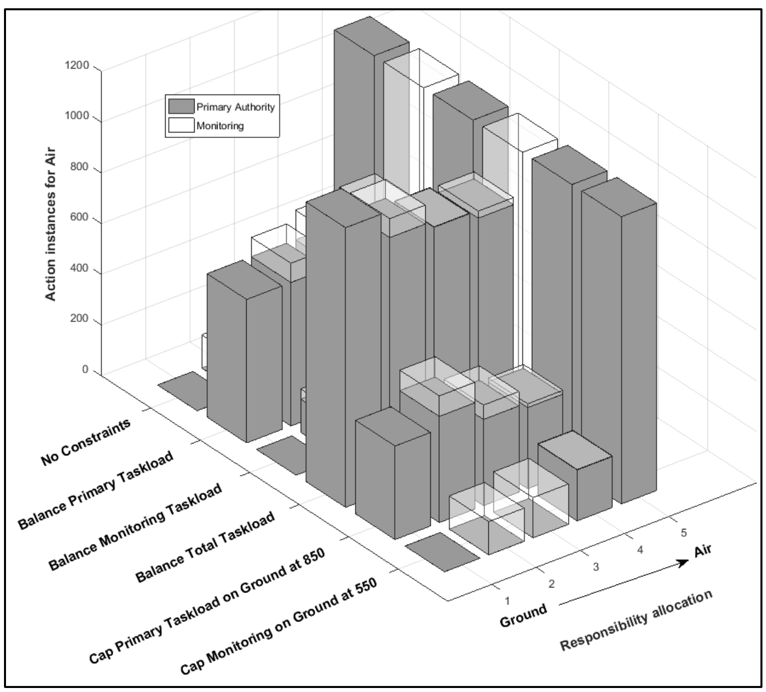

Figure 4.10: Taskload imposed upon Air cluster for minimizing inter-cluster information transfer with varying responsibility allocation and constraints

Figure 4.9 and Figure 4.10 portray the total taskload imposed on the Ground and Air clusters, respectively, with the same authority allocations as in Figure 4.8. As with information transfer, the taskload imposed on the clusters is also categorized into two types: primary authority taskload (grey) and monitoring taskload (transparent). The taskload due to primary authority in the two figures represent observations that are duals of each other. For every authority allocation, summing up the primary authority taskloads on the Air and Ground clusters results in the same constant value as it represents the taskload associated with the work that is necessary for the concept of operations. Thus, while different authority allocations may divide up the primary authority taskload differently between the Air and the Ground cluster, the total primary authority taskload remains the same across all authority allocations.

In the unconstrained taskload case, the three responsibility allocations giving the most responsibility to the Ground cluster result in an authority allocation that assigns authority for all the primary actions to the Ground cluster to minimize the inter-cluster information transfer requiring for monitoring in the face of authority-responsibility mismatches. Likewise, for responsibility allocations 4 and 5, the resulting authority allocation swaps to giving primary authority mostly to the Air cluster.

In one of the constrained cases, the primary authority taskload on the Ground cluster was capped. The value of the cap was fixed at 850, which is the mean between the unconstrained taskload value and the balanced primary authority taskload value on the Ground cluster. Under this capping constraint, with the responsibility allocated entirely to the Ground cluster, the authority for all the actions can no longer just be assigned to the Ground cluster to minimize the information transfer. Hence, some actions are allocated to the Air cluster, resulting in monitoring taskload for the Ground cluster. Moreover, since the actions that could not be allocated to the Ground cluster are allocated to the Air cluster, there is an associated primary authority taskload on the Air cluster. On the other hand, with the responsibility allocated entirely to the Air cluster, authority for all the actions is allocated to the Air cluster because the constraint does not impose any capping requirement on the primary authority taskload on the Air cluster.

The constraint balancing primary taskload across the Ground and Air clusters results in even taskload across all responsibility allocations on the Ground cluster and progressively reduces monitoring for the Ground cluster as the responsibility for more actions is allocated to the Air cluster. On the other hand, the constraint balancing total taskload results in allocations of authority that are significantly different in behavior as they impose much more monitoring load on the Ground cluster as compared to the other constraint cases for the responsibility allocations that allocate responsibility for more actions to the Ground cluster.

Table 4.1 reports the resulting allocations of authority for actions to the Air and Ground cluster obtained under the varying allocations of responsibility and constraints. Note that three copies of each action are listed, corresponding to each of the three aircraft in the scenario. In the unconstrained case, for the three responsibility allocations that allocate responsibility for more actions to the Ground cluster, the authority for the entire set of actions is allocated to the Ground agent. The two responsibility allocations that allocate responsibility for more actions to the Ground cluster allocate more actions to the Air cluster.

The addition of taskload constraints causes the resulting allocations of authority to differ from the unconstrained case. For instance, imposing the constraint of balancing primary authority taskload no longer allows an authority allocation for all actions to either Air or Ground entirely, even when one cluster has responsibility for all actions.

Finally, it is interesting to note that the same type of action, for different aircraft, can be allocated to different clusters. For example, in the balance primary taskload constrained case with responsibility allocated entirely to the Ground cluster, the actions "Calculate Distance to Runway" for both aircraft 1 and 2 are allocated to the Ground cluster, whereas for aircraft 3 it is allocated to the Air cluster. This illustrates the importance of the taskload imposed by actions as captured by the node weights in the network representation. Since aircraft 3 performs IM throughout the arrival approach, the "Calculate Distance to Runway" action is executed more often for aircraft 3. Thus, in this case, allocating the same type of action for different aircraft to different clusters facilitates the objective of minimizing information transfer under the constraint of balancing the resulting primary authority taskload on the clusters.

Table 4.1: Authority allocations for minimizing inter-cluster information transfer with varying constraints and responsibility allocations. (G=Ground cluster, A=Air Cluster)

Constraints	No constraints					Balance primary authority taskload					Balance monitoring taskload					Balance total taskload					Cap primary authority taskload on the Ground Cluster					Cap monitoring taskload on the Ground Cluster				
Actions\Resp	**1**	**2**	**3**	**4**	**5**	**1**	**2**	**3**	**4**	**5**	**1**	**2**	**3**	**4**	**5**	**1**	**2**	**3**	**4**	**5**	**1**	**2**	**3**	**4**	**5**	**1**	**2**	**3**	**4**	**5**
3manageWP	G	G	G	A	A	G	G	G	A	A	G	G	G	A	A	A	G	A	A	G	G	G	G	A	A	G	G	G	A	A
3calcIM	G	G	G	G	A	G	G	G	G	A	G	G	G	G	A	A	A	A	A	G	G	G	G	G	A	G	G	G	G	A
3setleadAC	G	G	G	G	A	G	G	G	G	A	G	G	G	G	A	G	G	G	G	A	G	G	G	G	A	G	G	G	G	A
3interceptGS	G	G	G	A	A	G	A	A	A	A	G	G	G	A	A	G	A	A	A	A	G	A	G	A	A	G	G	G	A	A
3setairspeed	G	G	G	G	A	G	G	G	G	A	G	A	A	G	A	A	A	A	G	G	G	G	A	G	A	G	G	G	A	A
2calcdistRW	G	G	G	G	A	A	A	A	A	G	G	G	G	G	A	A	A	A	A	G	A	A	G	G	A	G	G	G	G	A
3cleardescent	G	G	G	G	A	G	G	G	G	A	G	A	G	G	A	A	A	G	G	G	G	G	G	G	A	G	G	G	A	A
2manageWP	G	G	G	A	A	G	G	G	A	A	G	G	G	G	A	A	G	A	A	G	G	G	G	A	A	G	G	G	A	A
2cleardescent	G	G	G	A	A	A	G	G	A	G	G	G	G	A	A	G	G	A	G	G	G	G	G	G	A	G	G	G	A	A
1manageWP	G	G	G	A	A	G	G	G	A	A	G	G	G	A	A	G	G	G	A	A	G	G	G	A	A	G	G	G	A	A
1cleardescent	G	G	G	G	A	G	A	A	G	G	G	G	A	G	A	A	G	G	G	G	G	G	G	G	A	G	G	G	G	A
3directWP	G	G	G	A	A	G	G	G	G	A	G	G	A	G	A	A	G	A	A	G	G	G	A	A	A	G	G	G	A	A

Table 4.1 continued: Authority allocations for minimizing inter-cluster information transfer with varying constraints and responsibility allocations. (G=Ground cluster, A=Air cluster)

2directWP	G	G	G	A	A	G	G	G	G	A	G	A	A	G	A	A	A	A	A	G	G	A	A	A	A	G	G	G	A	A
1directWP	G	G	G	A	A	A	A	A	G	G	G	A	A	G	A	A	G	A	A	G	G	G	G	A	A	G	G	G	A	A
2interceptGS	G	G	G	A	A	A	A	G	A	A	G	A	A	A	A	G	A	A	A	G	G	G	G	A	A	G	G	G	A	A
2commOPD	G	G	G	A	A	A	G	A	G	A	G	A	G	G	A	A	G	A	G	G	G	G	G	G	A	G	G	G	G	A
2setairspeed	G	G	G	G	A	G	G	G	G	G	G	A	A	A	A	A	A	A	G	G	G	G	A	A	A	G	G	G	A	A
1interceptGS	G	G	G	A	A	G	A	G	A	A	G	A	A	A	A	A	A	A	A	A	G	A	G	A	A	G	G	G	A	A
1commOPD	G	G	G	G	A	G	G	A	G	G	G	G	A	A	A	G	A	A	G	A	G	G	G	A	A	G	G	G	A	A
1setairspeed	G	G	G	G	A	G	G	G	A	A	G	G	G	G	A	G	G	G	G	A	G	G	G	A	A	G	G	G	G	A
1calcdistRW	G	G	G	G	A	A	A	A	A	G	G	G	G	A	A	A	A	G	G	G	G	G	G	A	A	G	G	G	G	A
3calcdistRW	G	G	G	G	A	G	G	G	A	G	G	G	G	G	A	A	A	G	G	A	G	G	G	G	A	G	G	G	G	A
2setleadAC	G	G	G	G	A	G	G	G	G	G	G	G	G	G	A	G	G	G	G	A	G	G	G	G	A	G	G	G	G	A
1setleadAC	G	G	G	G	A	G	G	G	G	A	G	G	G	G	A	G	G	G	G	A	G	G	G	G	A	G	G	G	G	A
1calcIM	G	G	G	G	A	G	G	G	G	A	G	G	G	G	A	A	A	A	A	G	G	G	A	G	A	G	G	G	G	A

4.3.1.2 Maximize Inter-Cluster Information Transfer

While the previous section discussed the resulting authority allocations for minimizing information exchange between the Air and the Ground clusters, this section studies the synthesis of authority allocations with the objective function of maximizing the information transfer between the two clusters. Such an objective may stem from the need for error checking and redundancy to increase the overall safety of the concept of operations where 'shared situation awareness' or cross-checking between the agents is important.

Figure 4.11 summarizes the inter-cluster information transfer for the 30 allocations of authority resulting from the varying responsibility allocations and constraints. Predictably, the average information transfer across all 30 allocations is nearly triple that of the average information transfer obtained with the objective of minimization of information transfer.

For the case when no taskload constraints are imposed, and responsibility for all actions is allocated entirely to either the Air cluster or the Ground cluster, the resulting allocations of authority result in monitoring information transfer because all the actions are allocated to the cluster that is not given responsibility to maximize the need for information transfer for monitoring. Interestingly, this unconstrained allocation results in no primary authority inter-cluster information transfer because it would have required sacrificing authority-responsibility mismatches, thereby reducing the possible inter-cluster monitoring information transfer.

Other constraints result in allocations of authority that vary both the amount of monitoring and the amount of inter-cluster information transfer. For example, the constraint requiring capping the total monitoring on the Ground cluster results in the inter-cluster information transfer being composed of both primary authority and monitoring components. In this constraint, the allowable monitoring taskload on the Ground cluster is capped at 550, which is the mean value between the unconstrained case and the balanced monitoring case. Since the capping of allowable monitoring load on the Ground cluster prohibits maximizing authority-responsibility mismatches, and instead allocates authority for certain actions to the same cluster that has responsibility for them, the optimal allocation of authority (for those responsibility allocations that do not allocate responsibility entirely to the Air cluster), involves allocating authority to both the Air and the Ground clusters.

The constraint requiring total taskload to be balanced across the Air and Ground clusters results in the same information transfer behavior as the unconstrained case. Since the monitoring behavior studied in this case study imposes the same taskload on the cluster as the corresponding primary authority action, this constraint treats the monitoring actions and primary authority actions as equivalent from the taskload perspective. Thus, the maximum inter-cluster information transfer under this constraint is found where responsibility allocation is allocated entirely to one cluster. When total taskload is required to be balanced across both clusters, and the responsibility for all actions is allocated entirely to one cluster, the resulting authority allocation simply allocates primary authority for all actions to the cluster without responsibility for any actions.

Finally, whereas dividing up the authority allocation between two clusters is a possible way to increase the resulting inter-cluster information transfer, the figure shows that authority-responsibility mismatches also provide additional inter-cluster information transfer through monitoring requirements. Thus, a combination of dividing primary authority allocation between cluster to generate primary authority information transfer and authority-responsibility mismatches to generate commensurate monitoring information transfer can result in allocations of authority that enable the concept of operations to meet the objective of maximizing inter-cluster information transfer.

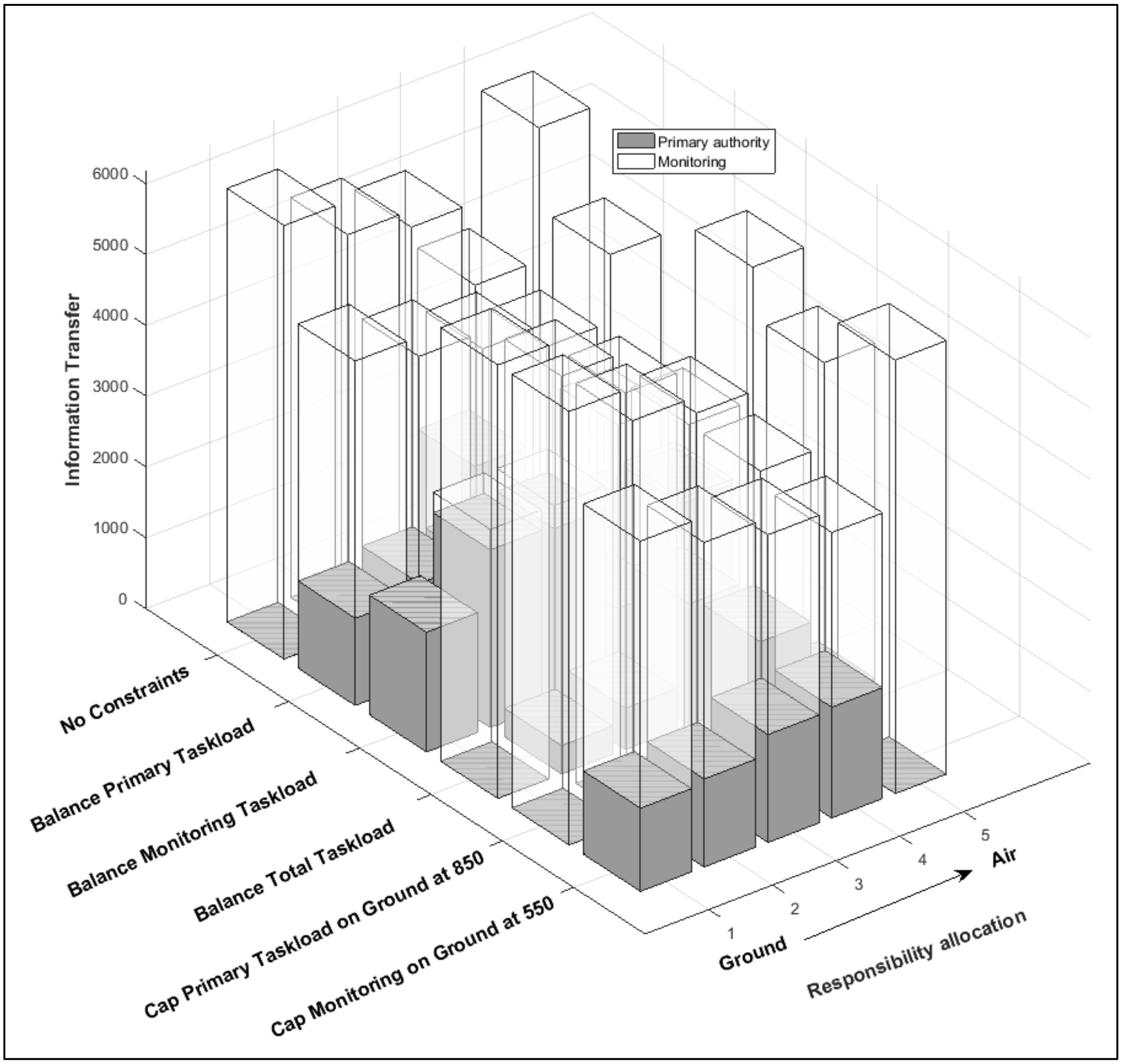

Figure 4.11: Information transfer obtained by maximizing inter-cluster information transfer with varying responsibility allocation and constraints on taskload

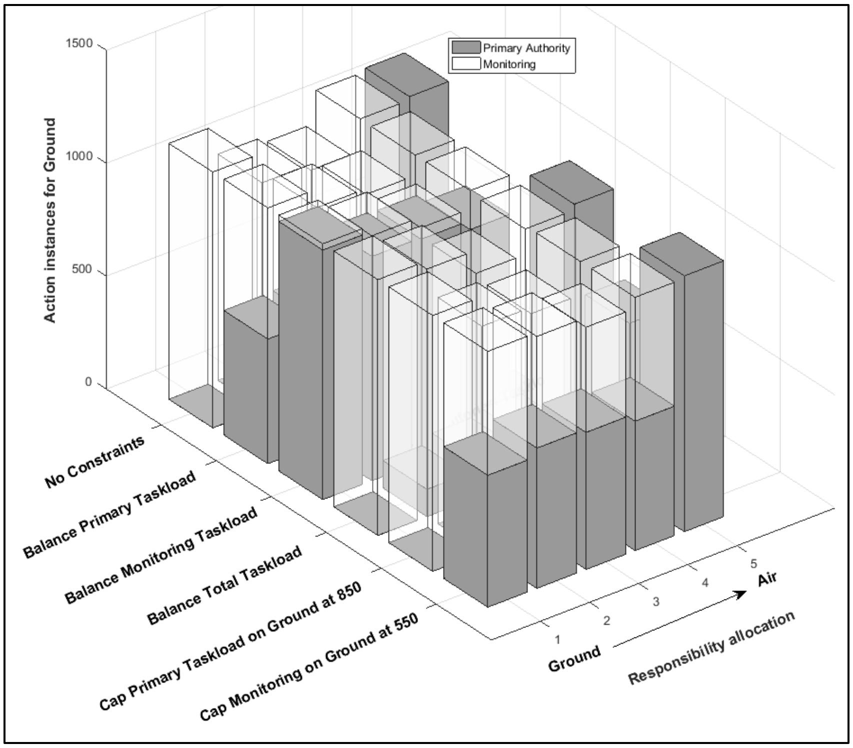

Figure 4.12: Taskload imposed upon the Ground cluster for maximizing inter-cluster information transfer with varying responsibility allocation and constraints

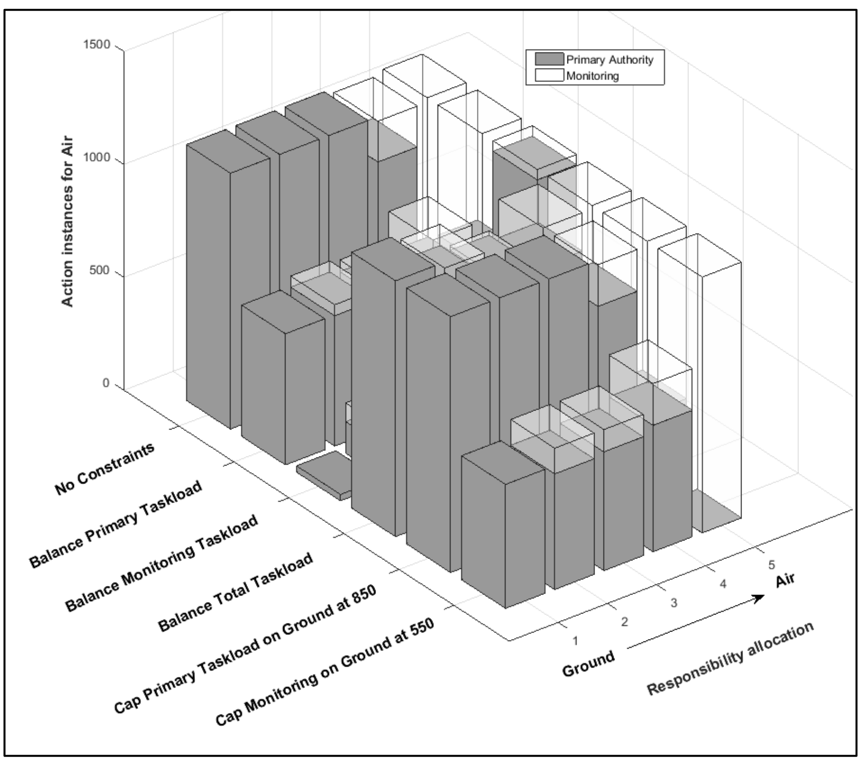

Figure 4.13: Taskload imposed upon the Air cluster for maximizing inter-cluster information transfer with varying responsibility allocation and constraints

Figure 4.12 and Figure 4.13 report the taskload imposed on the Air and Ground clusters with varying constraints and responsibility allocations. These figures show a marked difference in the taskload profiles of the allocations as compared to the profiles obtained with the objective of minimizing information transfer. Previously, the constraint that capped monitoring on Ground cluster to 550, with responsibility allocations that allocated the responsibility for majority of actions to the Ground cluster, resulted in the Ground cluster taking up the majority of the primary authority taskload and the Air cluster having very little taskload. In contrast, for the same responsibility allocations, these figures show the Ground cluster imposed with both more monitoring and primary authority taskload and the Air cluster imposed with more primary authority taskload.

Similarly, again considering the responsibility allocations that allocate responsibility for the majority of actions to the Ground cluster, the unconstrained case differs from the minimize information transfer objective studied in the previous section. While previously the Ground cluster was allocated the entire primary authority taskload and the Air cluster had no taskload at all, the Air cluster is now imposed with the entire primary authority taskload and the Ground cluster is imposed with monitoring taskload resulting from the actions being allocated to clusters such that the entire set of actions has authority-responsibility mismatches.

Overall, there is a higher total taskload created with maximal information transfer as compared to the case with the objective of minimizing information transfer. Since the total primary authority taskload has to be constant irrespective of authority allocation, constraints or responsibility allocation, the additional taskload can only arise from authority-responsibility mismatches. Thus, the overall higher taskload observed here is due to the monitoring taskload associated with authority-responsibility mismatches arising out of the allocations designed to maximize inter-cluster information transfer.

Table 4.2 reports the resulting allocations of authority obtained with the objective of maximizing inter-cluster information transfer under varying constraints and responsibility allocations. The impact of the choice of objective function on the resulting authority allocations is evident from this table. The case with no constraints on taskload and responsibility for the entire set of actions allocated to the Ground cluster serves as an example to illustrate this point. Whereas previously the authority for all actions was allocated to the Ground cluster, now with the objective of maximizing inter-cluster information transfer, the authority for all the actions is allocated to the Air cluster to generate the most authority-responsibility mismatches.

Overall, responsibility allocations that allocated more responsibility for more actions to the Air cluster result in the authority for more actions being allocated to the Ground cluster. Similarly, allocating responsibility for more actions to the Ground cluster results in authority for more actions being allocated to the Air cluster. This behavior is opposite to the allocations obtained with the objective of maximizing inter-cluster information transfer.

Table 4.2: Authority allocations for maximizing inter-cluster information transfer with varying constraints and responsibility allocations. (G=Ground cluster, A=Air Cluster)

Constraints	No Constraint					Balance Primary authority taskload					Balance monitoring taskload					Balance total taskload					Cap primary authority taskload on the Ground Cluster					Cap monitoring taskload on the Ground cluster				
Actions\Resp	**1**	**2**	**3**	**4**	**5**	**1**	**2**	**3**	**4**	**5**	**1**	**2**	**3**	**4**	**5**	**1**	**2**	**3**	**4**	**5**	**1**	**2**	**3**	**4**	**5**	**1**	**2**	**3**	**4**	**5**
3manageWP	A	A	A	G	G	A	A	A	G	G	G	A	A	G	A	A	A	A	G	G	A	A	A	G	G	A	A	A	G	G
3calcIM	A	A	A	A	G	A	A	A	A	G	G	G	G	A	A	A	A	A	A	G	A	A	A	A	G	A	A	A	A	G
3setleadAC	A	A	A	A	G	A	A	A	A	G	A	A	A	A	G	A	A	A	A	G	A	A	A	A	G	A	A	A	A	G
3interceptGS	A	A	A	G	G	A	G	A	G	G	A	A	A	G	G	A	G	A	G	G	A	A	A	G	G	A	G	A	G	G
3setairspeed	A	A	A	A	G	G	A	G	A	G	G	A	A	A	A	A	A	A	A	G	A	A	A	G	G	A	A	G	G	G
2calcdistRW	A	A	A	A	G	G	G	G	G	A	G	G	G	G	A	A	A	A	A	G	A	A	A	A	A	G	G	G	G	A
3cleardescent	A	A	A	G	G	A	A	A	A	G	G	A	A	G	G	A	A	A	A	G	A	A	A	G	G	A	A	A	G	G
2manageWP	A	A	A	G	G	A	A	A	G	G	A	A	A	G	G	A	A	A	G	G	A	A	A	G	G	A	A	A	G	G
2cleardescent	A	A	A	A	G	A	A	A	G	G	G	A	A	G	G	A	A	A	A	G	A	A	A	A	G	A	A	A	G	G
1manageWP	A	A	A	G	G	A	A	A	G	G	A	A	A	G	G	A	A	A	G	G	A	A	A	G	G	A	A	A	G	G
1cleardescent	A	A	A	G	G	G	A	A	G	G	G	A	A	A	A	A	G	A	G	G	A	A	A	G	G	A	A	A	A	G
3directWP	A	A	A	G	G	A	A	G	G	G	G	G	A	G	A	A	A	A	G	G	A	A	A	G	G	A	G	G	G	G

Table 4.2 continued: Authority allocations for maximizing inter-cluster information transfer with varying constraints and responsibility allocations. (G=Ground cluster, A=Air cluster)

2directWP	A	A	A	G	G	A	A	A	G	A	G	G	G	G	A	A	G	A	G	G	A	A	A	G	G	G	A	A	G	G
1directWP	A	A	A	G	G	A	G	A	G	G	G	A	A	G	A	A	G	A	G	G	A	A	A	G	G	G	G	G	G	G
2interceptGS	A	A	A	G	G	A	A	A	G	G	G	A	G	G	A	A	G	A	G	G	A	A	A	G	G	A	A	A	G	G
2commOPD	A	A	A	G	G	G	A	G	G	G	G	A	G	G	A	A	G	A	G	G	A	A	A	G	G	A	A	A	G	G
2setairspeed	A	A	A	G	G	G	A	A	G	A	G	A	G	G	A	A	G	G	A	G	A	A	A	A	G	A	A	A	A	G
1interceptGS	A	A	A	G	G	A	G	A	G	G	G	G	G	G	G	A	G	G	G	G	A	A	A	G	G	A	A	G	G	G
1commOPD	A	A	A	G	G	G	A	A	A	G	G	A	G	G	G	A	A	G	A	G	A	A	A	G	G	A	A	G	G	G
1setairspeed	A	A	A	G	G	A	A	A	A	G	G	A	A	G	G	A	A	G	G	G	A	A	A	G	G	A	A	G	G	G
1calcdistRW	A	A	A	A	G	G	G	G	A	A	G	G	G	G	A	A	A	G	A	G	A	A	A	A	G	G	G	G	A	G
3calcdistRW	A	A	A	A	G	G	A	A	A	G	G	A	A	A	A	A	A	A	A	G	A	A	A	A	G	A	A	A	A	G
2setleadAC	A	A	A	A	G	G	A	A	A	G	G	A	A	A	G	A	A	A	A	G	A	A	A	A	G	A	A	A	A	G
1setleadAC	A	A	A	A	G	A	A	A	A	G	G	A	A	A	G	A	A	A	A	G	A	A	A	A	G	A	A	A	A	G
1calcIM	A	A	A	A	G	A	A	A	A	G	G	G	G	G	A	A	A	A	A	G	A	A	A	A	G	A	A	A	A	G

4.4 Summary

This chapter described and demonstrated a methodology for systematic synthesis of authority allocations to reflect the performance and safety goals of concepts of operations. The methodology first framed the concept of operations as a network formed by its constituent actions. Then, the design of the authority allocation was cast as a network optimization problem with the goal of finding optimal clusters in the network of actions. The objective function and constraints of the optimal clustering were selected to reflect the performance and safety goals of the concept of operations.

The methodology was demonstrated in the case study of the previous chapter but now from the perspective of synthesizing allocations instead of analyzing them. The goal of the case study was to allocate authority for all actions to two clusters: Air and Ground. Two objective functions were studied: minimize inter-cluster information transfer and maximize inter-cluster information transfer. Five different allocations of responsibility were studied, ranging from responsibility for all actions allocated entirely to the Ground cluster to responsibility for all actions allocated to the Air cluster. Six different constraints were studied representing various requirements on the resulting taskload that could be imposed on the two clusters.

The network model of the concept of operations was parametrized by weighting its nodes and edges to represent scenarios of interest. This parametrization was made possible by a single run of the computational simulation demonstrated in the previous chapter. The simulation was only needed to be run once to capture the edge weights (representing information transfer between actions) and node weights (representing taskload associated with an action). The optimization problem was then framed in terms of metrics derived from these attributes of the network.

The optimization problem was solved in this case study using a Genetic Algorithm. The authority for the two clusters of actions could then be allocated to the Air and Ground agents. The resulting information transfer and taskload on these agents was recorded. This distribution of authority of clusters to agents depended upon the specific responsibility allocation and taskload constraints imposed on the concept of operations.

The resulting allocations of authority with associated metrics of inter-cluster information transfer and cluster taskload were found to vary significantly with the responsibility allocation and the different constraints on allowable taskload. Thus, the value of the methodology proposed here is reflected in its ability to quickly examine appropriate allocations of authority given required responsibility allocations, and to do so comparatively early in design, before the prototyping and detailed specification of training and procedures inherent to later-in-design testing methods. Further, since the objective functions and constraints were chosen to reflect the performance and safety goals of the concept of operations, the results indicate that there is no one optimum allocation of authority. Rather, the allocation of authority has to be guided by the performance and safety goals of the concept of operations.

CHAPTER 5 - CONCLUSIONS

5.1 Summary

This thesis examined air traffic concepts of operations calling for novel allocations of authority and responsibility. Specifically, this thesis proposed and demonstrated systematic methodologies to analyze and synthesize these allocations. The allocation of authority determines which agent will execute each action, and thus is the most directly observable; further, it generally drives what system-level behaviors and performance will emerge within the operation. However, a full understanding of the agents' collective work needs to also account for the monitoring required when different agents have authority and responsibility for the same action.

This thesis first proposed a methodology to analyze allocations of authority and responsibility. The methodology represents concepts of operation building on what is known early in their design (the constraints due to the physics of flight and the mid-level models of the actions required), and on what can be specified early in their potential allocations of authority and responsibility. This representation allows for the simulation of concepts of operation before more detailed design aspects are known (such as specific algorithms underlying various technologies). Thus, the "work" of the concept of operations is not yet tied to any one agent, and the authority and responsibility for actions can be fluidly assigned – at any time during the simulation – to simulate a wide range of possible function allocations.

While the specific demands on the agents can be predicted using simulation, the general trends in the results can also be predicted by the qualitative attribute of the coherence of the allocation. Coherent allocations assign actions such that each agent performs actions that contribute to the same general function using the same information, while incoherent allocations require agents to inter-leave their actions and to frequently transfer information back-and-forth. Thus, the coherence of allocations was identified as a construct to serve as a quick qualitative guiding principle for defining potential allocations.

Next, this thesis proposed a methodology for the systematic synthesis of authority allocations to reflect the performance and safety goals of concepts of operations. The methodology first frames the concept of operations as a network formed by its constituent actions. The network model of the concept of operations is parametrized by weighting its nodes and edges to represent scenarios of interest. This parametrization is made possible by a single run of the computational simulation methodology for analyzing allocations to capture the edge weights (representing information transfer between actions) and node weights (representing taskload).

The objective function and constraints of the optimal clustering are selected to reflect the performance and safety goals of the concept of operations framed in terms of metrics derived from attributes of the network. The optimization identifies clusters of actions in the network representation to subsequently guide the allocation of authority for actions to agents.

Both the analysis and the synthesis methodologies were demonstrated in case studies involving merging and spacing operations of aircraft in arrival operations. The case studies showed that emergent metrics, such as taskload and information transfer, can be used to assess allocations of authority and responsibility in concepts of operations. Further, capturing these emergent metrics in a network representation allowed the synthesis of allocations to meet the performance and safety goals of the concept of operations. These goals were represented as objectives and constraints in the synthesis of allocations of authority.

The resulting allocations of authority with associated metrics of inter-cluster information transfer and cluster taskload were found to vary significantly with the responsibility allocation and the different constraints on allowable taskload. Thus, the value of the methodology proposed here is reflected in its ability to quickly examine appropriate allocations of authority given required responsibility allocations, and to do so comparatively early in design, before the prototyping and detailed specification of training and procedures inherent to later-in-design testing methods. Further, since the objective functions and constraints were chosen to reflect the performance and safety goals of the concept of operations, the results indicate that there is no one optimum allocation of authority. Rather, the allocation of authority has to be guided by the performance and safety goals of the concept of operations

5.2 Contributions

The first objective of this thesis was achieved by creating methodologies and tools for the analysis of allocations of authority and responsibility in the design of future ATM concepts of operations. The analysis of allocations extracts emergent metrics of taskload and information transfer using computational simulation. While this methodology can be applied at any stage of the design process by modeling the agents and actions to a level of detail commensurate with the design stage, this thesis showed how analysts can predict and compare, early in design, the system-level-performance and agent-level requirements associated with different proposed allocations of authority and responsibility. Further, the rapid re-configurability of the simulation framework makes the methodology amenable to what-if experiments to test out a range of constructs that can help designers conceive of new operations and estimate their performance. Likewise, while the simulations can replace SME estimates with more systematic predictions, the SMEs have the unique capability of then valuing these predictions, and flagging results that may be problematic (e.g., excessive task load or reliance on information transfer, assumptions about monitoring).

The second objective of this thesis was achieved by demonstrating a methodology for the synthesis of allocations of authority. The synthesis problem is framed as a network optimization problem with the concept of operations modeled as a network of actions which is then parametrized using the metrics obtained from simulation. This methodology can be used to synthesize allocations of authority for a wide range of concepts of operations as it allows the performance and safety goals of the concept of operations to be represented as objective functions and constraints in terms of the attributes of the network model. Thus, the value of the methodology proposed here is reflected in its ability to quickly identify appropriate allocations of authority given required responsibility allocations, and to do so comparatively early in design, before the prototyping and detailed specification of training and procedures inherent to later-in-design testing methods.

5.3 Future Work

The work reported in this thesis can be extended in several interesting directions. While this thesis demonstrated the proposed methodologies in case studies focusing on "early in design" analysis and synthesis, the methodologies can continue to contribute throughout the later design process. In the simulations used here, the actions were executed perfectly, identifying an upper bound on what is achievable within the concept of operation. However, once the fundamental framework of the concept of operation has been explored with perfect execution, the models of the actions can then emulate in detail how specific agents or technologies might actually execute them. This might define specific requirements for technologies, or might analyze the in situ performance of specific algorithms, communication structures, or procedures. The synthesis process could then take into account these specific requirements to refine allocations of authority.

Such explorations can also help define the role of the human agents in air traffic concept of operations. For example, assumptions for human intervention in off-nominal situations can be articulated, modeled and computationally simulated. Where effective courses of action are identified in the computational simulations, they can then be further codified into the specific procedures and training accompanying the concept of operation. Further, when the clusters of actions identified in the network representation of the concept of operations are to be allocated to agents, the allocation could be guided by the identified roles of the human agents.

Likewise, as noted earlier, related studies have incorporated key aspects of human performance into the agent models such that a simulated human agent might, under conditions of high task load, choose to delay or interrupt some of their assigned actions. Such effects can then impact system performance in potentially unpredictable ways; at an extreme, low-priority or long-delayed actions may be forgotten or otherwise overcome by events.

It is important to also analyze the allocation of responsibility and its likely impact on human agents, particularly where their roles are more that of a "manager," "supervisor" or "monitor." The results in the case studies in this thesis highlighted how frequently an authority-responsibility mismatch can implicitly demand additional monitoring actions of human agents, which themselves also can require significant information transfer. Subsequent studies could explore this monitoring further, seeking to explicitly articulate what this monitoring work really demands of the human. For example, what information should the monitoring agent base the monitoring on, and how will this agent get this information? Such questions would benefit from the designer specifying not only how the execution of the primary actions should be modeled, but also how they should be monitored (and how often), for inclusion in the analysis and synthesis process.

Finally, the approach demonstrated in the case studies at the early-in-design stage might also continue to the later stages involving human-in-the-loop (HITL) testing. Such HITL testing can be expensive to develop and conduct. The computational simulations here can help in their development by exploring the range of conditions or design variables that merit HITL testing. Further, the models and simulations here, given their focus on examining the work of the agents in the concept of operations, aid in defining and describing the tasks that the humans will need to perform in the HITL. Likewise, the models and simulations of the broader concept of operations used here can also be synchronized with real-time simulators so that a test with a single (real) human operator can be embedded in a controllable representation of the broader context.

REFERENCES

[1] RTCA, Inc., “Final Report of RTCA Task Force 3: Free Flight Implementation,” RTCA, Inc., Washington DC, Oct 1995.

[2] Jardin, M.R., “Toward Real-time En Route Air Traffic Control Optimization,” Ph.D. Dissertation, Dept of Aeronautics and Astronautics, Stanford Univ., Stanford CA, 2003.

[3] Feigh, K.M. and Pritchett, A.R., “Requirements for Effective Function Allocation: A Critical Review,” *Journal of Cognitive Engineering and Decision Making*, Vol 7, No. 1, pp. 179-197, 2014.

[4] Bilimoria, K.D., "Survey of Air/Ground and Human/Automation Functional Allocation for Separation Assurance,” *12th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference and 14th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, AIAA, Indianapolis, IN, 2012.

[5] Hoffman, E., Zeghal, K., Cloerec, A., Grimaud, I., and Nicolaon, J.-P., “Is Limited Delegation of Separation Assurance Promising?” *3rd USA/Europe Air Traffic Management R&D Seminar*, Napoli, Italy, 2000.

[6] Grimaud, I., Hoffman, E., Rognin, L., and Zeghal, K., “Spacing Instructions in Approach: A Stepwise Design,” *Air Traffic Control Quarterly*, Vol 13, No. 2, 2005.

[7] Bone, R.S., “Cockpit Display of Traffic Information (CDTI) Assisted Visual Separation (CAVS): Pilot Acceptability of a Spacing Task During a Visual Approach,” *6th USA/Europe Air Traffic Management R&D Seminar*, Baltimore, Maryland, U.S.A, 2005.

[8] Mundra, A.M., Domina, D.A., Helleberg, J.R., and Smith, A.P., “Feasibility and Benefits of a Cockpit Traffic Display-Based Separation Procedure for Single Runway Arrivals and Departures,” *8th USA/Europe Air Traffic Management R&D Seminar*, Napa, California, U.S.A, 2009.

[9] Callantine, T.J., Lee, P.U., Mercer, J.S., Palmer, E.A., and Prevot, T., “Simulation of Terminal-Area Flight Management System Arrivals with Airborne Spacing,” *Air Traffic Control Quarterly*, Vol 14, No. 1, 2006.

[10] Penhallegon, W.J. and Bone, R.S., "Evaluation of a Flight Deck–Based Merging and Spacing Concept on En Route Air Traffic Control Operations," *7th USA/Europe Air Traffic Management R&D Seminar*, Barcelona, Spain, 2007.

[11] Barff, A., "Summary of the Results of the Mediterranean Free Flight (MFF) Programme," *Air Traffic Control Quarterly*, Vol 15, No. 2, 2007.

[12] Barmore, B.E., Bone, R.S., and Penhallegon, W.J., "Flight Deck-based Merging and Spacing Operations," *Air Traffic Control Quarterly*, Vol 17, No. 1, 2009.

[13] Wing, D. et al. "Pilot and Controller Evaluations of Separation Function Allocation in Air Traffic Management," *10th USA/Europe ATM R&D Seminar*, Chicago, USA, 2013.

[14] Lee, P. U., et al. "Free Maneuvering, Trajectory Negotiation, and Self-Spacing Concepts in Distributed Air-Ground Traffic Management," *5th USA/Europe ATM R&D Seminar*, Budapest, Hungary, 2003.

[15] Ruigrok, R., and Valenti, C., "The Impact of Aircraft Intent Information and Traffic Separation Assurance Responsibility on En-Route Airspace Capacity," *5th USA/Europe ATM R&D Seminar*, Budapest, Hungary, 2003.

[16] Martensson, J. and Rekkas, C., "Airborne Traffic Situational Awareness: Flight Trials of the In Trail Procedure Project," *Air Traffic Control Quarterly*, Vol 17, No. 1, 2009.

[17] Prevot, T., et al. "Air/Ground Simulation of Trajectory-Oriented Operations with Limited Delegation," 7th USA/Europe Air Traffic Management R&D Seminar, Barcelona, Spain, 2007.

[18] Lohr, G.W., Oseguera-Lohr, R.M., Abbott T.S., and Capron, W.R., "A Time-Based Airborne Inter-Arrival Spacing Tool: Flight Evaluation Results," *Air Traffic Control Quarterly*, Vol 13, No. 2, 2005.

[19] Henley, T. and Pywell, M., "ASAS: From Concept to Reality," *Air Traffic Control Quarterly*, Vol 13, No. 2, 2005.

[20] DiMeo, K., Kopardekar, P., Ashford, R., Lozito, S., and Mackintosh, M. A., "Shared-Separation: Empirical Results and Theoretical Implications," *4th*

USA/Europe Air Traffic Management R&D Seminar, Santa Fe, New Mexico, U.S.A, 2001.

[21] Lozito, S., Mackintosh, M.-A., DiMeo, K., Kopardekar, P., "Air-Ground Integration Experiment," *Air Traffic Control Quarterly*, Vol 9, No. 4, 2001.

[22] Callantine, T.J., Prevot, T., Smith, N., and Palmer, E., "Simulation of CTAS/FMS Air Traffic Management," *4th USA/Europe Air Traffic Management R&D Seminar*, Santa Fe, New Mexico, U.S.A, 2001.

[23] Prevot, T., et al. "Integrated Air/Ground System: Trajectory-Oriented Air Traffic Operations, Data Link Communication, and Airborne Separation Assurance," *Air Traffic Control Quarterly*, Vol 13, No. 2, 2005.

[24] Barhydt, R. and Kopardekar, P., "Joint NASA Ames/Langley Experimental Evaluation of Integrated Air/Ground Operations for En Route Free Maneuvering," 6th USA/Europe Air Traffic Management R&D Seminar, Baltimore, Maryland, U.S.A, 2005.

[25] Kopardekar, P., et al. "Feasibility of Integrating Automated Separation Assurance with Controller-Managed Aircraft Operations in the Same Airspace," *Air Traffic Control Quarterly*, Vol 17, No. 1, 2009.

[26] Mueller, E., "Experimental Evaluation of an Integrated Datalink and Automation-Based Strategic Trajectory Concept," AIAA Aviation Technology, Integration, and Operations Conference, Belfast, Northern Ireland, U.K, 2007.

[27] McNally, D. and Gong, C., "Concept and Laboratory Analysis of Trajectory-Based Automation for Separation Assurance," *Air Traffic Control Quarterly*, Vol 15, No. 1, 2007.

[28] McNally, D. et al. "A Near-Term Concept for Trajectory-Based Operations with Air/Ground Data Link Communication," *27th International Congress of the Aeronautical Sciences*, Nice, France. 2010

[29] Erzberger, H. and Paielli, R.A., "Concept for Next Generation Air Traffic Control System," *Air Traffic Control Quarterly*, Vol 10, No. 4, 2002.

[30] Andrews, J.W., Erzberger, H., and Welch, J.D., "Safety Analysis for Advanced Separation Concepts," *Air Traffic Control Quarterly*, Vol 14, No. 1, 2006.

[31] Farley, T. and Erzberger, H., "Fast-Time Simulation Evaluation of a Conflict Resolution Algorithm under High Air Traffic Demand," *7th USA/Europe Air Traffic Management R&D Seminar*, Barcelona, Spain, 2007.

[32] Coppenbarger, R.A., Mead, R.W., and Sweet, D.N., "Field Evaluation of the Tailored Arrivals Concept for Datalink-Enabled Continuous Descent Approach," *Journal of Aircraft*, Vol. 46, No. 4, pp 1200-1209, 2009.

[33] Hoekstra, J.M., van Gent, R., and Groeneweg, J., "Airborne Separation Assurance Validation with Multiple Humans-in-the-Loop," *5th USA/Europe ATM R&D Seminar*, Budapest, Hungary, 2003.

[34] Reitenbach, O., "Traffic Awareness for General Aviation (TAGA)," *Air Traffic Control Quarterly*, Vol 13, No. 2, 2005.

[35] Wing, D.J., Adams, R.J., Barmore, B.E., and Moses, D., "Airborne Use of Traffic Intent Information in a Distributed Air-Ground Traffic Management Concept: Experiment Design and Preliminary Results," *4th USA/Europe Air Traffic Management R&D Seminar*, Santa Fe, New Mexico, U.S.A, 2001.

[36] Barhydt, R., Eischeid, T.M., Palmer, M.T., and Wing, D.J., "Regaining Lost Separation in a Piloted Simulation of Autonomous Aircraft Operations," *5th USA/Europe ATM R&D Seminar*, Budapest, Hungary, 2003.

[37] Barmore, B., Johnson, E., Wing, D.J., and Barhydt, R., "Airborne Conflict Management within Confined Airspace in a Piloted Simulation of DAG-TM Autonomous Aircraft Operations," *5th USA/Europe ATM R&D Seminar*, Budapest, Hungary, 2003.

[38] Wing, D.J., Krishnamurthy, K., Barhydt, R., and Barmore, B., "Pilot Interactions in an Over-constrained Conflict Scenario as Studied in a Piloted Simulation of Autonomous Aircraft Operations," *5th USA/Europe ATM R&D Seminar*, Budapest, Hungary, 2003.

[39] Ruigrok, R.C.J., de Gelder, N., Abma, H.J., Obbink, B.K., and Scholte, J.J. "Pilot Perspective of ASAS Self-Separation in Challenging Environments," *6th*

USA/Europe Air Traffic Management R&D Seminar, Baltimore, Maryland, U.S.A, 2005.

[40] Consiglio, M.C., Hoadley, S.T., Wing, D.J., and Baxley, B.T., "Safety Performance of Airborne Separation: Preliminary Baseline Testing," *AIAA Aviation Technology, Integration, and Operations Conference*, Belfast, Northern Ireland, U.K, 2007.

[41] Consiglio, M., Hoadley, S., Wing, D., Baxley, B., and Allen, D., "Impact of Pilot Delay and Non-Responsiveness on the Safety Performance of Airborne Separation" *AIAA Aviation Technology, Integration, and Operations Conference*, Anchorage, Alaska, U.S.A, 2008.

[42] Blom, H.A.P., Obbink, B.K., Bakker, G.J., "Simulated Safety Risk of an Uncoordinated Airborne Self Separation Concept of Operation," *Air Traffic Control Quarterly*, Vol 17, No. 1, 2009.

[43] Wing, D.J. et al. "Comparison of Ground-Based and Airborne Function Allocation Concepts for NextGen Using Human-In-The-Loop Simulations," *AIAA Aviation Technology Integration, and Operations Conference*, Fort Worth, Texas, U.S.A, 2010.

[44] Vu, K. et al. "Pilot performance in trajectory-based operations under concepts of operation that vary separation responsibility across pilots, air traffic controllers, and automation." *International Journal of Human-Computer Interaction*, Vol 28, No. 2, pp 107-118, 2012.

[45] Sweiringa, K., "An Evaluation of Interval Management using Task Analysis and Work Domain Analysis," Masters' Thesis, School of Aerospace Engineering, Georgia Institute of Technology, Atlanta, USA, 2013.

[46] Bigelow, M., "Examining the relative costs and benefits of shifting the locus of control in a novel air traffic management environment via multi-agent dynamic analysis and simulation." Masters' Thesis, School of Aerospace Engineering, Georgia Institute of Technology, Atlanta, USA, 2011.

[47] Pritchett, A., Bhattacharyya, R. P., and IJstma, M., "Computational Assessment of Authority and Responsibility in Air Traffic Concepts of Operations," *Journal of Air Transportation*, In Press, 2016.

[48] IJstma, M., Bhattacharyya, R. P., and Pritchett, A., "Computational Simulation of Authority- Responsibility Mismatches in Air-Ground Function Allocation" *18th International Symposium on Aviation Psychology*, Dayton, Ohio, 2015.

[49] IJstma, M., Bhattacharyya, R. P., Pritchett, A., and Hoekstra, J., "Computational Assessment of Different Air-Ground Function Allocations" *11th USA/Europe Air Traffic Management R&D Seminar*, Lisbon, Portugal, 2015.

[50] Pritchett, A., and Bhattacharyya, R. P., "Modeling the Monitoring Inherent within Aviation Function Allocations," *International Conference on Human-Computer Interaction in Aerospace*, Paris, France, 2016.

[51] Clegg, C., Gray, M., & Waterson, P., "The charge of the byte brigade and a socio-technical perspective," *International Journal of Human-Computer Studies*, Vol 52, pp. 235–251, 2000.

[52] Sheridan, T. B., "Function allocation: Algorithm, alchemy or apostasy?" *International Journal of Human-Computer Studies*, Vol 52, pp. 235–251, 2000.

[53] Bailey, R. W., *Human Performance Engineering*. Englewood Cliffs, NJ: Prentice Hall, 1982.

[54] Fitts, P. (Ed.), "Human engineering for an effective air- navigation and traffic control system", (DTIC Accession No. ADB815893). Washington, DC: National Research Council, Committee on Aviation Psychology, 1951.

[55] Billings, C., *Aviation Automation: The Search for a Human- Centered approach*. Mahwah, NJ: Lawrence Erlbaum, 1997.

[56] Parasuraman, R., Sheridan, T. B., & Wickens, C. D., "A model for types and levels of human interaction with automation," *IEEE Transactions on Systems, Man, and Cybernetics– Part A: Systems and Humans*, Vol 14, 286–297, 2000.

[57] Pritchett, A., "Reviewing the role of cockpit alerting systems," In D. Harris & H. C. Muir (Eds.), *Contemporary issues in human factors and aviation safety*, pp. 47–80, London, UK: Ashgate, 2005.

[58] Pritchett, A., "Aviation automation: General perspectives and specific guidance for the design of modes and alerts," *Reviews of Human Factors and Ergonomics*, Vol 5, 82–113, 2009.

[59] Norman, D. A., "The problem of automation: Inappropriate feedback and interaction, not over-automation," *Philosophical Transactions of the Royal Society of London, B*, Vol 327, 585–593, 1990.

[60] Bainbridge, L., "Ironies of automation," *Automatica*, Vol 19, 775–779, 1983.

[61] Wiener, E. L., & Curry, R. E., "Flight-deck automation: Promises and problems," *Ergonomics*, Vol 23, 995–1011, 1980.

[62] Lee, J., & Moray, N., "Trust, control strategies and allocation of function in human-machine systems," *Ergonomics*, Vol 35, 1243–1270, 1992.

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

[64] Endsley, M., & Kiris, E., "The out-of-the-loop performance problem and level of control in automation," *Human Factors*, Vol 37, 381–394, 1995.

[65] Wiener, E., "Beyond the sterile cockpit," *Human Factors*, Vol 27, 75–90, 1985.

[66] Bass, E. J., & Pritchett, A. R., "Human–automated judge learning: A methodology for examining human interaction with information analysis automation," *IEEE Transactions on System Man and Cybernetics, Part A: Systems and Humans*, Vol 38, 759–776, 2008.

[67] Muir, B., "Trust in automation: Part I. Theoretical issues in the study of trust and human intervention in automated systems," *Ergonomics*, Vol 37, 1905–1922, 1994.

[68] Muir, B., & Moray, N., "Trust in automation: Part II. Experimental studies of trust and human intervention in a process control simulation," *Ergonomics*, Vol 39, 429–460, 1996.

[69] Woods, D., “Cognitive technologies: The design of joint human–machine cognitive systems,” *AI Magazine*, Vol 6, No. 4, 86–92, 1985.

[70] Woods, D. D., & Hollnagel, E., *Joint cognitive systems: Patterns in cognitive systems engineering*. Boca Raton, FL: CRC Press, 2006.

[71] Christofferson, K., and Woods, D., “How to make Automated Systems Team Players,” In E. Salas (Ed.), *Advances in Human Performance and Cognitive Engineering Research*, Vol. 2, pp. 1-12, Bingley, UK: Emerald Group, 2002.

[72] Miller, C. A., & Parasuraman, R., “Designing for flexible interaction between humans and automation: Delegation interfaces for supervisory control,” *Human Factors*, Vol 49, 57–75, 2007.

[73] Ockerman, J., & Pritchett, A., “A review and reappraisal of task guidance: Aiding workers in procedure following,” *International Journal of Cognitive Ergonomics*, Vol 4, 191–212, 2000.

[74] Feigh K., and Pritchett, A., “Requirements for Effective Function Allocation: A Critical Review,” *Journal of Cognitive Engineering and Decision Making*, Vol. 8, No. 1, pp. 23-32, 2014.

[75] Pritchett, A., Kim, S., and Feigh, K., “Modeling Human–Automation Function Allocation,” *Journal of Cognitive Engineering and Decision Making*, Vol. 8, No. 1, pp. 33-51, 2014.

[76] Pritchett, A., Kim, S., and Feigh, K., “Measuring Human–Automation Function Allocation,” *Journal of Cognitive Engineering and Decision Making*, Vol. 8, No. 1, pp. 52-77, 2014.

[77] Kantowitz, B. "Attention and mental workload." *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, Vol. 44, No. 21, 2000.

[78] Pritchett, A. R., van Paassen, M. M., Wieland, F. P., and Johnson, E. N., "Aerospace Vehicle and Air Traffic Simulation." *Applied System Simulation: Methodologies and Applications*, edited by M.S. Obaidat and G.I. Papadimitriou, Springer, New York, 2003, pp. 365-389.

[79] Johnson, E.N., and Pritchett, A.R., "Generic pilot and flight control model for use in simulation studies." *AIAA Modeling and Simulation Technologies Conference and Exhibit*, AIAA, Monterey, CA, 2002, AIAA-2002-4694.

[80] Pritchett, A.R., Christmann, H.C., and Bigelow, M.S., "A Simulation Engine to Predict Multi-agent Work in Complex, Dynamic, Heterogeneous Systems." *2011 IEEE First International Multi-Disciplinary Conference on Cognitive Methods in Situation Awareness and Decision Support (CogSIMA)*, Miami, FL, 2011, pp. 136-143.

[81] Feigh, K.M. Pritchett, A.R., Mamessier, S., and Gelman, G., "Generic Agent Models for Simulations of Concepts of Operation," *Journal of Aerospace Information Systems*, Vol. 11, No. 10, 2014, pp. 623-631.

[82] Pritchett, A.R., Feigh, K.M., Kim, S.Y. and Kannan, S.K., "Work Models that Compute to Describe Multiagent Concepts of Operation: Part 1," *Journal of Aerospace Information Systems*, Vol. 11, No. 10, 2014, pp. 610-622.

[83] Mesbahi, M., and Egerstedt, M., *Graph Theoretic Methods in Multiagent Networks*. Princeton University Press, 2010.

[84] Godsil, C., and Royle, G., *Algebraic Graph Theory*. Springer Science & Business Media, 2013.

[85] Buluc, A., et al. "Recent Advances in Graph Partitioning," In *Algorithm Engineering: Selected Results and Surveys*, LNCS 9220. Springer-Verlag, 2015

[86] Kahng, A. B., Lienig, J., Markov, I. L., and Hu, J., *VLSI Physical Design - From Graph Partitioning to Timing Closure*. Springer, 2011.

[87] K. S. Camilus and V. K. Govindan. A Review on Graph Based Segmentation. IJIGSP, 4:1–13, 2012.

[88] Newmanm M., *Networks: An Introduction*. Oxford University Press, Inc., New York, NY, USA, 2010.

[89] J. Li and C.-C. Liu. Power System Reconfiguration based on Multilevel Graph Partitioning. PowerTech, pp 1-5, 2009.

[90] Junker, B., and Schreiber, F., *Analysis of Biological Networks*. Wiley, 2008.

[91] Newman, M., “Community Detection and Graph Partitioning,” *Europhysics Letters*, Vol. 103, No. 2, 2013.

[92] Donath, W. E., and Hoffman, A. J., “Algorithms for Partitioning of Graphs and Computer Logic Based on Eigenvectors of Connection Matrices,” *IBM Technical Disclosure Bulletin*, Vol 15, No. 3, pp 938–944, 1972.

[93] Donath, W. E., and Hoffman, A. J., “Lower Bounds for the Partitioning of Graphs,” *IBM Journal of Research and Development*, Vol 17, No. 5, pp 420–425, 1973.

[94] Fiedler, M., “A Property of Eigenvectors of Nonnegative Symmetric Matrices and its Application to Graph Theory,” *Czechoslovak Mathematical Journal*, Vol 25, No. 4, pp 619–633, 1975.

[95] Land, A. H., and Doig, A. G., “An Automatic Method of Solving Discrete Programming Problems,” *Econometrica*, Vol 28, No. 3, pp 497-520, 1960.

[96] Ford, L. R., and Fulkerson, D. R., “Maximal Flow through a Network,” *Canadian Journal of Mathematics*, Vol 8, No. 3, pp 399-404, 1956.

[97] Kim, J., Hwang, I., Kim, Y. H., and Moon, B. R., “Genetic Approaches for Graph Partitioning: A Survey,” In *13th Genetic and Evolutionary Computation (GECCO)*, pages 473–480. ACM, 2011.