System Modeling to Identify Agent Functions in Complex Human-Integrated Systems: An Application to Air Transportation

Steven J. Landry

School of Industrial Engineering
School of Aeronautics and Astronautics (by courtesy)
Purdue University
West Lafayette, IN 47907, USA

ABSTRACT

A general method for identifying function allocation in human-integrated systems is discussed and applied to the conflict detection and resolution function in air traffic control. The method involves creating a top-down, hybrid state model, where either human or automated agents must perform the functions of detecting the current system state, controlling the current system state, detecting the minimum time available for controlling the current system state, and controlling the minimum time available for controlling the current system state. These functions are considered necessary and sufficient. Allocation of the functions between human and automation can then be made based on the relative abilities of the humans and automation with respect to system performance and system safety.

Keywords: System Modeling, Human Systems Integration, Systems Safety, State-based Modeling, Function Allocation

INTRODUCTION

There is no accepted, definitive method for function allocation in complex human-integrated systems, although many methods have been proposed (Dearden, Harrison, & Wright, 1998; Drury, 1994; Hancock & Scallen, 1998; A. Lagu & Landry, 2011; A. V. Lagu & Landry, 2013; Parasuraman, Mouloua, Molloy, & Hilburn, 1993; Scallen & Hancock, 2001; Sheridan, 1998; Wright, Dearden, & Fields, 1998). Moreover, arguments over whether such systems can or should be fully automated have not been fully resolved. That is, it is generally accepted that, as a society, we are uncomfortable with fully automated safety-critical systems, but it is not clear in such systems exactly what safety benefit is provided by the human, since humans are poor at monitoring the automation that often exists at the heart of complex systems. The identification of engineering principles for determining the allocation of function would be helpful for system design, as would an argument that conclusively identifies a limit to what can be automated.

https://openaccess.cms-conferences.org/#/publications/book/978-1-4951-2097-8

A formal, hybrid state model can be built in a top-down fashion for complex human-integrated systems, consisting of a set of states, a criterion that determines when the system is in that state, events that precipitate changes to those conditions, and transitions, both passive and active, between states (Landry, Lagu, & Kinnari, 2010). Within the model there are potentially any combination of goal states, desirable states, undesirable states, neutral states, and failure states. From the state model, all necessary agent (human or automated) functions can be comprehensively identified to correspond to one of four types: (1) detection of state, (2) control of state, (3) detection of "intensity", and (4) control of "intensity," where intensity is a measure of how quickly the system could transition to a fail state from a desirable state (Landry, 2012).

In this paper, the modeling method is described, along with its ability to identify agent functions according to the four categories given above. In addition, criteria for deciding on function allocation are discussed, with the focus of the remainder of the paper being on the control of intensity. It is argued that this function is often where humans cannot be replaced by automation, identifying a safety-critical function that must be performed by a human operator. Moreover, the technology required to automate this function, in many practical applications, does not appear to be available, which means this represents a limit on our ability to fully automate complex human-integrated systems such as air transportation, driving, process control, and medicine. An analogy to driving, along with a detailed example from the conflict detection and resolution function in air traffic control, is provided.

BACKGROUND

Function allocation

As an increasing amount and sophistication of automation has been introduced into human work systems, the question of how to best integrate humans and automation has been growing in importance. Initially, and to a large extent still, automation was used to perform tasks that the human was either physically or conceptually able to accomplish. This includes automation that replaces human physical functions such as lifting, sorting, and joining. While the automation may be stronger or faster at these tasks, the tasks are still easily conceivable to the human. Other forms of automation have been introduced to offload tasks such as monitoring and control that, again, human agents are perfectly capable of performing.

For such automation, several methods have been proposed to allocate functions between humans and automation. These methods include static, *a priori* allocation methods such as the Fitts' list (de Winter & Dodou, 2014), instance-based allocations (Dekker & Woods, 2002), and adaptable or adaptive methods that change function allocation dynamically to try to meet some performance objective (e.g., Sheridan, 2011). All these methods generally follow the guideline that humans and automation should work together, and, typically, that the human should understand the automation and be capable of supervising it. In addition, these methods are either subjective or lack a sufficiently rigorous engineering foundation that would permit implementation of the methods.

In addition, more advanced automation has been proposed or implemented that is intended to allow a system to operate in ways beyond the comprehension of the human operators. One example of this is the flight stability computers on board many modern fighter aircraft. These computers stabilize the aircraft so that it can be flown. This type of automation is inscrutable to the human in that the human is completely unable to perform the task the automation is performing. In such cases, by definition the human cannot monitor the automation. In the case of the flight stability computer, the pilot is not expected to monitor the automation. If the computer(s) fail(s), the pilot must eject.

A similar situation is occurring with the air traffic control system, where automation is being conceived to allow the air traffic control system to handle 2 – 3 times as many aircraft without incurring additional delay or degrading safety (Erzberger, 2004). However, it has been shown that controller performance at conflict detection and resolution above approximately 1.5 times the current traffic load without substantial automation support and above 2.0 times with automation support, drops precipitously (Prevot, Homola, Martin, Mercer, & Cabrall, 2012; Prevot, Homola, & Mercer, 2008). Therefore, in such a system, the ability of a controller to supervise such automation is questionable, but no function allocation concepts have been proposed that would clearly articulate the roles and responsibilities of automation vs. human operator.

State-Based Modeling of Functions in a Human-Integrated System

Consider a generic system in which humans and automation can be variously allocated function to achieve a goal of the system, and where the agents, who can be either human or automated, are also trying to ensure the system does not end up in some failure state. We can very frequently model such a system as an instance of Figure 1.

In Figure 1, the goal state α is a terminal state, meaning there is no transition from that state, indicating the goal of the system has been achieved, and the fail state Ω is a terminal state indicating system failure. Some systems may have either a goal or fail state, but not both, whereas other systems may have both goal and fail states. The system also potentially, but not necessarily, has desirable (D), neutral (N), and undesirable (U) states.

The system starts in the desirable state D. Being in state D is <u>defined as</u> condition D being true, whatever that condition is, and that, after some time $t_{goal} - t$ units of time, the system will passively transition into the goal state (α). If some control action q is applied by an agent, the system may move into a neutral (N) state or an undesirable (U) state. Neutral states are <u>defined as</u> those states where transitions cannot occur directly to goal or fail states, should either or both exist. An undesirable state is <u>defined as</u> a state where a transition to a fail state will occur after $t - t_{fail}$ units of time.

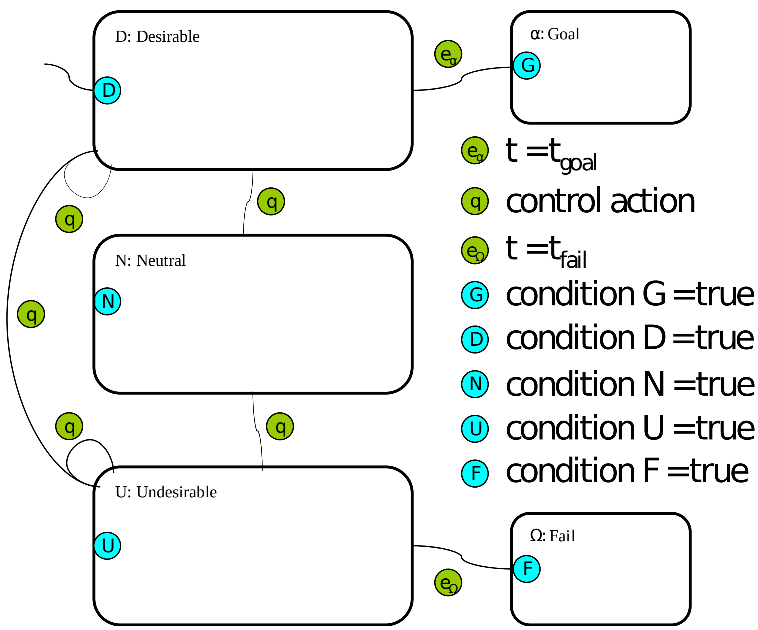

Figure 1. Model of generic human-integrated system.

An important modeling requirement is that the conditions G, D, N, U, and F, if they exist in the model, be mutually exclusive and exhaustive. This ensures that the model is complete. However, there may be multiple different complete models that can be developed, and not all models are necessarily useful.

In such a system, control actions can occur endogenously or exogenously. Exogenous control actions can cause the system to move between desirable, undesirable, or neutral states. Agents must then detect the new state and act accordingly, such as by applying an endogenous control action q to move the system back to the desirable state. Such control actions work by potentially changing the values of the conditions D, N, and U; they can also affect the values of t_{goal} and t_{fail}, including causing them to become undefined. Should the control action change the values of D, N, and/or U, they can cause the system to transition to another state. (They also may cause the system to remain in its current state.) Should the system be in the desirable state at $t = t_{goal}$, then the system will achieve its goal. Should the system be in the undesirable state at $t = t_{fail}$, then the system will fail. Note that the system cannot arrive at its goal state without being in state D, and cannot arrive at the fail state without being in state U.

https://openaccess.cms-conferences.org/#/publications/book/978-1-4951-2097-8

In this system, a basic requirement is that agents must ensure that the system move to the goal state without moving to the fail state. Moreover, there may be quality measures associated with the time to move to the goal state, the number of control actions required, the time spent in neutral or undesirable states, the time it takes for an agent to detect a neutral or undesirable state, and so on. To accomplish the basic requirement, agents must be able to do the following:

1. detect the system state;
2. if the system state is neutral or undesirable, apply a control action to move the system to a desirable (or neutral) state;
3. detect whether there exist exogenous control actions that can move the system from its current state to the fail state in less than t_{fail} - t – t_{crit} units of time, where t_{crit} is the amount of time needed to detect and resolve an undesirable system state; and
4. if such a condition exists, apply a control action such that no such exogenous control action exists.

These four functions are, we argue, a necessary and sufficient set of conditions for the system to meet the basic requirement of achieving its goal state while avoiding system failure. Such functions often also contribute greatly to the overall quality of the performance of the system, given by quality measures such as those indicated above. For brevity, function 3 is referred to as "intensity detection" and function 4 is referred to as "intensity control."

EXAMPLE APPLICATION: TRUCK PASSING ON A TWO-LANE HIGHWAY

We begin with a simple task: driving on a two-lane highway. The goal state is to reach your exit. The fail state is a collision. Note that both states are terminal; a collision cannot be "undone," and reaching the exit likewise cannot be undone. In this simple system we consider two alternative states: that a collision will occur at some time in the future, and that a collision will not occur at some time in the future. The former is our desirable state; the latter is our undesirable state. (There is no neutral state in this system.) The model is shown in Figure 2. Note that, for simplicity, I am assuming that *fuel_available* >> *fuel_required*, so only the collision/non-collision states are modeled. A complete model should include potential fuel states that could preclude the system from reading the goal state.

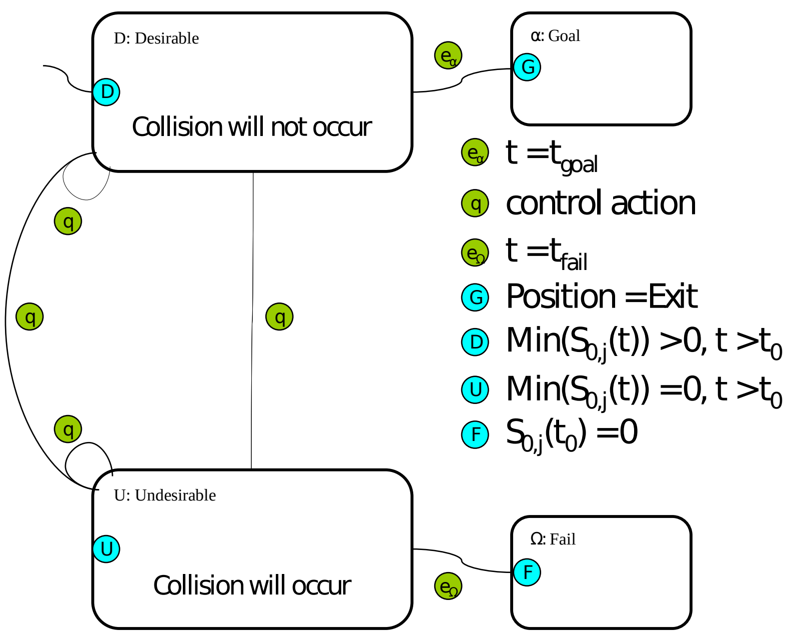

Figure 2. State model for two-lane highway driving.

In this simplified example, the conditions specified in G, D, U, and F are mutually exclusive and exhaustive, if one assumes that the separation between the "own" car and some other car j, as given by $S_{0,j}(t)$ cannot be negative. Condition D indicates that the minimum separation between the own vehicle and any other car j is greater than zero. Condition U indicates that the minimum separation at some $t > t_0$ equals zero. Condition F indicates that the current separation (at time t_0) equals zero. One of these conditions must be true, and only one of them can be true. Moreover, when the system moves to the exit, the goal state is reached and the model effectively "stops."

Consider our functions (1) – (4). Typically, the only agent performing these functions in a car is the human driver. Next, consider that the car enters the highway at 65 miles per hour (mph), intending on exiting in 20 miles. If there are no other cars on the highway, then the system is in state D. Now suppose a truck enters the highway 1 mile ahead of our "own" vehicle, just after the own vehicle enters the highway, and the truck is traveling 10 mph less than the own vehicle. Using simple algebra, we estimate that the own vehicle will overtake and collide with the truck in approximately 0.1 hours (6 minutes), after traveling a distance of 6.5 miles. The system is therefore in state U.

However, the driver of the own vehicle may not know that the truck has entered the highway 1 mile ahead, that the truck is traveling more slowly than the own vehicle, or both. In this case the driver believes the system is still in state D. At some point, it is likely that the driver of the own vehicle will recognize that the system is in state U, once seeing the truck and noticing the closure rate. (This constitutes accomplishing Function 1.) If the driver of the own vehicle does nothing, i.e. does not apply a control q, then the collision will occur. However, it is likely that the driver will apply a control q, such as slowing to match the speed of the truck or move into the next lane to pass the truck. (This constitutes accomplishing Function 2.) Once that occurs, the system moves back into state D. If the driver applies a control q such that the own vehicle speed still exceeds that of the truck, the system would remain in state U until a more effective control is applied.

Note that function 1 corresponds very specifically to monitoring for condition U. The ability of the agent to detect the true value of condition U is exactly equal to the ability of the agent to determine the correct state of the system. For function 2, the agent must identify and implement a q such that $q \rightarrow$ D. Using the dynamics of the vehicles and of the agents applying the control, the possible range of q that could accomplish this can be precisely identified.

Assume for a moment that the control action q is to slow to match the speed of the truck (55 mph). At that point functions 1 and 2 have been accomplished and the system is in state D. If the system remains in that state, the driver will eventually arrive at the goal state (the exit) after some time. This is the expected behavior of the system, should no other control actions, endogenous or exogenous, be applied. However, for Function 3 (intensity detection), the driver must determine whether there is an exogenous control that can be applied that could move the system to the fail state in less time than required for the driver of the own vehicle to detect and respond. With respect to following the truck, this corresponds to an action the truck could take that would induce a collision, e.g., slamming on its brakes when the own vehicle is "following too close." We can define this more specifically; if we assume that the maximum (constant) deceleration of both the own vehicle and the truck is equal to 15 ft/sec/sec, and the response time of the own vehicle driver is no more than 1 second, then the distance one must follow in order to ensure that such an endogenous control action cannot cause an unavoidable transition to the fail state can be computed using simple differential equations of motion. Solving those equations, one finds that the own vehicle must be at least 32.5 feet behind the truck so that a maximum deceleration by the truck will not result in an unavoidable collision. This calculation is an example of Function 3. If the own vehicle is inside of 32.5 feet and recognizes that there exists an endogenous control that would result in an unavoidable collision, and then applies a control (e.g., deceleration) to move back outside of 32.5 feet, that is an example of Function 4 (intensity control).

Accomplishing these four functions ensure that there is always a control available to prevent the system from entering the fail state. Moreover, they are necessary and sufficient with respect to preventing the system from entering the fail state.

In this example, the only agent is the human operator (driver). If automation were available, however, it would be necessary to consider how to integrate automation and human in a way that results in superior, or even acceptable, system performance. Automation that detects imminent collisions ("collision detection automation"), and even brakes automatically ("collision avoidance automation"), have been implemented, although in limited fashion.

Such automation can be mapped to the functions it performs. Collision detection automation performs function 1 by attempting to identify when the system is in state 1. Collision detection automation performs function 2 by applying control to move the system out of the undesirable state and into the desirable state. With respect to system integration, engineers should ensure that the system with the automation outperforms the system without the automation in terms of the measures of quality of interest. For our highway example, some likely quality measures would be fuel used, time, and probability of collision. The benefit in terms of probability of collision can be estimated by a quantitative comparison of the ability of the automation (vs. human) to identify condition U and to implement q such that $q \rightarrow$ D.

With respect to intensity detection, we could imagine automation that detects when one is following too closely, which, by our example calculation above, was inside of 32.5 feet. (This needn't be a static number, of course, it could be dynamically computed.) Inside of 32.5 feet, an exogenous control can result in a collision before functions 1 and 2 could be completed. Further, for intensity control, that automation could also apply brakes (q) to keep the own vehicle outside of 32.5 feet. However, keeping cars separated by 32.5 feet may reduce the throughput of the highway system; a trade-off must be made between safety and throughput. If other cars would routinely insert themselves in to the 32.5 foot gap, such an operation may even be infeasible. It is unlikely that automation would be capable of making such tradeoff decisions, since it would be difficult for automation to estimate the likelihood that the particular exogenous control would be applied, whereas human operators do this type of estimation of risk routinely. It therefore seems likely that functions 3 and 4 must be accomplished by the human operator; moreover, these functions are safety critical.

APPLICATION TO THE AIR TRANSPORTATION SYSTEM

The modeling of the air traffic system begins by identifying the functions of interest. An abstraction-decomposition hierarchy (Rasmussen, Pjeterson, & Goodstein, 1994) can be constructed to identify the purposes, abstract functions, and general functions of the air traffic system. This abstraction-decomposition hierarchy is shown in Figure 3. The hierarchy is created using the following definitions for purpose (top of the hierarchy), abstract functions (middle of the hierarchy), and general functions (bottom of the hierarchy); the physical function and physical forms sections are not modeled since they are specific to the actual form of the system, and we would prefer our model to be abstracted from the current physical form.

Table 1. Abstraction-decomposition hierarchy definitions.

Means-Ends relations	Properties represented
Goals	At this level are the goals of the system. Broad, high-level goals can be broken down into sub-goals of increasing specificity.
Abstract functions	The functions at this level are abstract in the sense that they cannot be tied to specific physical actions or a/c states. That is, one would not be able to identify a particular physical action that specifies the function. The high-level abstract functions can be broken down into sub-functions.
General functions	These functions can be tied to specific physical actions, but are generalized across particular instantiations of the system. That is, the function is not dependent on the aircraft or other sub-system in which it is executed. The high-level general functions can be broken down into sub-functions.
Physical functions (not modeled)	These functions are specific instances of the general functions within a particular instantiation of the system. That is, the function is dependent on the aircraft or other sub-system in which it is executed. The high-level physical functions can be broken down into sub-functions.
Physical form (not modeled)	These are things—nouns—in the system. The high-level physical form can be broken down into sub-systems and sub-entities.

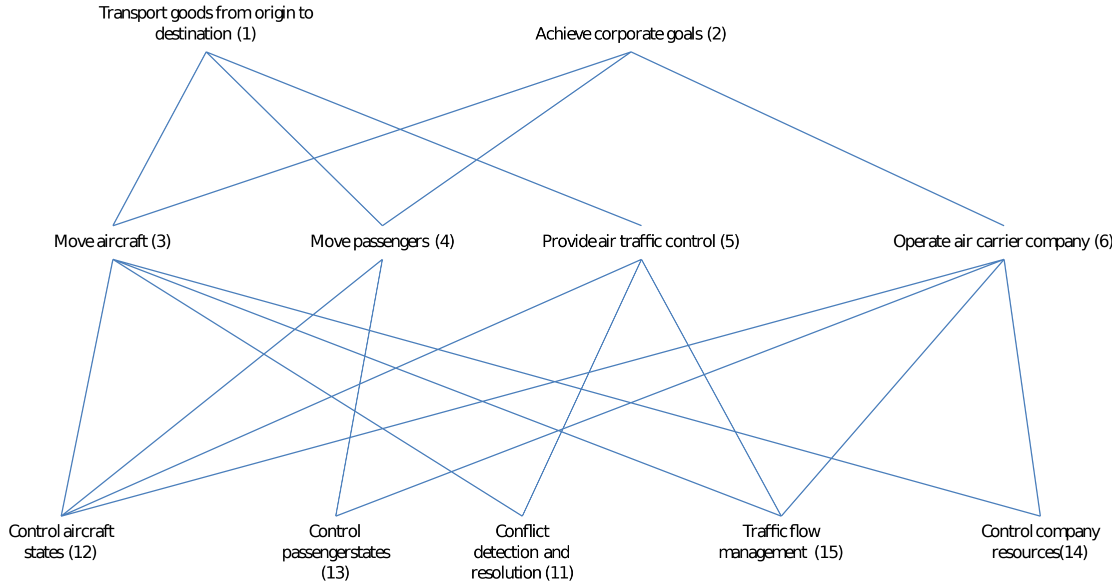

Figure 3. Abstraction-decomposition hierarchy for the National Airspace System.

A state model is then constructed for each of the general functions, as shown in Figure 4. These state models interact since control actions related to one function can affect states in other functions, thereby acting as exogenous control for the related function state models. For this paper, we will focus on the conflict detection and resolution function, whose state model is shown in full detail in Figure 5.

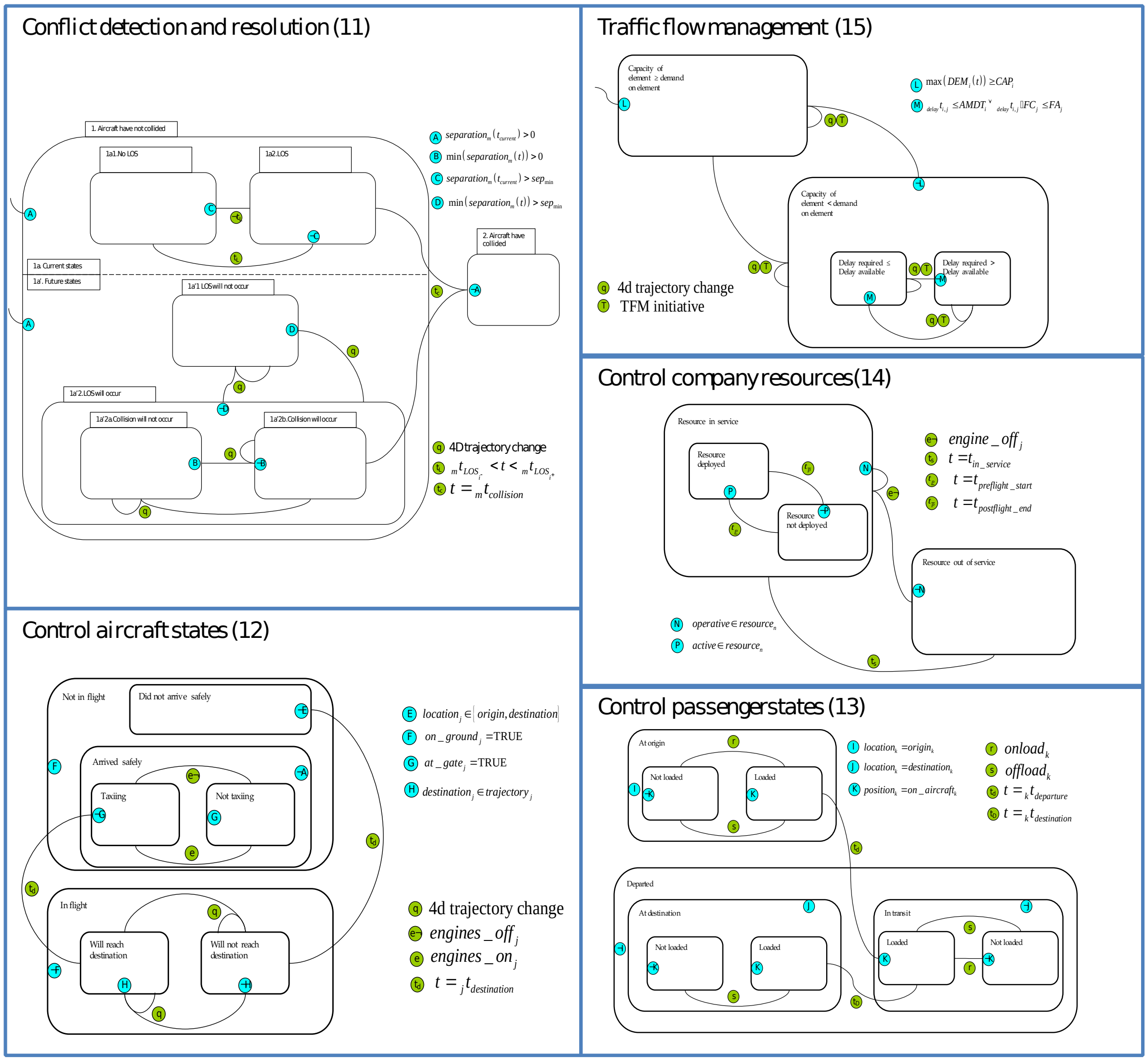

Figure 4. State model for the National Airspace System.

In Figure 5, the system has either had a collision, as approximated by a “near midair collision” (NMAC), defined by centers of mass passing within 500 feet of one another, (state 2) or not (state 1). If not, two orthogonal states must be considered for the aircraft pair – current states (states 1a) and future states (states 1a’). The aircraft pair either has proper separation (1a1) or not (1a2) and, simultaneously, either will have proper separation (1a’1) or not (1a’2). If not, one must further decompose the state into will have an NMAC (1a’2b) or not (1a’2a). This model can be shown to be a complete model of the conflict detection and resolution function (Landry, et al., 2010).

Mapping our four functions to Figure 5, the agents in the system must (1) detect state, (2) control state, (3) detect intensity, and (4) control intensity. These functions are accomplished in the current system by the agents indicated in Table 2Table 2. Current and proposed automation mapping to conflict detection and resolution functions.. Proposed automation for accomplishing these functions is also shown in Table 2.

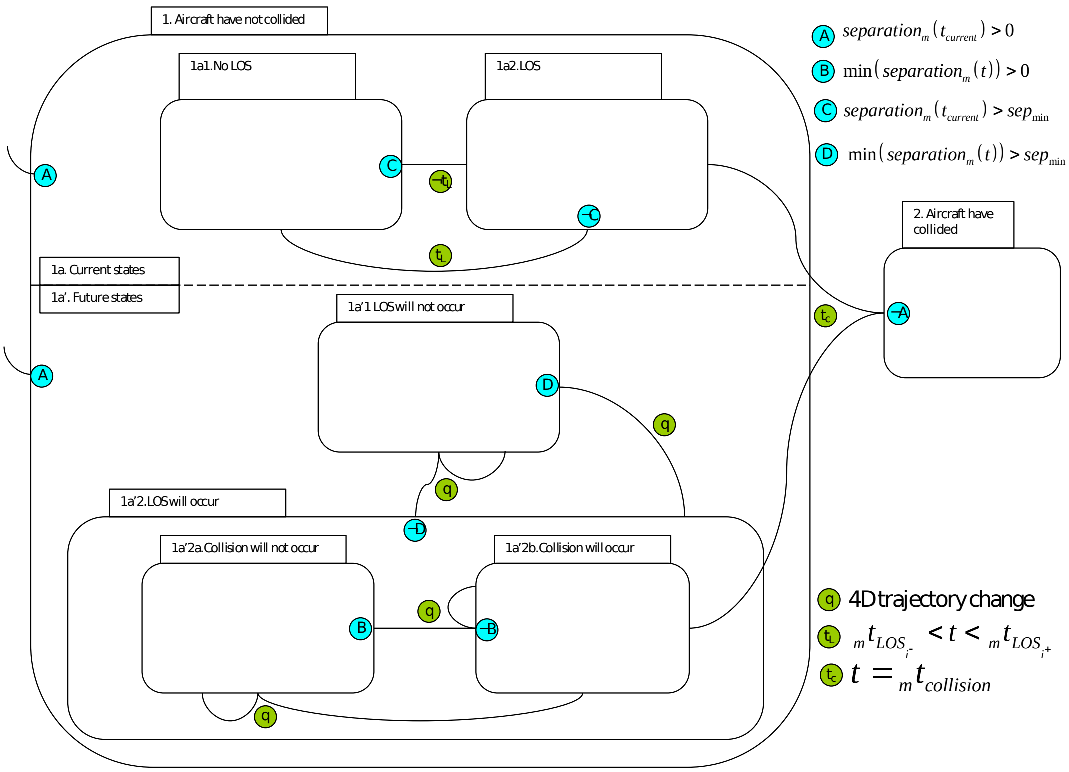

Figure 5. State model for the conflict detection and resolution function.

Table 2. Current and proposed automation mapping to conflict detection and resolution functions.

Current system							
1			2			3	4
1a2	1a'2	1a'2b	1a2	1a'2	1a'2b		
Controller, OEDP	Controller, conflict alert, conflict probe	Pilot, ACAS	Controller, pilot	Controller, pilot	Pilot, ACAS	Controller	Controller

Additional proposed automation							
1			2			3	4
1a2	1a'2	1a'2b	1a2	1a'2	1a'2b		
None	Improved conflict probes	None	None	Autoresolver, TSAFE, airborne systems	None	None	None

The "OEDP" system (Operational Error Detection Program) is a system designed to detect losses of separation. "ACAS" (Airborne Collision Avoidance System) is designed to detect excessive closure rate between aircraft and provide verbal resolution instructions to the pilot. "Conflict alert" is an existing system to predict losses of separation based on dead reckoning predictions of future positions, although it is highly error prone. "Conflict probe" is an existing system, although not fully implemented or utilized, to better predict losses of separation; improved versions are proposed. The "autoresolver" is a system that provides resolutions to predicted losses of separation (approximately) 5 – 12 minutes in advance of the predicted loss of separation (Erzberger & Paielli, 2002), and "TSAFE" (Tactical Separation Assured Flight Environment) provides tactical resolutions to predicted losses of separation (approximately) 5 minutes in advance (Paielli, Erzberger, Chiu, & Heere, 2009). Numerous airborne systems have also been proposed (e.g., Barhydt, Eischeid, Palmer, & Wing, 2003; Bilimoria, Sheth, Lee, & Grabbe, 2003; Brooker, 2004; Consiglio, Carreno, Williams, & Munoz, 2008; Krozel, Peters, & Bilimoria, 2000).

In the current system, the agents' responsibility for a particular function can be compared to that agents' capability to perform that function, and changes or additions to those allocations can be examined for their effect on the quality measures of the system. In the future system, these allocations may no longer be possible or advisable, and so the entire allocation portfolio must be reconsidered. Consider functions 1 and 2 for state 1a'2. In the current system, the controller performs function 1 with little help from automation, and identifies and instructs the pilot on the resolution (function 2). In the new system, this function, as indicated above, cannot be performed by the controller. Moreover, the controller cannot supervise the automation that performs it. Therefore, the proposed automation must be capable of performing the task without fail, or a "graceful" degradation path must be identified so that the controller can resume control when capable of doing so. To date, no such claim can be made for the proposed automation, suggesting that the concept is, as yet, incomplete.

In considering intensity detection and resolution, controllers currently appear to perform these functions manually, although the extent to which they do this is not well known or studied. In the future system, it is not possible for controllers to perform this function, as implied by the research results referred to previously. Moreover, automated methods to perform this task have been shown to likely yield reduced airspace capacity (Kim, Landry, & Torjek, 2009). This suggests that these functions <u>cannot</u> be automated; they are safety-critical functions that must be allocated to the human.

DISCUSSION: INTENSITY DETECTION AND CONTROL

As indicated in the above examples, it appears that intensity detection and control, where one is attempting to ensure that there is always an available control to prevent the system from entering a fail state, is problematic from an automation standpoint. Specifically, it appears that, frequently, attempting to automate intensity detection and control results in infeasible or undesirable system behavior. In the case of the highway example, ensuring control of intensity may result in a reduced highway throughput. In the case of the air traffic example, automating intensity control results in reduced air traffic capacity. Neither of these are feasible.

There is therefore a tradeoff between ensuring system safety through intensity detection and control and the quality or feasibility of the system to achieve its performance goals. Such tradeoffs seem to necessarily be the purview of the human operators of the system, and may indicate a limit to where automation is useful.

CONCLUSIONS

A method has been proposed to rigorously model a complex human-integrated system. That method clearly articulates four functions that must be performed by agents in the system. Different combinations of automated and human agents can be considered for these four functions to obtain feasible and desirable allocations of functions between humans and automation. Moreover, these four functions can be associated with quantifiable criteria and the relative abilities of agents to identify those quantifiable criteria.

Analysis of the subsequent function allocations can yield insight into functions that must be, or cannot be, automated. In the examples above, several functions were identified that would be required to be automated, such as the conflict detection and resolution functions in the air traffic system, and several functions were identified that could not be automated, such as the intensity functions (functions 3 and 4) associated with the highway and air traffic examples.

REFERENCES

Barhydt, R., Eischeid, T. M., Palmer, M. T., & Wing, D. J. (2003). Use of a prototype airborne separation assurance system for resolving near-term conflicts during autonomous aircraft operations. *Proceedings of the AIAA Guidance, Navigation and Control Conference, Austin, TX, USA, AIAA*.

Bilimoria, K. D., Sheth, K. S., Lee, H. Q., & Grabbe, S. R. (2003). Performance evaluation of airborne separation assurance for free flight. *Air Traffic Control Quarterly, 11*(2), 85-102.

Brooker, P. (2004). Airborne separation assurance systems: Towards a work programme to prove safety. *Safety Science, 42*, 723-754.

Consiglio, M. C., Carreno, V., Williams, D. M., & Munoz, C. (2008). Conflict prevention and separation assurance in small aircraft transportation systems. *Journal of Aircraft, 45*(2), 353.

de Winter, J. C. F., & Dodou, D. (2014). Why the Fitts list has persisted throughout the history of function allocation. *Cognition, Technolgoy, & Work, 16*, 1-11.

Dearden, A., Harrison, M., & Wright, P. (1998). Allocation of function: scenarios, context and the economics of effort. *International Journal of Human Computer Studies, 52*(2), 289-318.

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

Drury, C. G. (1994). Function allocation in manufacturing. *Contemporary Ergonomics*, 2-2.

Erzberger, H. (2004, August). *Transforming the NAS: The next generation air traffic control system.* Paper presented at the 24th International Congress of the Aeronautical Sciences, Yokohama, Japan.

Erzberger, H., & Paielli, R. A. (2002). Concept for next generation air traffic control system. *Air Traffic Control Quarterly, 10*(4), 355-378.

Hancock, P. A., & Scallen, S. F. (1998). Allocating functions in human-machine systems. In R. Hoffman, M. Sherrick & J. Warm (Eds.), *Viewing Psychology as a Whole: The Integrative Science of WIlliam M. Dember* (pp. 509-539). Washington, D.C.: American Psychological Association.

Kim, T., Landry, S. J., & Torjek, G. (2009, September). *Efficient control of collision risk through mitigation.* Paper presented at the 10th Annual Aviation Technology, Integration, and Operations Technical Forum, Hilton Head, SC.

Krozel, J., Peters, M., & Bilimoria, K. (2000). A decentralized control strategy for distributed air/ground traffic separation. *AIAA Guidance, Navigation, and Control Conference*.

Lagu, A., & Landry, S. J. (2011). Roadmap for the next generation of dynamic function allocation theories and strategies. *Human Factors and Ergonomics in Manufacturing, 21*, 14-28.

Lagu, A. V., & Landry, S. J. (2013). Adaptive function allocation stabilization and a comparison of trigger types and adaptation strategies. *International Journal of Industrial Ergonomics, 43*, 439-449.

Landry, S. J. (2012). *Intensity control: A concept for automated separation assurance safety and function allocation in NextGen*. Paper presented at the 13th Annual Aviation Technology, Integration, and Operations Technical FOrum.

Landry, S. J., Lagu, A., & Kinnari, J. (2010). State-based modeling of continuous human-integrated systems: An application to air traffic separation assurance. *Reliability Engineering and Safety Science, 95*, 345-353.

Paielli, R. A., Erzberger, H., Chiu, D., & Heere, K. (2009). Tactical conflict alerting aid for air traffic controllers. *Journal of Guidance, Control, and Dynamics, 32*(1), 184-193.

Parasuraman, R., Mouloua, M., Molloy, R., & Hilburn, B. (1993). *Adaptive function allocation reduces performance costs of static automation.* Paper presented at the Proceedins of the Seventh International Symposium on Aviation Psychology, Columbus, OH.

Prevot, T., Homola, J., Martin, L., Mercer, J., & Cabrall, C. D. (2012). Toward automated air traffic control - Investigating a fundamental paradigm shift in human/systems interaction. *International Journal of Human-Computer Interaction, 28*, 77-98.

Prevot, T., Homola, J., & Mercer, J. (2008, 18 - 21 August). *Initial study of controller/automation integration for NextGen separation assurance.* Paper presented at the Guidance, Navigation, and Control Conference, Honolulu, HI.

Rasmussen, J., Pjeterson, A. M., & Goodstein, L. P. (1994). *Cognitive Systems Engineering*. New York: John Wiley & Sons.

Scallen, S. F., & Hancock, P. A. (2001). Implementing adaptive function allocation. *The International Journal of Aviation Psychology, 11*, 197-221.

Sheridan, T. B. (1998). Allocating functions rationally between humans and machines. *Ergonomics in Design, 6*(3), 20-25.

Sheridan, T. B. (2011). Adaptive automation, level of automation, allocation authority, supervisory control, and adaptive control: Distinctions and modes of adaptation. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans, 41*, 662-667.

Wright, P., Dearden, A., & Fields, B. (1998). Function allocation: A perspective from studies of work practice.

International Journal of Human Computer Studies, 52(2), 335-355.