Proceedings of the Twenty-Fourth Innovative Appications of Artificial Intelligence Conference

Multi-Agent Simulation of En-Route Human Air-Traffic Controller

David Šišlák, Přemysl Volf, Michal Pěchouček,
Agent Technology Center, FEE, Czech Technical University in Prague
david.sislak@fel.cvut.cz

Christopher T. Cannon, Duc N. Nguyen and **William C. Regli**
Applied Informatics, College of Information Science and Technology, Drexel University
cannon@drexel.edu

Abstract

The Next-Generation Transportation program coordinates the evolution and transformation of the current air-traffic management (ATM) system for the National Airspace System (NAS). Currently the NAS has a limited capacity and cannot handle the increasing future air traffic demands. However, before newly proposed ATM concepts are deployed they must be rigorously evaluated under realistic conditions. This paper presents AGENTFLY, an emerging NAS-wide high-fidelity multi-agent ATM simulator with precise emulation of the human controller operation workload model and human-system interaction. The simulator is validated using a flight scenario developed by the U.S. Federal Aviation Administration that is based on real data. We present preliminary results focusing on the accuracy of the simulated controllers within AGENTFLY.

Introduction

The air-traffic management (ATM) system used in the National Airspace System (NAS) of the United States is one of the most complex aviation systems in the world (Nolan 2004) involving thousands of people and systems. Human controllers organize air-traffic flow to maintain safe airplane distances for assigned airspace sectors. The capacity of ATM depends on many factors, such as controller cognitive capacity, weather conditions, air-space availability, and airport capacities. However, during peak hours the ATM system reaches its limits. During these times, airplanes are placed in holding patterns to keep them from entering congested airspace, or delayed from takeoff. To further exacerbate this issue, Boeing has predicted that the number of cargo flights will triple within the next 20 years. The U.S. Federal Aviation Administration (FAA) estimates that the NAS and weather caused 606,500 delays (513,420 hours of delays) in 2008, leading to unnecessary fuel consumption and increased atmospheric pollution (Chin and Melone 1999).

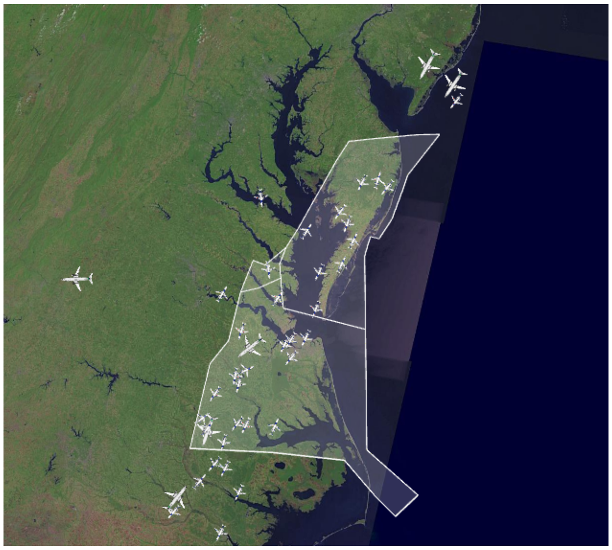


Figure 1: AGENTFLY highlighting sectors 34 and 54.

The Next-Generation Air Transportation System (NextGEN) (Wickens et al. 1998) program is designed to coordinate the evolution of ATM systems to satisfy future growth of air-traffic without losing efficiency with the aviation community. ATM tools developed within NextGEN would lower the controller's cognitive load, maintain safety, and increase efficiency. Before deployment these concepts must be rigorously tested through simulation. Therefore there is a need for high-fidelity simulators to evaluate new concepts and compare them to current ATM procedures.

This paper presents AGENTFLY, a NAS-wide high-fidelity distributed multi-agent simulator (Volf, Šišlák, and Pěchouček 2011) shown in Figure 1. The overall goal of AGENTFLY is to provide a platform to study NextGEN concepts and perform scenario analysis to assess how to handle future air-traffic. The multi-agent approach (Wooldridge 1999) has been chosen for its natural mapping of system elements to autonomous intelligent agents (*e.g.*, each entity is an agent). This paper also uses a comprehensive evaluation methodology to identify goal-based metrics and precisely measure the performance and effectiveness of the system.

This paper is organized as follows. First we compare existing ATM simulators followed by a description of current air-traffic controller duties. We explain the architecture of AGENTFLY followed by a description of air-traffic controller models. Finally, we present our validation methodology, experimental results, conclusions and future work.

ATM Simulators

The most precise ATM simulations are carried out with human-in-the-loop (HITL) simulations where human interaction is integrated into the simulation model. HITL simulations in ATM require many people (*e.g.*, human controllers and pilots). Additionally, HITL simulations run in real-time and therefore must be limited in duration and scope. New approaches have to be studied within NAS-wide area as minor local delays can potentially cascade into large regional congestion (Tumer and Agogino 2009).

In addition to real-time HITL simulations, the simulation and analysis team at the FAA uses ATM simulators to validate new aviation concepts, technologies, and system capacity issues to evaluate the performance of both emerging and existing systems within NAS. We present three ATM simulators in use by FAA that either focus on a small piece of the NAS or provide simplified models of existing components.

The Airspace Concept Evaluation System (ACES) (George and Wieland 2011) is a non-real-time modeling and simulation environment for NAS, developed by NASA Ames Research Center. ACES is composed of interoperable models that represent the gate-to-gate actions in NAS. The ACES prototype uses an agent-based modeling framework and a distributed simulation approach called High Level Architecture (HLA). HLA is a set of processes, tools and middleware software developed to support plug-and-play assembly of independently developed models.

The National Airspace System Performance Analysis Capability (NASPAC) (Reddy 2010) is an integrated set of computer modules designed to model the entire NAS, the en-route structure and traffic flows, as a network of interrelated components, reflecting the effects of weather conditions, air-traffic control procedures, and air-carrier operating practices. The NASPAC simulation flies individual aircraft through daily itineraries and provides statistical reports on delays and observed flow rates. The NASPAC includes simplified models of en-route sectors and airports.

The Reorganized ATC Mathematical Simulator Plus (RAMS Plus) (Dvojakovski 2004) is a fast-time discrete-event simulation software package providing functionality for the study and analysis in ATM. The RAMS Plus package contains an integrated editor and display tool, rapid data development, stochastic traffic generation, 4D flight profile calculation, sectorisation, conflict detection and rule-based resolution, airspace routing, and free-flight.

The key distinction between AGENTFLY and these ATM simulators is the agent-based approach in designing high-fidelity models and interactions of human controllers and pilots. AGENTFLY can perform in real-time and/or faster-than-realtime distributed NAS-wide simulations.

En-Route Human Controller Duties

The controlled airspace in the U.S. NAS is divided into several area control centers that provide air navigation services for flights within a particular airspaces. In the NAS, there are 22 Air Route Traffic Control Centers (ARTCC) that provide services for flights in their en-route stage. ARTCCs are further subdivided horizontally and vertically into areas. Each area contains several sectors due to controller staffing purposes. A Sector is a three-dimensional volume of airspace with defined boundaries and airplane radio coverage.

Each sector with high air-traffic is covered with primary or secondary radar[1]. Every ARTCC position is equipped with a radar display covering a sector in the system. The En-Route Automatization Modernization (ERAM) system is a computer system that displays the sector map, and airplanes positions to linked textual information containing key flight data (*e.g.*, flight ID, altitude, ground speed, *etc*.). Each position is staffed by a radar controller (a.k.a. R-side controller) and often an associate (a.k.a. D-side controller) to assist with non-radar duties (*e.g.*, maintain separation between aircraft too low or far away to be displayed on the radar). In this paper, we only model the R-side controller.

The R-side controller (henceforth referred to as *controller*) monitors an en-route sector through ERAM. All duties performed by the controller are based on situation awareness gathered from ERAM or from communication with airplanes and other controllers. A controller is not able to work directly with these precise airplane dynamic models (*e.g.*, weight, fuel burn, *etc*.) as these are complex and not readily available. Controllers interact with airplanes by amending flight plans through the ERAM.

Each controller performs four key duties: (i) scanning, monitoring, and analysis, (ii) handoff, (iii) standard operating procedures, and (iv) resolving detected future conflicts. *Scanning* is the most frequent task done by a controller. During scanning a controller monitors the whole radar display to update its situational awareness in and around its sector. The controller *analyzes* new information observed from his display to identify possible future loss of separation (*e.g.*, future conflict) among airplanes. The controller *monitors* his previous clearances or requests issued to airplanes in the display.

A *handoff* procedure transfers an airplane horizontally or vertically to an adjacent sector and includes the transfer of communication frequency. The handoff is a crucial task because the controller must ensure that a conflict does not occur before an aircraft enters a sector. An airplane can enter a new sector only if either handoff is accepted or a point-out[2] is previously approved by that sector. Handoff is initiated for airplanes close to the sector boundary for which the controller has no other traffic and pending operations in its sector. Each controller must respect a sector's *Standard Operating Procedure* (SOP), which describe traffic flow restrictions among ATM components (*e.g.*, altitude restrictions).

When a controller identifies a possible conflict situation,

---

[1] In low air-traffic areas not covered by radar, non-radar procedures provide ATM services.

[2] A point-out is a procedure where an airplane can enter another sector without a previous handoff and transfer of communications.

he applies *conflict resolution* procedures to find a suitable manoeuvre to maintain airplane separation. For en-route sectors, a minimum separation is 5 nautical miles horizontally and 1,000 feet vertically airspace. Depending on the situation, a controller can resolve a conflict via changes in altitude, vectoring (*i.e.*, heading changes with a diversion from the flight plan track) and/or flight speed. The selection of a proper resolution manoeuvre considers positive separation (*i.e.*, safety assumptions about selected manoeuvres) and SOPs. As a precaution a controller always has a backup plan if something is wrong with the applied plan.

Combined Time-Stepped and Event-Driven Multi-Agent Simulation

AGENTFLY uses a multi-agent approach for simulation of the NAS system with en-route human controllers. The system simulates the physical entities controlled by agent-based models. Agent-based simulations are popular for modeling of complex multi-actor systems (Borshchev and Filippov 2004). Such multi-actor simulations can be used for studying complex emergent system behaviors from the micro-behaviors of its individual actors.

In AGENTFLY, controllers, pilots, airplanes and the ERAM are implemented as actor agents. The airplanes have high-fidelity representations based on performance models from the Base of Aircraft Data. Actor agents interact through communication channels or through the simulated environment. An agent can perceive the state of the the environment using its sensors and it can make changes in the environment through its effectors. There are two communication paths between a controller agent and a pilot agent: (i) direct (*e.g.*, radio communication) and (ii) indirect (*e.g.*, pilot movements being reflected back to the controller via the radar display). Additionally, there are a set of non-actor agents in the simulation that address environment modeling, simulation management, visualization and data collection.

AGENTFLY combines two simulation approaches: (i) time-stepped and (ii) event-driven. The time-stepped simulation advances by predefined equally-sized time steps (Wu and Gong 2001). The new states of the simulation are computed after each time step. Each round of simulation begins with sensor computation and ends with gathered agents actions. In AGENTFLY, a time-stepped approach is used for the simulation of the environment (*e.g.*, movement of airplanes, weather, *etc*). The time-step value is determined by the precise sensor reading state from the environment.

The current radar system updates its state every twelve seconds. The time-stepped simulation can be executed in real-time, suitable for HITL simulations. For faster-than-realtime execution, AGENTFLY also operates as an event-driven simulation as a basis for execution (Nicol and Yan 2004). Each event is scheduled with a time-stamp and the simulation framework processes events in time order. Events scheduled for the same time-stamps are processed based on their mutual priorities. An agent processing an event can advance non-processed events scheduled within a given interval to a later time, which simulates the duration of processed a action. By using events, the simulation is deterministic and can integrate controlled randomness into the simulation through proper random seeds. It is possible to make simulation only time-stepped (*i.e.*, no events) or purely event-driven (*i.e.*, time-steps never occur).

For large-scale simulations an effective distribution scheme is required to maintain high simulation fidelity and maximum simulation speed. In AGENTFLY, each processor in a network of computers manages the events from pilot agents and controller agents in a geographically partitioned area. The partitioned area, controlled by a single “master” process, dynamically changes based on the number of pilot agents in that area. The simulation details in AGENTFLY and results describing simulations consisting of over 50,000 flights during a single day in the NAS are presented in (Volf, Šišlák, and Pěchouček 2011).

Model of En-Route Human Controller

The en-route human controller models in AGENTFLY emulate controller operation and workload models. The workload model is based on Multiple Resource Theory (MRT) (Wickens 1984). MRT proposes that the human operator have several different pools of resources that can be tapped simultaneously. The operator must process information sequentially if tasks require the same pool of resources or in parallel if the task requires different resources. MRT theory views decreasing performance as a shortage of these different resources and models humans as having limited capability for processing information. Cognitive resources are limited by supply and demand when the individual performs two or more tasks that require a single resource. Excess workload caused by a task using the same resource can cause problems and result in slower task performance. The controller agent’s operations are emulated through the Visual, Cognitive, Auditory and Psychomotor (VCAP) (McCracken and Aldrich 1984) workload model. The visual and auditory components in the model are external stimuli. The cognitive component describes the required level of information processing. The psychomotor component describes required physical actions.

The en-route human controller duties are modeled as procedures with actions in AGENTFLY. Actions are organized into dependency chains and procedures. The procedures branch actions into several chains to be executed under different circumstances. Each particular action defines which components from the VCAP model it requires, its duration and its priority. An action can be performed if its predecessor(s) are completed and their respective VCAP components are available. When multiple actions are ready for execution at the same time, the action with the higher priority is selected. Ready actions are automatically postponed until they are selected. The duration of each action can be a fixed or based on a probabilistic model. Long-running activities (*e.g.*, a controller’s conflict resolution task) is decomposed into many actions with a short duration. The action-decomposition and processing is implemented using event-driven simulation. Each event is an action that needs to be processed. At each time-step event, an agent’s subsequent actions may be postponed by its previous actions’ execution duration and priority. The priority and duration for each action is configured externally. The FAA Human Factors Laboratory provided values to these parameters.

AGENTFLY emulates controller interactions with a simulated radar display system based on ERAM. The visual stimuli and psychomotor actions are sensor inputs of the controller model and are connected to the ERAM model. For more human-realistic models, the controller model includes the inability to scan and monitor the entire ERAM display. Internally, the radar display is partitioned into several regions and the controller's focus cycles among these regions. The time spent in the region depends on the number and complexity of performed visual stimuli provided by experts at the FAA. Whereas, the selection of the next region for focus is based on the priority model.

The controller model performs cognitive actions only based on information obtained from the available ATM tools. The controller does not have access to the internal states and plans of other components in the system. For tasks working with the airplane flight trajectories (*e.g.*, handoff, conflict detection, conflict resolution), the controller model builds a mental flight information model for each flight which is updated based on external stimuli and planned ATM control actions. This mental flight model also integrates controller predictions and uncertainness.

Besides ERAM, there is sector radio communication module. The sector radio is a half-duplex medium where only one participant can transmit at a time. To minimize the number of radio interferences, a controller checks whether the channel is free (*i.e.*, no other station is transmitting) and follows a backoff procedure to acknowledge transmission. If there is no acknowledgement (*e.g.*, the receiver is overloaded by other higher priority tasks), the message is repeated.

Validation of Designed Model

This section describes the validation process used to determine the accuracy of the ATC model within AGENTFLY. As the foundation of validating AGENTFLY we incorporate Roche's (Roche 1994) five step measurement process: (i) *formulation*, (ii) *collection*, (iii) *analysis*, (iv) *interpretation*, and (v) *feedback*. We refer to and explain these steps throughout the validation process.

To evaluate AGENTFLY, the FAA provided a system scenario, shown in Figure 1, designed to accurately simulate events and information exchanges that would occur in real air traffic control situations. The goal of the scenario is to safely navigate each aircraft to its en-route destination. The scenario takes place in Sectors 34 and 54 which represent sectors that exist in the U.S. NAS. Surrounding these sectors are Ghost Sectors for controller agents which provide external interactions. To break down the FAA scenario into its key elements we use previous work by Carroll (Carroll 2000) who identifies four important components to a scenario: *goals*, *setting*, *actors* and *plot*. *Goals* describe what users are trying to accomplish with the system. The *setting* is the operational environment. *Actors*, agents or people, initiate actions in the system. A *plot* is made up of actions and events that happen to an actor or the environment. Although Carroll's methodology is used to design a product, we propose to extend Carroll's process to evaluate an existing system after its creation. An example of applying Carroll's scenario definition to the FAA scenario is as follows:

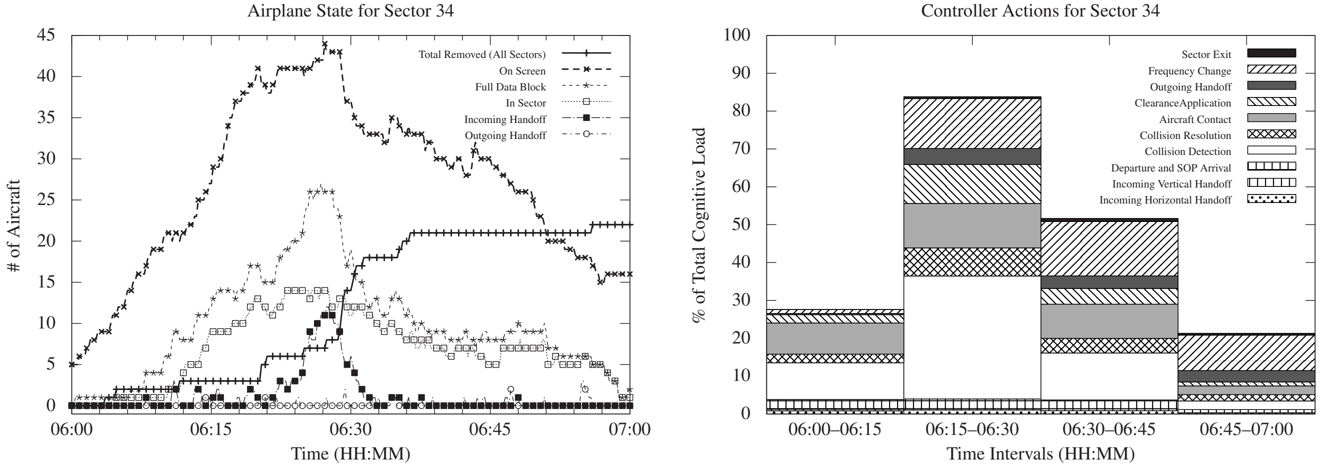
- **Goals:** Detect air traffic control events in the actively monitored sector as quickly as possible, produce the least complex resolution plan if a conflict exists, and continually monitor the execution of the resolution plan and adjust the plan when necessary to avoid a conflict.
- **Setting:** Sectors 34 and 54 within the U.S. NAS.
- **Actors:** Pilot agents and controller agents.
- **Plot:** Each controller agent has to monitor its assigned sector to ensure that each aircraft safely reaches its en route destination. The controller agent accomplishes this by coordinating with each aircraft's pilot and other surrounding controller agents.

Next in the *formulation step*, we derive goal-based metrics for the scenario by using the Basili's *et al.* (Basili, Caldiera, and Rombach 1994) Goal Question Metric (GQM) approach. Metric selection using the GQM approach begins by extracting the goals or high-level objectives from the specifications of the system. In this case, we extract these goals from Carroll's scenario definition. In the GQM approach, there are four parts to a goal: *purpose*, *issue*, *object*, and *viewpoint*. The *purpose* describes how the goal is effected. The *issue* is the measurement of the event(s). The *object* is the item that the goal is centered around. The *viewpoint* is the perspective of the entity that is affected by the goal. Next, the evaluator identifies questions—usually with quantifiable answers—that when answered, decide whether or not the system meets the goal. Finally, the metric is a set of data associated with the questions. Due to the lack of space, we apply the GQM approach to one evaluation goal with its subsequent questions and metrics:

Goal 1: Effectively issue, monitor, and/or repair (p) the execution of flight plans and/or conflict resolution plans (i) of an aircraft (o) from the view of the controller agent (v).

- **Question 1.1:** How does the state of the aircraft traffic change over time?
  - **Metric 1.1.1:** Measure the number of aircraft in each state at each time tick of the scenario.
- **Question 1.2:** What is the cognitive load of the controller over time?
  - **Metric 1.2.1:** Measure the percentage of time used by a controller for its tasks over 15 minute intervals.
- **Question 1.3:** How does a controller's cognitive load affect its ability to manage air traffic?
  - **Metric 1.3.1:** Measure the number of unaccepted handoffs over 15 minute intervals.
  - **Metric 1.3.2:** Measure the number of unresolved collisions over 15 minute intervals.

In the *collection step*, these metrics were instrumented into the AGENTFLY system that produced log files at run time with their results. The *analysis* and *interpretation steps* are discussed in the experiments section including empirical results. The *feedback step* is not included in this paper because AGENTFLY is still an emerging application and has not been put in front of the final end users.



(a) Airplane state over time.

(b) Controller actions over 15 minute intervals.

Figure 2: AGENTFLY airplane and controller simulation results in Sector 34.

Experiments

To validate the AGENTFLY system we measure the metrics identified in the validation section to determine how well AGENTFLY simulated human controllers. For these experiments we only examine one simulation in Sector 34 as it contains more air-traffic congestion than Sector 54.

First we examine Metric 1.1.1 to determine the state of the traffic being handled by the controller over time. Figure 2a shows the number of airplanes within Sector 34 at different states during the period of 6AM to 7AM. Airplanes in a sector can be in several states. *On screen* airplanes are ones currently visible on the screen. *In sector* airplanes are positioned within Sector 34's boundaries. Airplanes in *full data block* require more attention from the controller. Once an airplane enters full data block, it will remain in that state until it leaves the sector. Airplanes in transition between two sectors are in *incoming* or *outgoing* handoff state. Airplanes can be *removed* from the simulation if the controller (a) cannot resolve a future collision in time (only vertical collision resolution is implemented now), (b) receives a clearance it is not able to fulfill (expedite descent not yet implemented) or (c) the airplane enters a new sector before the handoff procedure is completed (executing holding patterns is not yet implemented). The number of airplanes *in handoff* peak between 06:15 and 06:30. During this time, the number of removed airplanes increase signalling that the controller is overloaded and unable to manage all airplanes in the sector.

Next we examine Metric 1.2.1 to determine the cognitive load of the controller. Figure 2b shows the distribution of controller's actions aggregated over 15 minute time intervals. The controller actions follow the VCAP model, some actions are processed in parallel (*e.g.*, screen scanning and radio communications) while others must be executed in sequence. The *total cognitive load* is the sum of all actions excluding screen scanning. Screen scanning is the default action and consumes most of the controller's time. Non-scanning actions (*e.g.*, collision detection/resolution and radio communication) require the controller's cognitive abilities over a period of time. During collision detection and resolution, the controller must retrieve and process a lot of information. Specifically radio communication consumes a lot of time because messages are repeatedly acknowledged.

Last we examine Metrics 1.3.1 and 1.3.2 to see how the controller's cognitive load affected its ability to manage air traffic. We analyze two critical controller events: (i) *unaccepted handoffs* which is when a controller agent is overloaded with other actions and cannot accept a handoff (*i.e.*, the aircraft cannot enter the sector and is removed) and (ii) *unresolved collisions* which is when a controller agent did not resolve a conflict using vertical resolution. The results are shown in Figure 3 which overlay total cognitive load with the number of unaccepted handoffs and unresolved collisions aggregated over 15 minute time intervals. The results show that as cognitive load increases so does the number of unaccepted handoffs and unresolved collisions, which peaks between 6:15AM–6:30AM. Unaccepted handoffs increase because the cognitive load of the agent is at such a high level that it cannot process the incoming requests as quickly as they are arriving. Unresolved collisions also increase due to the peak in cognitive load which correlates to the increase in aircraft on screen during the same time interval shown in Figure 2a. These results show that controller agents act in a similar fashion to human ATCs if they were put in the same situation and cognitive load.

Conclusions & Future Work

This paper presented AGENTFLY, a NAS-wide high-fidelity multi-agent simulator integrating precise emulation of human en-route controller operation with detailed interaction and fully configurable models with respective ATM tools. AGENTFLY allows for large-scale testing of future ATM concepts from the NextGEN program and the ability to study new methods for future air-traffic control. Previous

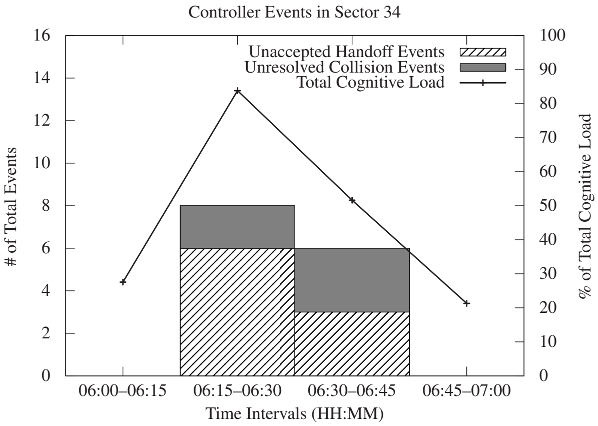


Figure 3: The number of unaccepted handoff and unresolved collision events comapred to total cognitive load.

studies relied on HITL simulations which cannot scale to whole NAS scenarios. This paper also presented the validation process used to evaluate AGENTFLY that utilized common software engineering techniques to derive goal-based metrics and produced results which accurately describe the performance and effectiveness of the system. Through these results we also demonstrated that agents reactions mimic human ATCs when under the same scenario and cognitive load.

In the future, we will compare the quality of the simulation against real human operation on the same sector in cooperation with FAA Human Factors Laboratory. Additionally, the system will be extended with other crucial roles in the current ATM system. For example, the inclusion of D-side controller activities to provide higher fidelity simulations during peak travel periods. It is also planned, to integrate other ATM tools used by controllers like tactical ERAM conflict probe and strategic conflict probe (*e.g.*, User Request Evaluation Tool). The system is able to model precisely en-route ATM control now, but the goal is to extend the coverage of the functionality through Terminal Control Centers to Airport operations. This would allow the system to simulate the gate-to-gate ATM cycle.

Acknowledgements

The work has been supported by the Federal Aviation Administration (FAA) under project numbers DTFACT-10-A-0003 and DTFACT-08-C-00033 and by the Czech Ministry of Education under grant number 6840770038. The underlying AGENTFLY system was supported by the Air Force Office of Scientific Research, Air Force Material Command, USAF, under grant number FA8655-06-1-3073. The views and conclusions contained herein are those of the author and should not be interpreted as representing the official policies or endorsements, either expressed or implied, of the Federal Aviation Administration, the Air Force Office of Scientific Research, the U.S. Government or the Czech Government.

References

Basili, V. R.; Caldiera, G.; and Rombach, H. D. 1994. *Goal Question Metric Paradigm*. Wiley & Sons, Inc. 528–532.

Borshchev, A., and Filippov, A. 2004. From system dynamics and discrete event to practical agent based modeling: Reasons, techniques, tools. In *Proc. of the 22nd Intl. Conf. of the System Dynamics Society*, 25–29.

Carroll, J. M. 2000. *Making Use: Scenario-Based Design of Human-Computer Interactions*. MIT Press.

Chin, D. K., and Melone, F. 1999. Using airspace simulation to assess environmental improvements from free flight and CNS/ATM enhancements. In *Proc. of the 1999 Winter Simulation Conf.*, 1295–1301.

Dvojakovski, B. 2004. Fast-time simulation. Technical Report CRDS/SIM/FTS/2662, EUROCONTROL: CEATS Research Development and Simulation Centre.

George, S., and Wieland, F. 2011. Build 8 of the airspace concept evaluation system (ACES). In *Proc. of Integrated Communications, Navigation and Surveilance Conf.*

McCracken, J., and Aldrich, T. 1984. Analyses of selected LHX mission functions: Implications for operator workload and system automation goals. Technical Report ASI479-024-84, U.S. Army Research Inst. Aviation Research and Development Activity.

Nicol, D., and Yan, G. 2004. Discrete event fluid modeling of background TCP traffic. *ACM Trans. on Modeling and Computer Simulation* 14(3):211–250.

Nolan, M. S. 2004. *Fundamentals of Air Traffic Control*. Belmont, CA, USA: Thomson Brooks/Cole, fourth edition.

Reddy, R. 2010. Advances in NASPAC architecture. In *Innovations in NAS-wide simulation in support of NextGEN benefits analysis*.

Roche, J. M. 1994. Software metrics and measurement principles. *Software Engineering Notes* 19(1):77–85.

Tumer, K., and Agogino, A. 2009. Improving air traffic management with a learning multiagent system. *IEEE Intelligent Systems* 24(1):18–21.

Volf, P.; Šišlák, D.; and Pěchouček, M. 2011. Large-scale high-fidelity agent-based simulation in air traffic domain. *Cybernetics and Systems* 42(7):502–525.

Wickens, C. D.; Mavor, A. S.; Parasuraman, R.; and McGee, J. P. 1998. *The Future of Air Traffic Control: Human Factors and Automation*. National Academy Press.

Wickens, C. 1984. *Varieties of attention*. New York: Academic Press. chapter Processing Resources in Attention.

Wooldridge, M. 1999. *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*. Massachusetts London, England: The MIT Press Cambridge.

Wu, Y., and Gong, W. 2001. Time-stepped simulation of queueing systems. In *Proc. of SPIE*, volume 4367, 262.