Interaction Design Considerations for an Aircraft Carrier Deck Agent-based Simulation

Miles Aubert
Humans and Autonomy Lab
144 Hudson Hall
Durham NC 27708
(919) 619-5078
Miles.Aubert@duke.edu

Weston Ross
Humans and Autonomy Lab
144 Hudson Hall
Durham NC 27708
(505) 385-5867
Weston.Ross@duke.edu

Steven Mazzari
Humans and Autonomy Lab
144 Hudson Hall
Durham NC 27708
(352) 256-7455
Steven.Mazzari@duke.edu

Alex J. Stimpson
Humans and Autonomy Lab
144 Hudson Hall
Durham NC 27708
(352) 256-7455
Alexander.Stimpson@duke.edu

Mary L. Cummings
Humans and Autonomy Lab
144 Hudson Hall
Durham NC 27708
(919) 660-5306
Mary.Cummings@duke.edu

Abstract— **Agent-based modeling techniques have been utilized for a variety of environments within the aerospace domain. For these models, there exist a diverse range of potential users with domain knowledge that ranges from little (e.g. casual gamers) to high (e.g. academic or professional researchers), each with different interests and objectives. Such models allow for both descriptive representations of complex systems that help to explain historical behaviors and outcomes, but they also help in the prospective analysis of futuristic system architectures. Thus the use of agent-based models will be particularly useful in the planning for future unmanned systems. One key issue with such agent-based simulation engines is the complexity of creating an interaction environment that can span the user expertise gap and allow for the intuitive and useful interactions, while retaining high fidelity of information. In order to achieve an interaction environment that can span both the domain and modeling knowledge gap, we propose that the setup, management, and visualization of a given agent-based simulation should be distilled into cognitively simple components that allows users of various degrees of subject matter expertise to effectively understand and manage the simulation. Such an environment should allow users of all skill levels the ability to set up various models and hypotheses, as well as understand the results. To this end we propose an interaction design framework in the context of an interaction engine built on top of an existing agent-based model of flight deck launch operations of a Navy aircraft carrier deck. In this paper, we will discuss how the design framework influenced the design of the interaction environment and how the resultant interaction environment spans the user groups that represent different levels of subject matter expertise.**

TABLE OF CONTENTS

1. INTRODUCTION

Recent advancements in agent-based simulations (ABS), such as the Multi-Agent Safety and Control Simulation [1], have yielded promising results regarding the ability of ABS to provide meaningful and valuable data about operating environments with complex interactions between humans and machines. Such simulations can allow the user to observe and predict emergent behaviors in the system. This information is useful over a diverse range of users whose domain knowledge of the simulated environment varies.

Many recent manifestations of agent-based simulations have focused on the research sector with very little consideration of the accessibility by users that have relatively little or no domain-specific knowledge or development skill. Examples of users that fall in to this category include gamers, educators and professionals. These user types could benefit from the ability to compile and execute agent-based simulations in their domain for reasons such as training, optimization and recreation. However, the current development trends in ABS imbue a level of complexity that makes these simulations difficult for these users to utilize.

To this end we propose a set of interaction design considerations for developing interaction environments of agent-based simulations that will allow users with little domain knowledge the ability to design, execute, and analyze these simulations. The design considerations should provide an immersive and transparent interaction environment to users that allows them to extract valuable information from the simulations with very little effort.

The paper is structured as follows. Section 2 presents an overview of the agent-based simulation technique with specific reference to the important aspects relevant to interaction design. Section 3 proposes our generic interaction design considerations that can be applied across interaction environments for all agent-based simulations. Section 4 provides an application of our interaction design considerations to a model ABS of a US Navy Carrier Deck. Section 5 summarizes the interaction design considerations and presents future steps in refining this work.

2. AGENT-BASED SIMULATIONS

Agent-based simulations are often used for their ability to simulate the decisions, motion and structure of environment agents within a given complex operating environment. Such simulations have been crafted to simulate multiple

978-1-4673-7676-1/16/$31.00 ©2016 IEEE

aerospace operating environments with success such as the carrier deck of a US Navy Aircraft Carrier [1] or air transportation [2]. Agents within ABS are modeled individually each with their own set of behaviors to be executed under certain stimuli.

ABS can offer a wealth of information regarding its agents, how they interact and the potential impact of these interactions at the system level. However in most existing simulation environments this information is buried in lines of code and large data files [3], diminishing the ability of users to locate and utilize this information in a meaningful way.

Each agent-based simulation has three separate phases from start to completion. The first is the setup phase, which consists purely of setting up variables and parameters that define the simulation. For a typical ABS, these include number of agents, the respective agent types and underlying information that govern agents' actions. The second phase is the execution, which consists of the actual simulation of each agent and their respective actions, behaviors, and interactions. The third and final phase is the analysis. This is where the simulation computes specified results from the interactions that occurred during the simulation. These three phases are fundamental for users to develop, run and analyze agent-based simulations.

3. INTERACTION CONSIDERATIONS FOR AGENT-BASED SIMULATIONS

In order to provide users with an intuitive interaction environment for ABS representations, we propose a framework of interaction design considerations based on the three core phases of agent-based simulations. The considerations for each of these three steps are as follows:

Setup

In the setup phase, the user should be given the ability to specify all simulation parameters and attributes, such as the number and type of agents, behavioral aspects of the agents such as task occurrence rates and agent movement rates, and environmental factors such as simulation speed. In order to allow users with minimal domain-specific knowledge to successfully setup and compile such a simulation, there are two key design considerations that need to be addressed.

The first design consideration that needs to be addressed is variable association, which is how well a given user is able to associate an action with its impact on the agents within the simulation. This is especially important for users with little domain-specific knowledge, as they may not understand which variables correspond to which agents or whether a variable is agent-specific or global to the system. In order to address this issue during setup, agents with similar attributes and behaviors should be grouped together into separate interactive domains (such as different screens or panels within a screen). This segmentation process can be described by a tree-like structure as shown in Figure 1, each square node in the tree represents a control group or page that contains controls for setting associated represented by the circular nodes. This tree structure allows users to associate the variables with their respective agents or agent groups.

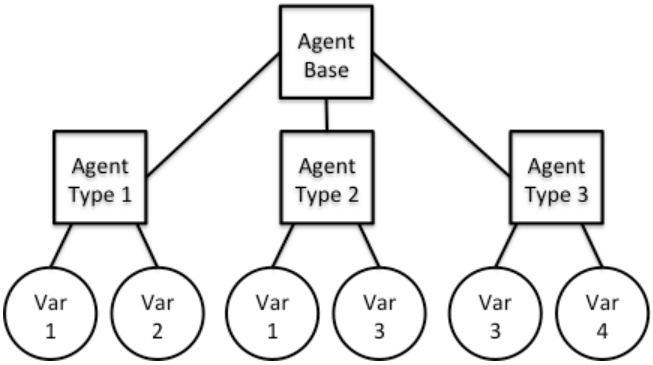


Figure 1: Variable Association Grouping Tree

The second key design consideration is that of setup progress. Setup progress is the ability of the setup environment to allow the user to determine if all required variables and parameters have been initialized. In order to provide this feedback to users of the simulation environment a summary of the specified agent specific simulation variables can be presented to the user in a format similar to the tress shown in Figure 1. By presenting this summary information to the user they will have the ability to quickly identify that all variables have been initialized correctly before executing the simulation.

In summary, in order to create an intuitive setup environment that invariant of developer skill or domain-specific knowledge the user must be able to understand how to setup the simulation they desire, including the numbers and types of agents, and ensure the input settings are correct.

Execution

The execution phase of agent-based simulations consists of providing the user real-time access to the simulation. This real-time access allows the user to both monitor the development of events within the simulation and introduce changes in behavior at run time to investigate how these changes impact agent and system-level performance. In order to support the user's ability to inject events in real-time, there are three key interaction design considerations: situational awareness, environment manipulation and interaction impact.

The first key interaction design consideration is simulation situational awareness. Situational awareness within an agent-based simulation means that a given user must at any time step be able to perceive, comprehend and project [4] the behavior of any given agent within the simulation. This awareness is extremely important for users to gain confidence in the system. To this end, we propose a "fully interactive" visualization of the simulation environment.

The core features include a visualization of each agent within the environment, the ability to query any agent and view its current behavior, and providing constant feedback to the user about agents' states in the current simulation.

During execution users must be able to initiate and comprehend the impact of a user-initiated behaviors. Simulations that provide no interaction component in real time appear as black boxes to users. A better approach is to allow users to introduce events to stress test the system or determine a specific response for verification and validation. In order to achieve this, there are two design considerations. First, the user must be able to directly perceive the impact of the initiated event of the simulation environment. This can be achieved by highlighting all affected agents within the simulation environment. The second consideration requires linking the visualization of the impact of the user's interaction, thereby providing confirmation feedback.

In summary, during live execution, each user must have awareness of each agent within the simulation and understand the impact of each interaction on live behavior of the simulation environment and all of its modeled agents. In addition, it is critical to allow the user the ability to initiate events in real-time to better understand the simulations.

Analysis

The analysis phase of agent-based simulation is centered on the ability for a user to analyze the results of a previously executed simulation. During the analysis phase users should be able review simulations at varying levels of detail that range from overview summaries to low level temporal behaviors of all simulation agents. To provide a user access to both high and low level information for analysis, we propose three key interaction design considerations (simulation playback, event tagging and simulation summary).

During a low-level analysis of a simulation, it is essential to provide users the ability to examine the behavior of each simulation agent throughout the simulation lifecycle in order to promote a deeper understanding of agent behaviors. Thus we propose a replay function to allow users to scroll through a simulation and analyze each time step. The replay environment should be flexible, allowing the user to move both forwards and backwards through simulation time, allowing for the selection of a particular moment in time in order to view agent states.

The second design consideration concerns key event tagging. Key event tagging is a method of providing users a list of key events within the simulation to allow for efficient traversal through the simulation. By tagging key events within an agent-based simulation, users will be able to quickly locate relevant parts of the simulation. These events indicate key turning points within a simulation and therefore provide users information regarding how these events impacted the simulation.

The final design consideration concerns how to present results relevant to analysis to the user. When presenting summary information, it is important to ensure that the user can understand the presented information in the context of the simulation. Special consideration needs to be placed on using controls that can effectively relate this information back to the simulation. Such user controls should generally consist of visual plots of information that can directly be related back to the simulation environment.

In summary, during the analysis phase of agent-based simulations it is important to provide users the necessary information to abstract and reason upon what happened in the simulation through exploring the simulation events and providing information regarding the local and global impact of each agent behavior within the simulation.

4. CARRIER DECK SIMULATION APPLICATION

In order to illustrate the proposed interaction design considerations in the context of a real agent-based simulation, they are applied to an agent-based simulation of flight deck operations on a US navy aircraft carrier.

The implemented interaction environment is design to span four key user groups. First with zero domain specific knowledge are non-military personnel looking to learn about carrier deck operation. Second with low domain specific knowledge are baby carrier deck personnel looking to evaluate their tasks during certain carrier deck operations. Thirdly with high level domain specific knowledge are ranking US Navy personnel that are looking for an overview of current carrier deck operations. Finally with high domain specific knowledge Navy researchers looking to preform detailed analyses of carrier deck operations.

The section is structured with an overview of the implemented carrier deck simulation environment first followed by a summary of the implemented interaction environment for the simulation based on the proposed interaction design considerations.

Overview of the Carrier Deck Simulation Environment

The simulation of US navy aircraft carrier deck operations called Personnel Multi-Agent Safety and Control Simulation (PMASCS) [1,2,6] is a stochastic ABS of carrier deck operations during launch cycles. The simulation is built upon the Golden T Game Engine (GTGE) programming library and developed in Java. Three agent types are modeled: aircraft, catapults, and crewmen. Included are two aircraft types, F-18 (fighter jet) and E-2 (support propeller), nine crewman types: aircraft directors, aircraft captains, chocks and chains, weight-board checkers, maintenance crew, top side petty officers, fueling crew, safety officers, and ordnance crew, and two failure types: catapult failures and maintenance failures.

Each agent has their behavior defined as a set of actions that are a function of the state of the simulation and the states of

other agents. Two types of actions exist, abstract and basic actions. Basic actions include simple movements such as "Walk", "Run", or "Wait." Abstract actions are more complex and consist of a queue of basic actions. Common abstract actions are "Fuel Aircraft", "Perform Maintenance", or "Taxi to Catapult." For example, the maintenance crew actions include maintenance on aircraft and catapults [5], each of which is triggered respectively when an aircraft failure is discovered or a catapult fails. They can occur at random times, determined from sampling a uniform distribution over the entire preparation cycle, or at specific times as desired by the user.

For example, a maintenance agent recognizes that a failure has occurred and performs a repair. The time to perform a repair is pulled from a log-normal distribution, which has been shown to estimate well the task completion time for people [6,7]. In a similar way, all agents in the simulation interact with one another to handle eleven event types; failures, maintenance, fueling, taxiing, un-chocking aircraft, chocking of aircraft, ordnance attachment, weight board checks, holdback bar installation, and launches.

The simulation records two primary types of metrics for safety and operation efficiency: halo violations and sortie launch times, respectively. Halo violations are a measure of risk to personnel and aircraft determined by their proximity to one another. When an operator comes within a range of an aircraft determined to be dangerous, called the primary halo radius of the aircraft, a person-to-aircraft halo violation occurs. A primary halo radius is defined by the circle of radius R from the center of the aircraft, where R is the wingspan of the aircraft. Similarly, when an aircraft comes within one primary halo radius of another aircraft, an aircraft-to-aircraft halo violation occurs. The duration of each halo violation is recorded as well. Sortie launch time is the total time required to launch all aircraft from parked positions in the launch cycle, beginning with the launching of the first aircraft and ending with the launching of the last aircraft, and is a primary metric of operational efficiency.

Overview of the ABS Carrier Deck Interaction Environment

Building upon PMASCS, we developed an interaction environment that provides separate interfaces to handle the setup, execution, and analysis of the ABS, as described in Section 3. We call this interactive environment "PriFly," which simulates the primary flight control environment that exists onboard carriers today. The simulation setup interface sets and compiles all relevant variables for the carrier deck agents. The execution interface shows real-time visualizations of the simulation, and allows for user interaction. The analysis interface provides both summary and review information to the user.

Application of Setup Interaction Considerations

The first interaction window handles the setup of the carrier deck simulation. According to the proposed design considerations, an important aspect is to partition the agent types and their respective variables in to a tree like structure as shown in Figure 2. The square nodes in the structure represent agent groups and agents which are implemented as pages or control groups, the circular nodes represent agent variables, those with solid lines are fixed variables and not visually represented in the interaction environment and those with a dashed line represent variables that can be set by the user and are implemented with an active user control.

This structure shows there are three simulation attributes that need to be set up (aircraft, personnel and failures) and each agent within the agent type groups has its respective variables (represented by the dashed circles) that control its presence and behavior within the simulation.

Visually the agent group partition layer of the tree is shown in Figure 3. This interaction window has three clickable areas for each of the agent type groups that take you to the respective agent group setup page. Also this window contains a control to set the number of times this simulation will execute.

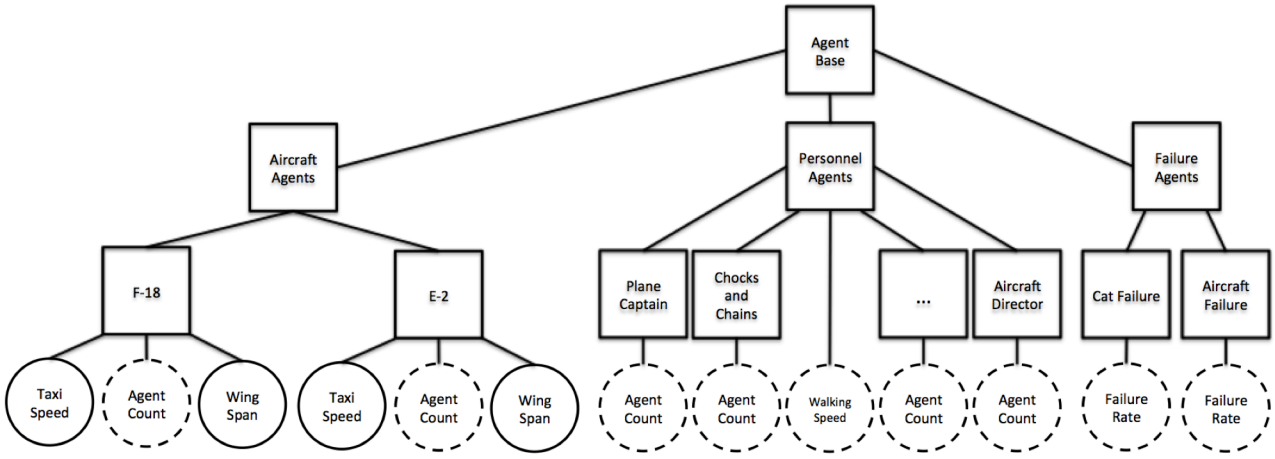


Figure 2: Agent Type-Variable Tree Structure

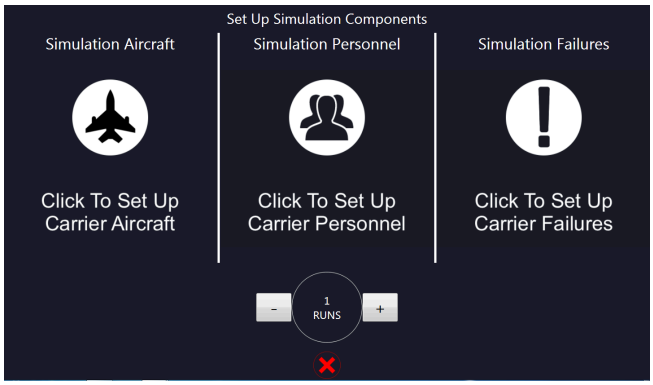


Figure 3: Agent Group Partitions Page

The aircraft and failure setup group pages in Figure 4 show how the individual agent types are separated with their relevant variables within the group setup page. For Aircraft setup there are two aircraft types (F-18 and E-2), each aircraft has two permanent variables (taxi speed and wingspan), which are unseen, and one flexible variable (agent count), which each aircraft group has a control to set. For failures, the two failure types (Catapult Failure and Maintenance Failure) each have a control to set the failure probability for the simulation.

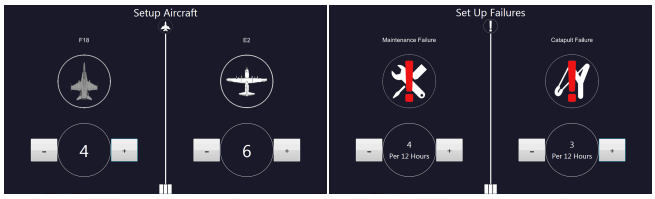


Figure 4: Aircraft Agent Group Setup Page (Left) and Failure Agent Group Setup Page (Right)

The personnel setup group page (shown in Figure 5) offers a more complex variable association problem. The walking speed is a global variable across all personnel types, as shown in Figure 5. As discussed in Section 3, the user must understand the setup progress and allow for clear understanding of the dependency of agent group types on each other. When applied to the carrier deck simulation this can be achieved by summarizing the agent settings on the agent group partition page demonstrated in Figure 6.

Figure 5: Personnel Agent Group Setup Page

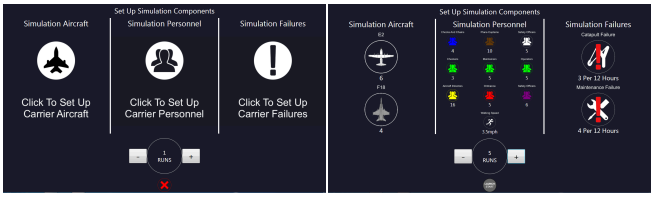


Figure 6: Unset (Left) and Set (Right) Partition Groups

Application of Execution Interaction Considerations

The primary design consideration we proposed for this phase was promotion of situational awareness. In order to meet the three components of situational awareness (perception, comprehension and projection) for the carrier deck simulation, an interactive visualization window was implemented as shown in Figure 7 and Figure 8. This visualization window is available in the live mode (Figure 7) and shows all of the assets on deck in motion and how they are interacting with each other per the ABS. This allows for users to perceive the active behaviors of all agents within the simulation.

Figure 7: Visualization Page in Live Stream Mode

In order to achieve meaningful comprehension, users can pause the simulation to enter agent exploration mode as shown in **Figure 8**. When in the agent exploration mode, which can only be accessed in a paused live simulation, users are able to click on any agent within the simulation to reveal the agent type and information regarding their current behavior, specifically their current task and the current associated error. This information allows users to project what each agent will be doing in the next time step.

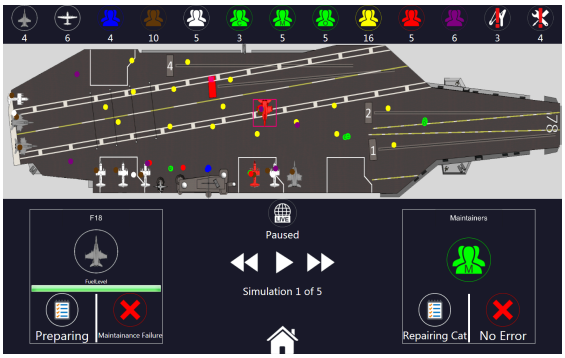


Figure 8: Visualization Page in Agent Exploration Mode

The implemented carrier deck simulation allows the user to initiate two real-time failures (catapult failures and maintenance failures), represented on the visualization window (Figure 7). According to the proposed design consideration framework, each of these controls when clicked triggers a failure that is processed by the simulation environment. This failure is in addition to the failures already generated in the setup mode, and is introduced via a state machine. This state machine starts with the control in an inactive state in an active state, and once a failure control is clicked it transitions to a state that visually represents the failure (Figure 9).

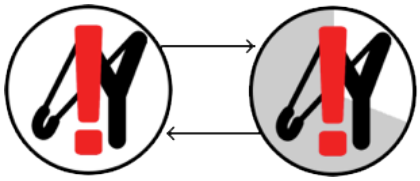


Figure 9: Failure Control States (Inactive (left) In progress (right))

Application of Analysis Interaction Considerations

The final interaction paradigm handles the results and analysis phase of the ABS. In order to provide users an intuitive environment for reviewing the simulation results, Figure 10 was implemented (Figure 10). The interaction window satisfies the two previously defined design considerations that relate to the low-level visualization. First the simulation has standard playback controls (forward/backward, play/pause) that allow users to visualize every time step within a given simulation, providing users the ability to analyze every frame of the simulation.

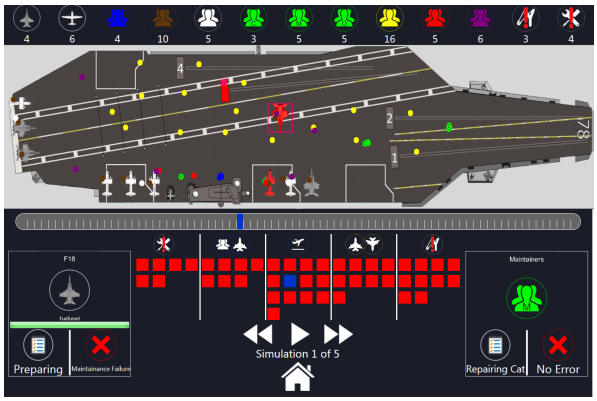


Figure 10: Visualization Page in Review Mode

Moreover, a summary of the user-specified simulation variables is provided at the top of the simulation window to allow users to recall the initial settings. A further design consideration is providing the user the ability to traverse the simulation by key events within the simulation. For the carrier environment, the key events include maintenance errors, person to aircraft halo violations (occurrences of two agents coming too close to one another), aircraft launches, aircraft to aircraft halo violations and catapult failures. These key features are represented by the clickable red controls in Figure 10. When each of these controls are clicked, users are presented with a visualization of the time steps surrounding the key event. This information allows users to efficiently examine the conditions under which each of these events occur and therefore learn strategies to optimize the simulation.

Figure 11: Results Summary Page

Additionally, a window summarizing the results was implemented (Figure 11). The page is segmented to clearly illustrate the two key performance metrics for this carrier deck simulation (launch time and halo violations). Basic statistics for the aggregation of launch times per simulation are displayed (mean, maximum, minimum, and standard deviation), as well as a trend graph.

In order to display results regarding the two halo violation types (person to aircraft and aircraft to aircraft), which are hard to contextualize purely with statistics, an additional visualization was implemented to ensure that users can relate these events back to the simulation. This visualization control plots each halo violation in terms of deck location and type as shown in Figure 11. This visualization spatially relates each halo violation back to the simulation, which in turn allows users to abstract upon the summary information and determine problem areas on the carrier deck, potentially uncovered problematic areas that increase the probability of a serious incident.

5. SUMMARY

This paper presented a framework of interaction design considerations that allow for the creation interfaces for agent-based simulations that are suitable for a diverse range of users with varying levels of development and domain-specific knowledge.

The proposed interaction design pattern focuses on how to develop interaction environments for the setup, execution and analysis of agent-based simulations. The core components of the interaction design pattern include segmenting simulation agents during setup to ensure an effective understanding of the simulation initial conditions, providing the user situational awareness, and effective ways to traverse key events to allow for detailed analyses of agent behaviors and the operating environment.

Future work based on this interaction design considerations has two core stages. First is a detailed analysis of the performance of the PriFly to refine the proposed framework of interaction design considerations. Second is the application of the refined framework to other agent-based simulations to validate the generalizable nature of the proposed considerations.

ACKNOWLEDGEMENTS

We would like to thank the Office of Naval Research for funding this research initiative through contract #N000140910625.

REFERENCES

[1] J. C. Ryan and M. L. Cummings, 'Development of an Agent-Based Model for Aircraft Carrier Flight Deck Operations', *M&S Journal*, No. Spring, 2014, pp. 5-15.

[2] A. Pritchett, S. Lee, M. Abkin, A. Gilgur, R. Bea, K. Corker, S. Verma and A. Jadhav, 'Examining air transportation safety issues through agent-based simulation incorporating human performance models', Proceedings. The 21st Digital Avionics Systems Conference, 2002.

[3] S. Sanchez and T. Lucas, 'Exploring the world of agent-based simulations: simple models, complex analyses', Proceedings of the Winter Simulation Conference, 2002.

[4] M. Endsley, 'Toward a Theory of Situation Awareness in Dynamic Systems', *human factors*, vol. 37, no. 1, pp. 32-64, 1995.

[5] W. Ross, A. J. Stimpson, and M. L. Cummings, "Optimizing Maintenance Crew Staffing on Aircraft Carrier Decks," *Submitted to: AIAA SciTech*. 2016.

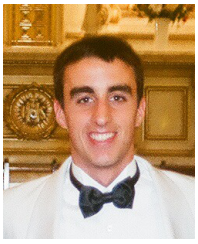
[6] J. C. Ryan, 'Evaluating Safety Protocols for Manned-Unmanned Environments through Agent-Based Simulation', *Aeronautics and Astronautics*. Vol. Doctor of Philosophy in Human Systems Engineering, Massachusetts Institute of Technology, 2014.

[7] D. Schmidt, 'A Queuing Analysis of the Air Traffic Controller's Work Load', IEEE Transactions on Systems, Man, and Cybernetics, vol. 8, no. 6, pp. 492-498, 1978.

BIOGRAPHY

Miles Aubert *received a B.Sc in Robotics from the University of Reading in 2014. His background is focused in developing accessible technologies that minimize cognitive workload for use in unskilled or high cognitive workload domains. He is currently in the process of completing a PhD as part of the Human and Autonomy Laboratory at Duke University.*

Weston Ross *Weston Ross received his B.S. in Computer Engineering from the University of New Mexico, NM, in 2012. He is currently pursuing a PhD in Mechanical Engineering and Materials Science at Duke University, NC, in the Humans and Autonomy Laboratory. His research focus is on optimal manning in heterogeneous manned-unmanned environments.*



Steven Mazzari *graduated from Regis High School in 2013 and attends Duke University's Pratt School of Engineering. Double majoring in electrical and computer engineering and computer science, he started research in April of his sophomore year in the Human and Autonomy Laboratory, headed by Dr. Missy Cummings. Mazzari currently is still working as an undergraduate researcher specializing mainly in simulation development.*

Alexander Stimpson *received the B.S. degree in biological engineering from the University of Florida, Gainesville, FL, USA, in 2007, and the S.M. degree in Aeronautics and Astronautics from the Massachusetts Institute of Technology (MIT), Cambridge, MA, USA, in 2011. His dissertation work focused on the application of machine learning models to inform training assessment and intervention. His current research interests include human supervisory control, decision support systems, artificial intelligence, and data mining. He is currently a postdoctoral researcher for the Humans and Autonomy Laboratory at Duke University.*

Mary Cummings *received her Ph.D. in Systems Engineering from the University of Virginia in 2004. She is currently an associate professor in the Duke University Department of Mechanical Engineering and Materials Science, the Duke Institute of Brain Sciences, and the Duke Electrical and Computer Engineering Department. She is the director of the Duke Humans and Autonomy Laboratory.*