



**UNDERSTANDING EFFECTS OF AUTONOMOUS AGENT TIMING ON
HUMAN-AGENT TEAMS USING ITERATIVE MODELING, SIMULATION
AND HUMAN-IN-THE LOOP EXPERIMENTATION**

THESIS

Tyler J. Goodman, 2d Lieutenant, USAF

AFIT-ENV-MS-16-M-154

**DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY**

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

DISTRIBUTION STATEMENT A.
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

The views expressed in this thesis are those of the author and do not reflect the official policy or position of the United States Air Force, Department of Defense, or the United States Government. This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States.

AFIT-ENV-MS-16-M-154

UNDERSTANDING EFFECTS OF AUTONOMOUS AGENT TIMING ON HUMAN-
AGENT TEAMS USING ITERATIVE MODELING, SIMULATION AND HUMAN-
IN-THE LOOP EXPERIMENTATION

Presented to the Faculty

Department of Systems Engineering and Management

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

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

Tyler J. Goodman, BS

2d Lieutenant, USAF

March 2016

DISTRIBUTION STATEMENT A.
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

UNDERSTANDING EFFECTS OF AUTONOMOUS AGENT TIMING ON
HUMAN-AGENT TEAMS USING ITERATIVE MODELING, SIMULATION AND
HUMAN-IN-THE LOOP EXPERIMENTATION

Tyler J. Goodman, BS

2d Lieutenant, USAF

Committee Membership:

Dr. M. E. Miller
Chair

Maj C. F. Rusnock, PhD
Member

Capt J. M. Bindewald, PhD
Member

Abstract

Recent U.S. Air Force Research Laboratory strategy documents have suggested the need for research in human-agent teaming. Teaming supports a dynamic shift in roles between the human and the agent, depending upon human performance and mission needs. Further, because the performance of these agents will be highly dependent upon the state of the human and the mission, this strategy suggests the need for increased use of modeling to provide a broader understanding of the automated agents' behavior. This thesis applies a combination of static modeling in SysML activity diagrams, dynamic modeling of human and agent behavior in IMPRINT, and human experimentation in a dynamic, event-driven environment. The dynamic models and human experiments are used to understand the effects of agent delay time on human behavior, performance, and workload, as well as team dynamics. The models and experiments illustrate that agent delay time has a significant effect upon team behavior, performance, and the roles assumed by the human and agent. Therefore, it is proposed that the consequences of agent timing are significant in the context of human agent teaming and that models, which incorporate the human and agent within a common modeling environment, can be useful in understanding this effect.

Acknowledgments

I would like to thank my research advisor and committee chair, Dr. Miller for his guidance. I greatly appreciate the amount of time and effort he sacrificed in assisting me. I am also grateful to my other committee members, Maj Rusnock and Capt Bindewald, for their support and investment. I value the assistance from fellow students, Olivia Ashworth, Bryan Zake, and Jayson Boubin. Also, I would like to thank the Air Force Office of Scientific Research for their contributions.

Tyler J. Goodman

Table of Contents

	Page
Abstract.....	1
Acknowledgments.....	2
Table of Contents.....	3
List of Figures.....	6
List of Tables.....	8
I. Introduction.....	9
General Issue.....	9
Problem Statement.....	14
Research Objectives.....	14
Investigative Questions.....	15
Methodology.....	16
Assumptions and Limitations.....	16
Expected Contributions.....	17
Preview.....	18
II. Incorporating Automation: Using Modeling and Simulation to Enable Task Re- Allocation.....	19
Abstract.....	19
Introduction.....	19
Method.....	24
Results and Discussion.....	28
Discussion.....	37
Conclusions.....	38

Future Research	39
III. Timing Within Human-Agent Interaction and its Effects on Team Performance and Human Behavior	41
Abstract.....	41
Introduction	41
Method for Previous Experiment	45
Experiment Results and Discussion	49
Space Navigator IMPRINT Model.....	51
Simulation Results.....	59
Conclusions and Future Work.....	61
IV. Timing and its Effects on Human Agent Teaming.....	64
Abstract.....	64
Introduction	64
Space Navigator IMPRINT Model.....	70
Simulation Results.....	79
Experimental Method	82
Experiment Results.....	83
Discussion.....	86
Conclusion	90
V. Conclusions and Recommendations	92
Chapter Overview.....	92
Research Motivation.....	92
Research Objectives	94

Investigative Questions	95
Study Limitations	101
Recommendations for Action.....	102
Recommendations for Future Research.....	102
Significance of Research	115
Bibliography	117
Appendix.....	117
IMPRINT Simulation Description	Error! Bookmark not defined.

List of Figures

<i>Figure 1: Pictorial representation of the Space Navigator application environment.</i>	<i>27</i>
<i>Figure 2: Activity Diagram for the Manual System.....</i>	<i>29</i>
<i>Figure 3: Imprint model illustrating the Manual System. Although not shown, the final model will also likely include the system events (e.g., ship spawn frequency), which are likely to influence human performance.</i>	<i>30</i>
<i>Figure 4: Activity Diagram for Automation with Direct Re-Allocation, with first order task allocation</i>	<i>31</i>
<i>Figure 5: IMPRINT model for Automation with Direct Re-Allocation, displaying computer and human control by blue and lavender nodes, respectively.....</i>	<i>32</i>
<i>Figure 6: Model displaying automation with handover including a red outline highlighting the communication between the human and automation.....</i>	<i>34</i>
<i>Figure 7: Partial display of the activity diagram for automation with supervision, incorporating communication between the automation and human, as well as the human’s decision to update situation awareness.....</i>	<i>36</i>
<i>Figure 8: Screen capture from Space Navigator, highlighting spaceships, planets, trajectories, bonuses, and no-fly zones.....</i>	<i>47</i>
<i>Figure 9: Activity diagram representing the actors and actions in the IMPRINT model. Vertical swimlanes are used to designate actions.....</i>	<i>53</i>
<i>Figure 10: Graph displaying the probability of the agent drawing a route with respect to the number of ships on the screen. The third order regression line, with equation, was used in calculating the reliance algorithm in the IMPRINT model.</i>	<i>54</i>
<i>Figure 11: Graph displaying average score and workload per agent delay time.</i>	<i>60</i>
<i>Figure 12: Graph displaying mean human draws, agent draws, and redraws per agent delay time.....</i>	<i>61</i>
<i>Figure 13: Screen capture from Space Navigator, highlighting spaceships, planets, trajectories, bonuses and no-fly zones.....</i>	<i>73</i>

<i>Figure 14: Activity diagram representing the actors and actions in the IMPRINT model. Vertical swimlanes are used to designate actions performed by the specified actor.....</i>	<i>75</i>
<i>Figure 15: Graph displaying the probability of the agent drawing a route as a function of the number of ships on the screen. The third order regression line, with equation, was used in calculating the reliance algorithm in the IMPRINT model.</i>	<i>77</i>
<i>Figure 16: Graph displaying model predictions of mean score and workload as a function of agent delay time.....</i>	<i>81</i>
<i>Figure 17: Graph displaying model predictions of mean human draws, agent draws, and redraws as a function of agent delay time.</i>	<i>81</i>
<i>Figure 18: Graph displaying experimentally derived mean standardized NASA TLX workload values, as well as and score as a function of agent delay time.</i>	<i>84</i>
<i>Figure 19: Graph displaying experimentally derived mean human draws, agent draws, human redraws, and agent redraws as a function of agent delay time.....</i>	<i>85</i>
<i>Figure 20: Graph displaying percent of human draws as a function of ship arrival rate and delay time as predicted from the IMPRINT model.</i>	<i>105</i>
<i>Figure 21: Graph displaying percentage of possible score as a function of ship arrival rate and delay time as predicted from the IMPRINT model.</i>	<i>105</i>

List of Tables

Table 1: Manual and Straight Line Agent Data 50

UNDERSTANDING EFFECTS OF AUTONOMOUS AGENT TIMING ON HUMAN-AGENT TEAMS USING ITERATIVE MODELING, SIMULATION AND HUMAN-IN-THE LOOP EXPERIMENTATION

I. Introduction

General Issue

Autonomous systems have provided a significant impact on modern warfare. This can be observed in the recent advancement of systems such as Unmanned Aerial and Ground Vehicles. According to the Department of Defense (DoD), “autonomy is a capability (or set of capabilities) that enables a particular action of a system to be automatic or, within programmed boundaries, ‘self-governing’ ” (*The role of autonomy in DoD Systems*, 2012). The purpose of autonomous systems is not to replace humans in military systems, but to complement human ability to improve system performance. Autonomy has the potential to impact several domains within the Air Force, including manned and unmanned aircraft, space, cyber, intelligence, surveillance, and many more operations. The benefits autonomy can provide to the Air Force include:

- Increasing range and speed of operations
- Reducing unnecessary manual labor and reducing system manning costs
- Reducing the time required to conduct time-critical operations
- Providing increased levels of operational reliability, persistence and resilience
- Removing the human operator from harm’s way (M.R. Endsley, 2015; *The role of autonomy in DoD Systems*, 2012).

As the use and sophistication of autonomy increases, the presence of human interaction will still be necessary (*The role of autonomy in DoD Systems*, 2012). Developing autonomous systems introduces new levels of complexity and opportunity for failures, bugs and vulnerabilities. When these systems leave the development and testing environment and are introduced into a real, wartime environment, the systems may encounter situations that the developers never considered (M.R. Endsley, 2015). Therefore, it is believed that the development of autonomy will not result in the exclusion of human presence, but rather future operations will require human and autonomy collaboration to achieve mission success.

The role of the autonomous system is evolving from a tool, simply providing aid, to a fully functional teammate that engages and interacts with the human operator. The Air Force has recognized the evolution of autonomy and has put an emphasis on teaming to approach humans and autonomy working together (M.R. Endsley, 2015). The fundamental aspect of teaming is that humans and autonomy will “interchange initiative and roles across mission phases to adapt to new events, disruptions and opportunities as situations evolve” (*The role of autonomy in DoD Systems*, 2012).

This dynamic relationship between humans and automated systems has not been fully realized in current systems due to numerous challenges associated with autonomous system development. Two specific challenges are addressed in this research. An anticipated difficulty in system design is a similar issue that has been experienced in previous development of automated systems. Automation and automatic capabilities are designed with the intent of assisting the operator, but to some extent systems have caused

adverse effects towards human workload and situation awareness (M.R. Endsley, 2015). Another challenge within autonomous system design is the ability to properly test and evaluate the system. As the potential actions conducted by the autonomy expands within a dynamic environment, traditional methods of test and evaluation are inefficient and impractical.

The ability to effectively use automation in past operations was hindered by several factors including reduced human situation awareness and undesirable workload levels. It has been suggested that these similar issues may arise in autonomous systems as well (M.R. Endsley, 2015). Maintaining proper levels of situation awareness is essential for the human to ensure the autonomy is operating properly and responding to situations as desired. When people supervise automation, it can be easy for the human to become “out-of-the-loop”, in which case, they can become slow to detect and diagnose a problem (Endsley and Kaber, 1999). Another challenge presented by autonomy is managing workload levels for the human. Low workload levels, which may arise from tasks such as monitoring automation, may cause the operator to become complacent and “out-of-the-loop”. High workload levels result in strain upon the operator. They are unsustainable for extended periods of time and are likely to result in errors or omissions as the human is unable to respond appropriately. Roles, responsibilities, and tasks should be allocated between the human and autonomy to sustain the operator’s awareness and manage their workload levels.

One aspect of autonomous system design that may have considerable impact on team member roles and initiative, as well as human situation awareness and workload, is

the autonomy's task timing. The timing of task execution in highly dynamic, event-driven domains is assumed to influence the performance and behavior of the team. Considering that automated systems have the potential to respond much faster than their human counterparts, it is posited that their response time can affect task responsibility. If the autonomy's response time is too quick, the human operator may assume a supervisory role as the automation will always respond to an event faster than its human counterpart is capable of responding. If its response is excessively delayed, the automation will be incapable of a timely response and the human is likely to assume responsibility for the event and attempt to respond before the automation. However, the proper timing and changes in the behavior of human team members as a function of automation response time is not apparent in the literature. The influence of task timing within the range of the two extreme times, too quick or too slow, is uncertain, yet potentially significant to understanding human and autonomy interactions.

The Air Force Research Laboratory has identified enduring problems regarding autonomous system development (Clark, Kearns, & Overholt, 2014). In addition to issues in human autonomy teaming, another enduring problem is the proper testing, evaluation, verification, and validation of the system. This issue arises as the range of actions that could potentially be performed by autonomy is exponentially greater than previous automation systems, which do not significantly adapt their response to environmental stimulus. As autonomy's software is adaptive and learns to respond to a large range of environmental conditions, autonomy has several potential outputs per input it receives. Traditional methods of test and evaluation involved placing the automation into a scripted

scenario and observing how it responds. It is not feasible to perform this same style of testing as the space of autonomous actions cannot be “exhaustively searched, examined or tested” (Clark et al., 2014). Inserting these systems into unpredictable and unknown environments compounds this problem. Therefore, there is great uncertainty in the consequences of the behavior of the system and its interactions with the human operator.

The vision that has been proposed to address this issue relies upon modeling and simulations to understand autonomous systems and the consequences of their actions (Clark et al., 2014). Identifying effective methods of reusing test and evaluation results has been a challenge. Modeling and simulation provides the developers the opportunity to efficiently examine expected responses from the system in a wide range of environments. Through iterative, continuous and evolutionary modeling and simulation, it may be possible to evaluate a greater range of autonomy responses and actions. In autonomous systems that adapt the human’s task environment, developers may be able to understand human autonomy interactions and the effects of system design on human behavior, workload, situation awareness and performance through the use of models which include human and automation behavior.

Overall, this thesis examines the effects of an autonomous agent’s task timing on the distribution of roles and responsibilities within the human agent team, as well as the effects of this variable on human behavior and workload. The human behavior that typically results from automation executing actions too soon or too late is fairly understood. However, research is lacking as to the effects of autonomy’s timing of execution within that timeframe. To examine the effects of autonomy task timing,

research is conducted using modeling and simulation techniques, in alignment with AFRL's vision for autonomy testing and evaluation.

Problem Statement

This thesis addresses a primary and secondary problem in the field of human agent interaction. The main problem is the uncertainty of the effects of agent task timing on human autonomy teaming. This research seeks to understand the effect of agent's timing on team roles, responsibilities, and performance, as well as, the change in human behavior and workload. The secondary problem is developing an approach to modeling and simulation that contributes to AFRL's goal of establishing effective autonomy design methods using progressive sequential modeling, simulation, test and evaluation.

Research Objectives

The task force report by the DoD refers to the human autonomy team and a need to understand team dynamics (*The role of autonomy in DoD Systems*, 2012). A significant aspect of team dynamics is the dynamic allocation of roles and responsibility amongst team members. It appears that the agent's task timing, as a contributing teammate, may have significant effects on human behavior and team performance. However, there is uncertainty in how agent's timing, within the context of teaming in a changing environment, affects team dynamics as well as, human behavior and workload. Therefore, the primary objective within this research is to assess the human agent team and the effects of agent timing.

Model Based Systems Engineering (MBSE) is often used to help developers perceive and understand a system in the conceptual phases of a system's lifecycle. The use of models and simulations early in the product lifecycle can provide a cost effective method to understand and influence several aspects of the project including budgeting, scheduling, requirements, construction, and operational capabilities of the system. Therefore, it is critical to establish accurate representations of the system when incorporating MBSE in a design process. With the integration of autonomous systems into military operations being relatively new, modeling guidelines and principles are limited. The secondary objective of this research is to understand the considerations and requirements needed to properly model interaction between the human and autonomous agent.

Investigative Questions

Understanding the primary objective for this research will provide insight and contribute to answering the following investigative questions:

1. What are the considerations needed when modeling a process that involves human-agent interaction?
2. How can modeling and simulation tools be used to infer agent timing that simultaneously improves operator performance and reduces workload?
3. How does the timing of an agent affect operator behavior and workload, as well as team performance and dynamics?

Methodology

The application environment for this research is a route generation game called Space Navigator. Space Navigator provides an environment that can be performed solely by a human operator or include an automated agent. This research consisted of three phases. The first phase involved modeling the process of an operator playing the game through an activity diagram. Then, the automated agent was introduced and the models were reconfigured to more accurately represent human behavior. The models were evaluated to understand the significant differences between them, as well as the requirements needed to develop a model that contains human-agent interaction. The second phase included the development of a workload simulation model that was used to estimate operator workload and performance across varying agent trigger times. The final step included human test subjects' experiment where participants operated the game with varying agent delay times and the results were collected. Performance and workload data from the simulations were compared to the test subject data with the intent of validating the models that were developed.

Assumptions and Limitations

The biggest limitations to the research are that the human test subjects do not truly represent the population of military operators, and the game used in these experiments does not replicate the use of a militarized autonomous system. Nonetheless, Space Navigator provides a controlled representation of a highly-dynamic, event-driven environment. The environment also permits the control of the event rate and other potentially confounding variables, logging of human response, and the creation of

automations that can be enabled to assist the operator during high event rate conditions. Additionally, the environment includes a single, clearly defined, top level goal (i.e., score the most points) as opposed to most games which provide multiple, often conflicting goals (e.g., leveling up and score). The use of the relatively intuitive game environment simplifies participant recruitment and training.

Thus, the primary assumption of this research is that although the subjects and environment do not directly represent the types of autonomous systems that would be used by the DoD, the results will apply to the general field of human-agent interaction.

Expected Contributions

All results and conclusions from this research will be able to contribute to the research and development of autonomous systems for the Department of Defense. The Department of Defense and the Air Force have identified autonomous systems as a key contributor to militaristic efforts (M.R. Endsley, 2015; *The role of autonomy in DoD Systems*, 2012) and is in need of further research regarding this new technological frontier. Therefore, if considerations extracted from the development of human-agent modeling are validated, design tools can be established to help create accurate models of human-agent interaction. Statistical analysis from the simulations and human test subject experiments can provide further insight as to how people interact with automated agents. Understanding gained from these experiments can aid future research in the field of autonomy.

Preview

This document consists of three individual, yet interrelated, articles that provide in depth processes, results, and applications from this research. Chapter 2 includes the article, “Incorporating Automation: Using Modeling and Simulation to Enable Task Re-Allocation” (Goodman, Miller, & Rusnock, 2015) which provides insight to modeling considerations for human-agent interaction. Chapter 3, “Timing Within Human-Agent Interaction and its Effects on Team Performance and Human Behavior” (Goodman, Miller, Rusnock, & Bindewald, 2016) details the development of a simulation to predict human performance and workload with respect to agent trigger time. Chapter 4, “Timing and its Effects on Human-Machine Teaming” (manuscript in preparation for Journal of Cognitive Engineering and Decision Making) compares the simulation results to human test subject experimentation and discusses some insights gained in regards to humans’ interactions with agents. The Conclusion, Chapter 5, addresses the research objectives, answers investigative questions, and discusses the role of timing within a human-agent team and application to military systems.

II. Incorporating Automation: Using Modeling and Simulation to Enable Task Re-Allocation

Abstract

Models for evaluating changes in human workload as a function of task allocation between humans and automation are investigated. Specifically, SysML activity diagrams and IMPRINT workload models are developed for a tablet-based game with the ability to incorporate automation. Although a first order model could be created by removing workload associated with tasks that are allocated away from the human and to the computer, we discuss the need to improve the activity diagrams and models by capturing workload associated with communicating state information between the human and the automation. Further, these models are extended to capture additional human tasks, which permit the user to maintain situation awareness, enabling the human to monitor the robustness of the automation. Through these model extensions, it is concluded that human workload will be affected by the degree the human relies upon the automation to accurately perform its allocated tasks.

Introduction

In Systems Engineering, a significant step during preliminary system design involves the allocation of functions to various subsystems (Blanchard and Fabrycky 2000). At the highest level, this allocation decision involves assigning functions to a human operator or a machine. Because the quality of this allocation decision is subject to many constraints and considerations, and this decision is typically made very early in the

system lifecycle, before system prototyping and in-depth understanding of the system is acquired, this process is often considered an art which cannot be addressed by analytic means (Fuld 2013; Dekker and Woods 2002). Due to the uncertainty inherent in this decision, low risk solutions, for example employing allocations similar to that employed in legacy systems, are often pursued. While low risk, such solutions are not particularly desirable when a primary goal of the system development is to improve human performance or reduce manpower to reduce operational costs. Fortunately, the quality of these allocation decisions can be improved through modeling and simulation of the system and human performance (“Improved Performance Research Integration (IMPRINT) Tool,” 2010). For example, modeling of operator workload can provide insight into the system performance consequences of various allocation decisions. While it is acknowledged that optimizing task allocation based upon workload is only one of many potential criteria (Older, Waterson, and Clegg 2010), this paper explores the use of SysML diagrams and human workload models to aid the allocation decision. Specifically, this paper seeks to address the effect that potential changes in allocation, or re-allocations, have upon the structure of task representations within a workload model. By addressing this issue, this paper improves the robustness of human workload models and allows for more accurate and effective task re-allocations.

Perhaps the most frequently cited reference in the function allocation literature is a technical report, which acknowledges that machines perform certain types of tasks better than humans and that humans perform other tasks (e.g., inductive reasoning, flexibility, judgment, selective recall) better than machines (Fitts et al. 1951). Equally

important, however, Fitts and his colleagues acknowledge that humans cannot employ their capabilities properly when overloaded due to excessive task demands or when they are unable to maintain alertness due to underactivity, for example when not actively participating in system control. The relationship between human performance and perceived workload resulting from a level of task demand, commonly referred to as the Hebb-Yerkes-Dodson Law, indicates that human performance follows an inverted-U-shaped function with maximum performance occurring at moderate levels of arousal, which permit the human to concentrate on relevant cues within the environment (Teigen, 1994). This relationship has been extended to explain the impact of stress and perceived workload on human performance, with human performance nearing an optimal for moderate perceived workload levels (de Waard, 1996). Perceived workload generally increases with an increase in the number or complexity of tasks to be performed by the human and as the time available to perform these tasks decreases (Hart & Staveland, 1988; Reid & Nygren, 1988). The level of perceived workload is thus highly linked to the allocation of tasks between the human and computer, which in turn has a significant impact on the performance of the human operator and therefore the performance of the entire system. As a result, Kaber and colleagues have suggested that a decision regarding the level of automation to be applied should be made to minimize a cost function which includes a nonlinear function of workload (Kaber et al. 2009).

Importantly, task load and the resulting perceived workload is not constant during system operation. Instead, changes in the environment can influence the number and complexity of cues that an operator must process to correctly perceive the environment.

For example, consider the number of potential hazards one can encounter when driving on a deserted rural highway versus driving in a crowded city center. The number and complexity of the tasks that must be performed also differ as goals change. For example, consider the complexity of maintaining level flight versus performing a landing, particularly in clear versus adverse weather conditions. This variability in workload is particularly important when investigating automation as the tendency of the automation designer is to automate the functions which are the easiest to automate, potentially creating systems in which the human operator is relegated to a monitor during times that they are easily capable of controlling the system, while performing unassisted during times that they experience peak workload (Colombi et al., 2011). Therefore, it is necessary for any model used for allocation to consider this variability within the context of the work to be performed by the human operator within the allocated system (Dearden, Harrison, and Wright 2000).

To account for this variability, this study uses Improved Performance Research Integrated Tool (IMRINT) a discrete event simulation environment (Army Research Laboratory 2010). This environment models human workload and performance as a function of time by tracking activities performed by a human or a machine. These activities are described in a task network, which captures the task sequencing and decision points.. The frequency of the tasks, as well as the time necessary to perform each task result from a stochastic process, permitting the modeler to represent the variability within the system. Different task networks can be derived for different goals and a workload level is assigned to each task performed by the human operator. Various

system allocations can then be modeled by allocating specific tasks to be performed by the human operator or machine (hardware or software) component. However, to employ this tool to accomplish this goal, the modeler must begin with a activities to be performed by the system, allocate these activities to the human or machine and then derive the tasks or actions necessary to perform these functions. Once these activities are allocated to a component, human or machine, other inherent tasks may become necessary to facilitate communication of system state as control is passed between the human and machine (Bindewald, Miller, & Peterson, 2014).

IMPRINT enables the quick re-allocation of tasks by simply changing the “assignee” for the task from a human operator to an automated component. However, attempts to incorporate automation from a simple re-allocation of tasks previously performed by a human operator to the automated system are unlikely to be sufficient. The current paper develops function and task networks to explore the impact of task re-allocation on changes in the task networks. Specifically, this paper demonstrates that re-allocating tasks previously handled by a human operator to a machine results in the necessary creation of new tasks. This creation of new tasks has implications for the design of the system as well as impacts to the operator’s expected workload. While a simple re-assigning of tasks is expected to reduce operator workload and enhance system performance, to be truly accurate workload modeling must account for additional tasks caused by required communications and operator attempts to maintain situation awareness. Through this process we seek to understand and explain the considerations necessary when modeling human workload to support function allocation.

Method

Systems Modeling Language

Recent developments in Systems Engineering have led to increased adoption of Model-Based Systems Engineering (MBSE), which commonly includes a modification of the Unified Modeling Language referred to as the Systems Modeling Language (SysML) (Delligatti, 2013). SysML captures process allocation through activities and actions within Activity Diagrams. Allocation decisions are captured in Activity diagrams with each actor indicated by unique partition--each partition is colloquially referred to as a “swim lane”.

Elements within the activity diagram include action nodes, control nodes, pins, and flows. The actions are the “building blocks” of the diagram which accept inputs and transform them to outputs. The input and output buffers on each activity are pins. Flows connect the output pin of one action to the input pin of another action to enable the passage of information or objects. In the constructed diagrams, the control nodes consist of the decision, merge, fork, and join nodes.

Within this paper, activity diagrams were created within a systems modeling tool, called Enterprise Architect. As appropriate, these diagrams include not only the actions and control logic necessary to depict the necessary “functions,” they also include swim lanes to depict particular allocations of these actions to performing entities. These diagrams provide the basis for task networks within IMPRINT.

IMPRINT

As noted earlier, the Improved Performance Research Integrated Tool (IMPRINT) provides an environment to enable discrete event modeling of human workload. The task networks developed in the activity diagrams were transferred to this modeling environment, capturing the flow of actions and decision logic. Completion of these models would then require development of task time probability distributions for each action and a mental workload value for each action performed by the human operator. Other values, such as action completion accuracy may also be captured for the human or computer as well. While we acknowledge that completion of a model requires the development of these distributions and workload values, as the focus of this paper is to understand the changes in function and task networks necessary to capture changes in allocation, neither the development of these model inputs or the results of the modeling activity are discussed within the current paper.

Application Environment

To explore the decision to re-allocate tasks from a human to an automated component, it was necessary to select an application environment which was simple enough to permit the task network to be depicted in small activity diagrams and complex enough to provide a series of activities which could be allocated to either a human or a machine. The environment employed in this paper is a tablet computer based game called Space Navigator, which includes a number of activities that can be allocated to a human or a machine (Bindewald, Peterson, and Miller 2015). The game contains four stationary planets present on the screen. Each planet has one of four colors: red, green, blue, or

yellow. Spaceships appear at a set interval from a random location at the side of the screen. Each spaceship is red, green, blue, or yellow. The player must direct each spaceship to the destination planet of the same color by drawing a trajectory line on the game touch screen using their finger. The spaceship then follows this line at a constant rate. Spaceships continue to appear until an allotted time of five minutes is over. If desired, trajectories may be re-drawn, to avoid a collision and account for dynamic changes in the environment. Points are earned when a ship successfully reaches its destination planet or traverses any of a number of small bonuses that appear throughout the play area. Upon reaching its destination planet, a spaceship disappears from the screen. When spaceships collide, points are lost and each spaceship involved in the collision is lost.

Additionally, points are lost when a spaceship traverses one of several “no-fly zones” that move to different random locations on the screen at a set time interval. The objective of the game is to earn as many points as possible in five minutes. Figure 1 shows an annotated screen capture from Space Navigator, which illustrates various elements of the game.

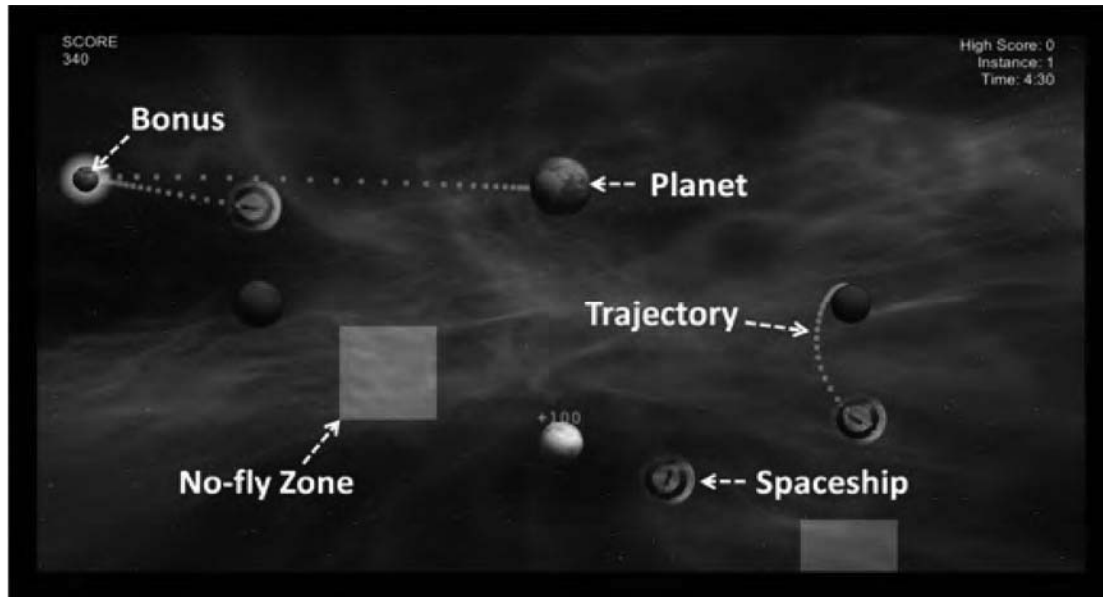


Figure 1: Pictorial representation of the Space Navigator application environment.

Procedure

We coordinated SysML, IMPRINT, and the Application Environment by creating task networks in Activity Diagrams for the application environment and transferring these task networks to IMPRINT. Initially, these activity diagrams were constructed with the assumption that the human operator was to perform all tasks associated with playing the game. This provided a baseline model, referred to as the Manual System, that accurately demonstrates the actions that were necessary for the human to successfully play the game. A second model, Automation with Direct Re-Allocation, introduces swim lanes to indicate the allocation of actions to the computer or human operator. In this model, the actions were split such that the machine was responsible for indicating the ship which required the most immediate attention by the human operator, while the human operator remained responsible for generating the ship's route. This task allocation

model was revised into a third model, Automation with Handover, to recognize that automation would change the human's task management strategy, with the operator fully reliant on the machine to perform its target selection actions appropriately. This handover model provides additional actions which permits the automated system to communicate with the human operator. A final task network, Automation with Supervision, is explored for a condition where the human is not fully reliant on the machine but instead monitors the environment to maintain situation awareness, enabling the human to monitor and override the machine in the event of an error.

Results and Discussion

The SysML model activity diagram of the Manual System, as displayed in Figure 2, displays one complete instance of two high level activities: 1) determining which ship to move and 2) which route to draw for it. The operator attempts to attain awareness of the current state of the game environment by identifying all bonuses, ships, likely collisions, no-fly zones, and ships heading for no-fly zones. Based on the operator's priorities, he or she will determine the best ship to move. Potential routes are created by the operator and one is chosen based on earning the highest amount of points possible. The desired ship is selected and the route is drawn. As shown in this diagram, each action performed by the human operator is depicted within an round-tangle. Parallel actions are enabled through the use of the horizontal bars within the figure, depicting splits and joins. The arrows (flows) show the information or control logic which is created within one action and is necessary for the performance of the receiving action. An IMPRINT model corresponding to the activity diagram in Figure 2 is shown in Figure 3.

As shown, each of the actions represented in the activity diagram are depicted in the IMPRINT task network.

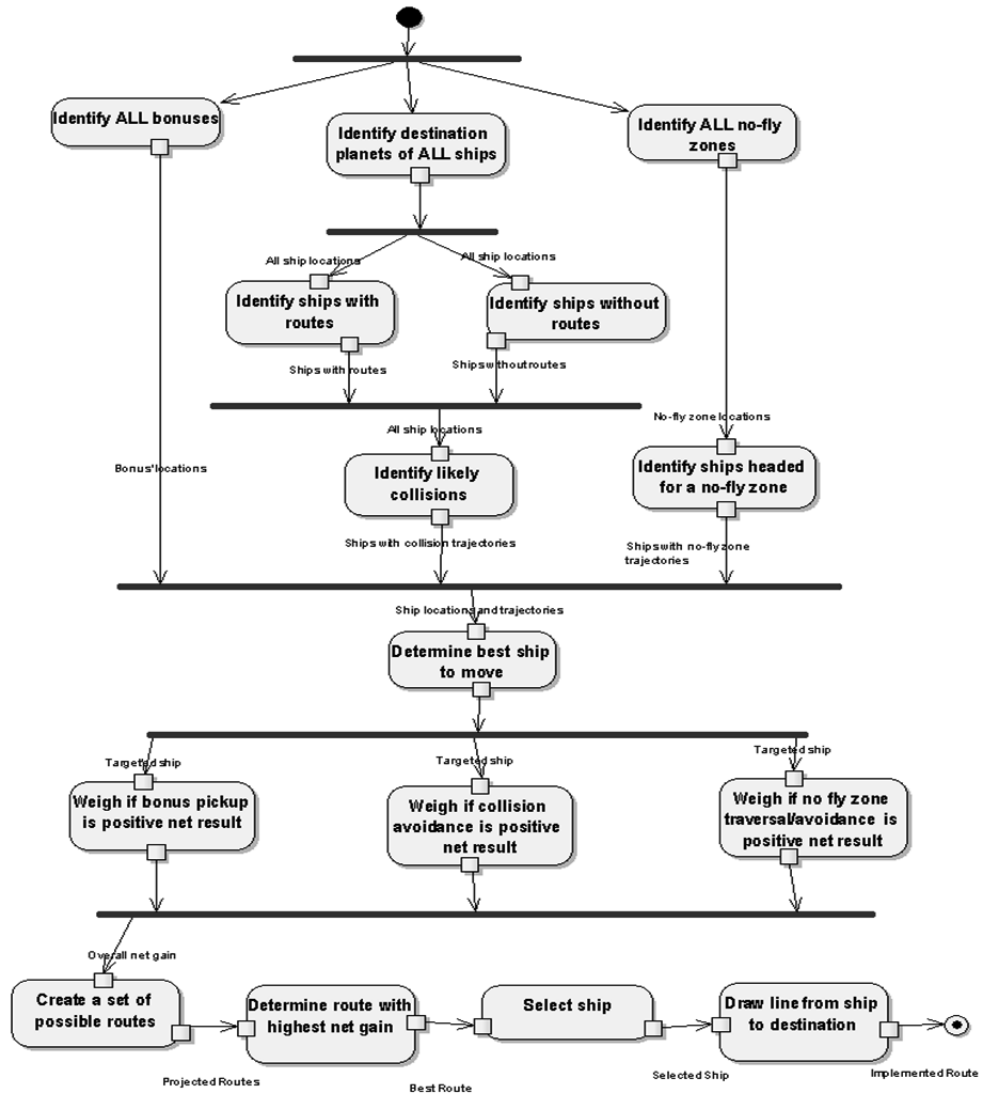


Figure 2: Activity Diagram for the Manual System.

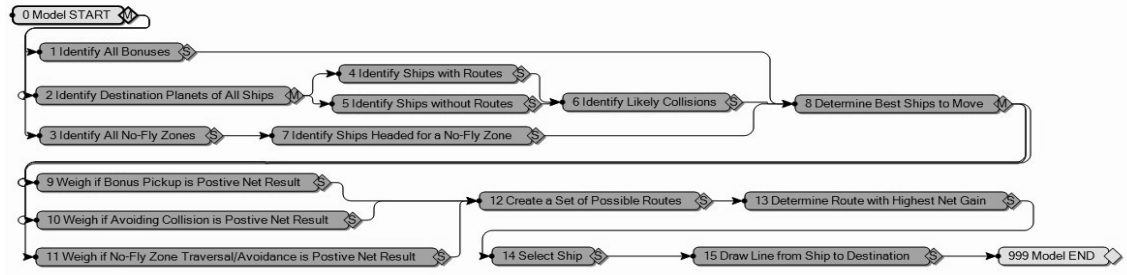


Figure 3: Imprint model illustrating the Manual System. Although not shown, the final model will also likely include the system events (e.g., ship spawn frequency), which are likely to influence human performance.

To explore the implementation of automation through re-allocation of some of the actions, we assumed that the first high level activities, i.e., determine which ship to move, was allocated to the computer and the second high level activity, i.e., determine which route to draw for it, remains with the human. As a default, this change in allocation can be depicted by simply introducing swim lanes to Figure 2 as shown in Figure 4, Automation with Direct Re-Allocation, which indicates the allocation through a “Computer” swimlane in the top of the diagram and a “Human” swimlane on the bottom of the diagram.

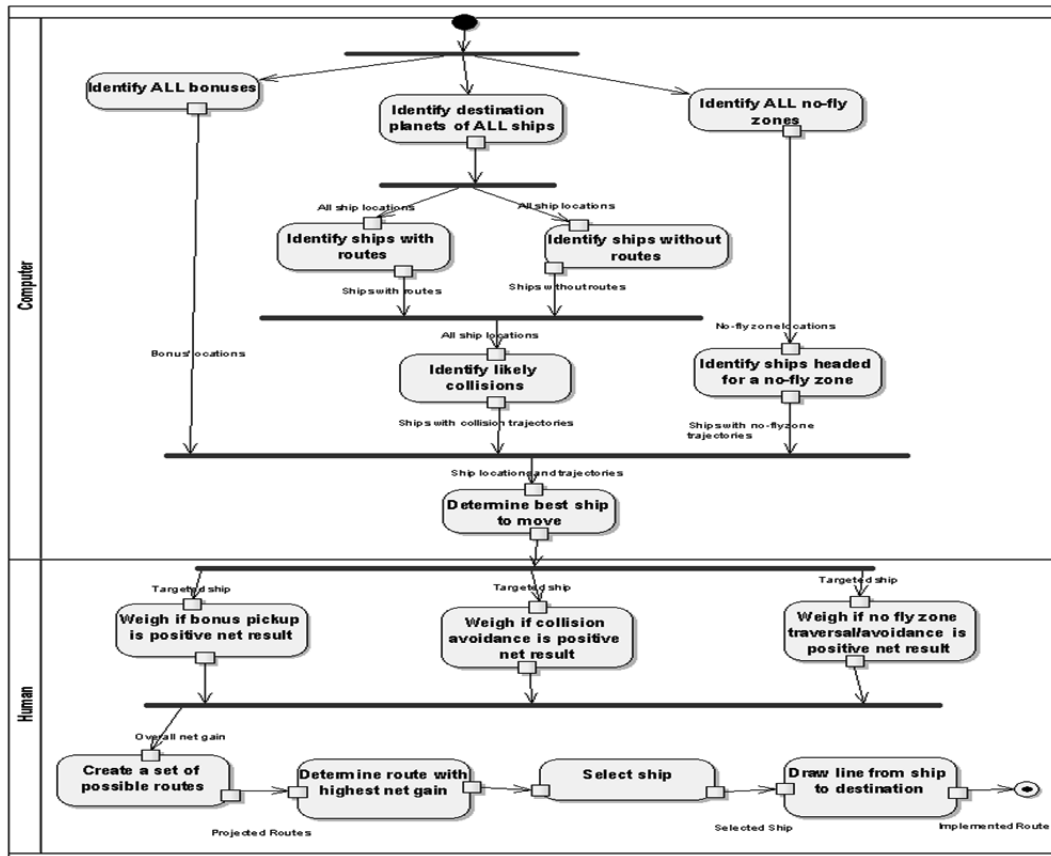


Figure 4: Activity Diagram for Automation with Direct Re-Allocation, with first order task allocation

This reallocation can then be indicated in IMPRINT by assigning the actions associated with the Computer to the new entity, with this change indicated by the difference in the color of the nodes within Figure 5, where blue and lavender indicates computer and human control, respectively. In these diagrams, the computer is responsible for determining the best ship to move. Afterwards the human operator decides which route to implement.

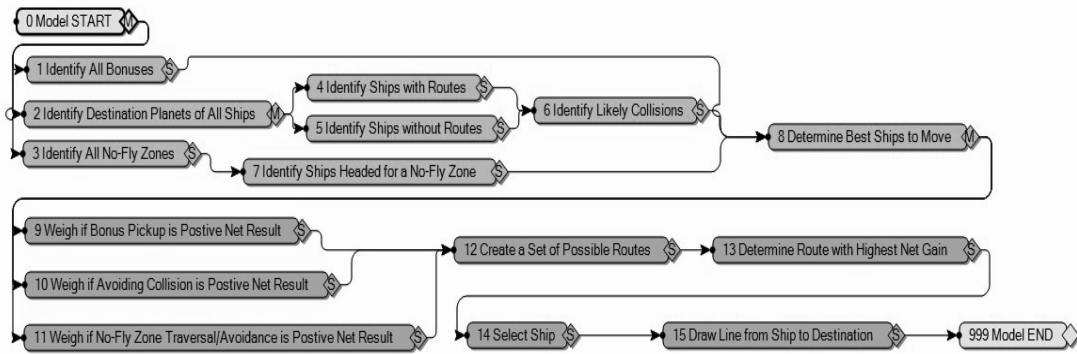


Figure 5: IMPRINT model for Automation with Direct Re-Allocation, displaying computer and human control by blue and lavender nodes, respectively.

Note, however, that in the Manual System model, the human scanned the entire set of objects on the screen and assembled all of the knowledge necessary to know for which ship to draw a route and the reason that a new route was necessary (i.e., new ship without a route, impending collision, heading for new no fly zone, new nearby bonus available). In the Automation with Direct Re-Allocation model, the human has no way of knowing which ship to move or why such a move is important, as the computer has assembled this knowledge but the information has not been transmitted to the human. The need to capture the communication of this information is inserted into a third set of models shown in Figure 6, Automation with Handover. Key adjustments to note are the replacement and addition of action nodes capturing human-computer communication, with the computer relaying to the human why it targeted a specific ship and which ship it targeted (for example by flashing a light around the targeted ship with the color of the light corresponding to the matter that is pressing, e.g., red is a collision, yellow is a no-fly zone, etc). When given this information from the computer, the human identifies the

relevant information surrounding the targeted ship, and then creates a set of routes after perceiving the environment around that particular ship, not for the entire screen. As such, this automation aid has the ability to reduce the human's workload as he or she does not need to assess the state of the entire game, only the portion of the game relevant to the highest priority ship, as determined by the game. Unlike the Automation with Direct Re-Allocation model, this task network appropriately identifies additional communication nodes required to ensure an effective handoff between the automation and the human. However, the addition of these communication tasks adds workload beyond what is captured by the simple re-allocation in the Automation with Direct Re-Allocation model.

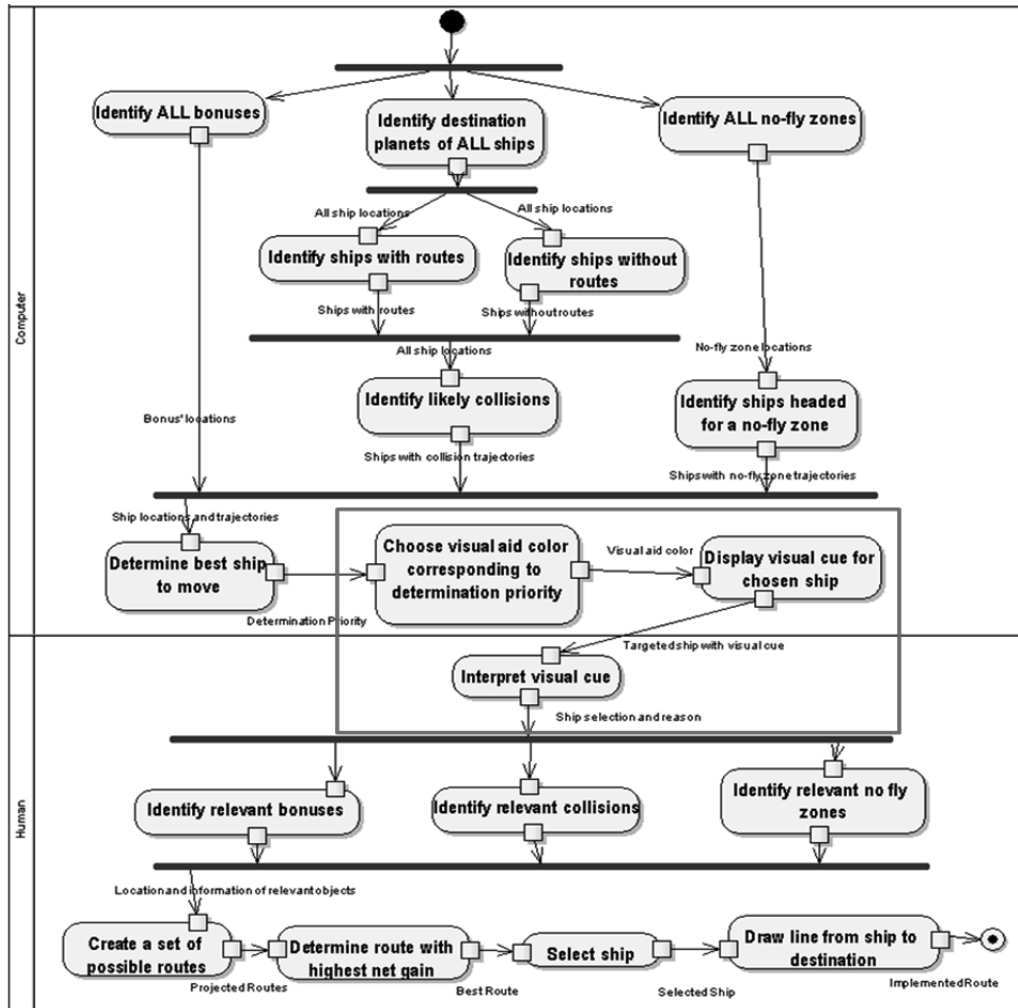


Figure 6: Model displaying automation with handover including a red outline highlighting the communication between the human and automation.

Although the Automation with Handover has the potential to improve the user's performance, assuming that the computer accurately identifies the most important ship to be addressed and the human and computer perform in complete symbiosis, this interaction has the potential to result in less than ideal performance. As (Stensson and Jansson, 2014), has indicated, human interaction with automation is necessary since the computer cannot be held responsible, while humans which have the ability to feel

remorse among other emotions are assumed to be responsible, particularly for catastrophic outcomes. As such, it will often be necessary for the user to maintain overall situation awareness of the environment to maintain supervisory responsibility over the actions. Unfortunately, as the human relinquishes all ability to verify that the computer has in fact chosen the most important ship to route, the human is unable to maintain responsibility for the task. To enable sufficient situation awareness, many of the functions allocated to the computer in the Automation with Handover model, must also be performed with some degree of regularity by the human to enable the necessary awareness, as shown in Figure 7, Automation with Supervision. Note that in this case, the human is performing as many actions as in the manual system, including actions from the first high level activity, determine which ship to move, which was allocated to the computer. In this scenario, which is not uncommon for automated systems under human-supervisory control, the automated system is unlikely to produce the expected workload benefits.

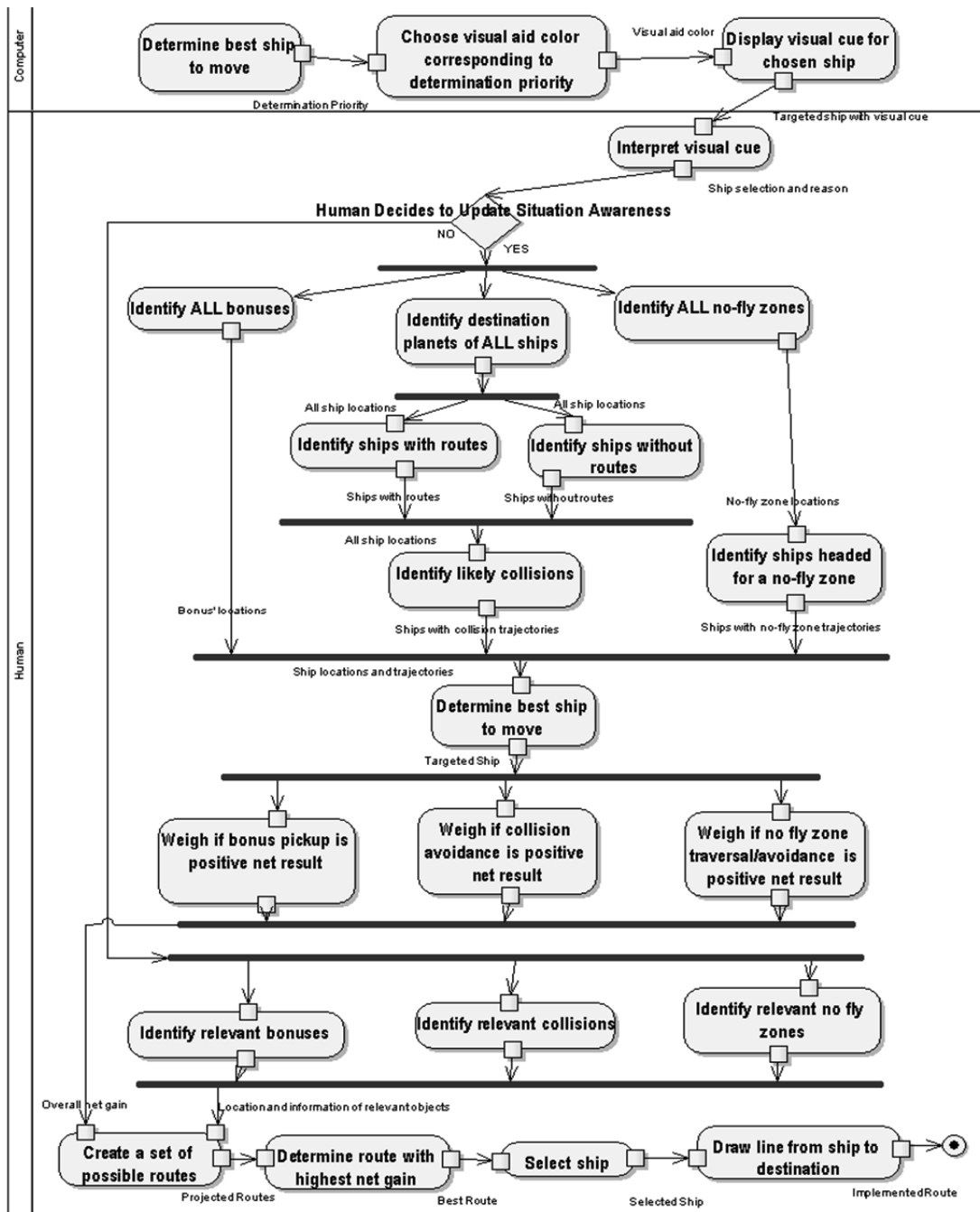


Figure 7: Partial display of the activity diagram for automation with supervision, incorporating communication between the automation and human, as well as the human's decision to update situation awareness.

Discussion

The Automation with Direct Re-Allocation Model appears to be a simple and efficient method in adapting a workload model to account for task re-allocations, as it only involves the inclusion of “swimlanes” allocating necessary actions to the human or computer. This provides a model that makes it easy to comprehend which actor is in control of specific tasks. Although this simple modification appears beneficial, it does not accurately capture the true system interactions that will result for incorporating automation.

The major pitfall in the Automation with Direct Re-Allocation Model is that it displays the human operator as seamlessly interacting with the computer without gaining the knowledge necessary to perform the actions assigned to the human. This is a significant issue as it is recognized that the human must sense their environment, perceive relevant information from the environment, decide upon a course of action given this information, and then take action, with each of these phases requiring both mental resources and time (Parasuraman, Sheridan, & Wickens, 2000). The lack of additional communication actions in the baseline model of the Manual System is accurate as the human operator completes each of these four steps on his/her own. However, in the Automation with Direct Re-Allocation, the computer gains awareness of all information necessary to select the best ship to move, and then the human implements a route without perceiving the information necessary to select or implement a route. If the operator lacks awareness of his or her surroundings and does not know why a ship is deemed the most

critical to move, then he or she will not be able to draw a route that properly addresses the problem at hand.

The Automation with Handover Model fills in these assumptions by including actions which permit the computer to communicate the necessary information to the user during the exchange in responsibility and actions necessary for the user to gain awareness of the situation enabling the decision. This automation reduces human workload by reducing the number of objects in the environment that the human must attend. Unfortunately, this action reduces the user's situation awareness. The final model, Automation with Supervision, then adds additional actions the human must perform to regain this situation awareness. In the final environment, the time allotted by the human for gaining situation awareness versus route creation will depend on the human's trust in the automation, system reliability, time available, and the relative importance he or she assigns to each of these higher level activities, all of which will need to be captured in the workload model.

Conclusions

This paper has illustrated the potential use of SysML together with IMPRINT to illustrate the construction of models to assess task re-allocation. Although initial allocation of actions within these models appears simple and intuitive, only requiring designating responsibility for existing actions, key assumptions are not explicitly depicted in the model. Adaptation of a model to include task re-allocation requires careful consideration in the areas of human-automation communication and adjustments in behavior. It is significant for the developer to understand that task flow between a human

and computer involves some type of input or output from both. Any adjustments in human behavior, arising as a result of automation, need to be addressed and input into an adapted model. Revision of action nodes and the inclusion of human-computer communication, as well as human monitoring to gain situation awareness, results in an activity diagram and IMPRINT model that is able to more accurately represent the system and project the workload of the human operator.

In the development of a new system, the accuracy of a model, or set of models, is critical to the further development of the system. Models and simulations are often made in the conceptual phase of system development, capturing the fundamental elements of projected system attributes and behavior in a cost efficient manner. Conceptual modeling is the cornerstone for Model Based Systems Engineering (MBSE), affecting nearly every aspect of the development and implementation of the system. If the model neglects certain aspects of the system, this could have a negative impact on the project's budget, schedule, requirements, functionality, and feasibility. Therefore, in the context of modeling human-computer interaction, one needs to apply careful consideration regarding communication, situation awareness, and behavior to properly capture system behavior and avoid undesired costs.

Future Research

The current research primarily focused on modeling theory when considering human-computer interaction. The next step would be the application of these theories by using the models to estimate system performance and human workload for each of the system designs discussed (Manual System, Automation with Direct Re-Allocation,

Automation with Handover, and Automation with Supervision). Model outputs regarding predicted system performance and workload could be validated using human test subjects for each of the system designs. This would enable a quantification of the negative impacts from inadequate automation task re-allocation.

III. Timing Within Human-Agent Interaction and its Effects on Team Performance and Human Behavior

Abstract

Current systems incorporating human-agent interaction typically place the human in a supervisory role and the agent as a subordinate. However, a key aspect of teaming is the dynamic shift in roles. Depending on the situation at hand, teaming could lead to a peer relationship where the human and agent are working together on the same task. This research investigates how the timing of agent actions impacts team performance, as well as human workload and behavior. A human-in-the-loop experiment demonstrated that when the agent performs tasks faster than the human, the human tends to become reliant upon the automation and assumes a supervisory role. A human performance model predicts that extending agent execution time will decrease human reliance on the automation. However, in the environment under investigation, a tradeoff exists between team performance and human involvement.

Introduction

Human Machine Teaming

The growing development and use of semi-autonomous systems has been beneficial in accomplishing tasks that would otherwise be error prone, dangerous, unmanageable, or simply impossible for humans (Milot, 2014). Research efforts in this field have also increased in response to the rapid rise in technological capabilities. Significant and foundational pieces of literature have described autonomous systems as

having several levels of automation when performing tasks typically allocated to a human operator (Endsley & Kaber, 1999; Parasuraman et al., 2000) This description of automation coincides with the design of several team-based descriptions of humans and autonomous coordination systems, including function allocation, supervisory control, adaptive automation, and dynamic task allocation (Johnson et al., 2011).

One determining factor that separates a team from an ordinary group is a shared goal by all members (Bruemmer, Marble, & Dudenhoeffer, 2002) where cooperation is needed to limit interference between members during goal completion (Hoc, 2001). The purpose of teaming is to “increase the level of task performance by leveraging the unique capabilities of each performer, taking advantage of each member’s strengths and available resources” (Bruemmer et al., 2002). Each team member’s unique capabilities can help build interdependency when tasks cannot be performed by any individual alone (Arthur et al., 2005). To use each team member’s strengths appropriately, teamwork is needed to facilitate interactions.

Current human-machine teams typically allocate responsibility such that the machine is subordinate to the human, thereby limiting the potential to which the team can leverage each member’s unique strengths. Comparatively, effective human teams implement dynamic allocation of roles, responsibility, and authority dependent upon members’ capabilities, availability, and task load. It is suggested that human-machine teams should model this schema to maximize performance in a dynamic environment. In classic systems, the machine usually fulfills the role of tool or subordinate, never reaching the status of a peer or leader. By allowing the machine to attain higher status, an

emphasis on interdependence and communication emerges as each becomes more reliant upon the other (Bruemmer et al., 2002).

Significant differences exist between humans and machines as team members. These differences not only include machine deficiencies, such as the limited ability to reason within the current context and respond to surprises in a robust manner (Huntsberger, 2011), machine difficulty in communicating priorities (Klein, Woods, Bradshaw, Hoffman, & Feltovich, 2004), and lack of machine accountability (Anderson, Anderson, & Armen, 2004); but also seemingly pedestrian issues, such as ill-defined temporal requirements for operations.

The human information processing loop extending from perception through completion of an action often requires at least one third of a second and, depending upon the size of the muscle movements involved, can require multiple seconds. However, an agent, embedded in a computing system can perform a similar sequence of events in a much shorter period of time. Therefore, a designer may automate a process to improve system performance and decrease human workload. However, the incorporation of an automated tool can lead to the human adopting a supervisory role, which can be harmful to production. It has been documented that humans are poor monitors, a role they often assume when acting in a supervisory capacity, because they lose vigilance and are prone to fatigue (Parasuraman, 2008). Loss in vigilance can result in the human being “out of the loop”, ultimately losing situation awareness. Consequently, it can be difficult for a person to understand the full context of a situation, possible actions, and consequences if

they have lost situation awareness when an unusual situation arises to which the automation cannot respond appropriately.

The idea of the human and machine working together as peers suggests that the human does not assume a supervisory role, but rather, the two are working alongside one another. There is a desire for the two to cooperate in such a manner where they are attaining adequate performance, yet the human is “in-the-loop” and maintaining situation awareness.

Triggers permit the automation to respond to events in the environment and actions by its team members. Triggers are developed to afford the automated system the ability to sense, observe, or model the environment to create a relative understanding of the events taking place around it and alter its behavior based upon this information. The goal of the automated agent is to receive relevant information from the environment and act accordingly (Feigh, Dorneich, & Hayes, 2012a). Therefore, the trigger affects the automation’s timing, i.e., time at which a task is initiated. Logically, the timing of task execution in highly dynamic, event-driven domains must influence the performance and behavior of the team. Considering that automated systems have the potential to respond much faster than their human counterparts, their response time can affect task responsibility. If the automation’s response time is too short, the human operator may assume the supervisory role as the automation will always respond to an event faster than its human counterpart. However, if its response is excessively delayed, the human is likely to assume responsibility for the event and attempt to respond before the

automation. However, the proper timing and changes in the behavior of human team members as a function of automation response time is not apparent in the literature.

Therefore, this research aims to understand the effect of automation's task timing on the performance of the human-machine team. This effect is examined using a combination of human-in-the-loop experimentation and human performance modeling within an environment employing an autonomous agent. A previous experiment is described which incorporated an autonomous agent that was triggered by the co-occurrence of an environmental event (i.e., appearance of a new task) and human inactivity in addressing this task (Bindewald et al., 2014). The time frame at which the agent considered human inactivity to be excessive was static throughout the experiment. However, based upon the results of this experiment, it is assumed that variation in task timing of the automation will have a significant impact on user behavior. Thus, this research was conducted to explore the type of effects task timing has on team performance, as well as, human behavior and workload.

Method for Previous Experiment

Participants

The experiment involved 36 volunteers with an average age of 32.5 years and a range of 22 to 39 years. A total of 30 males and 6 females participated.

The experiment involved the use of a computer based tablet game environment. Thus, each participant was asked how often they use laptops, tablets, desktops, phones, and gaming consoles. On average, they used tablets roughly 1-3 times a week and gaming

consoles 1-3 times a month. Other computer based platforms, including smart phones, were reported being used 3-7 times a week.

Apparatus and Environment

Space Navigator is a tablet-based computer trajectory-generation game which was constructed to provide a controlled representation of a highly-dynamic, event-driven environment. In these environments, the operator has little, if any, control of the event rate and there is no guarantee that the human will be capable of responding should unexpectedly high event rates occur. Similar environments might include air defense systems and certain command and control environments. The game, while not providing a high fidelity simulation of these environments, permits the control of the event rate and other potentially confounding variables, logging of human response, and the creation of automations that can be enabled to assist the operator during high event rate conditions. The use of the Space Navigator game for this study simplifies participant recruitment and training.

Figure 8 displays a screen capture from the game and identifies several key objects within the game. Spaceships appear at set intervals from the screen edges. The player directs each spaceship to its destination planet, designated through color, by drawing a line on the game screen using his or her finger. The spaceship then follows the entire drawn trajectory unless the player draws a different route for the ship. Points accumulate when a ship encounters its destination planet or one of a number of small bonuses that randomly appear throughout the play area. Points decrement when spaceships collide, and each spaceship involved in the collision is lost. Points are also lost

when a spaceship traverses one of several “no-fly zones” that move to random locations within the play area at a set time interval. For every second a spaceship traverses a no-fly zone, the player loses points. The game ends after five minutes.

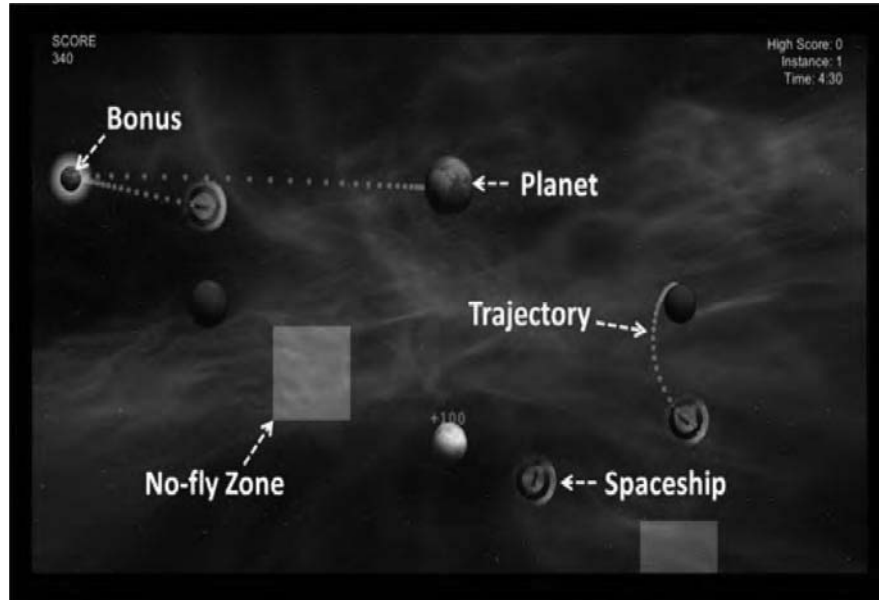


Figure 8: Screen capture from Space Navigator, highlighting spaceships, planets, trajectories, bonuses, and no-fly zones.

In addition to drawing the routes manually, the subjects also work in human-agent teams in which both the subjects and the agents draw routes. There were three types of automated agents: straight line, similar to the user, and dissimilar to the user. The straight line automation draws straight-line routes from the ship to the corresponding planet. The similar to the user automation uses a player model developed based on manual game play to draw routes predicted to be similar to those that the user would draw under similar circumstances. The dissimilar agent, selects random trajectories from the past game-play database. To provide the human with an opportunity to draw routes,

the agent does not draw routes instantaneously, rather the automation triggers after a specified amount of on-screen time for a ship has elapsed without the subject interacting with that ship.

Experimental Design and Procedure

The experimental procedure consisted of a within subjects design in which each participant completed 16 five-minute instances of Space Navigator. The initial five instances contained no interaction from an automated agent and were used as participant training sessions. Following the training, participants completed three experimental sessions. Experimental sessions included four five-minute instances and each instance attributed one trajectory type to the agent throughout the entirety of a five-minute game. The four types of trajectories were either similar to the user, dissimilar to the user, straight line, and none (participant performed the task without an automated agent as a partner). Ships appeared on screen at a fixed rate of one ship appearing every two seconds. Bonuses and no-fly zones repopulated every thirty seconds.

Data Analysis

Game play and NASA-TLX (Hart & Staveland, 1988) data were collected to assess user performance and workload per agent type. The Space Navigator environment actively stored information every time a ship-related action occurred. These actions included trajectory draws, collisions, bonus pickups, destinations reached, no-fly zone traversal, and off-screen movements. Subjective workload values were input by participants after completing each five-minute instance. Users were asked questions related to workload, frustration, and agent trust at the conclusion of the experiment.

Although data was collected for three different agents, which performed differently from one another, the data analysis for the current paper was constrained to include only the manual condition in which there was no agent and the straight line agent, which drew a straight line from the ship to the appropriate planet anytime a ship resided on the screen for 2 s during which the participant did not draw a trajectory.

Experiment Results and Discussion

The expected result from this experiment was that the participants would continue drawing routes, relying on the automation to draw routes only when they were overloaded to the point that they could not draw routes quickly enough to be successful. The rationale behind this assumption was that this agent would be able to work alongside the user, but work less effectively and therefore not be trusted to draw routes unless the individual was task saturated to the point that they could not draw routes quickly enough. Therefore, it was expected that the majority of trajectories would be drawn by the participant. However, participants' behavior unanimously differed from this reasoning.

As shown in Table 1, when interacting with the game in a manual mode, without the agent, the human participants drew an average of 126.26 routes for the 150 ships that were generated during the 5 minutes of game play. Further, they redrew 21.83 routes for ships that they had already designated routes. However, when the straight line agent was employed, the humans drew less than 1/5th as many trajectories on average (i.e., 23.19) than they did when playing the game manually. Additionally, when the agent was present, the participants redrew just over twice as many routes (mean of 43.97) as they did when operating in manual mode.

Table 1: Manual and Straight Line Agent Data

	Fully Manual		Agent Assistance	
	Mean	St Dev	Mean	St Dev
Score	5801.57	2327.62	8043.06	1573.72
Hum. Draws	126.26	12.58	23.19	24.26
Redraws	21.83	11.94	43.97	15.55

Initially, it appeared counter-intuitive that the human participants would relinquish most of their initial path planning to an agent when the agent is incapable of making decisions based upon obvious obstructions or bonuses in the environment. However, this behavior becomes more understandable when one computes the average human ship-selection cycle-time. A full ship-selection cycle for the human involves identifying a ship to select, physically selecting a ship with their finger, and drawing a designated path. Analysis of this data reveals that an average of 2.6 s is required for a participant's ship-selection cycle-time whereas a new ship is spawned every 2 s. Therefore, it is implausible for the average human to successfully generate paths fast enough to provide a path for every ship. Conversely, the agent draws a route at the same speed as the ship spawn rate, drawing a route for the previously generated ship when the subsequent ship appears.

In this environment, with intuitive ship movement and a predictable agent, the participants were able to predict the behavior of the agent and then adjust undesirable paths. Consequently, it would appear that users began to initiate fewer trajectories, supervising the agent and redrawing paths to improve performance. As seen in Table 1, the addition of the agent increased the average score by roughly 2250 points (a 39%

improvement) by having the human draw 103 fewer routes and doubling the number of redrawn routes.

Given this interaction, we therefore sought to better understand the interaction of the autonomous agent's timing within this environment. The trigger time employed in this experiment created an agent that assumed the human was overloaded if the human was unable to address an incoming ship within 2 s. It appeared that the automation's task timing exceeded the human operator's ability, relegating the operator to more of a supervisory role. Therefore, the participant game play data from this human-in-the-loop experiment was leveraged to construct a model of human-machine interaction. The model was used to examine how variation in the automation's timing affects team performance, human behavior, and workload, within the teaming environment.

Space Navigator IMPRINT Model

IMPRINT Simulation Software

To examine timing in the context of a human-machine team, this study uses the Improved Performance Research Integrated Tool (IMPRINT), a discrete-event simulation environment ("Improved Performance Research Integration (IMPRINT) Tool," 2010). This environment models human workload and performance as a function of time by tracking activities performed by a human or a machine. These activities are described in a task network, which includes task sequencing and decision points. The frequency of the tasks, as well as the time necessary to perform each task result from a stochastic process, permitting the modeler to represent the variability within the system. Different task networks can be derived for different goals and a workload level is assigned to each task

performed by the human operator. Various system allocations can then be modeled by allocating specific tasks to be performed by the human operator or machine (hardware or software). However, to employ this tool to accomplish this goal, the modeler must begin with activities to be performed by the team, allocate these activities to the human or machine and then derive the tasks or actions necessary to perform these activities. Once these activities are allocated to a component, human or machine, other inherent tasks may become necessary to facilitate communication of system state as control is passed between the human and machine (Bindewald et al., 2014; Goodman et al., 2015).

IMPRINT Task Network

The IMPRINT model is depicted in the SysML Activity Diagram (Delligatti, 2013) shown in Figure 2. This diagram divides the activities among three primary sections, separated by vertical lines known as “swim lanes”, which separate the activities of the environment, the human operator, and the agent. The environment nodes in Space Navigator are responsible for starting the model, generating ships, altering no-fly zones and bonus locations, operating the timer, and halting the model as shown in the center “swim lane” of the activity diagram.

The player’s attention and actions during game play are facilitated through a loop, continuously repeating two high level functions; determining which ship to select and drawing a trajectory for a ship. However, the loop is completed both for ships that have no drawn trajectories and for those that have a non-optimal trajectory. A view of ideal game play may include the person working to their capacity as they try to earn the highest score, leveraging the agent to draw paths they do not have time to draw. This behavior is

depicted through the path in Figure 9 within the Human swim lane, which includes identifying background items, identifying ships without routes without waiting for the agent, selecting a ship and drawing a route. However, as demonstrated in the experiment, the human could permit the agent to draw some initial paths permitting them to attend to other tasks within the game. Thus, a task load node, indicated by the first decision node in the human swim lane, is used to represent a human's decision to either initiate ship selection or monitor the environment, allowing the human to observe the agent as it creates routes. The decision to monitor is based on a reliance algorithm derived from the experimental data, as seen in Figure 10.

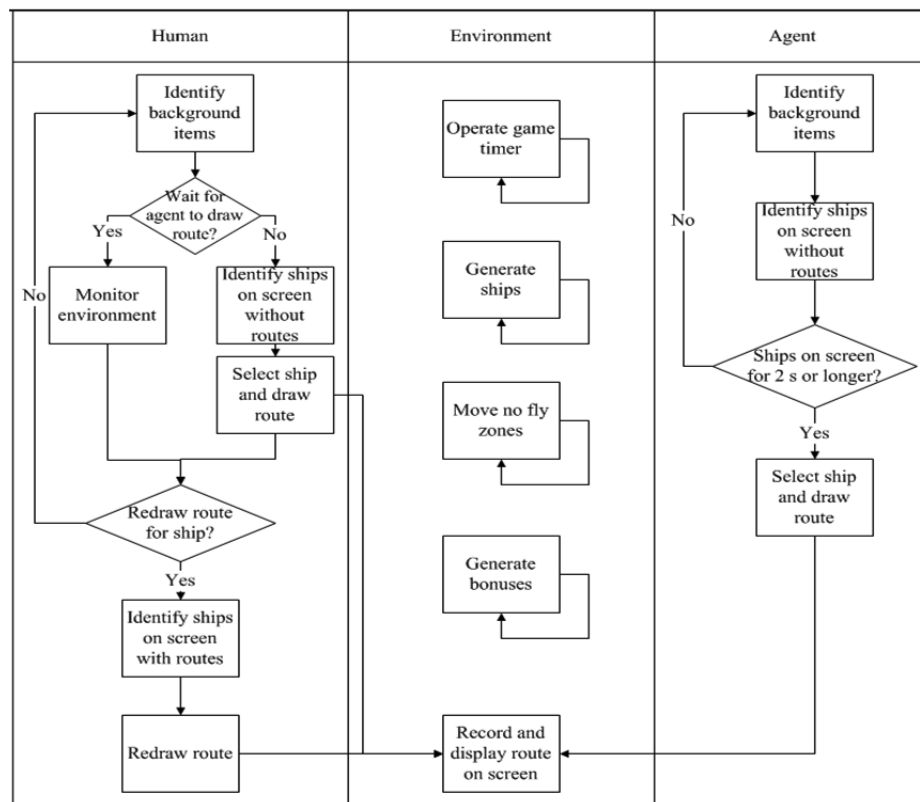


Figure 9: Activity diagram representing the actors and actions in the IMPRINT model. Vertical swimlanes are used to designate actions.

The reliance algorithm produced a probability that the human would permit the agent to draw a trajectory. Analysis of the experimental data indicated that the probability of the agent drawing an initial route as a function of the number of ships on screen produced a parabolic curve. The participants performed more route draws when the number of ships on screen was low as they likely had ample time to interact with the system. They also appear to have drawn more routes when larger numbers of ships were on screen to help avoid collisions, given the agents' inability to react to neighboring ships, no-fly zones, and bonuses. The regression curve in Figure 10 accounted for the reliance of the operator on the automated agent with respect to the number of ships on screen.

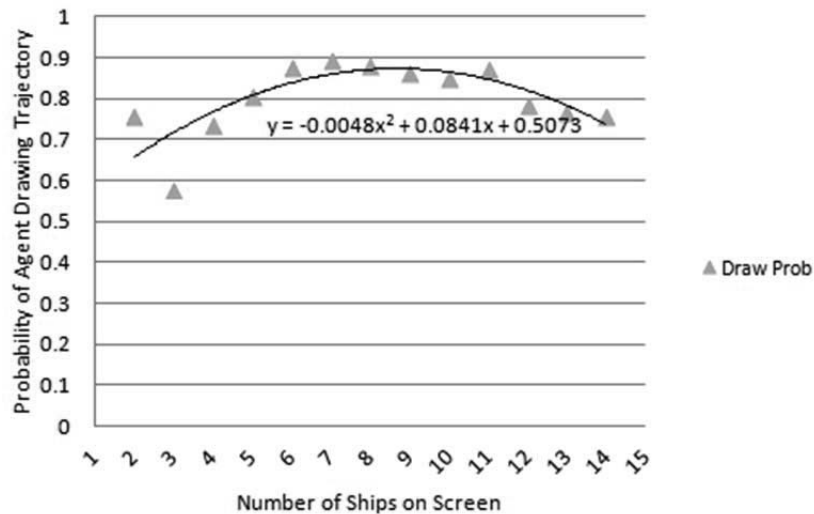


Figure 10: Graph displaying the probability of the agent drawing a route with respect to the number of ships on the screen. The third order regression line, with equation, was used in calculating the reliance algorithm in the IMPRINT model.

The other factor that was necessary to include in the reliance algorithm was the trigger time of the agent. While no data exists to construct this function, it was assumed that the longer the agent takes before assigning a route, the more likely the human will initiate tasks to avoid losing points. At the lower limit, if the agent drew the line as soon as the ship appeared, the person would never have time to initiate a route. However, in the case that the agent requires an infinite amount of time before drawing the route, the human cannot rely upon the agent to draw any route. The operators' average cycle-time, time between initiating routes on separate ships, and standard deviation were derived from the experiment's fully manual gameplay. Using three standard deviations above and one standard deviation below the mean of 2.6 s, it was assumed that a human would be unlikely to initiate a route for a ship at 0.1 s and the agent would be unlikely to initiate a route at 11.6 s. This assumption was used to determine points on a linear equation relating delay time to probability of agent draws. This linear model was used to shift the third order regression line shown in Fig. 10 downwards as the agent's time delay increased and shift the regression line upwards as the agent's time delay decreased. For every second that the agent's delay changed, the baseline probability value was incremented or decremented by 0.1058, within the bounds that the probability must be between 0 and 1.

Returning to the task network, if the operator decides to draw a route based upon the reliance algorithm, they will continue to identify ships on screen. Afterwards, they can draw a route for a ship that does not have a route, or they can "redraw" a route for a

ship that has an existing route. Following the draw route node, the human attention loops back to determine the background items, where the number of items impose a taskload, which is modeled as the number of ships on screen. As shown, when the human draws a route, the environment is updated, permitting both the agent to be informed by recording the route and displaying the route for the human.

Simultaneously, the agent is selecting ships and drawing trajectories for them as well, as indicated in the Agent swim lane of Figure 9. Unlike the human, the agent does not have the option to perform fewer tasks. The agent is constantly monitoring all ships on screen and drawing a route once the time trigger has occurred. However, unlike the human, the agent can only draw trajectories for ships that have not received a trajectory, and the agent does not redraw non-optimal trajectories. As the agent draws a path, this information is provided to the environment.

After the human or agent has designated a route for a ship, a new entity is created in the model, representing the ship with a route. The ship continues along its path for a length of time drawn from a distribution representing game play time-on-screen and is removed from the simulation after the time has elapsed (not depicted in Figure 2). There are three possible end results for a ship: collision, destination reached, and off-screen traversal. Ships arrive to these nodes according to probabilities associated with the number of ships on screen and the human or agent that drew the route. Once again these distributions are developed from the human-in-the-loop data discussed earlier.

Model Validation

To validate the model, the model was exercised for conditions that matched the conditions of the previously explained human-subjects experiment and the results were compared. The IMPRINT model replicated the experimental trials by having the agent create routes for ships that were on screen, and without a route, for two seconds or longer. The results applied for model validation were scores, number of automation trajectories drawn, and number of “redrawn” trajectories by the operator. These specific aspects of the model were chosen to ensure that performance and behavior, as predicted by the model, was similar to the data from the human-in-the-loop experiment. To compare score values and trajectories, two sample t-tests with 95% confidence intervals were performed. For score, the average from the experiment was 8043 (sd 1574) while the mean from the model was 8053 (sd 871). The t-test indicated that these values were not statistically different ($t(1,169) = -0.06, p=0.955$). The average number of agent-drawn trajectories from the experiment was 126.9 (sd 24.4) and the mean from the model was 122.4 (sd 3.24). The t-test indicated that these values were not statistically different ($t(1,109) = -1.91, p=0.06$). The average number of human redraws from the experiment was 44 (sd 15.5) and the mean from the model was 45.59 (sd 6.08). The t-test indicated that these values were not statistically different ($t(1,141) = -1.00, p=0.318$). Overall, there was no evidence of statistical differences between the model and the experimental data, and thus the model is considered validated.

The workload values collected in the human-subjects experiment were NASA-TLX values, whereas the workload inputs in IMPRINT are from the Visual, Auditory,

Cognitive, and Psychomotor (VACP) workload assessment tool. Consequently, workload could not be directly validated. Thus, validation was conducted with a subject matter expert.

As the slope of the linear equation relating agent delay time and probability of an agent draw was assumed during model construction, it is important to understand the sensitivity of the model to this slope. Simulations were run with a 10% increase and decrease to this slope. At the lower bound, on average the human drew 2.5 fewer trajectories and scored 24 fewer points. At the upper bound, the average score increased by 18 points and the human drew 2.14 more trajectories. The change in both values was greatest during the 8.6 s delay time where in the lower bound the human drew 10 fewer trajectories and in the upper bound the human drew 8.35 more trajectories. The difference in score for both fluctuated and had no distinct pattern. The change in workload and redraws was negligible. Therefore, it is believed that changes in this slope will significantly affect the model results for delay times near the intersection of the linear model with the delay time axis. However, the characteristic shape of model output as a function of timing delay is likely to be robust.

Simulation Procedure

A series of simulations were conducted in which the trigger time of the automated agent was altered in each simulation. Trigger times were selected based upon participant performance. As noted earlier, the participant required an average of 2.6 seconds between the time a ship is spawned, appearing on screen, and the time the human selects the ship to draw a trajectory. The associated standard deviation of this time was 3.0

seconds. Six conditions are evaluated: the mean time for a participant to select a ship (2.6 s), plus one-half, one, two and three standard deviations (ie., 0.1, 5.6, 8.6 and 11.6 s), as well as the original 2 s delay employed in the human-in-the-loop experiment. The six scenarios were each simulated 100 times, having the same random seed value for each condition. At the end of each scenario, the average scores, workload, and trajectories drawn were calculated. A one-way Analysis of Variance (ANOVA) was used to determine whether agent delay-time had a significant effect on any of the model outputs and Tukey Pairwise Comparisons were used to test for differences between individual means.

Simulation Results

The results from the IMPRINT simulations displayed an inverse relationship between performance and workload, as shown in Figure 11. As the trigger time increased beyond the average time of 2.6 s, the operator's workload increased and overall team performance decreased. Although performance, in terms of overall score, was recorded for each of 100 model runs, workload is shown for a typical single model run.

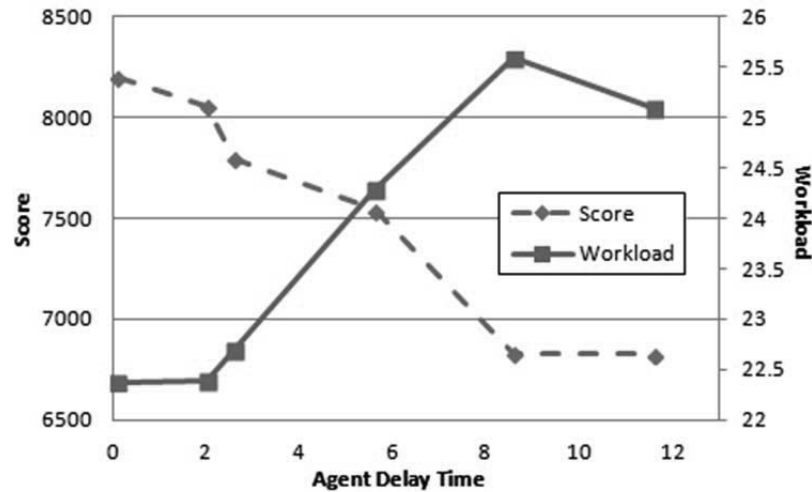


Figure 11: Graph displaying average score and workload per agent delay time.

The ANOVA indicated that the effect of agent trigger time on overall score is statistically significant ($F(5,594) = 43.28$; $p < 0.001$). Tukey pair-wise comparisons indicated that there were four groups of scores that were significantly different from one another. These groups in terms of agent time delay were (0.1, 2.0), (2.0, 2.6), (2.6, 5.6) and (8.6, 11.6). It was shown in these pairings that as the time delay increased the average score significantly decreased.

As shown in Figure 12, the human and agent draws were also inversely related. The agent's trigger time significantly affected agent draws ($F(5,594) = 35784$; $p < 0.001$), human draws ($F(5,594) = 31975$; $p < 0.001$), and redraws ($F(5,594) = 174$; $p < 0.001$). The ANOVA for agent and human draws produced similar results. The number of human draws were statistically different for all agent redraw conditions. The effect of time on redraws generated three different groupings, with 0.1 s producing the most redraws, followed by 2.0 and 2.6 s conditions and 5.6, 8.6, 11.6 s conditions.

These simulations indicate that human behavior will change as a function of the agent's trigger time. When the agent created routes at the same speed or faster than the human, the human initiated routes between 2% and 20% of the time. When the trigger time is one to three standard deviations slower, the number of human initiated routes increased from 50% to 95%. Furthermore, the model anticipated that the largest shift in performance would occur when the trigger time was adjusted from 5.6 to 8.6 s, decreasing the score by 10%. The greatest increases in workload should occur when delay times change from 2.6 to 5.6 s and 5.6 to 8.6 s, with a 7% and 5% increase, respectively.

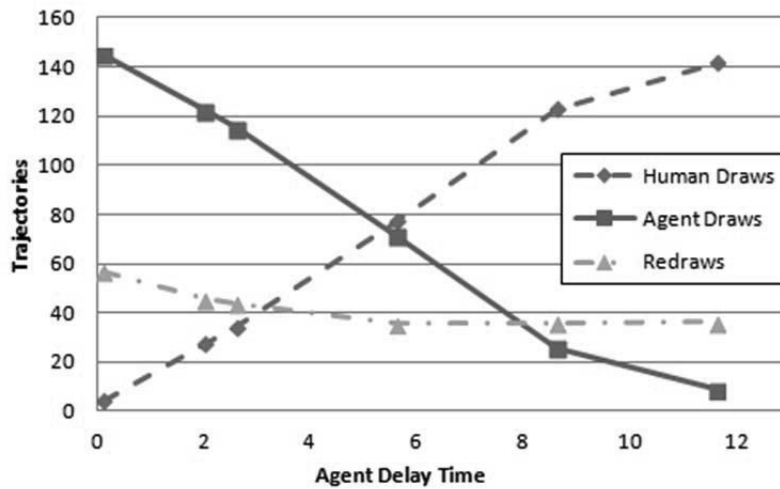


Figure 12: Graph displaying mean human draws, agent draws, and redraws per agent delay time

Conclusions and Future Work

According to the simulations, timing of the interaction between the human and automated agent significantly affects system performance, human workload, and

behavior. As a result, the agent's task time can be determined to support system objectives. For example, if the only objective is to obtain the highest score possible, it seems appropriate to place the agent trigger at 0.1 s to obtain the best possible team score. However, if there is an added objective, such as keeping the user engaged in drawing a portion of the initial routes, the approach should vary. For the operator to respond correctly to any error that might occur, they need to detect and understand the context of the error. By having the agent trigger too quickly, the human is likely to learn to redraw paths without drawing initial paths. Under conditions of low task load, as might occur as the spawn rate of the ships is reduced, the user may fall into performing a vigilance task and potentially lose the ability to maintain situation awareness. Therefore, while it may be optimal to have a quick trigger to earn higher points, this same trigger could be detrimental if the human is unable to maintain alertness and therefore be unable to detect agent errors. The purpose of keeping an operator "in the loop" is to ensure they are capable of making appropriate decisions when tasked accordingly. Keeping the operator "in the loop" appears to correlate with the timing of human-agent interaction.

Future studies could investigate how an individual's tendency to trust an automated agent affects performance, workload, and behavior. This can be evaluated by adjusting the reliance function to represent varying levels of trust. Furthermore, the results from this model suggest that human-subjects experiments should be performed to validate the behavior predicted in this research. If those experiments affirm this research, it could provide insight into the significance of timing when human and agents work together on the same task. Finally, one might expect that agent delay time is not only

dependent upon the human's response time but also upon the taskload modulated as a function of ship spawn rate. Understanding interactions between these variables may provide an understanding of human-machine teaming based upon agent timing.

IV. Timing and its Effects on Human Agent Teaming

Abstract

The research and development of automation has led to the creation of systems that can vary their level of automation, which is known as adaptive automation. As technology becomes more sophisticated, the use of autonomous agents increases within the context of a human-agent team. Teaming allows for the dynamic shift in roles between the human and agent, whereas previous technologies contained a static relationship between the two. A previous simulation model was used to investigate how the timing of agent actions impacts team performance, as well as human workload and behavior. A human-in-the-loop experiment is performed, and results are compared to the model. The agent delay time has a significant effect upon team behavior, performance, and roles assumed by the human and agent. Therefore, it is postulated that understanding the consequences of agent timing is significant in the context of human agent teaming.

Introduction

Adaptive Automation

As technology's sophistication continues to exponentially increase, automated systems will continue to infiltrate and influence daily human operations. Automated systems and human operators bring unique qualities, abilities, strengths, and weaknesses to any working environment. Automated systems can successfully execute monitoring tasks, as well as generate, select, and implement alternatives (Endsley, and Kaber, 1999), whereas, humans have been documented as being poor monitors, (Parasuraman and

Manzey, 2010) yet are not limited to adhere to a strict underlying code, enabling them to be flexible decision makers in response to unusual or unforeseen circumstances. Understanding these differing abilities may result in a clear direction to employ either an automation or human to accomplish simple tasks. However, as task complexity increases, task assignment to human or automation may not be as clear. Rouse originally proposed that it is “reasonable to expect humans and computers to have overlapping or intersecting abilities and responsibilities” (Rouse, 1977). Thus, some tasks may result in suitable outcomes, regardless of the actor performing the action.

Overlapping abilities and responsibilities can be arranged through the development of an automated system containing several levels of automation, signifying that a task does not need to be addressed either fully manual or fully automated. Rather, these are two extremes on a spectrum including variations in automation’s purpose and interactions with the human operator (Parasuraman et al., 2000; Parasuraman, Bahri, Deaton, Morrison, & Barnes, 1992). For instance, an intermediate level of automation could be a system that provides the human with a set of alternatives and allows the human to select a decision. In this circumstance, the automation only completes part of the task, generating alternatives, and proceeds to relinquish control of the next phase to the human. Although an automated system may contain varying levels of automation, it is still a static system conducting behavior changes through user input or system design. Consequently, automated systems do not attempt to understand when the human would benefit from assistance and therefore, do not assist the human at an optimal level in naturalistic environments where human task demands and capabilities are constantly

changing. The need for the automation to respond to human task demands and capabilities has led to research and increasing discussion of and implementation of adaptive automation.

Adaptive automation generally refers to the technological component of joint human and automation systems where the automation's level of control and behavior adjusts in response to real-time and context specific information (Feigh et al., 2012; Sheridan, 2011). The change in behavior attempts to respond to situational demands to meet user needs often without explicit instructions. Adaptive automation reacts to perceived circumstances by tracking and sensing information about the operator, tasks, and environment (Feigh et al., 2012). The adaptive approach aims to achieve optimal system performance through the dynamic regulation of automation, thus maintaining automation's benefits while reducing costs (Feigh et al., 2012). Parasuraman provided an example of adaptive automation within the context of an air defense system that alerted the user with a specific automated sequence if critical events occurred. In this setting, the automation is considered adaptive because it is scanning the environment and is invoked when the critical events occur; otherwise, the automation does not intervene (Parasuraman, Sheridan, & Wickens, 2008).

Dynamic Task Allocation and Triggers

Dynamic task allocation is employed in system design to actively assign and reassign tasks between the human and automation (Feigh et al., 2012). Dynamic task allocation creates an environment where the automation performs tasks contextually and the distribution of task responsibility, between the human and automation is concurrently

dependent upon real time circumstances. As a result, the task environment can be restructured to designate responsibilities (Byrne & Parasuraman, 1996).

This class of dynamic task allocation requires an administrative agent that has the authority to dictate the level of control (Sheridan, 2011). Allocation authority can either be allocated to the human operator, automation, or other mechanism.. In circumstances where the allocation authority is allocated to the automation, decisions are made in response to a trigger. A trigger is a criteria, state, threshold or event that causes the allocation authority to implement relevant adaptive automation behavior. Triggers are designed relative to information that can be sensed, observed, or modeled by the adaptive automation to establish an understanding of the current context. An adaptive automation system in which changes in the level of control is allocated to the machine uses triggers to recognize when, and how long, to engage and disengage certain adaptation behaviors (Feigh et al., 2012).

Triggers are generally classified as being based on operator, system, environment, task or mission, or spatiotemporal metrics (Feigh et al., 2012). Triggers are designed to recognize important information and implement change according to critical events, operator performance, physiological data, cognitive and task models, or other feedback. The trigger implemented in this research does not assess a singular condition, but tracks the state of multiple criteria. Nonetheless, the metric used as the independent variable in this study is time. Time is a simple mechanism to manage the engagement and disengagement of automation. In previous literature, researchers have found the benefits and costs of short-cycle versus long-cycle adaptive automation (Hilburn, Molloy, Wong,

& Parasuraman, 1993). Automation that was engaged for short periods of time, or short-cycle, only led to performance enhancement. Automation that was extended for a long period of time, or long-cycle, placed increased demand on the human operator to monitor for potential automation problems because humans are not well suited for extended monitoring tasks, as they are prone to lose vigilance (Parasuraman, 2008). It has been observed in this research that time triggers have limited applicability in effectively using adaptive automation; however, the research conducted by Hilburn was based solely upon triggers that alternated between fully manual and fully automatic control for predetermined lengths of time (Feigh, Dorneich, & Hayes, 2012; Hilburn et al., 1993). The literature doesn't contain depth as to the impact of automation's timing within a highly dynamic task environment and circumstance where human and automation interact and engage in a teaming construct.

The lack of understanding timing's impact within a human-automation team environment is significant because as technology increases in sophistication, the future direction for automation's application appears to be in the form of Human-Machine Teaming in which the machine will support the human in real time. Teaming is different from current adaptive automation considering that throughout dynamic task allocation, the automation remains subordinate to the human, thereby limiting the potential to which the team can leverage each member's unique strengths. Comparatively, effective human teams not only dynamically allocate tasks, but also roles, responsibility, and authority dependent upon each team members' capabilities (Bruemmer et al., 2002). In classic systems, the machine usually fulfills the role of tool or subordinate, never reaching the

status of a peer or leader. By allowing the machine to attain higher status, an emphasis on interdependence and communication emerges as each becomes more reliant upon the other. Timing may be a considerable factor within this team context, as humans and automated systems respond to events at different rates. Humans can require several seconds to perceive an event occurrence, process a course of action, and implement the desired action. An automation embedded in a computing system could potentially perform the same sequence of events almost instantaneously. The dichotomy in process time is a significant aspect when considering team member capabilities with respect to team performance, as many operations are time sensitive. Given the differing abilities of the human compared to the automation, it is assumed that the time in which automation executes an action has a significant impact upon the team. There is uncertainty how variation in automation timing affects human behavior and the team as a whole. Therefore, this paper aims to provide some insight to the impact of automation timing on a human-machine team performing within a dynamic environment.

Hypothesis

This research investigated a scenario where an automation and human interacted within the same environment to examine the effects of automation's task timing upon their relationship. A human operator and automated agent were placed in an environment where they were completing some of the same tasks alongside one another with a particular team objective. To fulfill the team objective, we recognized that optimization of the agent must consider multiple objectives. The agent was designed to complete tasks, keep the human involved in the task (to help overcome the agent's failings), and maintain

an acceptable level of human workload. We assumed that variation in the timing of the automation's task initialization would have significant effects upon team performance, human behavior, and workload.

To address this research objective, we employed an instrumented task environment which included an automated agent capable of performing the human's primary task. This environment permitted control of event generation to which the human must respond from rates which were clearly manageable by a human operator to rates that clearly exceeded the human's ability to respond, creating highly dynamic environments. Human performance modeling was applied based upon existing data to predict human behavior within the environment and provide a deeper insight into the reasons for this behavioral change (Goodman et al., 2016). A human-subjects experiment was conducted to assess model performance and to further understand human behavior within the target environment.

Space Navigator IMPRINT Model

To examine timing in the context of a human-machine team, this study uses the Improved Performance Research Integrated Tool (IMRINT), a discrete event simulation environment ("Improved Performance Research Integration (IMPRINT) Tool," 2010). This environment models human workload and performance as a function of time by tracking activities performed by one or more humans or machines. These activities are described in a task network, which captures the task sequencing and decision points. The frequency of the tasks, as well as the time necessary to perform each task result from a stochastic process, permitting the modeler to represent the variability within the system.

Different task networks can be derived for different goals and a workload level is assigned to each task performed by the human operator. Various system allocations can then be modeled by allocating specific tasks to the human operator or machine (hardware or software) component. However, to employ this tool to accomplish this goal, the modeler must begin with activities to be performed by the system, allocate these activities to the human or machine and then derive the tasks or actions necessary to perform these functions. Once these activities are allocated to a component, human or machine, other inherent tasks may become necessary to facilitate communication of system state as control is passed between the human and machine (Bindewald et al., 2014; Goodman et al., 2016).

Environment

The environment used for this experiment was a route creation, tablet-based game, called Space Navigator. Space Navigator was constructed to provide a controlled representation of a highly-dynamic, event-driven environment. In these environments, the operator has little, if any, control of the event rate and there is no guarantee that the human will be capable of responding should unexpectedly high event rates occur. Similar environments might include air defense systems and certain command and control environments. The game, while not providing a high fidelity simulation of these environments, permits the control of the event rate and other potentially confounding variables, logging of human response, and the creation of automations that can be enabled to assist the operator during high event rate conditions. Additionally, the environment includes a single, clearly defined, top level goal, (i.e., score the most points),

as opposed to most games which provide multiple, often conflicting goals (e.g., leveling up and score). The use of the relatively intuitive game environment simplifies participant recruitment and training.

Figure 13 displays a screen capture from the game and identifies several key objects within the game. Spaceships appear at set intervals from the screen edges. The player directs each spaceship to its destination planet, designated through color, by drawing a line on the game screen using his or her finger. The spaceship then follows the entire drawn trajectory unless the player draws a different route for the ship. Points accumulate when a ship encounters its destination planet or one of a number of small bonuses that randomly appear throughout the play area. Points decrement when spaceships collide, and each spaceship involved in the collision is lost. Points are also lost when a spaceship traverses one of several “no-fly zones” (NFZs) that move throughout the play area to random locations at a set time intervals. For every second a spaceship traverses a NFZ, the player loses points. The game ends after five minutes.

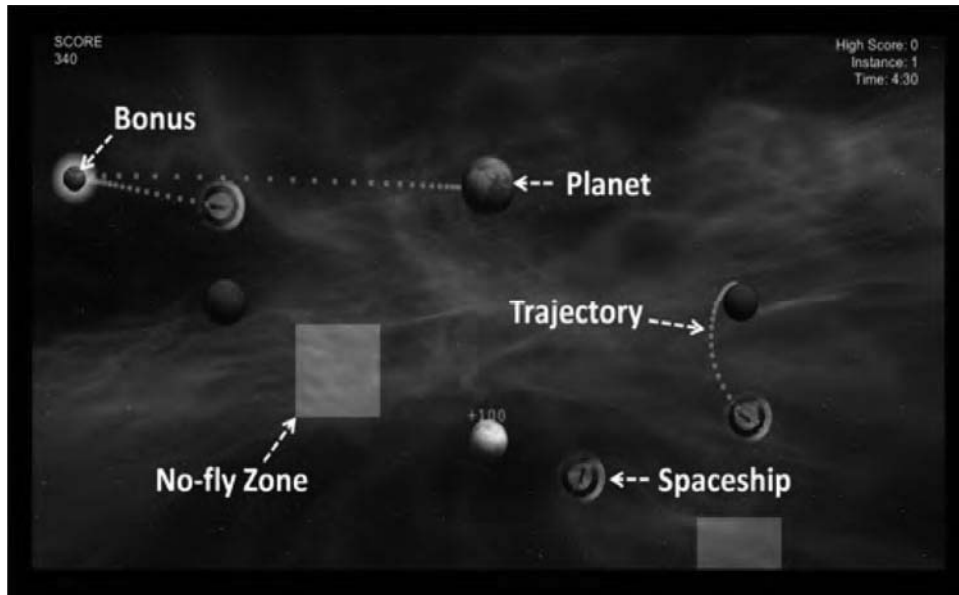


Figure 13: Screen capture from Space Navigator, highlighting spaceships, planets, trajectories, bonuses and no-fly zones.

The automated agent presented in this experiment draws straight-line routes from ships to their corresponding planet, ignoring the presence of bonuses or no-fly zones. The trigger used for the agent considers the arrival of a new ship, human response to this event, and human inactivity. The automated agent only draws a route for a ship if the human operator had not given the ship an initial route after a specified period of time. This design was presumably aiding the user if the user was unable to respond to the environmentally generated event in a timely fashion. Thus, if the user was highly task saturated or unable to complete ship routes for other reasons, the agent would help the user by drawing routes for the ships left unattended. The agent used the time that a ship was on the display without being assigned a route by the human operator to trigger its

action. The time before agent activity was the independent variable of interest in the current study, therefore, its value varied throughout the experiment.

IMPRINT Task Network

The IMPRINT model is depicted in the SysML Activity Diagram (Delligatti, 2013) shown in Figure 14. This diagram divides the activities among three primary sections (i.e., “swim lanes”). Each section represents the activities of the environment, the human operator, or the agent. The environment nodes in Space Navigator are initially responsible for starting the model, generating ships, altering no-fly zone and bonus locations, operating the timer, and halting the model, as shown in the center swim lane of the activity diagram.

The player’s attention and actions during game play are facilitated through a loop, continuously repeating two high level functions; determining which ship to select and drawing a trajectory for a ship. However, the loop is completed both for ships that have no drawn trajectories and for those that have non-desirable trajectories. A possible player strategy would be for the person to work to their capacity as they try to earn the highest score, leveraging the agent to draw paths they do not have time to draw. This behavior is depicted through the path in the Human swim lane of Figure 14, which includes identifying background items, identifying ships without routes without waiting for the agent, selecting a ship, and drawing a route. However, the human could decide to permit the agent to draw some initial paths freeing capacity to attend to other tasks within the game. Thus, a task load node, indicated by the first decision node in the human swim lane, is used to simulate a human’s decision to either initiate ship selection or monitor the

environment, allowing the human to observe the agent as it creates routes. The decision to monitor is based on a reliance algorithm derived from previously-collected experimental data (Goodman et al., 2016), as shown in Figure 15.

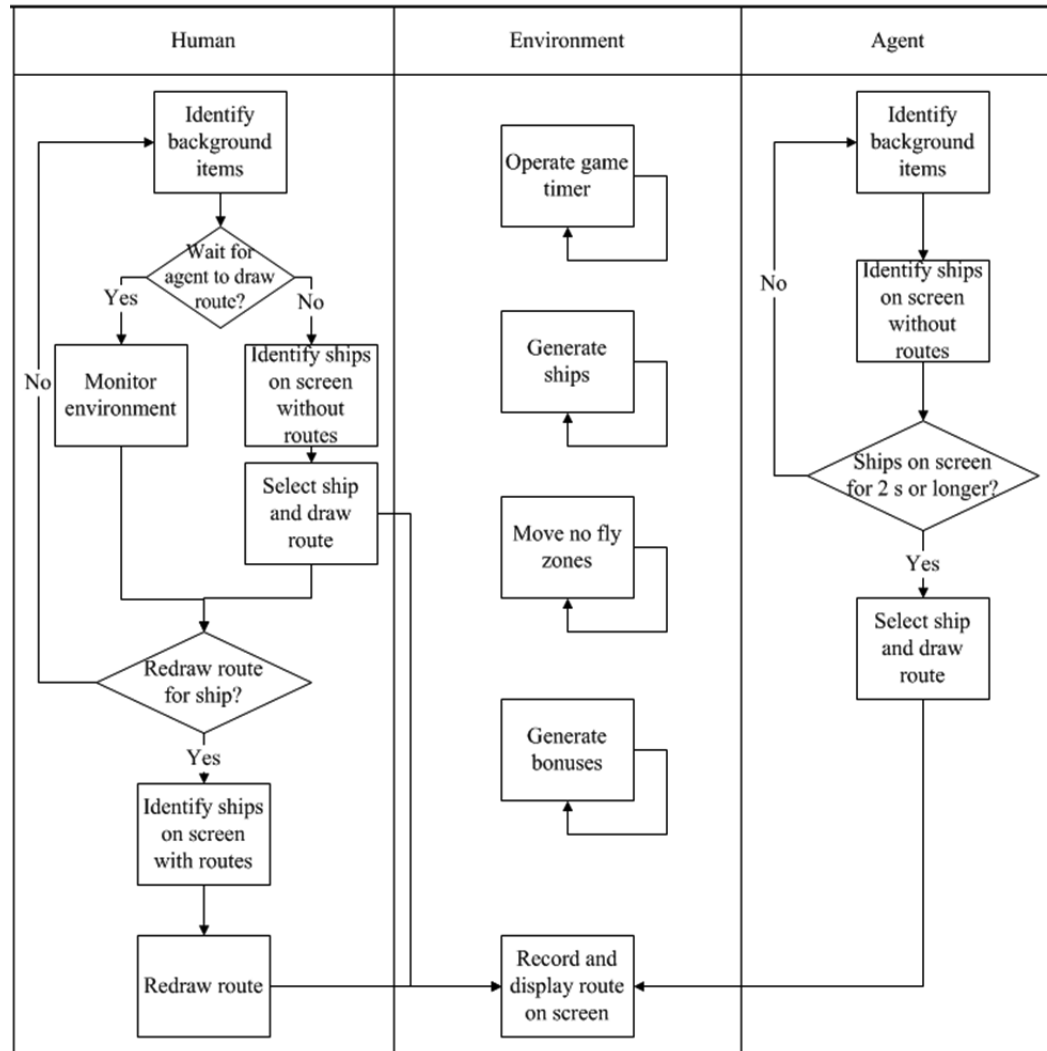


Figure 14: Activity diagram representing the actors and actions in the IMPRINT model. Vertical swimlanes are used to designate actions performed by the specified actor.

If the operator decides to draw a route based upon the reliance algorithm, they will continue to identify ships on screen. Afterwards, they can draw a route for a ship that does not have a route, or they can “redraw” a route for a ship that has an existing route. Following the draw route node, the human attention loops back to determine the human’s task load (modeled as the number of ships on screen). As shown, when the human draws a route, the environment is updated, informing the agent not to draw a route for that ship and displaying the route for the human.

Simultaneously, the agent is selecting ships and drawing trajectories, as indicated in the Agent swim lane of Figure 14. Unlike the human, the agent does not have the option to perform fewer tasks. The agent is constantly monitoring all ships on screen and drawing a route once the time trigger has occurred. Unlike the human, the agent can only draw trajectories for ships that have not received a route. As the agent draws a path, the environment is updated.

After the human or agent has designated a route for a ship, a new entity is created in the model. The entity represents a ship with a route, with human and agent-generated routes differentiated by color. The model assumes a ship continues along its path for a length of time drawn from a time distribution representing time-on-screen and is removed from the simulation after the time elapses (not depicted in Figure 14). There are three possible end results for a ship: collision, destination reached, and off-screen traversal. Ships arrive at these nodes according to probabilities associated with the number of ships on screen and the operator that drew the route. Once again these distributions were collected from human-in-the-loop experimental data discussed earlier.

The reliance algorithm produced a probability that the human would permit the agent to draw a trajectory. Analysis of data from an earlier experiment (Goodman et al., 2016) indicated that the probability of the agent drawing an initial route as a function of the number of ships on screen produced a parabolic curve. The participants performed more route draws when the number of ships on screen was low as participants likely had ample time to interact with the system. Participants also appear to have drawn more routes when larger numbers of ships were on screen to try to avoid collisions, given the agents' inability to react to neighboring ships, no-fly zones, and bonuses. The regression curve in Figure 15 was used to account for the reliance of the operator on the automated agent with respect to the number of ships on screen.

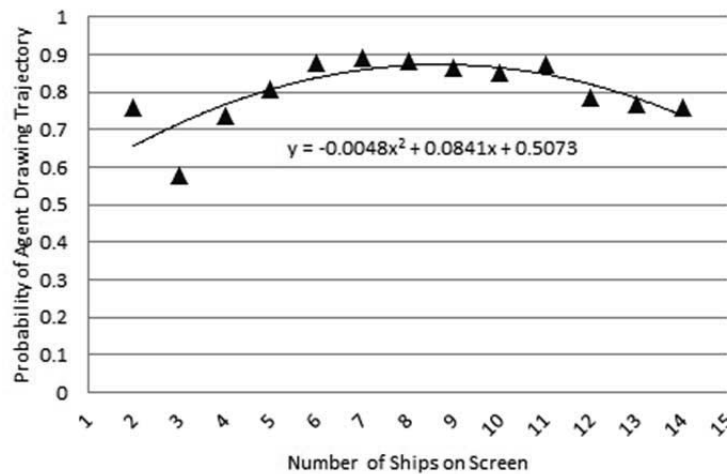


Figure 15: Graph displaying the probability of the agent drawing a route as a function of the number of ships on the screen. The third order regression line, with equation, was used in calculating the reliance algorithm in the IMPRINT model.

The other factor necessary in the reliance algorithm was the trigger time of the agent. While no data exists to construct this function, it was assumed that the longer the

agent takes before assigning a route, the more likely the human will initiate tasks to avoid losing points. At the lower limit, if the agent drew the line as soon as the ship appeared, the person would never have time to initiate a route. However, in the case that the agent requires an infinite amount of time before drawing the route, the human cannot rely upon the agent to draw any route. To understand an agent delay time that would initiate routes for ships about the same time as a human, the operators' average ship selection cycle times were calculated using data from fully manual gameplay in previous experiments. The cycle time was considered the time between initiating routes for separate ships, and was determined to be 2.6s with a standard deviation of 3s. Using three standard deviations above and one standard deviation below the mean of 2.6 s, it was assumed that a human would be unlikely to initiate a route for a ship at 0.1 s and the agent would be unlikely to initiate a route at 11.6 s. This assumption was used to determine points on a linear equation relating delay time to probability of agent draws. This linear model was used to shift the third order regression line shown in Figure 15 downwards as the agent's time delay increased and shift the regression line upwards as the agent's time delay decreased. For every second that the agent's delay changed, the baseline probability value of agent draws was incremented or decremented by 0.1058, within the bounds that the probability must be between 0 and 1. This model was validated against previous gameplay data as described elsewhere (Goodman et al., 2016).

Simulation Procedure

A series of simulations were conducted altering the trigger time of the automated agent in each simulation. Trigger times were selected based upon participant

performance. As noted earlier, the participant required an average of 2.6 seconds between the time a ship appears on screen and the time the human selects the ship to draw a trajectory. The associated standard deviation was 3.0 seconds. Six conditions were evaluated for the agent delay time: the mean time for a participant to select a ship (2.6 s), plus one-half, one, two and three standard deviations (ie., 0.1, 5.6, 8.6 and 11.6 s), as well as the 2 s delay from the earlier human-in-the-loop experiment (Goodman et al., 2016). The six scenarios were each simulated 100 times, having the same random seed for each condition. At the end of each scenario, the average scores, workload, and trajectories drawn were calculated. A one-way Analysis of Variance (ANOVA) was used to determine whether agent delay-time had a significant effect on model outputs and Tukey Pairwise Comparisons were used to test for differences between individual means.

Simulation Results

The results from the IMPRINT simulations displayed an inverse relationship between performance and workload (shown in Figure 16), as well as human draws and agent draws (shown in Figure 17). As the trigger time increased beyond the average time of 2.6 s, the operator's workload increased and overall performance decreased. Although overall score was recorded for each of 100 model runs, workload, calculated using VACP, is shown for a typical single model run.

The ANOVA indicated that the effect of agent trigger time on overall score is statistically significant ($F(5,594) = 43.28$; $p < 0.001$). Tukey pair-wise comparisons indicated that there were four significantly different groups of scores. These groups of agent time delays were (0.1, 2.0), (2.0, 2.6), (2.6, 5.6) and (8.6, 11.6). It was shown in

these pairings that as the time delay increased the average score significantly decreased. However, neighboring pairs of values were not statistically different from one another.

As shown in Figure 17, the human and agent draws also showed an inverse relationship. The agent's trigger time had a significant effect on agent draws ($F(5,594) = 35784$; $p < 0.001$), human draws ($F(5,594) = 31975$; $p < 0.001$), and redraws ($F(5,594) = 174$; $p < 0.001$). The agent and human draws ANOVA produced similar results where all times were significantly different from one another. The effect of time on redraws generated three different groupings, with 0.1 s producing the most redraws, 2.0 and 2.6 s conditions producing fewer redraws and 5.6, 8.6, 11.6 s conditions producing the fewest redraws.

Through these simulations it is observed that human behavior is expected to change as a function of the agent's trigger times. When the agent created routes at approximately the same speed or faster than the human, the human initiated routes between 2% and 20% of the time. When the trigger time is one to three standard deviations slower, the number of human initiated routes increased from 50% to 95% of all routes drawn. Furthermore, the model anticipated that the largest shift in performance would occur when the trigger time was adjusted from 5.6 to 8.6 s, decreasing the score by 10%. The greatest gains in workload were predicted to occur as the agent's delay increased from 2.6 to 5.6 s and 5.6 to 8.6 s, with a 7% and 5% increase, respectively.

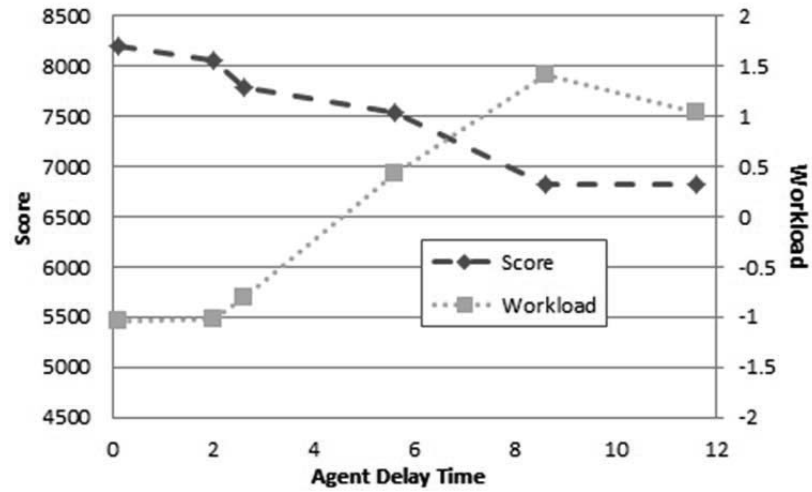


Figure 16: Graph displaying model predictions of mean score and workload as a function of agent delay time.

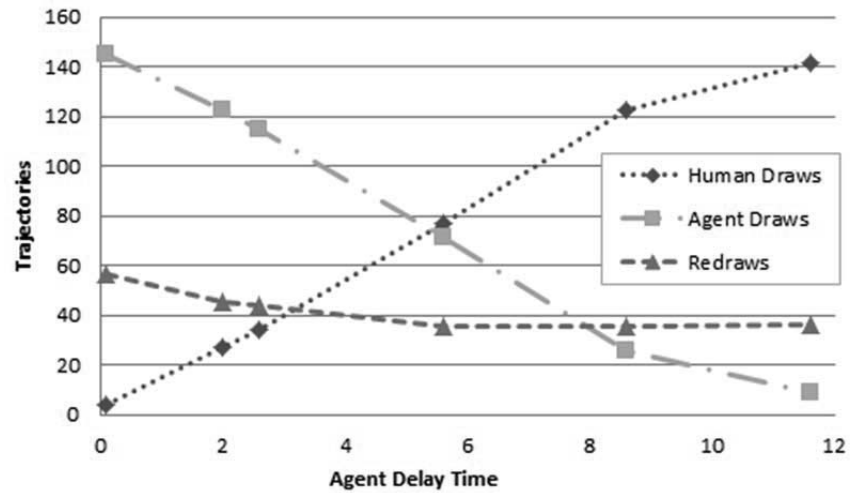


Figure 17: Graph displaying model predictions of mean human draws, agent draws, and redraws as a function of agent delay time.

Experimental Method

A human-subjects study was designed to investigate the model's findings of significant changes in team performance, human workload, and human behavior as a function of agent delay times between 0.1 and 11.6s.

Participants

The experiment involved 4 female and 16 male volunteers with an average age of 26.5 years, range of 21 to 38 years. The participants reported average use of tablets between 1-2 times a week and gaming consoles 1-3 times a month. Other computer based platforms, including smart phones and laptops were reported being used 3-7 times a week.

Experimental Design and Procedure

The experiment included a within subjects design in which each participant completed two phases of training and an experimental phase including 15 five-minute trials of Space Navigator. The first phase of training consisted of two or more fully manual (no automated agent) trials. The first phase training was terminated when participants developed a consistent strategy and performance, which was assumed to be after two or more completed trials. Participants were given the opportunity to play the game fully manually as many times as they wished before starting the second phase of training. The second phase was used to familiarize the participant with the automated agent. This phase contained two five-minute trials where the agent delay time of the first trial was 2s and the delay time for the second trial was 6s. Following the second phase of training, participants completed five experimental blocks, each containing three, five-

minute trials of one designated delay time throughout the entire block. The delay times per block were assigned to participants through a Graeco-Latin Square Design. The five delay times evaluated in the experiment were 0.1s, 2.6s, 5.6s, 8.6s, and 11.6s. Ships were spawned on screen at a fixed rate of one ship appearing every two seconds. Bonuses and no-fly zones repopulated every thirty seconds.

Participants completed the experiment in the confines of a laboratory and were permitted breaks between blocks. Player data collection used a set of Microsoft Surface Pro 3 tablet computers running the Windows 8 operating system. Workload information was collected through NASA-TLX (Hart and Staveland, 1988) and Instantaneous Self-Assessment (ISA) (Tattersall and Foord, 1996) questionnaires. Following each trial, participants indicated their ISA rating, and after each experimental block, participants indicated their NASA TLX workload values.

Experiment Results

The results from the experiment produced an inverse relationship between score and NASA-TLX, as displayed in Figure 18 NASA-TLX values were calculated by averaging participant scores across the individual subscales and standardizing the resulting values according to z score. The average scores over the delay times 0.1s to 8.6s have a negative linear relationship with a slope of -247 points per second and R^2 value of 0.99. After the 8.6s delay time, the average score plateaus. The NASA-TLX value also has a positive linear relationship over delay times 0.1s to 5.6s.

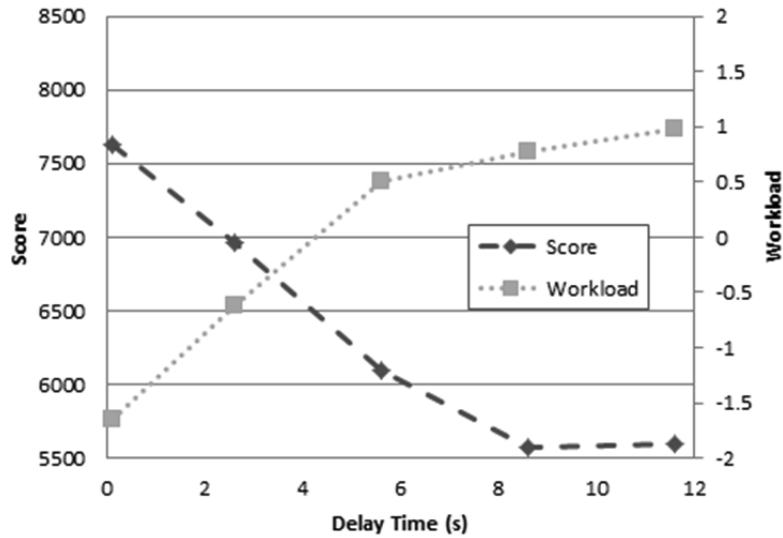


Figure 18: Graph displaying experimentally derived mean standardized NASA TLX workload values, as well as and score as a function of agent delay time.

The ANOVA indicated that the effect of agent trigger time on overall score is statistically significant ($F(4,193) = 19.36$; $p < 0.001$). Tukey pair-wise comparisons indicated that there were two groups of scores that were significantly different from one another. These groups in terms of agent time delay were (0.1, 2.6) and (5.6, 8.6, 11.6). It was shown in these groupings that there is a significant decrease in score when the delay time is greater than 2.6 s.

As seen in Figure 19 as the delay time increases, there is a shift in human behavior as indicated by the increasing number of trajectories initiated by the operator and decreasing redraws of agent trajectories. Redraws performed by the human were categorized into two types of redraws: human redraws, where the participant drew a route for a ship they had already drew a route, and agent redraws, where the participant drew a new route for a ship with a route drawn by the agent. The data shows a relatively steady

decrease in agent redraws, whereas the human redraws stay relatively the same as the delay time increases.

Although the human never drew initial routes for ships at the 0.1s delay time, a sequence could occur where the agent drew the first route for a ship followed by the human drawing a second and third route for the same ship. This sequence would be counted as 1 Agent Draw, 1 Agent Redraw, and 1 Human redraw since the agent drew the initial route, the human redraw over an agent's route, and the human redraw over their own route.

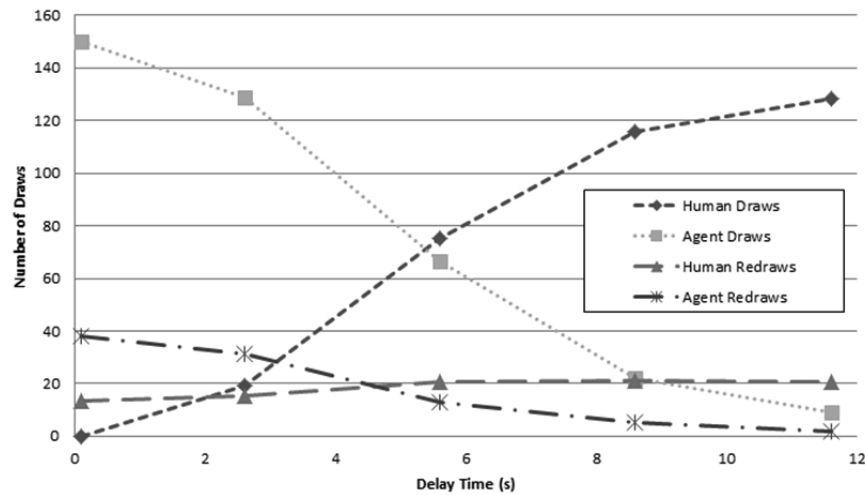


Figure 19: Graph displaying experimentally derived mean human draws, agent draws, human redraws, and agent redraws as a function of agent delay time.

The ANOVA indicated that the trigger time had a statistically significant effect on agent draws ($F(4,193) = 254.66$; $p < 0.001$), human draws ($F(4,193) = 209.54$; $p < 0.001$), human redraws ($F(4,193) = 10.20$; $p < 0.001$), and agent redraws ($F(4,193) =$

30.77 ; $p < 0.001$). Tukey pair-wise comparisons conducted for human draws, agent draws, and agent redraws led to five groupings per response according to the delay time. Human redraws were grouped as (0.1, 2.6) and (5.6, 8.6, 11.6). Another notable factor revealed in the ANOVA was the block number. Each participant completed five blocks, each block consisting of three five-minute trials.

Altogether, there were twenty participants; therefore, there were a total of sixty trials per block, with each delay time representing twelve of those trials. The ANOVA did not indicate statistical significance for agent draws and human draws, but there was a consistent trend. The blocks completed early in the experiment, blocks one and two, had ten to fifteen more human draws than blocks three through five. This indicated that as the experiment progressed, the participants let the agent engage more routes.

Discussion

The IMPRINT simulation predicted a significant impact of agent delay time upon team performance, as well as human behavior and workload. The human experiment displays similar results and trends over the spectrum of agent delay times. The standard deviations of values from the IMPRINT simulation are small compared to standard deviations from the human subjects study. A Regression Analysis was applied to assess the ability of the simulation to produce mean values that were predictive of the experimental means across all delay times, with the assumption that model results which perfectly predict the results of the human subjects data would provide regressions having a slope and a coefficient of determination (R^2) equal to 1. The regressions indicated a very strong relationship between predictions and measurements of human and agent

draws with R^2 values of 0.99 and 0.985 and slopes of 1.049 and 0.917, respectively. Redraws and scores had slightly weaker correspondence with R^2 values of 0.747 and 0.777 and slopes of 0.89 and 0.86, respectively.

The expectation of human-performance simulations, such as IMPRINT, is to provide outputs that are general estimates within the context of application. For this research, IMPRINT represented the general effect of changing the delay time factor on performance, behavior, and workload. The regression values for agent draws and human draws indicate that the simulation is able to predict this behavior according to the agent delay time with good accuracy within the range of agent delay times explored. The redraws and score were not as accurate as human and agent draws, as the simulation consistently projected higher values for each. Nonetheless, the simulation conveyed similar results and trends as the experiment. The simulation redraws were very accurate for the 0.1s, 2.6s, and 5.6s delay times, but plateaued after that point and didn't incur a sharp decrease as observed in the experiment. The experiment redraws observed a plateau in redraws from delay times 8.6s to 11.6s. Thus, the simulation captured a similar redraw pattern over delay time, but projected this transition to occur sooner. The simulation scores were higher than the experimental means at every delay time, but the simulation anticipated a similar trend. The simulation and experiment mean scores display a steady, linear decrease as delay time increases from 0.1s to 8.6s and plateaus at 11.6s. Considering the values used to influence scores in the model were from a previous experiment, it is plausible that the participants from that experiment were higher

performing players and this difference in performance may have produced the differences in score.

The model predicted similar results to the experiment based on the primary assumption that a human, when afforded the opportunity to shed tasks, will take advantage of this opportunity as long as it is not detrimental to their performance. This assumption was expressed through the agent draw probability function, displayed in Figure 15. This function has a relationship between agent delay time and human engagement in initial routes, where the quicker delay times resulted in the human being more likely to wait for the agent to draw an initial route for a ship. In turn, as the delay time was longer, the human would initialize more routes to avoid a decline in performance. At the 0.1s delay time, the human was unable to draw initial routes for any ships, but at the 2.6s delay time it was still projected that over 80% of ships would have initial routes provided by the agent. At the longer delay times, the human filled the gap and didn't rely upon the agent to the same extent because the human operator desired to maintain adequate performance. As a whole, the human test subjects followed this principle with the 2.6s delay time recording an average of 128 agent draws, or 86% of possible initial routes.

However, this assumption wasn't uniformly observed as some participants disregarded the agent. The 2.6s delay time is a significant value because this was the average human cycle time (time between drawing a route on one ship and initializing a route for another) calculated from the prior Space Navigator experiment. If participants had approached the environment by drawing routes as quickly as possible and allowing

the agent to draw routes that the human couldn't address, then it is very possible that agent draws would be almost equal to human draws at the 2.6s delay time. In the experiment, there were a minority of participants who did not prefer to have the agent active. For these participants, the average agent and human draws at the 2.6s delay time were 74 and 75 draws, respectively. If these participants are excluded from the total average agent draws at the 2.6s delay time, this value increases from 128 to 139 agent draws, or 86% to 93% of possible initial routes. Although the minority of participants collaborated with the agent in contrast to the other participants and the general assumption of human behavior, they still achieved their two highest scores when the agent was most active at the 0.1s and 2.6s delay times. Therefore, it appears that the agent maintains utility regardless of the human's perception of the agent.

It is evident from the simulation and experiment that task initiation is a function of agent delay time, but it may also be suggested that team member "roles" are assumed according to delay time as well. At the 0.1s and 2.6s delay times, the number of agent draws and agent redraws are significantly greater than the number of human draws and human redraws, respectively. It is observed at the 5.6s delay time the human draws slightly more initial routes, while the human redraws their routes and the agent's routes roughly the same. At the 8.6s and 11.6s delay times, the human initiated routes for almost every ship, causing agent involvement to be minimal.. The change in behavior over the course of the varying delay times suggests that the agent's timing significantly affects the assumption of team member roles. At the early delay times, 0.1s and 2.6s, the human adopted a "supervisory" role, allowing the agent to draw a substantial number of routes

while intervening to attain higher net points. The human operator would oftentimes redraw the agent routes to gain higher point values by avoiding collisions, no-fly zones, and collecting bonuses. As the delay time exceeds the 5.6s delay, the human does not continue to operate with the same strategy, recognizing that waiting for the agent to respond would likely lead to lower system performance. At the 5.6s delay time, the human operator behaves almost as a peer to the agent with regard to assigning initial routes where they drew approximately the same number of initial routes as the agent and performed redraws on their own routes as frequently as they redrew the agent's routes. At the longer delay times, the human continues the trend of initiating routes for more ships. At 8.6s and 11.6s, the agent is almost non-existent, in terms of its participation, as the human initiates most of the routes. At these delay times it could be assumed that the human discounts the agent as the agent is unable to provide a timely response to a new ship requiring the human to respond to the ship to maintain acceptable performance.

Conclusion

Given these findings, as human-agent teaming conceptually matures, timing needs to be thoroughly evaluated; for, it is a significant factor to the human-agent team. Although specific timing constructs between a human and agent may be task dependent, there are general principles that can be applied in the development of a human-machine team. Within the context of the team, timing affects the relationship between the human and machine, thereby influencing the behavior of each. In the experiment, there were two primary relationships, supervisor-subordinate and peer-peer, that were assumed according to the timing of the agent. The supervisor-subordinate relationship was generally

observed during the quickest and slowest agent delay times. As the agent drew routes quicker, the human generally assumed a supervisory role, intervening primarily to alter preexisting routes determined by the agent. At the slower delay times, the human took the initiative to draw most of the routes while the agent picked up ships when the human was unable to address them. The peer-peer relationship occurred when the human and agent drew routes at the same pace.

The aspect of teaming that makes it different from adaptive automation is that teams have a shared goal. In the attempt to reach this goal, it is realized that each team member possesses unique qualities, which leads to interdependence and the dynamic facilitation of roles according to team members' capabilities. Further pursuing an understanding of the role of timing within dynamic environments may prove advantageous for human-agent teams.

V. Conclusions and Recommendations

Chapter Overview

This chapter addresses the need for research in the field of human-agent interaction and teaming. The research objective is restated and the research summarized. The investigative questions are individually answered, followed by recommendations for future work and final conclusions.

Research Motivation

The Department of Defense (*The role of autonomy in DoD Systems*, 2012) and the Air Force (M.R. Endsley, 2015) understand the potential benefit of autonomous software working synergistically with military members in a vast range of operations. Autonomous systems provide an opportunity to enhance future Air Force operations by “potentially reducing unnecessary manning costs, increasing the range of operations, enhancing capabilities, providing new approaches to air power, reducing the time required for critical operations, and providing increased levels of operational reliability, persistence and resilience” (M.R. Endsley, 2015). Further, it is recognized that these systems have application across a larger number of Air Force Domains as the Autonomous Horizons document states: “Increased levels of autonomy can be brought to bear to enhance operations in both manned and unmanned aircraft, and in operations in space, cyber, command and control, intelligence, surveillance and reconnaissance, readiness, and sustainment across the Air Force” (M.R. Endsley, 2015).

However, these potential advances in military operations will only be successful if there is sufficient research and understanding in the realm of human-agent teaming. It was noted that past developments and the structure of automation created fragile systems that contained limited capabilities and consideration of the human operator (M.R. Endsley, 2015). Two issues within human autonomy teaming motivated this research: uncertainty in the effects of autonomy task timing on team dynamics and the need for effective modeling and simulation methods for autonomous system test, evaluation, verification, and validation. .

The fundamental aspect of teaming is that humans and autonomy will “interchange initiative and roles across mission phases to adapt to new events, disruptions and opportunities as situations evolve” (*The role of autonomy in DoD Systems*, 2012). This dynamic relationship between humans and automated systems has not been fully realized in current systems due to numerous challenges associated with autonomous system development. One aspect of autonomous system design that may have considerable impact on team member roles and initiative, as well as human situation awareness and workload, is the autonomy’s task timing. The timing of task execution in highly dynamic, event-driven domains is assumed to influence the performance and behavior of the team. Considering that automated systems have the potential to respond much faster than their human counterparts, it was posited that their response time can affect task responsibility.

In addition to issues in human autonomy teaming, another problem is the proper testing, evaluation, verification, and validation of the system. This issue arises as the

range of actions that could potentially be performed by autonomy may be exponentially greater than previous automation systems, which do not significantly adapt their response to environmental stimulus. As autonomy's software is adaptive and learns to respond to a large range of environmental conditions, Autonomy has several potential outputs per input it receives. Traditional methods of test and evaluation involved placing the automation into a scripted scenario and observing how it responds. Through iterative, continuous and evolutionary modeling and simulation, it may be possible to evaluate a greater range of autonomy responses and actions. In autonomous systems that adapt the human's task environment, it was posited that developers may be able to understand human autonomy interactions and the effects of system design on human behavior, workload, situation awareness and performance through the use of models which include human and automation behavior.

Research Objectives

Two research objectives were posed. The primary objective was to assess the human-agent team, particularly, how agent timing affects human behavior, team performance, and relationship dynamics within the context of the team. The effects of an automated system that executes actions exceedingly fast or slow are well understood, where a human will almost never initiate tasks when automation acts too quickly, or a human will always initiate tasks when the automation responds too late. As autonomous systems' behaviors and roles begin to evolve from simple assistance tools to fully capable team members, there are many unknowns as to what effects autonomy task timing has on the human operator and team environment.

The secondary objective was to understand the considerations and requirements needed to properly model the interaction between the human and autonomy. This objective was fulfilled through the use of models and simulations. In this thesis, MBSE was applied through the use of SysML activity diagrams to discover significant considerations and assumptions that are needed when modeling a new human-agent team. This capability was applied in conjunction with discrete event simulation to create projections of human behavior and team performance in response to changes in the environment.

Investigative Questions

Responding to the research objectives, the following investigative questions were addressed.

1) What are the considerations needed when modeling a process that involves human-agent interaction?

The main considerations in modeling the incorporation of an agent are the changes in human behavior and communication between the human and agent. It should not be assumed that the human will continue to exhibit the same behaviors when the agent is introduced. It may seem logical that the human will continue the same course of action (COA) regardless of agent involvement or will simply relinquish a portion of their task to the agent, but actually the agent causes the human to consider several different COAs depending upon the context of the situation. It is important to model the entire spectrum of COAs and not be complacent with assumptions of how the human and agent will interact.

Within the model developed in this research, initial allocation of actions appeared simple and intuitive, only requiring designating responsibility for existing actions to either the human or the agent. The early model was adapted to include task re-allocation, which requires careful consideration in the areas of human-agent communication and adjustments in behavior. It is significant for the developer to understand that task flow between a human and agent involves some type of input or output from both. Any adjustments in human behavior, arising as a result of automation, needs to be addressed and input into an adapted model. Revision of human behavior nodes and the inclusion of human-computer communication leads to a model that more accurately represents the system. As a specific example, each of the models discussed in this thesis applied the concept of reliance, modeling this concept as a decision to be made by the human to either rely upon the automated aid or not. This reliance decision is influenced by a number of factors, including agent delay time, number of ships on screen, and number of ships without routes. Within the models shown throughout this thesis, many of these factors are shown to influence the behavior of the human within the human-machine team.

In the development of a new system, the accuracy of a model, or set of models, is critical to the further development of the system. Models and simulations are often made in the conceptual phase of system development, capturing the fundamental elements of projected system attributes and behavior in a cost-efficient manner to project the impact of design decisions upon later system performance. Conceptual modeling is the cornerstone for Model Based Systems Engineering (MBSE), affecting nearly every

aspect of the development and implementation of the system. If the model neglects certain aspects of the system, this could have a negative impact on the project's budget, schedule, requirements, functionality, and feasibility. Therefore, in the context of modeling human-computer interaction, one needs to apply careful consideration regarding communication, situation awareness, and behavior to properly capture system behavior and avoid undesired costs.

2) How can modeling and simulation tools be used to infer optimal agent timing that simultaneously improves operator performance and reduces workload?

The IMPRINT simulation was designed to assess the effects of agent timing on human behavior, workload, and team performance. Experiments were conducted prior to the development of the simulation, which was beneficial as they provided data for the IMPRINT simulation and the data from this experiment was useful to validate or reject the SysML model, at least under the range of conditions included in the experiment.

In the conceptual phase of model development, it was expected that the participants would continue drawing routes, relying on the agent to draw routes only when they were overloaded to the point that they could not draw routes quickly enough to be successful. Prior to conducting the baseline experiments, the rationale behind this assumption was that the agent would be able to work alongside the user, but work less effectively and therefore not be trusted to draw routes unless the individual was task saturated to the point that they could not draw routes quickly enough. Based on this assumption, it was expected that the majority of trajectories would be drawn by the participant. It was assumed that the agent would work less effectively because it was

expected that the agent's route drawing behavior would conflict with the user's strategy, causing the human to lose trust. However, most participants' behavior differed from this reasoning. Participants oftentimes assumed supervisory roles, allowing the agent to draw more initial routes while the human implemented strategies involving collision avoidance, no-fly zone avoidance, and bonus pickup.

The human test subjects' data was incorporated into the simulation for realistic probabilities, time distributions, behaviors, and performance. Data, in conjunction with updated assumptions of human behavior, resulted in an IMPRINT simulation that projected the significant impact of agent delay time upon team performance, as well as human behavior and workload. Follow-on human experiments displayed similar results and trends over the spectrum of agent delay times.

Therefore, the IMPRINT simulation was very effective at realistically capturing human-agent interaction and the effects of agent delay time. Simulations can be used as cost-effective means to predict and report projected metrics. When using a simulation to predict human-agent behavior, using a conceptual model as a guideline is necessary. In this case, adaptation of the SysML activity diagram revealed new human behaviors and choices, as well as the need for human-agent communication. The SysML diagram permitted the creation of the simulation to remain focused on capturing all of the tasks, behaviors and subtleties of the system that were described in the previous model. However, assumptions made within the original model were incorrect. Baseline experimentation helped reveal actual implications of system behavior and performance, resulting in a more robust simulation. As a result, the model can be used to predict how

changes in the system affect human behavior, workload, and performance with relative accuracy.

Note that this process is an embodiment of the general user centered design process. Although the general user centered design process does not necessarily include the use of models during the definition process, this research illustrates that this step has significant value. Understanding the spectrum of human-agent interactions, in response to a dynamic environment, occurs in the modeling process. Modeling accurate and detailed depictions of human-agent interactions are useful in developing a robust simulation because they provide an established framework. This framework leads to simulation development that is focused on capturing events and behaviors that reflect the human-agent team as closely as possible.

3) How does the timing of an agent affect operator behavior and workload, as well as team performance and dynamics?

It is evident from the simulation and experiment that task initiation is a function of agent delay time, but it may also be suggested that team member “roles” are assumed according to delay time as well. At the 0.1s and 2.6s delay times, the number of agent draws and agent redraws are significantly greater than the number of human draws and human redraws, respectively. It is observed that at the 5.6s delay time, the human draws slightly more initial routes, while the human redraws their routes and the agent’s routes roughly the same. At the 8.6s and 11.6s delay times, the human initiated routes for almost every ship, causing agent involvement to be minimal. The change in behavior over the course of the varying delay times suggests that the agent’s timing significantly affects

team member roles. At the short delay times, 0.1s and 2.6s, the human adopted a “supervisory” role, allowing the agent to draw a substantial number of routes while intervening to attain higher net points. The human operator would oftentimes redraw the agent routes to gain higher point values by avoiding collisions, no-fly zones, and collecting bonuses. As the delay time exceeds the 5.6s delay, the human does not continue to operate with the same strategy, recognizing that waiting for the agent to respond would likely lead to lower system performance. The human operator behaved almost at a “peer” state where they almost drew the same number of initial routes and performed redraws on their own routes as much as the agent’s routes. At the longer delay times, the human continues the trend of initiating routes for more ships. At 8.6s and 11.6s, the agent is almost non-existent, in terms of its participation, as the human initiates most of the routes. At these delay times it could be assumed that the human is adopting a more “subordinate” role where they are initiating for most ships and the agent is drawing routes when necessary.

It is important to note that the experiment cannot fully replicate supervisor-subordinate and peer-peer relationships because roles are not assigned solely upon who is completing certain tasks. Other factors, such as decision authority, influence team roles and relationships. Throughout these experiments, the human was able to override or redraw agent routes, whereas the agent could not redraw any routes. Final decision authority was given to the human; therefore the roles attributed in this discussion are not pure representations. Nonetheless, the change in behavior and strategy performed by the

human at different delay times appear to indicate an adoption of different roles according to agent delay time.

This experiment was conducted at a single event rate with a ship appearing on screen once every two seconds. At this event rate, agent delay time has a significant effect upon score and human workload. As the agent delay time increases, the score decreases and human workload increases significantly. Based on these experimental results alone, it appears that it is best for the agent to have the shortest delay time possible as score is highest and workload is lowest under this condition. It is suggested, however, that low workload in conjunction with supervisory tasks is not well suited for humans as their vigilance may decline and potentially lose the ability to maintain situation awareness. Thus, one might suppose that the agent delay time should be increased to keep the human operator “in the loop”. The purpose of keeping an operator “in the loop” is to ensure they are capable of making appropriate decisions when tasked accordingly. However, it should be noted, that while the human may assume the role of supervisor, redrawing non-optimal paths for short delay times, the resulting task does not resemble a vigilance task for the current event rate. Instead, the human is highly engaged with redrawing paths, redrawing as many as 10 paths per minute with an agent delay time of 0.1 s.

Study Limitations

The biggest limitation to the research is that the human test subjects do not truly represent the population of military operators. Also, the game used in these experiments does not replicate the use of a militarized autonomous system. Space Navigator presents

the user a very simple process with a small learning curve. Thus, the main assumption of this research is that although the subjects and environment do not directly represent the types of autonomous systems that would be used by the DoD, the results will apply to the general field of human-agent interaction.

Recommendations for Action

In the context of employing human-agent teams in dynamic environments, there is a wide range of possibilities and scenarios these teams may encounter. The roles and responsibilities assumed by each team member may have significant consequences due to the unique capabilities of each team member. This research proposed that the task timing of the autonomous agent impacts the allocation of roles between the human and autonomy and ultimately affects the team's performance. It is recommended that military installations designing autonomous systems investigate the effects of autonomy task timing on their human-agent teams. Understanding how variations in timing affect the human operator and the team may prove to be beneficial in accomplishing objectives more effectively.

Recommendations for Future Research

Future Agent Timing Research

While this experiment provides insight into the significance of agent response time and its effect on teaming, it only captures one state of event rates. Future research should examine the effects of the external event rate on team behavior to mirror the dynamic pace of real-world environmental events. Admittedly, the event rate, or time

between ship arrival, of one ship every 2s requires the human to designate trajectories at a rushed pace, actually faster than could be reliably performed by the majority of the participants, as indicated by the decrease in score as the agent's assistance was delayed. When the agent's delay time was set to 0.1s, participants, on average, were touching the tablet screen roughly every 6s. At the 11.6s delay time, that time interval was 1.5s. Performing an action once every 1.5s to 6s with a constant event rate is not very applicable to common circumstances in a real work environment. Most environments have unknown and variable event rates observed over an extended period of time, unlike the experiment which provided a constant, fast-paced rate for five minutes per trial.

To better understand what may happen when the event rate is slower, IMPRINT was used to predict these effects. Event rates were chosen to be 3s, 4s, 6s and the agent was given delay times of 2.6s, 5.6s, 8.6s per event rate, resulting in 9 scenarios. The length of each game remained 5 minutes, therefore, as the event rate extended, the number of ships and possible maximum score decreased. The simulation ran 10 times per instance and recorded average human draws, agent draws, redraws, and score. Figure 20 captures human initial draws per delay time across three event rates. The y-values, human draws and score, were input as percentages, rather than total values, due to the varying number of spawned ships. The total number of spawned ships was different per event rate because the total length of the game remained at 5 minutes. As seen in the graph, human draws increased almost linearly across all arrival rates for delay times of 2.6s and 5.6s, whereas the 8.6s delay time plateaued. As the event rate increased and the human initiated more draws, the agent became less involved at an inverse rate.

Performance was affected similarly to behavior, shown in Figure 21. As the event rate became slower, performance increased per delay time. The shortest agent delay time of 2.6s was not consistently the highest scoring, as optimal agent delay time performance seems to be dependent upon event rate. At the fastest event rate of 2s, the greater the agent involvement, the higher the score. As the event rate increases, it can be assumed that shorter agent delay times would perform better. However, as the event rate is extended, overlap is observed at the 3s delay time and all delay times produce similar results. As the event rate slowed down to 4s, the 8.6s delay time is the highest scoring. At the 6s event rate, scoring amongst all delay times are relatively the same, but this may be due to the simulation's internal probabilities being based upon human performance as the 2s event rate where most human operators found it difficult to avoid collisions and prioritized collision avoidance over attaining additional points through bonus collection or avoiding no-fly zones. During the slower event rates, there are generally fewer ships on the screen at a given moment which provides the human operator with more time to implement these strategies while avoiding collisions. Therefore, the model likely underpredicts human performance at these slower event rates.

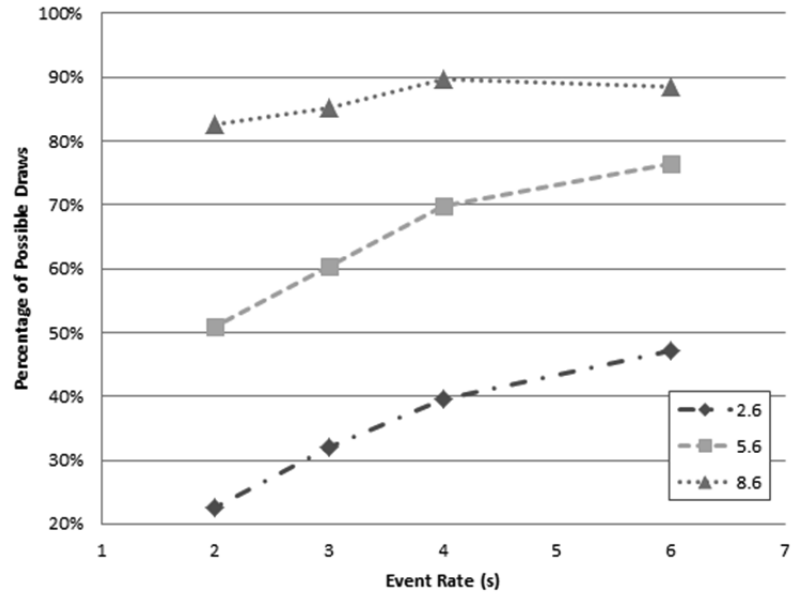


Figure 20: Graph displaying percent of human draws as a function of ship arrival rate and delay time as predicted from the IMPRINT model.

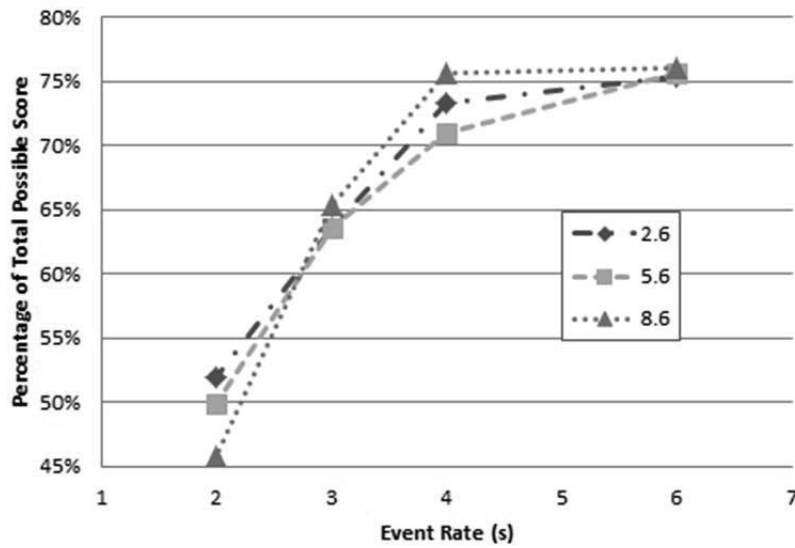


Figure 21: Graph displaying percentage of possible score as a function of ship arrival rate and delay time as predicted from the IMPRINT model.

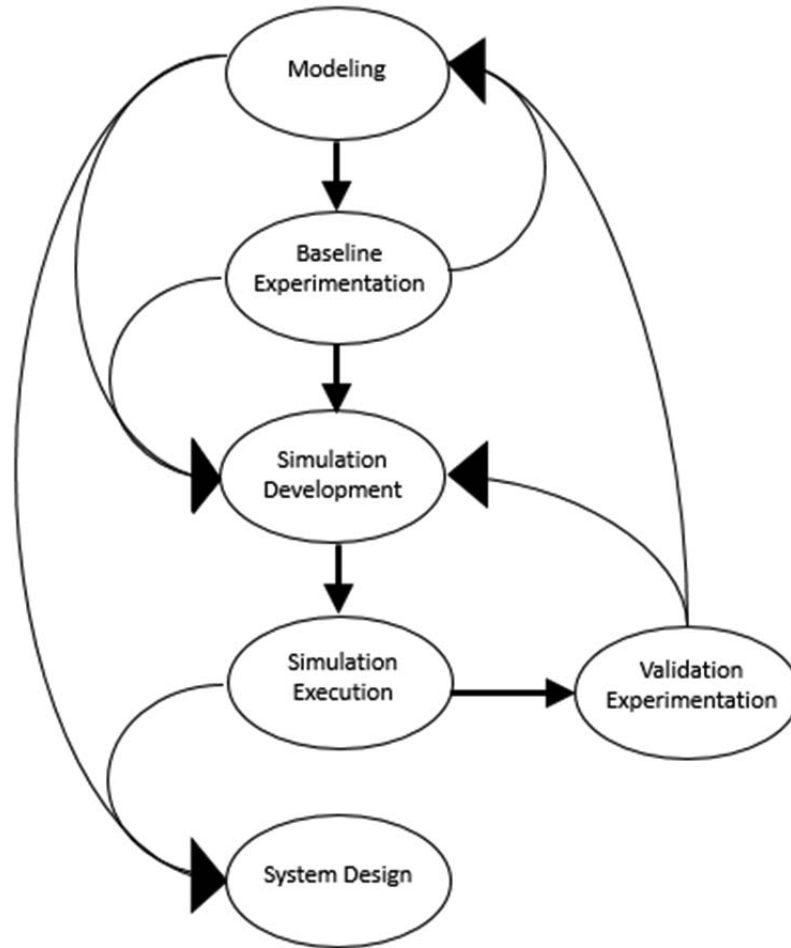
It is interesting to note that varying event rates may lead to increased or decreased agent involvement to achieve greater team performance. In this circumstance, as ships appear more rapidly and the team incurs a larger task load, heavy agent involvement leads to higher performance. At slower ship arrival rates, there may be a longer agent trigger time to maximize team performance. According to the IMPRINT simulation, this factor of event rate noticeably influences team dynamics and behavior. Therefore, agent timing may need to be a function of event rate. The role of an agent within a team may dynamically change according to the rate at which tasks need to be accomplished. For example, at slower event rates the agent could assume the role of supervisor while the human is performing a majority of the tasks, whereas when the event rate surpasses human capability, the human adopts the role of supervisor while the agent completes most of the tasks. At a moderate event rate, both could work together as peers. However, future research should be pursued through test subjects experiments to more accurately understand the effects of event rates and its implications on human-agent team performance.

Future Modeling Research

The range of actions that could potentially be performed by autonomy is extremely vast, especially when placed in an unpredictable environment. It is not feasible to perform traditional testing considering that the space of autonomous actions cannot be “exhaustively searched, examined or tested” (Clark et al., 2014). The future of autonomous system design relies upon progressive modeling and simulations to feasibly understand the potential actions by the autonomy and the consequences of those actions.

Through iterative, continuous and evolutionary modeling and simulation, it may be possible to evaluate a greater range of autonomy responses and actions. In autonomous systems that adapt the human's task environment, developers may be able to understand human autonomy interactions and the effects of system design on human behavior, workload, situation awareness and performance through the use of models which include human and automation behavior.

An approach to modeling and simulation has been proposed for future research regarding its potential to be an effective when addressing with the problems stated above. The modeling approach has four primary phases to be completed in sequential order and requires feedback to flow from the experiments to the models and simulation. Figure 22 displays four ovals as the six phases – Modeling Process, Baseline Experimentation, Simulation Development, Simulation Execution, Validation Experimentation, and System Design. The straight lines between the phases represent the order in which the phases must be completed. The curved lines represent feedback from one phase being provided to another.



a. Modeling Process

The modeling process is the initial phase in the design methodology. The purpose is to create an understanding of the environment, human, and agent through modeling behaviors, states, capabilities, assumptions, and other significant information. This produces a base network of information that is necessary to investigate how the human and agent dynamically allocate tasks, responsibilities, and roles through the methods in which they communicate and interact. Modeling is also used to identify the environmental factors that are subject to change

and affect human-agent interactions. This phase should establish the factors to include in the baseline experiment and to the factors to use as the independent variable(s) in the validation experiment.

In this thesis, the approach to the modeling phase was the development of an activity diagram to represent human decision making in response to agent delay time, events, and on-screen information. The activity diagram created a logical approach the human would take when interacting with the agent. This model helped guide the simulation development process, as well as identify assumptions to be validated in the following experiments and simulations.

b. Baseline Experimentation

The baseline experiment is a human test subjects experiment with the purpose of validating the model and providing data to the simulation. The baseline experiment provides context specific information about how the human and agent interact with one another. Interactions observed in the experiment can be used to examine how they compare with the model and recognize behaviors and assumptions that were validated and others that may need to be reassessed. The baseline experiment also provides real human performance data that can be incorporated into the simulation for probabilities, time distributions, tendencies, and other necessary information. It is important to acknowledge that this experiment is not

designed to test the effects of an independent variable, but to examine human-agent interaction and gain baseline data for the simulation.

The baseline Space Navigator experiment applied in this thesis had a static event rate and agent delay time, both equaling 2s. It provided desired data for the simulation and feedback for the model. Without the baseline experiment, the simulation would not have appropriately captured all the essential information relevant to human and agent behaviors. Notable feedback obtained was the human reliance upon the agent as a function of the number of ships on screen. Identifying this key component of human behavior was critical to the process as it provided data for the human's decision making process of either attempting to draw a route before the automation or letting the automation draw a route. It also validated an assumption in the modeling process that one of the factors that influence human reliance on an agent or automation is the taskload, which in this case was represented by the number of ships on screen. Reliance data provided a function that was used to simulate how a person would respond to the change in agent delay time, which was critical to the official simulation runs and the validation experiment.

c. Simulation Development

Simulation development consists of two steps: creation and validation. The creation step is simply the process of using the validated models to generate the desired network of task nodes. As mentioned

earlier, the models provide a crucial foundation and framework for the simulation. Data collected from the baseline experiment is input into the task nodes such as task time distributions, outcome probabilities, reliance functions, and behavior functions.

Following the completion of the model, it requires further validation by comparing the results of the simulation to the baseline experiment. Validation can be accomplished through comparing results of the model for the conditions included in the experiment through the use of comparison or statistical tests, including equivalence tests, t-tests, or other appropriate means. The values to validate should be significant to the human agent team, such as, team performance and human and agent behaviors. It is also extremely important to evaluate secondary measures in the simulation to ensure that the outcomes being validated are supported by similar underlying behaviors. For example, in the Space Navigator simulation validation, values that were used for validation were score, agent draws, human draws, and redraws. However, it was also ensured that secondary values, such as ships on screen, number of bonuses collected, time spent in no-fly zones, and destinations reached, were similar to the experiments. If the primary values are validated, but the secondary values are not representative of the experiments, then the simulation does not adequately reflect the system and is subject to report poor results during further testing.

d. Simulation Execution

Following Simulation Development, the next phase is to run official simulations to examine the effects of one or more independent variables beyond those explored in the validation experiment. The variables, number of runs, and other metrics should have been established prior to the creation step so that the independent variable(s) can be easily adjusted. Once the results are collected, they are evaluated as desired, but they do not provide feedback for the models. At this point, the simulation does not necessarily represent human behavior and interesting results need to be validated.

e. Validation Experimentation

The Validation Experimentation phase is another human test subjects experiment with the purpose of validating the simulation runs previously conducted, as well as providing further feedback to the conceptual models of the human-agent team. The same independent variable that was used in the simulation needs to be used in the validation experiments. The validation experiment will reveal how accurate the simulation projects adaptations in human-agent interactions and team performance in response to changes in the environment. Again, statistical significance testing is appropriate in this phase to validate the simulation and experiment.

Following validation testing, data and observations from the experiment should be used to assess any changes or additions to the conceptual models and further enhance the simulation's performance and stability. If the simulation is rejected by the statistical significance tests, then an iterative approach of simulation evaluation, adjustment, and runs should take place until the simulation is validated and accurately represents the system in a dynamic environment. It should be noted what values were not aligned with the experiment's results and what the underlying causes were within the construction of the simulation. It is possible that assumptions made in the development of the simulation were not valid and need to be changed in the simulation and conceptual models.

In the Space Navigator validation experiments, the simulation was validated in human draws and agent draws, however, redraws and score were not. The simulation was able to capture similar tendencies in redraws and score as a result of changing the independent variable, but it didn't predict the same magnitude of effects. As a result, it was necessary to investigate why these values were not as accurate. It was discovered that score was rejected because the participants in the baseline experiment were simply higher performers than the validation experiment's participants, causing disparity in the score results. The number of redraws in the simulation is a function of number of ships on screen and did not account for agent delay time. This function was adjusted to model how

human behavior changes with respect to agent delay time. Also, the validation experiment affirmed an assumption made in the conceptual model that a human, when afforded the opportunity to shed tasks, will take advantage of this opportunity as long as it is not detrimental to their performance. This feedback, as well as other observations, were applied to the simulation and models to enhance the simulation network and conceptual understandings of the human agent team in this context.

f. Iteration Influencing System Design

The purpose of the design tool is to develop a simulation that would adequately predict human-agent interactions, behaviors, and performance when testing other variables. After the first completed cycle of this design tool, it is possible to continue iterating through the modeling process, simulation development, simulation execution and validation experiments. However, the goal of this design tool is to create a simulation product that can reliably represent the human-agent team in a wide variety of environmental changes to save the extensive time, money and effort that is needed to conduct numerous human test subjects' experiments. It is suggested that one cycle through this design process can provide a reliable simulation, but it is also flexible enough to continue iterating if it is desired to ensure the model is producing accurate results with respect to other variables.

The emphasis of this tool is to apply modeling and simulation to inform system design. Constantly updating the representations and understanding of the system through progressive modeling influences the way the agent is constructed. As the models and simulations become more robust through iterative processes, it may be assumed that fewer human test subjects' experiments may be performed to understand human autonomy interactions and system behavior. However, this approach, as currently constructed, is only a proposal for one method to address a problem that is a challenging problem for the Air Force. Further research should be conducted to develop an approach that contributes to the problem of obtaining "effective methods to record, aggregate and reuse test and evaluation results" (Clark et al., 2014) for autonomy development.

Significance of Research

This research demonstrated a method to human-machine teaming that involved MBSE, simulation software development, and human test subjects experiments. The method was applied to the design of a human-machine teaming environment and revealed how timing is a significant factor to the human-machine team. Although specific timing constructs between a human and agent may be task dependent, there are general principles that can be applied in the development of a human-machine team. Within the context of the team, timing affects the relationship between the human and machine, thereby influencing the behavior of each. In the experiment, two primary relationships

were observed: supervisor-subordinate and peer-peer. These relationships were assumed according to the timing of the agent, as well as the human.

This observation of timing and its effects on team dynamics is significant because the aspect of teaming that makes it different from adaptive automation, or previous automation frameworks, is that teams have a shared goal. As a team attempts to accomplish their goal, it is realized that each team member possesses unique qualities, which leads to interdependence and the dynamic facilitation of roles according to team members' capabilities. Therefore, agent timing directly influences the relationship with the human operator and is an extremely important aspect to human machine teaming. This research begins to uncover how human-agent relationships respond to changes in the agent's timing and how agent timing should be dependent upon environmental event rates. As human agent teaming appears to be the future direction of the Air Force, this thesis contributes to foundational research of teaming to help develop successful new frameworks for autonomous systems.

Appendix A: IMPRINT Simulation Description

Environment Task Nodes

The **Generate Ship** task contains the ship interarrival rate in the task duration tab. In the effects tab, another ship is added through a tuple variable, ShipArray. ShipArray contains attributes for new ships, with each ship having a ship number, time in which it arrived on screen, and a Boolean variable for a route, where false is no route. This array is used to attribute routes to ships and trigger the automation. Counter variables are also used to calculate total number of ships to enter the game, number of ships on screen, and ships without routes.

The **Change No-Fly Zone Locations** and **Change Bonus Locations** task nodes perform similar tasks. Each performs their tasks iteratively during fixed time duration. The Change Bonus task node sets the number of bonuses to 3, as other task nodes have the ability to decrement that number if a ship collects a bonus. The **Operate Game Timer** node runs for the length of the game and executes the **Model END** node after the given length of time.

Human Loop Task Nodes

The human loop task nodes are different from the other nodes because they have an interface assigned to them labeled “Space Navigator Tablet”, in which the human operator interacts with. Workload demand is attributed to the operator each of the human loop tasks using VACP workload values. This interface The Human Loop begins with the **Identify All Planets** and **Identify Background** tasks. Both tasks only occur once and

are used to simulate the initial awareness process of the human operator. The iterative section of the human loop begins with the **Determine Taskload** node. This node represents a decision for the human operator to either proceed to identify a ship and draw a route, or to monitor the environment. There is no time duration for this task. Within the paths tab, the decision type is set to tactical meaning that it the human will proceed to do one or the other, not both at the same time. The decision to monitor is determined using a reliance algorithm calculated using the “Calculate Reliance Macro”. The algorithm was determined from the baseline experiments. If the monitor conditions are not met, then the human will proceed to identify a ship to draw.

In the **Monitor** task node, the time of the task duration is only 0.1s and its path is a “Multiple” decision which routes to the determine taskload and redraw nodes. By having the time so quick, the overall time between identifying ships is flexible. There could be instances where the human is monitoring for several seconds or only half of a second. The other path directs to the **Redraws** node. The redraws node simply increments the number of redraws and doesn’t have a path extending from it.

If the human operator does not monitor the environment, then they will begin the **Identify Ships** task node. The task time is based on the number of ships on screen. There are three different distributions set according to a low, medium, and high number of ships on screen. There are no other effects from this task. After identifying ships, the human operator proceeds to the **Select Ship and Draw Route** node. The time distributions for this task are also based on low, medium, and high number of ships on screen. In the beginning effect, the code searches the ShipArray and finds the ship that has been on the

screen the longest and also does not have a route. After it finds a ship, this ship is given a route. The ending effects increments the number of human draws and determines the probability of redrawing a ship. There are three paths leaving this node, set at a “multiple” decision type. The first path traces back to the Determine Taskload node, which permits the human loop nodes to execute iteratively. The second path connects to the Ship Path node, which simulates a ship following the route given by the human. The third path connects to the Redraw node.

Agent Loop

The **Automation Selection** task node simulates the agent identifying ships and drawing routes for ships that meet the given criteria. The agent loop is only one task node that iterates on itself. The time duration of the task is set to 0.01s to mimic the constant awareness of the real agent used in Space Navigator. In the effects tab, the main part of the code is found within the release condition. In the release condition, the code is looking for a ship that has been on screen without a route greater than or equal to the delay time. If it draws a route for a ship, then the task will release an entity to the Ship Path node and continue to reiterate. If it does not draw a route for a ship, the task will not release an entity and will continue to run the code within the release condition.

Ship Paths and Destinations

There are two task nodes that contain ship path time distributions. One task is for the human draws, **Ship Path**, and the other is for the agent, **Ship Path Auto**. Both tasks are responsible for keeping a ship entity on its designated route for a given length of time.

After the time has elapsed, then the ship is directed to the destination nodes. In the **Ship Path** node (the one designated for human draws), there are three different time distributions in the ship path nodes representing low, medium, and high number of ships on screen. This is used to represent how people may adjust their route drawing strategy according to the taskload they're experiencing. In the agent's ship path, there is only one time distribution because it only draws straight lines. At the end of this task, the decision of the ship's final destination is determined probabilistically according to results from the baseline experiment. There is an option to use tactical decision making, which incorporates the CalculateResult macro. This macro uses several equations to determine the result with the primary factor being the number of ships on screen. This macro wasn't used in the experiment because it didn't align with results from the baseline experiment. However, it could be refined and prove to be a valuable aspect of the simulation, especially when testing several different types of variables. The destination nodes, **Drawn Collision**, **Drawn Destination**, or **Drawn Off Screen**, indicate the final results of the ship entities. These nodes serve as counters which increment that type of destination variable, decrement the number of ships on screen, and change score accordingly.

Bibliography

- Anderson, M., Anderson, S. L., & Armen, C. (2004). Towards Machine Ethics. In *Proceedings of the AAAI 2005 Fall Symposium on Machine Ethics* (pp. 1–7). Crystal City, VA: AIAA.
- Arthur, W., Edwards, B. D., Bell, S. T., Villado, A. J., Station, C., & Bennett, W. (2005). Team Task Analysis : Identifying Tasks and Jobs That Are Team Based, *47*(3), 654–669.
- Bindewald, J. M., Miller, M. E., & Peterson, G. L. (2014). A function-to-task process model for adaptive automation system design. *International Journal of Human-Computer Studies*, *72*(12), 822–834.
<http://doi.org/10.1016/j.ijhcs.2014.07.004>
- Bruemmer, D. J., Marble, J. L., & Dudenhoeffer, D. D. (2002). Mutual initiative in human-machine teams. In *Proceedings of the IEEE 7th Conference on Human Factors and Power Plants* (pp. 22–30).
<http://doi.org/10.1109/HFPP.2002.1042863>
- Byrne, E., & Parasuraman, R. (1996). Psychophysiology and adaptive automation. *Biological Psychology*, *42*, 249–268. Retrieved from <http://www.sciencedirect.com/science/article/pii/0301051195051619>
- Clark, M., Kearns, K., & Overholt, J. (2014). *Air Force Research Laboratory Test and Evaluation, Verification and Validation of Autonomous Systems* (Vol. XX). Wright-Patterson AFB, OH.
- Colombi, J. M., Miller, M. E., Schneider, M., McGrogan, J., Long, D. S., & Plaga, J. (2011). Predictive Mental Workload Modeling for Semiautonomous System Design: Implications for Systems of Systems. *Systems Engineering*, *14*(3), 305–326. <http://doi.org/10.1002/sys>
- Delligatti, L. (2013). *SysML Distilled: A Brief Guide to the Systems Modeling Language*. Addison-Wesley.
- Endsley, M. R. (2015). *Autonomous Horizons: System Autonomy in the Air Force - A Path to the Future* (Vol. 1). Retrieved from
<http://www.af.mil/Portals/1/documents/SECAF/AutonomousHorizons.pdf?timestamp=1435068339702>
- Endsley, M. R., & Kaber, D. B. (1999). Level of automation effects on performance, situation awareness and workload in a dynamic control task. *Ergonomics*, *42*(3), 462–492. <http://doi.org/10.1080/001401399185595>
- Endsley, M.R., and Kaber, D. B. (1999). Level of Automation Effects on Performance, Situation Awareness and Workload in a Dynamic Control Task. *Ergonomics*, *42*, 462–492.
- Feigh, K. M., Dorneich, M. C., & Hayes, C. C. (2012a). Toward a Characterization of Adaptive Systems: A Framework for Researchers and System Designers. *Human Factors: The Journal of the Human Factors and Ergonomics Society*. <http://doi.org/10.1177/0018720812443983>

- Feigh, K. M., Dorneich, M. C., & Hayes, C. C. (2012b). Toward a Characterization of Adaptive Systems: A Framework for Researchers and System Designers. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 54(6), 1008–1024. <http://doi.org/10.1177/0018720812443983>
- Goodman, T., Miller, M. E., Christina, F., & Bindewald, J. (2016). Timing Within Human-Agent Interaction and its Effects on Team Performance and Human Behavior. In *Cogsim 2016*. IEEE.
- Goodman, T., Miller, M., & Rusnock, C. (2015). Incorporating Automation: Using Modeling and Simulation to Enable Task Re-Allocation. In *Winter Simulation Conference*. Huntington Beach, CA: IEEE. <http://doi.org/10.1073/pnas.0703993104>
- Hart, S. G., California, M. F., & Staveland, L. E. (1988). Development of NASA-TLX (Task Load Index): Results of Empirical and Theoretical Research. In P. a. Hancock & N. Meshkati (Eds.), *Human Mental Workload* (pp. 239–250). Amsterdam: North Holland Press. [http://doi.org/10.1016/S0166-4115\(08\)62386-9](http://doi.org/10.1016/S0166-4115(08)62386-9)
- Hart, S. G., & Staveland, L. E. (1988). Development of NASA-TLX (Task Load Index): Results of Empirical and Theoretical Research. *Advances in Psychology*, 52(C), 139–183.
- Hilburn, B., Molloy, R., Wong, D., & Parasuraman, R. (1993). *Operator versus computer control of adaptive automation*. Warminster, PA.
- Hoc, J.-M. (2001). Towards a cognitive approach to human–machine cooperation in dynamic situations. *International Journal of Human-Computer Studies*, 54(4), 509–540. <http://doi.org/10.1006/ijhc.2000.0454>
- Huntsberger, T. (2011). Cognitive architecture for mixed human-machine team interactions for space exploration. *IEEE Aerospace Conference Proceedings*. <http://doi.org/10.1109/AERO.2011.5747552>
- Improved Performance Research Integration (IMPRINT) Tool. (2010). Retrieved from [https://dap.dau.mil/aphome/das/Lists/Software Tools/DispForm.aspx?ID=58](https://dap.dau.mil/aphome/das/Lists/Software%20Tools/DispForm.aspx?ID=58)
- Johnson, M., Bradshaw, J. M., Feltovich, P. J., Hoffman, R. R., Jonker, C., Riemsdijk, B. Van, & Sierhuis, M. (2011). Beyond cooperative robotics: The central role of interdependence in coactive design. *IEEE Intelligent Systems*, 26(3), 81–88. <http://doi.org/10.1109/MIS.2011.47>
- Klein, G., Woods, D. D., Bradshaw, J. M., Hoffman, R. R., & Feltovich, P. J. (2004). Ten challenges for making automation a “team player” in joint human-agent activity. *IEEE Intelligent Systems*, 19(6), 91–95. <http://doi.org/10.1109/MIS.2004.74>
- Millot, P. (2014). *Designing Human-Machine Cooperation Systems*. (J.-C. Pomerol, Ed.). London, UK: John Wiley & Sons.
- Parasuraman, R. (2008). *Supporting Battle Management Command and Control: Designing Innovative Interfaces and*

Selecting Skilled Operators. Fairfax, VA.

- Parasuraman, R., Bahri, T., Deaton, J. E., Morrison, J. G., & Barnes, M. (1992). *Theory and design of adaptive automation in aviation systems*. Report No. NAWCADWAR-92033-60. Warminster, PA: Naval Air Warfare Center, Aircraft Division.
- Parasuraman, R., & Manzey, D. H. (2010). Complacency and bias in human use of automation: an attentional integration. *Human Factors*, 52, 381–410. <http://doi.org/10.1177/0018720810376055>
- Parasuraman, R., Sheridan, T. B., & Wickens, C. D. (2000). A model for types and levels of human interaction with automation. *IEEE Transactions on Systems, Man, and Cybernetics. Part A, Systems and Humans : A Publication of the IEEE Systems, Man, and Cybernetics Society*, 30(3), 286–297. <http://doi.org/10.1109/3468.844354>
- Parasuraman, R., Sheridan, T. B., & Wickens, C. D. (2008). Situation awareness, mental workload, and trust in automation: Viable, empirically supported cognitive engineering constructs. *Journal of Cognitive Engineering and Decision Making*, 2(2), 140–160. Retrieved from <http://edm.sagepub.com/content/2/2/140.short>
- Reid, G. B., & Nygren, T. E. (1988). The subjective workload assessment technique: a scaling procedure for measuring mental workload. In *Human Mental Workload* (pp. 185–218).
- Rouse, W. (1977). Human-Computer Interaction in Multitask Situations. *IEEE Transactions on Systems, Man & Cybernetics*, 7(5), 384–392. <http://doi.org/10.1109/TSMC.1977.4309727>
- 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(4), 662–667. <http://doi.org/10.1109/TSMCA.2010.2093888>
- Tattersall, A. J., & Foord, P. S. (1996). An experimental evaluation of instantaneous self-assessment as a measure of workload. *Ergonomics*, 39(5), 740–748.
- Teigen, K. H. (1994). Yerkes-Dodson: A Law for all Seasons. *Theory & Psychology*, 4(4), 525–547. <http://doi.org/10.1177/0959354394044004>
- Department of Defense, Defense Science Board. (2012). *Task Force Report: The role of autonomy in DoD Systems*. Office of the Under Secretary of Defense for Acquisition, Technology and Logistics. Washington, D.C.

REPORT DOCUMENTATION PAGE				<i>Form Approved</i> <i>OMB No. 074-0188</i>	
The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of the collection of information, including suggestions for reducing this burden to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.					
1. REPORT DATE (DD-MM-YYYY) 24-03-2016		2. REPORT TYPE Master's Thesis		3. DATES COVERED (From – To) September 2014 – March 2016	
TITLE AND SUBTITLE Understanding Effects of Autonomous Agent Timing on Human-Agent Teams Using Iterative Modeling, Simulation and Human-in-the-Loop Experimentation				5a. CONTRACT NUMBER	
				5b. GRANT NUMBER	
				5c. PROGRAM ELEMENT NUMBER	
6. AUTHOR(S) Goodman, Tyler J., 2d Lt, USAF				5d. PROJECT NUMBER 13RSL-AFIT1	
				5e. TASK NUMBER	
				5f. WORK UNIT NUMBER	
7. PERFORMING ORGANIZATION NAMES(S) AND ADDRESS(S) Air Force Institute of Technology Graduate School of Engineering and Management (AFIT/EN) 2950 Hobson Way, Building 640 WPAFB OH 45433-7765				8. PERFORMING ORGANIZATION REPORT NUMBER AFIT-ENV-MS-16-M-154	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) US Air Force Office of Scientific Research, Computational Cognition and Machine Intelligence Program Attn: James H. Lawton, PhD 4075 Wilson Blvd., Suite 350 Arlington, VA 22203 (703)696-5999 (DSN: 426-5999) James.Lawton.1@us.af.mil				10. SPONSOR/MONITOR'S ACRONYM(S) AFOSR	
				11. SPONSOR/MONITOR'S REPORT NUMBER(S)	
12. DISTRIBUTION/AVAILABILITY STATEMENT DISTRIBUTION STATEMENT A. APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.					
13. SUPPLEMENTARY NOTES This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States.					
14. ABSTRACT Recent U.S. Air Force Research Laboratory strategy documents have suggested the need for research in human-agent teaming. Teaming supports a dynamic shift in roles between the human and the agent, depending upon human performance and mission needs. Further, because the performance of these agents will be highly dependent upon the state of the human and the mission, this strategy suggests the need for increased use of modeling to provide a broader understanding of the automated agents' behavior. This thesis applies a combination of static modeling in SysML activity diagrams, dynamic modeling of human and agent behavior in IMPRINT, and human experimentation in a dynamic, event-driven environment. The dynamic models and human experiments are used to understand the effects of agent delay time on human behavior, performance, and workload, as well as team dynamics. The models and experiments illustrate that agent delay time has a significant effect upon team behavior, performance, and the roles assumed by the human and agent. Therefore, it is proposed that the consequences of agent timing are significant in the context of human agent teaming and that models, which incorporate the human and agent within a common modeling environment, can be useful in understanding this effect.					
15. SUBJECT TERMS Autonomy; Agent; Teaming; Modeling and Simulation; Timing					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT UU	18. NUMBER OF PAGES 128	19a. NAME OF RESPONSIBLE PERSON Dr. Michael E. Miller, AFIT/ENV
a. REPORT U	b. ABSTRACT U	c. THIS PAGE U			19b. TELEPHONE NUMBER (Include area code) (937) 255-3636, ext 4651 (michael.miller@afit.edu)

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. Z39-18