Proceedings of the 2016 Winter Simulation Conference
T. M. K. Roeder, P. I. Frazier, R. Szechtman, E. Zhou, T. Huschka, and S. E. Chick, eds.

PREDICTING THE EFFECTS OF AUTOMATION RELIABILITY RATES ON HUMAN-AUTOMATION TEAM PERFORMANCE

Anthony J. Hillesheim
Christina F. Rusnock

Department of Systems Engineering and Management
Air Force Institute of Technology
2950 Hobson Way, (AFIT/ENV)
Wright-Patterson AFB, OH 45433, USA

ABSTRACT

This study investigates the effects of reduced automation reliability rates on human-automation team performance. Specifically, System Modeling Language (SysML) activity diagrams and Improved Performance Research Integrated Tool (IMPRINT) models are developed for a tablet-based game which includes an automated teammate. The baseline model uses previously collected data from human-in-the-loop experiments where the automated teammate performs with 100% reliability. It is expected that team performance and user trust in automation will be affected if the automation is less reliable. The baseline model is modified to create alternate models that incorporate degraded automation reliability rates from 50% to 90%. This study finds that when automation reliability was 100% the automation was an effective teammate and enabled the human-automation team to achieve statistically improved performance over human-only scenarios. However, at reliability rates of 90% and less, the presence of the automated agent degraded system performance to levels less than achieved in human-only scenarios.

1 INTRODUCTION

Automation is prevalent in nearly every facet of today's military, transportation, industrial, and medical fields. These fields deal with many complex, repetitive tasks, which are well-suited to automation, thus enabling the human operator to focus his or her attention where it is needed (Hoff and Bashir 2015). The primary focus of many users' attention is on making decisions; automation consists of the technology that can select data, transform information, and make decisions or control processes (Lee, See, and City 2004). In order for the human-automation team to be effective or perform well, the cumulative effect of the team has to be greater than that of the human or automation individually. One of the factors that affects human-automation teaming performance is automation reliability (Wickens and Dixon 2007). The majority of research supports the notion that human-automation teaming performance is generally quite good when the automation is perfect (Wickens and Dixon 2007; Dixon and Wickens 2006; Rovira, McGarry, and Parasuraman 2007). For example, Dixon and Wickens used a perfectly reliable auditory alert system to help pilots detect system failures during military reconnaissance missions, and they discovered that the automation improved performance (Dixon and Wickens 2006). Unfortunately, most users of automation have experienced unreliable automation at some point. When conducting a complex and/or difficult task, performance will be limited and significantly reduced when automation is degraded; unreliable automation could actually harm performance relative to the human working without automation (Sheridan 1984). Highly reliable but imperfect automation has shown to be the cause of states of over-trust

(Parasuraman and Miller 2004), over compliance (Parasuraman and Manzey 2010) and reliance issues (Dixon, Wickens, and Mccarley 2006). While studies have examined the use of automation reliability as an aid to the user (Dixon and Wickens 2006; Wickens and Dixon 2007; Parasuraman and Manzey 2010) there is a lack of depth in the literature that addresses when an automated agent, who acts like a teammate rather than an aid, suffers from reduced reliability. Previous work found that systems using diagnostic automation experience positive performance when the automation had a reliability greater than 80%, neutral performance from 70% to 80% and negative performance when the automation's reliability was less than 70% (Dixon and Wickens 2006; Wickens and Dixon 2007; Maltz and Shinar 2003; Parasuraman and Manzey 2010). This paper extends this line of research by exploring the performance effects on human-automation teams when the automated agent is not perfectly reliable.

Two common forms of automation reliability errors are false alarms and misses (Dixon, Wickens, and Mccarley 2006). False alarms and misses directly affect both user reliance and compliance. Reliance pertains to the human operator's state when an alert or alarm is silent, meaning everything is "ok" (Dixon and Wickens 2006). Over-reliance on automation can create "automation-induced complacency," in which the automation is operating at a high level of reliability (but not perfectly), enabling the human operator to be lulled into a false sense of security, thus resulting in the human not detecting occasional automation failures (Singh, Tiwari, and Singh 2009). Inversely, compliance addresses the operator's response when the alarm sounds--whether true or false (Dixon and Wickens 2006). Wickens and Dixon examined false alarms and misses in terms of automation reliability as they pertain to unmanned aerial vehicles (UAVs). As previously discussed, perfect automation reliability had a beneficial effect on the human-automation system performance (Dixon and Wickens 2006). When using an automation reliability rate of 67%, the human-automation system performance was severely reduced and in some instances, the performance was worse than a human performing the tasks without the automation (Dixon and Wickens 2006). In an effort to further understand the impact on performance, the effects on user compliance and reliance were also examined. It was found that automation false alarms decreased system failure detection rates and increased system failure detection times compared to when the human performed the tasks without the automation (Dixon and Wickens 2006). Additionally, when exploring degraded automation reliability and reliance; it was found that an increased miss rate negatively affected user reliance and users became less trusting of the automation (Dixon and Wickens 2006). In a similar study, Roivra, McGarry, and Parasuraman conducted an experiment examining several levels of automation reliability and how performance was affected. They found that lower levels of automation reliability led to greater cost in decision-making accuracy and decreased performance (Rovira, McGarry, and Parasuraman 2007). Additionally, they found that as automation reliability increased, complacency increased (Rovira, McGarry, and Parasuraman 2007).

The focus of this paper is on how reduced reliabilities affect human-automation ***team*** performance, where the automation is not just a decision aid but an "equal" teammate. This research hypothesizes that because team members fill a unique role on the team, interdependence, reliance, and expectations are higher for team members than for decision aids. This relationship amongst team members will demand high reliability from each teammate, with reduced reliability affecting both the automation's actions and the team interactions. To ensure a teaming scenario, this paper focuses on pairing an autonomous agent with the human to create a synergistic effect in which the human-automation team out performs either the automation or the human alone.

The work conducted in this paper leverages the advantages provided by the Improved Performance Research Integrated Tool (IMPRINT) and simulation in general. Simulation provided a means with which to examine how reduced automation reliability rates have the potential to affect human-automation team performance. Additionally, the simulations were conducted at no cost and in a low-risk environment. Simulation was also able to provide practical and timely results for analysis. This allowed for an increase in efficiency when exploring multiple alternative models.

1.1 Purpose

The purpose of the research is to explore how human-automation team performance is affected by varying levels of automation reliability. Varying levels of automation reliability have been studied in previous experiments and research; however, little research has been conducted using an automated agent that works together with the human as a teammate, rather than working as an aid. By working as a teammate, the automation is able to complete tasks and make decisions without human supervision. It is hypothesized that when the automated teammate has reduced levels of reliability; the overall team performance will suffer. However, it is expected that just as imperfect human teammates can still be an asset to a team, imperfect (but highly reliable) automated teammates will also make a positive contribution to the human-automation team.

2 APPLICATION ENVIRONMENT

To explore the effect of automation reliability rates on team performance, it was necessary to select an application environment in which the human and the automation interact as a team, rather than the automation operating independently, as an aid, or under supervisory control. Thus, an environment in which tasks are highly integrated and there is a high level of human-automation interaction was necessary. The system selected for this research was the tablet computer game *Space Navigator*, a custom route-creation game similar to Harbormaster and Flight Control. The game consists of activities that are completed by the human, the automated agent, or both. The game contains four stationary planets present on the screen. Each planet is one of four colors: red, green, blue, or yellow. Spaceships are randomly generated on the sides of the screen at an interval of one spaceship every two seconds. Spaceships continue to appear until an allotted time of five minutes is over. Each spaceship is red, green, blue, or yellow. The player must direct each spaceship to the correct destination planet of corresponding color (e.g. red spaceship to red planet) by drawing a trajectory line on the game touch screen using his/her finger. The spaceship then follows this drawn trajectory route at a constant rate. If desired, trajectories may be re-drawn; this is often done to avoid a collision or 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, small bonuses appear in random locations throughout execution. If the path of a given spaceship crosses over one of these bonuses, it is `picked up' and a point bonus is given. The player loses points for allowing spaceships to traverse `no-fly zones' that move to different random locations on the screen at a set time interval. In the human-in-the-loop experiment, the game features 100% reliable straight-line automation, which draws straight lines from the spaceships to the planets with a trigger rate of 2 seconds (meaning that the automation draws the route if the spaceship has been on-screen for 2 seconds without the human creating a route). The environment allows for the automation's and the human's actions to affect each other. The environment also creates the opportunity for the human-agent team performance to be better than that of the human or automation alone. It is important to note that the automation does not feature collision avoidance, thus enabling the automation to serve as a team member and not a perfect solution capable of replacing the human player. Because the automation does not feature collision avoidance, human involvement results in higher performance than if the automated agent were to operate on its own. Figure 1 depicts the game environment with an annotated screen capture from Space Navigator, which illustrates the elements of the game described herein.

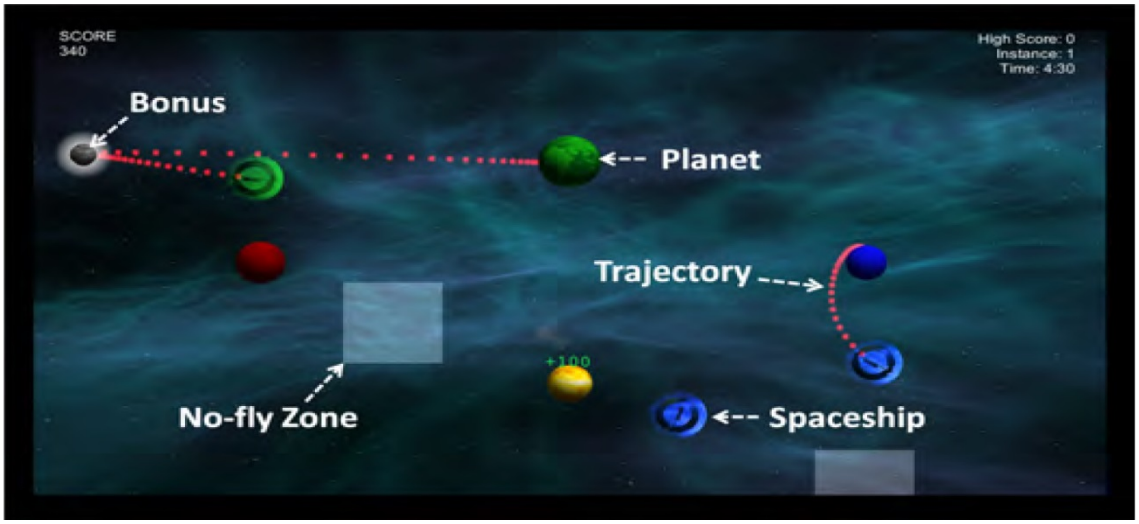

Figure 1: Space Navigator application environment.

3 METHOD

The first step in the procedure was conducting a human-in-the-loop experiment. This experiment was previously conducted and the results were used for this paper (Bindewald, Miller, and Peterson 2014). The second step consisted of modeling the application environment as a Systems Modeling Language (SysML) activity diagram. The activity diagram allowed for the conceptual model to be transferred to a diagram that consists of activities and functions with decision logic and flow. The activities and functions with decision logic and flow on the activity diagram were then transferred to an IMPRINT simulation model. IMPRINT is a discrete-event simulation software tool that allows for the modeling of human performance and analysis of human performance with a graphical user interface. The software allows for the use of task-network models that provide a visual representation of tasks performed by human operators (Mitchell 2009). This established a baseline model which captured all of the tasks that both the user and the automation completed during game play. The baseline model was validated using data previously collected on 36 subjects that played in the application domain with 100% reliable automation. Once the baseline model was validated, an alternate model was created. The alternate model addressed the reduction in automation reliability. IMPRINT simulated operator performance under automation reliability rates of 50%, 60%, 70%, 80%, 90%, and 100%. The results of the simulation were then compared against each other and the baseline model. Figure 2 below depicts an overview of the methodology for this study.

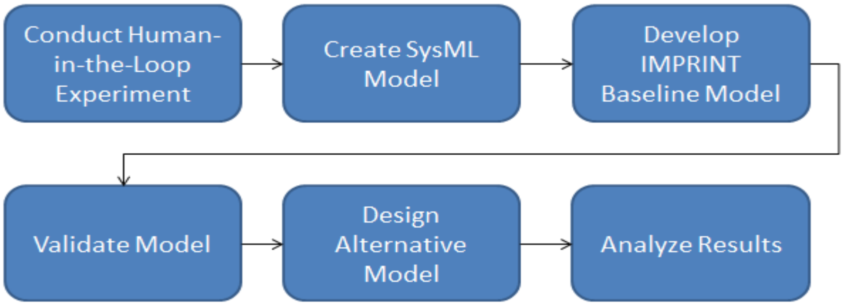

Figure 2 : Modeling and Simulation Process.

3.1 Human-In-The-Loop Experiment

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 experimental procedure consisted of a within subjects design in which each participant completed 16 five-minute games of *Space Navigator*. The first five games 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 4 five-minute games: one manual and then three with three differing automated agents, each with its own route-generation strategy. While three different automated agents were utilized in the experiment, only the agent with the straight-line strategy was of interest for this reliability simulation study. The straight-line strategy draws straight-line routes (i.e. shortest, most direct path) from the ship to the corresponding planet and was 100% reliable. Since only this agent is analyzed in this simulation study, the participant data from the three games (3 games x 36 participants = 108 games) that contained straight-line automation were used to populate and validate the model (Bindewald, Peterson, and Miller 2016).

3.2 SysML Model Development

After the human-in-the-loop experiment was conducted, an Activity Diagram was created using SysML. SysML is a general-purpose graphical modeling language that is useful for analysis, specification, design, verification and validation of complex systems (Steiner, Moore and Friedenthal 2014). The usability of the language extends to modeling human, automated, human-automation, and data centric systems (Steiner, Moore, and Friedenthal 2014). SysML facilititates the application of model-based systems engineering that provides several benefits: flow-based behavior, constraints on physical and performance properties, as well as structural classification of systems (Steiner, Moore and Friedenthal 2014).

Figure 3 shows the activity diagram which consists of multiple elements that provide clarity in understanding the activities and the flow of tasks. The elements include action nodes, control nodes, pins, and flows. The action nodes are the “transformers” of the process. The action nodes take inputs and transform them into outputs. The inputs and outputs are denoted by activity pins. The flows connect the output of one action and connect it to the input of another action.

The activity diagram includes the actions of both the automation and the human operator. The diagram includes all of the appropriate actions and decision necessary to accurately depict the functions of the system. This diagram provides the basis for task networks within IMPRINT.

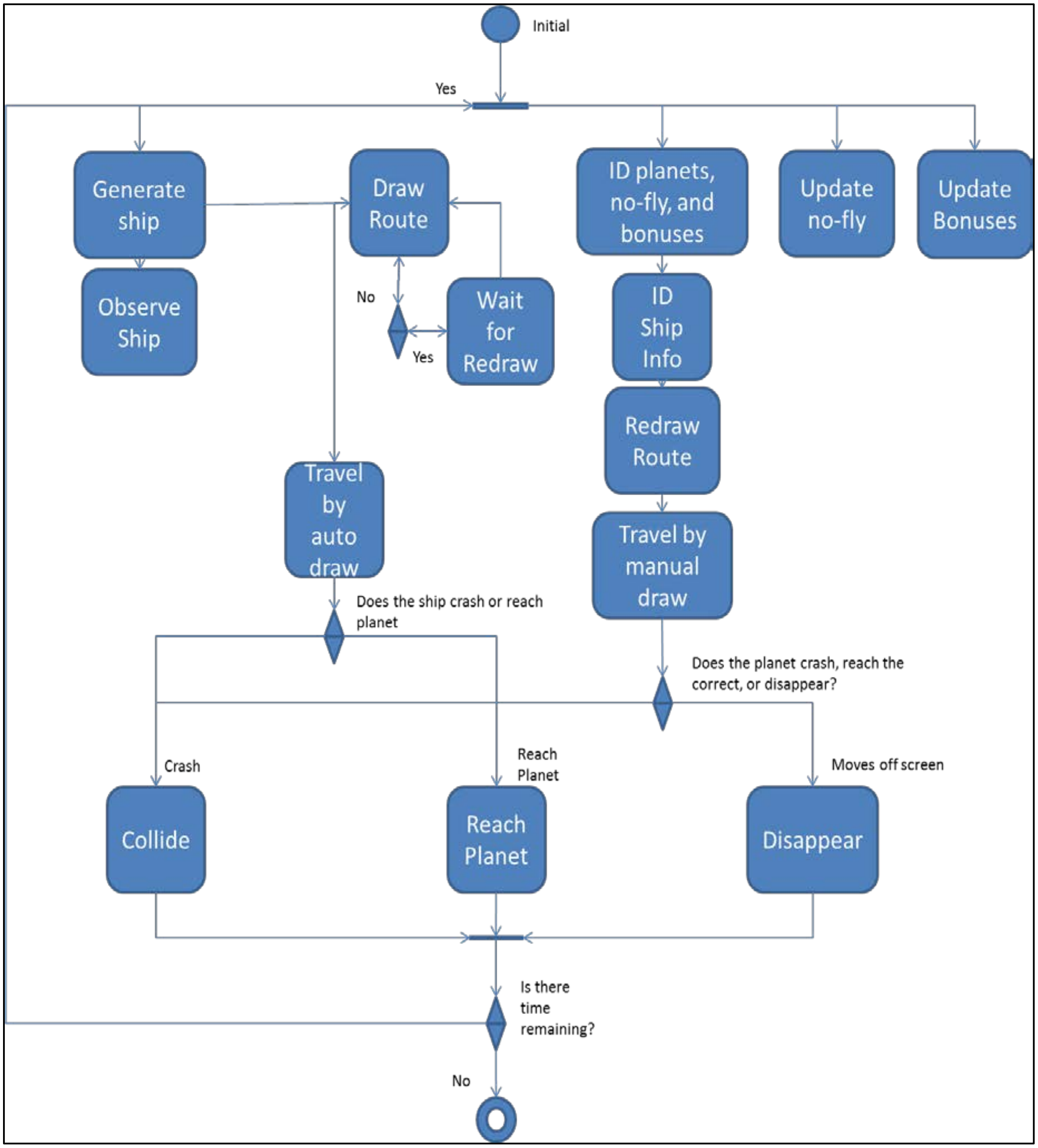

Figure 3: SysML Activity Diagram.

3.3 Imprint Baseline Model Development

As previously mentioned, IMPRINT provides an environment with which simulations can be performed to study human-system performance. The task networks developed in the activity diagrams were transferred to this modeling environment, capturing the flow of actions and decision logic. The completion of the IMPRINT model required determining task time probability distributions as well as probability functions relating to the successful completion or failure of certain tasks. Using the data from the human-in-the-loop experiment, the baseline model was created.

As seen in Figure 4, the baseline model task network is composed of three different task types. The purple tasks denote tasks performed by the automated agent, the blue tasks denote human player tasks, and the green tasks denote game environment tasks. Starting at the top of Figure 4, the first task flow is generating ships. The task depicts the system creating ships throughout the entire game. The “Draw Route” task does not begin until there are ships on screen without routes and will continue to loop on itself while there are ships on screen without routes. The task also accounts for the human user redrawing and will wait to redraw routes. The “Wait for Redraw” task stems from the need for the automation to wait for the human player while he or she redraws a route. The “Travel if by auto draw” task models a ship traveling on a route drawn by the automation. This task accounts for the ship picking up bonuses and/or flying through “no-fly zones”. The model also accounts for the background activities that occur throughout the game. The “Update no-fly zones” task models the system updating the zones every thirty seconds. The “Update Bonus Locations” task ensures that every thirty seconds the game is populated with three new bonus orbs. The “Operate clock” task is used to control the length of each game. The games are five minute in length.

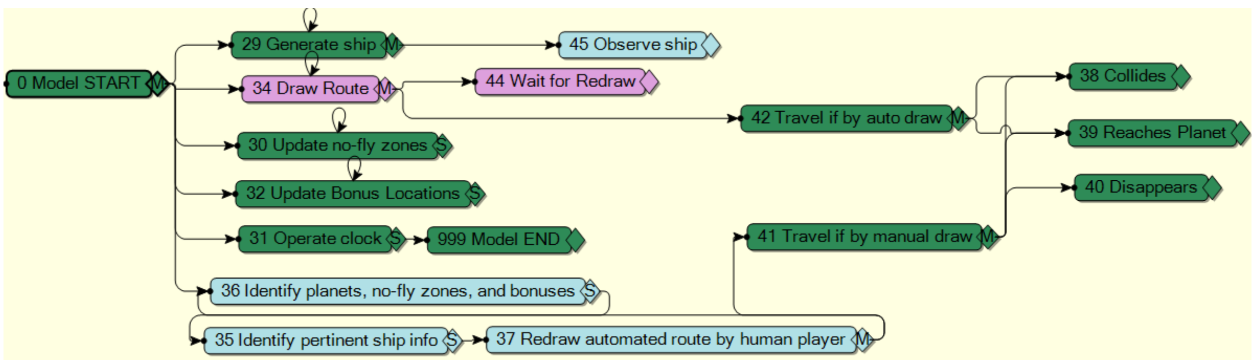

Figure 4: Baseline Model in IMPRINT.

The bottom of Figure 4 depicts the tasks the human user accomplishes to identify the game background and redraw routes if he or she does not agree with the automated routes drawn. The “Identify planets, no-fly zones, and bonuses” task accounts for the user examining all of the background game information. Once the human user has accomplished this task, he or she will move on to identifying all pertinent ship information such as the automated route drawn, if the ship is projected to collect bonuses, pass through a no fly zone, or potentially collide with another ship. The task also ensures there are ships on screen before it passes to redrawing routes. Before the “Redraw automated route by human player” task begins, the release condition ensures that there are ships with automatically drawn routes and that there are ships on screen.

After traveling, a ship has two possible outcomes if it is traveling on an automated route, collide or reach the correct planet. The automation will not draw a ship off-screen. However, if the ship is traveling on a human-generated route, the ship can collide, reach the correct planet, or disappear. If the ship goes to the “Collides” task, the task will account for the loss of 100 points. The “Reaches Planet” task increases score by 100 points. The final task encompasses when a ship disappears. The task “Disappears” accounts for when a ship with a manually draw route goes off-screen.

3.4 Model Validation

Before performing validation with the model predicted score data, we first confirm that the data meet the assumptions of normality. Figure 5 presents the normal probability plot to demonstrate that deviations are minimal and the data are normally distributed. The associated Shapiro-Wilk goodness-of-fit test yielded a p-value of .9914, thus the null hypothesis is not rejected and there is not significant evidence to state that the data are not normal.

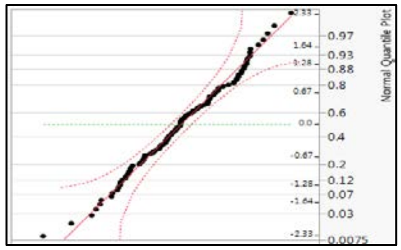

Figure 5: Simulated data results.

Figure 6 depicts the simulated predicted score data against the human-in-the-loop score data and shows that the means over-lap. The results of the two-sample t-test comparing simulated and real score data provide a p-value of .9965 for the two-tail test, indicating there is not a statistically significant difference between the predicted and measured score.. This means that there is insufficient evidence that the model produces results which differ from the real system; therefore, the baseline model is considered valid.

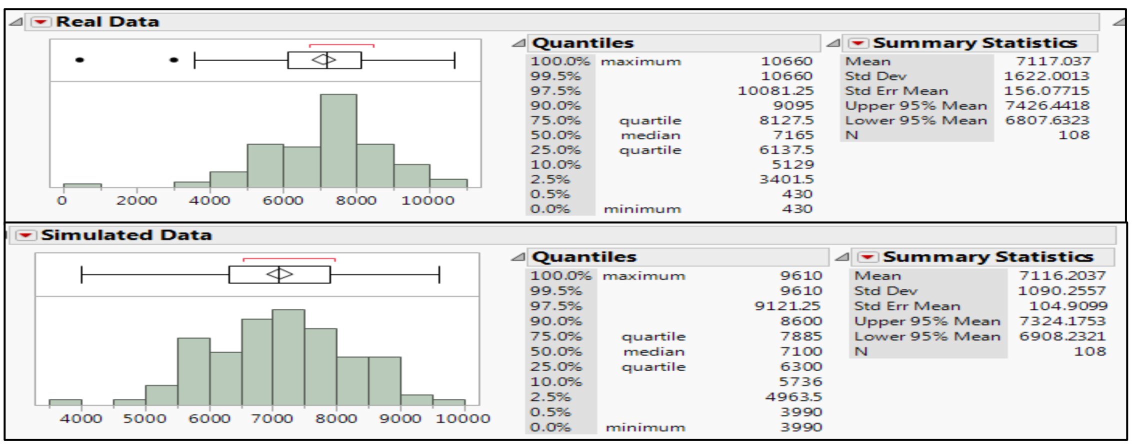

Figure 6: Simulated and Real Data Statistics

3.5 Experimental Design and Alternative Model Development

To capture differing reliability rates, the baseline model was modified in several ways. First, a new task node was created called “Travel to incorrect planet.” This task accounts for a ship traveling along an incorrectly drawn route. The node assumes that the time a ship spends traveling along an incorrect route is distributed as a Weibull distribution with the following parameters (5, 1). This task assumes that the amount of time it takes a ship to travel to an incorrect planet will be similar to the time it takes a ship to travel to the correct planet. The differing rates of reliability are captured in the “Travel if by auto draw” node. A random number is generated and compared against the following reliability rate setting. The model is run with six reliability rate settings: 50%, 60%, 70%, 80%, 90%, and 100%. The ending effects account for either correctly drawing the route or sending the ship to the “Travel to incorrect planet” node. The “Travel to incorrect planet” node feeds into the existing “Redraw automated route by human player” node (or the “Collision” node in the event that a collision occurs when traveling along the incorrect route, prior to human correction). The “Redraw automated route by human player” node captures the human redrawing a route to correct the erroneous route due to automation failure. This node has the same task time distributions and probability outcomes as any other human redraw. Thus, it is assumed that the redraw behaviors are the same as when the ship is traveling on a correct route. Figure 7 shows the alternative model.

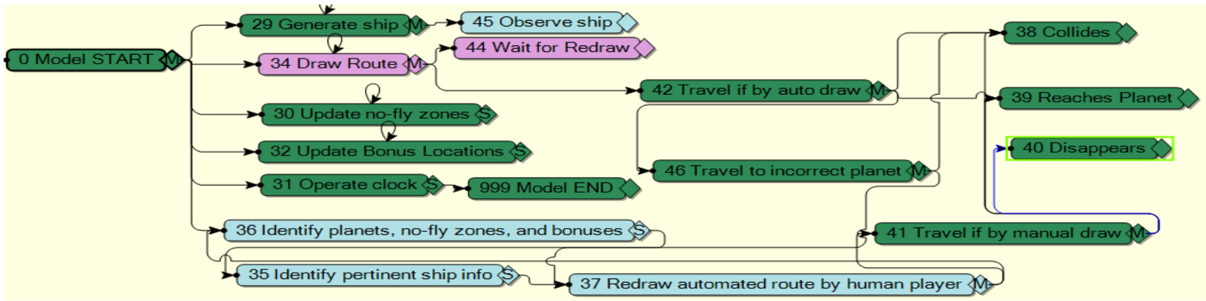

Figure 7: Alternate Model.

4 RESULTS AND DISCUSSION

A One-way ANOVA was performed to determine if there was a statistical difference between each of the score means of the automation reliability rates. The analysis of variances showed that the effect of automation reliability significantly influenced score, $F(5, 251) = 996.9143$, $MSE = 1.4277e+9$, $p < .0001$. Figures 8 and 9 provides the results of the Tukey's HSD for the team score at each of the predetermined reliability rates: 100% ($M = 7116.203$, $SD = 1090.26$), 90% ($M = 4119.6$, $SD = 1247.23$), 80% ($M = 2864.4$, $SD = 1246.55$), 70% ($M = 1394.9$, $SD = 1130.71$), 60% ($M = 311.0$, $SD = 1194.84$), and 50% ($M = -1250.5$, $SD = 1204.30$). Based on a Tukey's value of $CD = 2.8534$, all groups statistically differed from all other groups. As expected, with decreasing levels of automation reliability, the overall performance decreases.

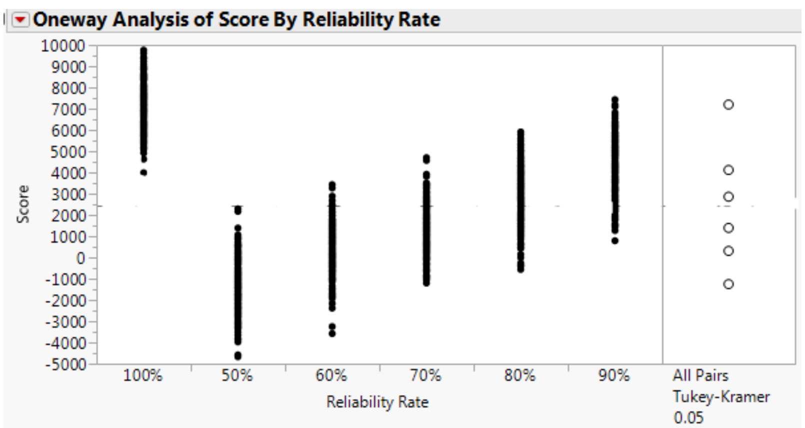

Figure 8: Means of reduced automation reliability rates.

Reliability Rate	Score	
One-way ANOVA	[F(5, 251) = 996.9143, p = 0.000]	
	95% CI	Tukey Groupings
100%	(6908,7324)	A
50%	(-1400,-1101)	B
60%	(162,459)	C
70%	(1254,1536)	D
80%	(2709,3019)	E
90%	(3964,4275)	F

Figure 9: Tukey Table for Varying Reliability Rates and Score.

The purpose of the research is to explore how human-automation team performance is affected by the level of automation reliability. Due to the inter-connected nature of teaming, this research hypothesized that reliability would negatively impact team performance, However, it was expected that imperfect, but highly reliable (e.g. 90% reliable) automation make a positive contribution to the human-automation team. Since each reliability rate produced statistically significant lower team performance, this indicates that the human-automation team performance is highly sensitive to the automation's reliability level. In the human-only trials, the players' the mean score was 5027. Thus only perfectly reliable (100% reliable) automation is effective in aiding human-automation team performance. All of the reduced reliability scenarios, produced mean scores lower than the human-only scenarios, thus the automated teammate was hindering, rather than helping, the team's performance.

Figure 10 provides the linear regression for predicted score based on reliability rate. From the regression equation we find that reliability rates of 92% and higher are expected to improve team score beyond human-only trials. This trend aligns with (Scerbo 1996) who found that with certain teaming tasks, the reliability of the automated teammate must be greater than 95% reliable (Scerbo 1996) in order to benefit from the automation. Although other researchers, such as Wickens and Dixon, have found benefits in automation with reliability levels as low as 70% (Dixon and Wickens 2006), these studies typically involve the use of an automated decision aid. The finding of this research supports the hypothesis that human-automation teaming may require higher levels of automation reliability than traditional automation aids.

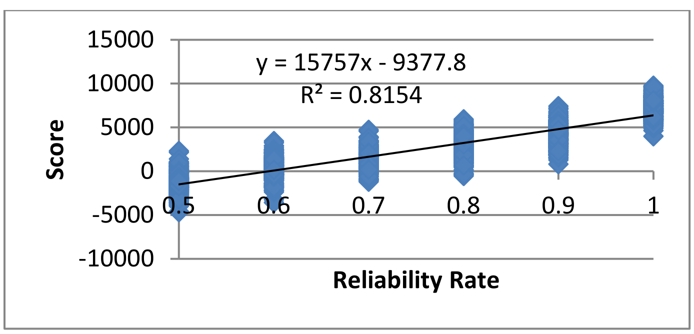

Figure 10: Linear Regression for Score and Reliability Rate.

5 CONCLUSIONS

This study sought to use simulation to understand the effects of automation reliability in a collaborative human-automation teaming scenario. The use of SysML modeling provided the framework for all tasks performed in the application environment. Implementing the SysML task network into IMPRINT enabled a simulation-based analysis of automation reliability. As anticipated, lower levels of automation reliability result in lower levels of human-automation team performance. While perfect automation enabled improved performance over human-only scenarios, reliability thresholds at 90% and below negated the value of the automation. The implication of this finding could affect autonomous system design, in that it may necessitate very high reliability requirements. If a system is being designed in which the automation acts as an independent teammate, it is likely that the automation will have to be designed and tested to have a reliability rate that is greater than 90%.

5.1 Future Research

The current IMPRINT model captures expected performance outcomes from degraded automation reliability, by modeling the human taking on the task of correcting routes which were incorrectly drawn by the automation. Thus the human is “re-doing” work that was erroneously performed by the automation. In addition, it would be quite likely that reduced automation reliability would reduce the human’s reliance on the automation, and thus instead of just “fixing” the automation’s errors, the human would pre-emptively perform some of the route-generation tasks currently performed by the automation. The current IMPRINT model does not include this additional operator-initiated trust behavior. Prior research demonstrates that reduced reliability results in reduced compliance and/or reliance, which in turn impacts human-automation performance (Dixon and Wickens 2006; Wright et al. 2013; Parasuraman and Miller 2004; Dzindolet et al. 2003; Hoffman, Lawson-Jenkins, and Blum 2006). Further iterations of this model could incorporate these trust-based behaviors.

In an effort to investigate the results presented in this analysis, a follow-on human-in-the-loop study that examines the relationship between reliability rates, performance, and user compliance is currently being conducted. The purpose of the follow-on human-in-the-loop study is to corroborate the findings of this paper, as well as highlight any shortcomings in the model created for reduced reliability. The study will also examine reliability levels between 90% and 100% to investigate the hypothesis from the linear regression analysis that high rates of reliability could yield improved performance over the human operating alone. The information gathered from this experiment will help to understand the relationship between trust and reliability for human-automation teaming, and how this relationship differs when the automation is not just a decision aid, but an autonomous teammate.

ACKNOWLEDGMENTS

The views in this article are those of the authors and do not necessarily reflect the official policy or position of the Department of the Air Force, Department of Defense, nor the U.S. Government. The authors gratefully acknowledge the support of the Air Force Office of Scientific Research.

REFERENCES

Bindewald, J. M., M. E. Miller, and G. L. Peterson. 2014. "A Function-To-Task Process Model For Adaptive Automation System Design." *International Journal of Human-Computer Studies* 822–834.

Bindewald, J. M., G. L. Peterson, and M. E. Miller. 2016. "Clustering-Based Online Player Modeling." *International Joint Conference on Artificial Intelligence (IJCAI) - Computer Games Workshop*.

Dixon, S. R., and C. D. Wickens. 2006. "Automation Reliability In Unmanned Aerial Vehicle Control: A Reliance-Compliance Model of Automation Dependence in High Workload." *Human Factors* 48(3): 474-86.

Dixon, S. R., C. D. Wickens, and J. S. Mccarley. 2006. "On The Independence of Compliance and Reliance : Are Automation False Alarms Worse Than Misses ?" *Human Factors* 49(4): 564–572.

Hoff, K. A., and M. Bashir. 2015. "Trust in Automation: Integrating Empirical Evidence on Factors That Influence Trust." *Human Factors* 57(3): 407–434.

Lee, J. D., K. A. See, and I. City. 2004. "Trust in Automation : Designing for Appropriate Reliance." *Human Factors* 46(1): 50–80.

Mitchell, D. K. 2009. "Workload Analysis of the Crew of the Abrams V2 SEP : Phase I Baseline IMPRINT Model." *Engineering*.

Parasuraman, R., and D. H. Manzey. 2010. "Complacency and Bias in Human Use of Automation: An Attentional Integration." *Human Factors* 52(3): 381–410.

Parasuraman, R., and C. A. Miller. 2004. "Trust and Etiquette in High-Criticality Automated Systems." *Communications of the ACM* 47(4): 51–55.

Rovira, E., K. McGarry, and R. Parasuraman. 2007. "Effects of Imperfect Automation on Decision Making in a Simulated Command and Control Task." *Human Factors* 49(1): 76–87.

Scerbo, M. W. 1996. "Theoretical Perspectives on Adaptive Automation." *Automation and Human Performance: Theory and Applications* 37-63.

Sheridan, T. 1984. "Supervisory Contol of Remote Manipulators, Vehicles and Dynamic Process: Experiments in Command and Display Aiding." *DTIC.* 49–137.

Singh, A. L., T. Tiwari, and I. L. Singh. 2009. "Effects of Automation Reliability and Training on Automation-Induced Complacency and Perceived Mental Workload." *Journal of the Indian Academy of Applied Psychology* 35(2009): 9–22.

Wickens, C. D., and S. R. Dixon. 2007. "The Benefits of Imperfect Diagnostic Automation: A Synthesis of the Literature." *Theoretical Issues in Ergonomics Science* 8(3): 201–212.

Wright, J. L., J. Y. C. Chen, S. A. Quinn, and M. J. Barnes. 2013. "The Effects of Level of Autonomy on Human-Agent Teaming for Multi-Robot Control and Local Security Maintenance." *DTIC*.

AUTHOR BIOGRAPHIES

ANTHONY J. HILLESHEIM is a Master of Science Student in the Department of Systems Engineering and Management at the Air Force Institute of Technology, Wright Patterson AFB, OH. He holds B.S. in Systems Engineering from the United States Air Force Academy. His thesis research involves examining trust and reliability in automation. His email address is anthony.hillesheim@afit.edu.

CHRISTINA F. RUSNOCK is an Assistant Professor of the Department of Systems Engineering and Management at the Air Force Institute of Technology, Wright Patterson AFB, OH. Dr. Rusnock earned her PhD in Industrial Engineering, Human Factors/Ergonomics Specialization, from the University of Central Florida, and a MS in Research and Development Management from the Air Force Institute of Technology. Her research interests include human performance modeling, mental workload, trust, and situation awareness, with a focus on applications in human-machine interaction and adaptive automation. She is a member of IIE and HFES. Her email address is christina.rusnock@afit.edu.