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.

SOURCES OF UNRESOLVABLE UNCERTAINTIES IN WEAKLY PREDICTIVE DISTRIBUTED VIRTUAL ENVIRONMENTS

Jeremy R. Millar
Communications and Information Directorate
Air Force Institute of Technology
2950 Hobson Way
Wright Patterson AFB, OH 45433, USA

Jason A. Blake
Simulation and Analysis Facility
Air Force Life Cycle Management Center
Bldg. 802, 2302 8th St., Area B
Wright Patterson AFB, OH 45433, USA

Douglas D. Hodson
Department of Electrical and Computer Engineering
Air Force Institute of Technology
2950 Hobson Way
Wright Patterson AFB, OH 45433, USA

J.O. Miller
Raymond R. Hill
Department of Operational Sciences
Air Force Institute of Technology
2950 Hobson Way
Wright Patterson AFB, OH 45433, USA

ABSTRACT

This work expands the notion of unresolvable uncertainties due to modeling issues in weakly predictive simulations to include unique implementation induced sources that originate from fundamental trade-offs associated with distributed virtual environments. We consider these trade-offs in terms of the Consistency, Availability, and Partition tolerance (CAP) theorem to abstract away technical implementation details. Doing so illuminates systemic properties of weakly predictive simulations, including their ability to produce plausible responses. The plausibility property in particular is related to fairness concerns in distributed gaming and other interactive environments.

1 INTRODUCTION

This paper considers two particular kinds of uncertainties that arise in distributed virtual environments and their relationship to weakly predictive simulation systems. We provide a review of uncertainties arising from insufficient knowledge about a system being modeled along with uncertainties arising due to the infrastructure of the experimental apparatus.

We consider the implementation-oriented aspects of distributed architectures from the perspective of the Consistency, Availability and Partition tolerance (CAP) theorem (Redmond and Wilson 2012) to abstract away details. Because the CAP theorem makes a strong statement concerning fundamental trades between three orthogonal aspects of any distributed system that manages a repository of data, it provides a means to reason about the quality of that data as a source from which an analysis often begins. Finally, we conclude with some observations concerning the use of weakly predictive simulated systems to support analysis-focused experiments.

This paper is organized as follows. Section 2 presents a review of uncertainty quantification and sources of uncertainty for computer-based experiment. Modeling human behavior in the presence of insufficient knowledge is the subject of Section 3 which illuminates the motivation to include humans within the simulation environment. Section 4 reviews how unresolvable uncertainties associated with modeling concerns fit within a topology of logical uses for simulation; of particular interest is the 'plausible outcomes' branch. Section 5 reviews fundamental tradeoffs made to create a highly interactive, responsive, distributed virtual environments and their associated issues with data consistency. We next characterize these issues as a CAP theorem problem. Section 6 highlights the concept of plausibility in relation to dynamic state space consistency issues. We conclude in Section 7 by organizing the two major sources of uncertainty (modeling and architecture) as two sub-branches to 'plausible outcomes' that lead to weakly predictive simulations.

2 UNCERTAINTY QUANTIFICATION

Uncertainty quantification (UQ) is the science of quantitative characterization and reduction of uncertainties in both computational and real world applications. UQ identifies and categorizes different sources of uncertainty with domains of interest. A casual literature search for areas of categorization and quantification of uncertainty reveals active work in finance, economics, manufacturing and even climate change.

The role of simulation is often the identification and modeling of uncertainty to understand its impact or effect on some aspect of system performance or operation. For the domain of computer-based experimentation, (Kennedy and O'Hagan 2001) categorized sources that have grown into a more exhaustive list that can be found here (Wikipedia 2016).

- *Parameter uncertainty*, which comes from the model parameters that are inputs to the computer model (mathematical model) but whose exact values are unknown to experimentalists and cannot be controlled in physical experiments, or whose values cannot be exactly inferred by statistical methods. Examples are the local free-fall acceleration in a falling object experiment, various material properties in a finite element analysis for engineering, and multiplier uncertainty in the context of macroeconomic policy optimization.
- *Parametric variability*, which comes from the variability of input variables of the model. For example, the dimensions of a work piece in a process of manufacture may not be exactly as designed and instructed, which would cause variability in its performance.
- *Structural uncertainty*, aka model inadequacy, model bias, or model discrepancy, which comes from the lack of knowledge of the underlying true physics. It depends on how accurately a mathematical model describes the true system for a real-life situation, considering the fact that models are almost always only approximations to reality. One example is when modeling the process of a falling object using the free-fall model; the model itself is inaccurate since there always exists air friction. In this case, even if there is no unknown parameter in the model, a discrepancy is still expected between the model and true physics.
- *Algorithmic uncertainty*, aka numerical uncertainty, which comes from numerical errors and numerical approximations per implementation of the computer model. Most models are too complicated to solve exactly.
- *Experimental uncertainty*, aka observation error, which comes from the variability of experimental measurements. The experimental uncertainty is inevitable and can be noticed by repeating a measurement for many times using exactly the same settings for all inputs/variables.
- *Interpolation uncertainty*, which comes from a lack of available data collected from computer model simulations and/or experimental measurements. For other input settings that do not have simulation data or experimental measurements, one must interpolate or extrapolate in order to predict the corresponding responses.

Structural and interpolation uncertainty are of particular interest in relation to this work. We consider the modeling of human behavior as analogous to the lack of understanding of the 'underlying true physics' of the system. In other words, we don't completely understand how humans make decisions; because of that, modeling them is an issue.

Uncertainties that arise in distributed virtual environments closely mirror interpolation issues. For our purpose, interpolation has less to do about modeling concerns, but is driven by fundamental trades associated with the simulation infrastructure that provides the experimental apparatus to conduct an experiment. We contend that both sources of uncertainty contribute to prediction outcomes that are classified as weak.

3 MODELING ISSUES

In modeling and simulation, abstraction is used to determine how details of a system are represented, while maintaining validity with respect to some purpose (Frantz 1995). Despite careful choices by the analyst to mitigate uncertainties, unresolvable uncertainties arise due to insufficient knowledge about a system being modeled (Bankes 1993). While these uncertainties are common across both distributed virtual environments and traditional self-contained discrete-event-based simulations; the implementation of a distributed virtual environment for analysis yields new sources of unresolvable uncertainty.

One particular modeling concern is representing humans and their behavior. Human behavior remains difficult to duplicate reliably; so in traditional simulations, modeling human behavior is often based on probabilistic draws from a defined or known distribution. This modeling approach is problematic when human behavior is not well understood, or if the purpose of the simulation experiment is to study that behavior.

Including humans within the simulated world is one of the key benefits of a distributed virtual environment. The goal of these systems is to create a shared sense of a represented world in which to interact. This leads to higher realism, as the human(s) are no longer models, they are real; but they risk adding noise to the system, especially if they are not clearly within the bounds of the system under study. As anyone with a family can attest, no two humans are exactly alike and even a reliable humans behavior will surprise you. This may be acceptable at family gatherings, but is an unfortunate occurrence for the analyst trying to draw conclusions from data collected from a distributed virtual environment used to conduct a study.

Introducing real humans into a simulation experiment presents challenges; if they are not directly part of the system under study, they can introduce unwanted noise, and if they don't know 'how they should act,' their very presence is considered to be an 'unresolvable uncertainity' or a modeling 'unknown.' This is a very real concern, especially if the system of interest is hypothetical - maybe representing the future or simply does not exist.

4 LOGICAL USES

In light of the fact that our simulations are not perfect reflections of reality, it must be determined how they might be useful. Considering only the types, sources and perceived magnitudes of uncertainty with a given simulation is not enough to make this determination. Only after considering the analytic purpose alongside simulation uncertainties can and educated opinion be formed regarding fitness for analytical use.

In (Dewar et al. 1996), a topology of logical uses for Distributed Interactive Simulation systems is presented (Figure 1). This topology divides the domain of logical uses into two subspaces; the first being experimental stimulus and the second as an analytical aid. The differentiator between these branches is defined by this question; for whose benefit is the study being executed (Dewar et al. 1996)? If a human-in-the-loop is the beneficiary, then the simulation is considered an experimental stimulus. Common examples of this type of simulation can be found in the training and education community where the purpose is to enhance three types of skills (i.e., motor, decision making, and operational skills) or entertainment (Topçu et al. 2016). Additional detail on simulation as experimental stimulus can be found in (Dewar et al. 1996).

Simulations where the beneficiaries are others beyond the participants in the simulation, then the system is categorized as an analytic aid (Dewar et al. 1996). This class of simulations will be familiar to most DoD analysts as this is where most constructive analysis tools fit within the topology. The key discriminator among the subclasses of analytic aids is the degree to which they are useful for prediction. Non-predictive uses of DIS do not require a high confidence fit between the simulation output and the real world, these

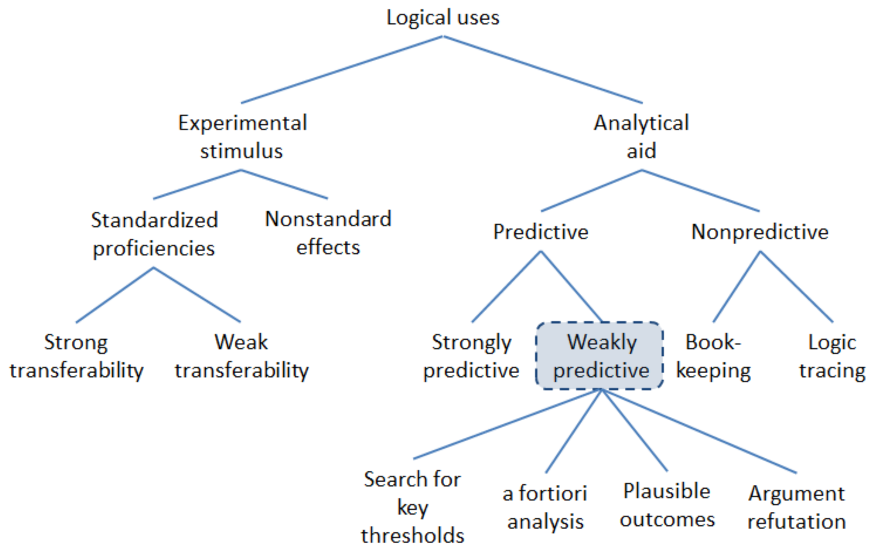

Figure 1: Topology of Logical Uses (Dewar et al. 1996).

uses include bookkeeping activities such as recording events and logic tracing. Predictive models place a higher requirement on the fit between simulation output and the real world system.

Strongly predictive models have a demonstrated capacity to forecast outcomes with a high degree of accuracy (Dewar et al. 1996). Weakly predictive models suffer from moderate to high levels of parametric, structural or unresolvable uncertainties, yet the model still describes some elements of the real system. This category is likely the most populated within the analytic aid class of simulations. Care should be taken to not overstate the predictive credibility of the analysis when using this type of simulation. Often the analyst is forced to use a weakly predictive model due to lack of data resulting from restrictions or nonavailability. Arguably, these system can play a supporting role in relation to experimentation, by generating hypotheses to investigate. In other words, use of weakly predictive simulations will not necessarily generate correct outputs, but might support a research strategy to understand and investigate the dynamics of a system.

Of particular interest is the 'plausible outcomes' branch in this topology, as it relates to unresolvable uncertainties due to modeling unknowns. One source of unknowns can be humans inserted into a virtual environment that really don't know 'how they should act or behave' to fairly represent a system of interest.

5 ARCHITECTURE ISSUES

It is fair to say that UQ is related to understanding known knowns and being tangentially aware of and accounting for, known unknowns. For a distributed virtual environment, this type of unknown or uncertainty arises from the software and/or hardware architecture that provides the experimental apparatus used to conduct an experiment. It is fair to say that we have known unknowns, but how those unknowns impact and influence outputs is not well understood.

Unresolvable uncertainties arise as a result of fundamental trades that must be made when the architecture of a distributed virtual environment is designed. Because these simulations by definition, include, interface and interact with humans and/or hardware, these systems are classified as real-time systems. Real-time systems define performance requirements in terms of timeliness; for example the time it takes to process new inputs to yield outputs is referred to as a response time. As an example, a human operator flying a simulated aircraft would expect (i.e., require) the system to respond to stick and throttle inputs by updating displays within a short period of time (e.g., 100ms).

This requirement is fundamentally at odds with the desire to maintain a consistent representation of the virtual world across all nodes in the distributed simulation. The challenge becomes problematic in

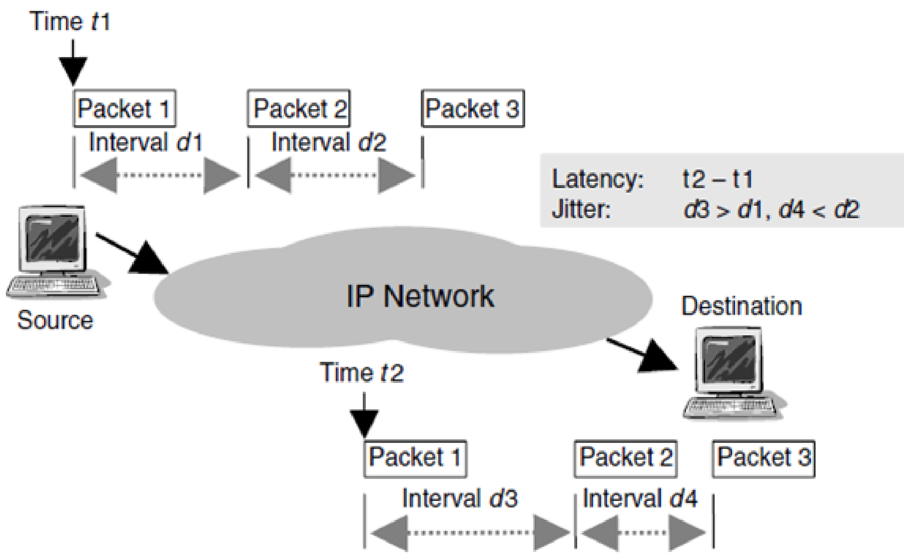

Figure 2: Latency and jitter affects (Armitage et al. 2006).

the presence of significant network latency and jitter. Latency is defined as the time required to deliver a network packet from a sender to receiver, and jitter measures the variation in that delivery time. As Figure 2 shows, latency and jitter affect the reliable delivery and timeliness of state data between nodes in the simulation. If the distributed nodes are connected via a network infrastructure with a relatively high latency, data being used by one simulation might be out of date or ‘old’ when compared to its current correct value as managed by another node. This inconsistency or error in state data is a distinguishing characteristic of distributed virtual environments and must be recognized and managed (Hodson and Baldwin 2009). The fundamental trade between state consistency and responsiveness is not new, it has existed since the advent of distributed interactive simulation; sometimes it is referred to as the consistency-throughput trade-off (Singhal and Zyda 1999). In the online gaming community, this issue manifests itself as an unfair game (e.g., a dead person shooting). How the concept of ‘fairness’ is defined, is an area of research, and relates to ‘plausibility’ or plausible responses which we discuss in Section 6.

If the purpose of the simulation is to serve as an analytical aid to support system study and understanding, then the system itself will most likely be derived from a conceptual model that defines what to represent. The conceptual model is mapped to entities which are created and managed by different simulation nodes (Hodson and Hill 2013). To illustrate this approach, Figure 3(a) presents a notional conceptual model that includes two interacting entities that exchange information related to an interaction. As shown in Figure 3(b), a possible architecture for the virtual environment might be two different simulations or simulators interconnected by a network. Given this arrangement, entity data created and locally managed will need to be shared using a network so that interactive environments can be created for each of the human participants. Figure 3(c) highlights the influence of network latency and jitter on the state data managed by each simulation. For this case, the state data associated with the location of remote entity(s) (i.e., dynamic shared state) most likely will not match their true position as managed by their local hosting application.

5.1 CAP Theorem

The complexities associated with the design and implementation a distributed virtual environment are well known. The underlying issues that create shared state inconsistencies are implementation specific and include the architecture of the software, such as single or multi-threaded, processing priorities, model execution/time advance assumptions, networking latency and jitter, model representation (fidelity), the

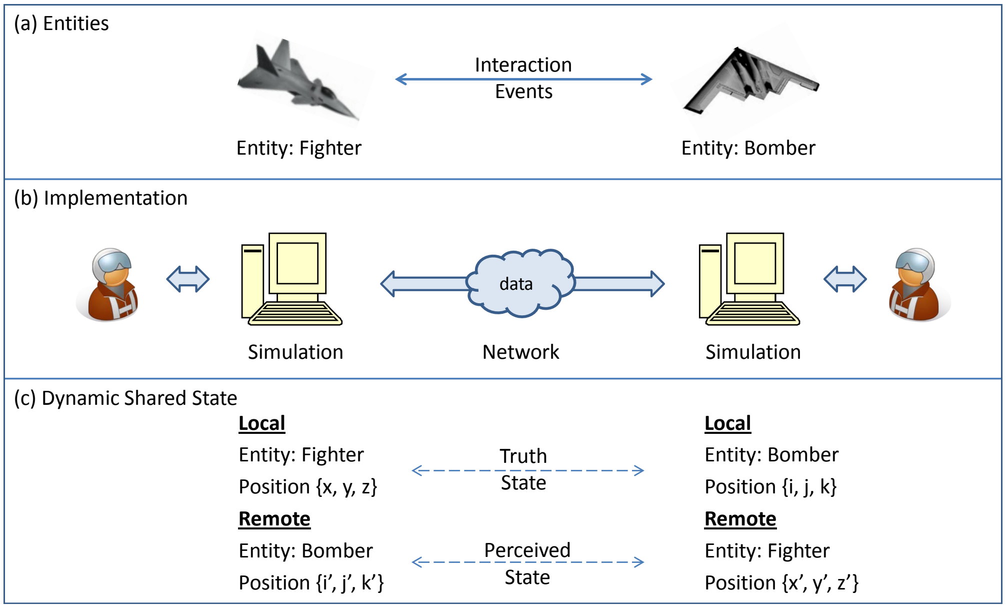

Figure 3: Entities, implementation, and dynamic shared state (Hodson and Hill 2013).

selection of various error threshold parameters, etc. To reason about these complexities, we leverage the CAP theorem - a useful abstraction from the domain of distributed databases.

The CAP theorem holds that you can create a distributed database that is *consistent* (writes are atomic and all subsequent requests retrieve the new value), *available* (the database will always return a value as long as a single server is running), or *partition tolerant* (the system will still function even if server communication is temporarily lost - that is, a network partition), but you can have only two at once (Redmond and Wilson 2012).

In other words, you can create a distributed database system that is consistent and partition tolerant, a system that is available and partition tolerant, or a system that is consistent and available (but not partition tolerant - which basically means not distributed). But it is not possible to create a distributed database that is consistent and available and partition tolerant at the same time (Redmond and Wilson 2012).

Our use of CAP hinges upon viewing the virtual environment in the same light as a distributed database system that attempts to keep its replicated data in sync (Millar et al. 2016). From this viewpoint, models are the consumers of state data that produce outputs. If inputs to the model(s) include dynamic shared state, then outputs may differ.

6 PLAUSIBLE RESPONSES

The CAP theorem has an important consequence for distributed virtual environments related to the plausibility of interaction outcomes as observed at the server and clients. Since virtual environments typically prioritize availability and partition tolerance above consistency, state divergence between the server and client is virtually guaranteed. In the main, this divergence is of little consequence. However, when the server and client must independently compute some result based on the same input variables at the same time, problems with the plausibility of outcomes can arise due to state divergence. For instance, if the interaction of interest is collision detection and the positional state of involved entities is not consistent, the server may detect a collision while the client does not. This problem is exacerbated when the server completes its outcome arbitration – the client's version of the state may well be overwritten by a completely different result, yielding an implausible outcome. Steed and Oliveira refer to this as joint plausibility, i.e., the notion that two or more users accept that they are viewing the same simulation of a shared space (Steed and Oliveira 2009).

Note that plausibility does not imply that the server and client compute identical interaction results. Rather, joint plausibility requires that the computed results be close enough that users are willing to accept the arbitrated results and not reject them out of hand as nonsensical. That is, dead men must not keep shooting (Mauve 2000) due to a divergence in states between client and server. Minimizing state inconsistency between nodes in a distributed virtual environment is a well-studied problem; Delaney *et al* provide an excellent overview of consistency maintenance algorithms (Delaney, Ward, and McLoone 2006a, Delaney, Ward, and McLoone 2006b).

The maximum tolerable state divergence resulting in plausible outcomes is interaction dependent (Itzel et al. 2010). None of the commonly employed consistency maintenance algorithms explicitly account for outcome plausibility, preferring instead to schedule state updates in a fashion designed to minimize inconsistency. However, minimal state inconsistency between nodes does not necessarily imply a plausible outcome.

7 CONCLUSIONS

The main contribution of this research is the unification of two well understood, but separately considered drivers that cause distributed virtual environments to produce at best plausible outcomes which lead to weak prediction. The first relates to modeling aspects of a system that are unknown, the second is a result of the simulation architecture or experimental apparatus used to perform the experiment. The first contributor to plausible outcomes might consistently yield the same agreed upon incorrect result; the second will most likely yield a range of results depending upon which node within the distributed system computes it.

ACKNOWLEDGMENTS

This research was supported by the Office of the Secretary of Defense, Operational Test and Evaluation (OSD DOT&E) and the Test Resource Management Center (TRMC) with the Science of Test Research Consortium.

DISCLAIMER

The views expressed in this article are those of the authors and do not reflect the official policy or position of the United States Air Force, Department of Defense, or the US Government.

REFERENCES

Armitage, G., M. Claypool, and P. Branch. 2006. *Networking and Online Games: Understanding and Engineering Multiplayer Internet Games*. Wiley.

Bankes, S. 1993. “Exploratory Modeling for Policy Analysis”. *Operations Research* 41 (3): 435–449.

Delaney, D., T. Ward, and S. McLoone. 2006a. “On Consistency and Network Latency in Distributed Interactive Applications: A Survey – Part I”. *Presence: Teleoperators and Virtual Environments* 15 (2): 218–234.

Delaney, D., T. Ward, and S. McLoone. 2006b. “On Consistency And Network Latency in Distributed Interactive Applications: A Survey – Part II”. *Presence: Teleoperators and Virtual Environments* 15 (4): 465–482.

Dewar, J. A., S. C. Bankes, J. S. Hodges, T. Lucas, D. K. Saunders-Newton, and P. Vye. 1996. “Credible Uses of the Distributed Interactive Simulation (DIS) System”. Technical report, Santa Monica, California.

Frantz, F. K. 1995. “A Taxonomy of Model Abstraction Techniques”. In *Proceedings of the 1995 Winter Simulation Conference*, edited by C. Alexopoulos, K. Kang, W. Lilegdon, and D. Goldsman, 1413–1420. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Hodson, D. D., and R. O. Baldwin. 2009. "Characterizing, Measuring, and Validating the Temporal Consistency of Live-Virtual-Constructive Environments". *Simulation: Transactions of The Society for Modeling and Simulation International* 85 (10): 671–682.

Hodson, D. D., and R. R. Hill. 2013. "The Art and Science of Live, Virtual, and Constructive Simulation for Test and Analysis". *Journal of Defense Modeling and Simulation: Applications, Methodology, Technology* 11:77–89.

Itzel, L., V. Tuttlies, G. Schiele, and C. Becker. 2010. "Consistency management for interactive peer-to-peer-based systems". In *Proceedings of the 3rd International ICST Conference on Simulation Tools and Techniques*. Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering).

Kennedy, M. C., and A. O'Hagan. 2001. "Bayesian Calibration of Computer Models". *Journal of the Royal Statistical Society. Series B (Statistical Methodology)* 63 (3): 425–464.

Mauve, M. 2000. "How to Keep a Dead Man from Shooting". In *Interactive Distributed Multimedia Systems and Telecommunication Services*, 199–204. Springer.

Millar, J. R., D. D. Hodson, G. L. Peterson, and D. K. Ahner. 2016, April. "Data Quality Challenges in Distributed Live-Virtual-Constructive Test Environments". *Journal of Data and Information Quality* 7 (1-2): 2:1–2:3.

Redmond, E., and J. R. Wilson. 2012. *Seven Databases in Seven Weeks: A Guide to Modern Databases and the NoSQL Movement*. The Pragmatic Bookshelf.

Singhal, S., and M. Zyda. 1999. *Networked Virtual Environments: Design and Implementation*. Addison Wesley.

Steed, A., and M. F. Oliveira. 2009. *Networked Graphics: Building Networked Games and Virtual Environments*. Elsevier.

Topçu, O., U. Durak, H. Oğuztüzün, and L. Yilmaz. 2016. *Distributed Simulation: A Model Driven Engineering Approach*. Springer.

Wikipedia 2016. "Uncertainty Quantification". Accessed April 28, 2016. https://en.wikipedia.org/wiki/Uncertainty_quantification.

AUTHOR BIOGRAPHIES

JEREMY R. MILLAR is a active-duty network operations officer in the United States Air Force. He is the Director, Communications and Information for the Air Force Institute of Technology (AFIT). He holds Bachelors and Masters degrees in Computer Science from the University of Tennessee and AFIT, respectively. He is currently completing a Ph.D. in Computer Science at AFIT. His research interests lie in distributed systems, virtual environments, and machine learning. Major Millar has served in a variety of Air Force assignments including deployments to Iraq, Afghanistan, and Guantanamo Bay, Cuba. His email address is jeremy.millar@afit.edu.

JASON A. BLAKE is a Systems Analysis Engineer for the U.S. Air Force Simulation and Analysis Facility (SIMAF) and a Ph.D. student in the Operations Research Department at the Air Force Institute of Technology. He has a B.S. in Industrial and Systems Engineering from The Ohio State University and a M.S. in Operations Research from the Air Force Institute of Technology. His research interests include combat modeling, distributed simulation and design and analysis of experiments. He can be reached at jason.blake.3@us.af.mil.

DOUGLAS D. HODSON is an Assistant Professor of Software Engineering with the Air Force Institute of Technology. He received a B.S. in Physics from Wright State University in 1985, and both an M.S. in Electro-Optics in 1987 and an M.B.A. in 1999 from the University of Dayton. He completed his Ph.D. at the Air Force Institute of Technology in 2009. He has over 25 years of experience in the domain of modeling and simulation and has a research interest in characterizing the consistency of shared simulation state data in terms of its temporal properties to estimate Live-Virtual-Constructive and Distributed Virtual Simulations performance, cloud computing and modeling quantum key distribution systems. He is the technical lead developer for the open-source OpenEaagles simulation framework that has been used to develop a wide variety of standalone and distributed simulation applications. He is also a DAGSI scholar and a member of Tau Beta Pi. His email address is douglas.hodson@afit.edu.

J.O. MILLER is a 1980 graduate of the U.S. Air Force Academy (USAFA) and retired from the Air Force as a Lt. Colonel in January 2003. In addition to his undergraduate degree from USAFA, he received an MBA from the University of Missouri at Columbia in 1983, his M.S. in Operations Research from the Air Force Institute of Technology (AFIT) in 1987, and his Ph.D. in Industrial Engineering from The Ohio State University in 1997. He is an Associate Professor of Operations Research in the Department of Operational Sciences at AFIT. His research interests include combat modeling, computer simulation, and ranking and selection. He can be reached at john.miller@afit.edu.

RAYMOND R. HILL is a Professor of Operations Research with the Air Force Institute of Technology. He has a Ph.D. in Industrial and Systems Engineering from The Ohio State University. His research interests include applications of simulation, applied statistical modeling, mathematical modeling, decision analytical methods, the design and analysis of heuristic optimization methods, and agent-based modeling. He is an Associate Editor for Quality Engineering, Naval Research Logistics, Military Operations Research, the Journal of Defense Modeling and Simulation, and the Journal of Simulation. He was a proceedings co-chair for both the 2008 and 2009 Winter Simulation Conferences and was the General Chair for the 2013 conference. His email address is rayrhill@gmail.com.