BENEFITS AND CONSEQUENCES OF AUTOMATED LEARNING IN COMPUTER GENERATED FORCES SYSTEMS

Mikel D. PETTY

Introduction

Should computer generated forces (CGF) systems include automated learning capabilities? The CGF research literature contains many statements by CGF experts that the ability to learn will be generally valuable, even necessary, in future CGF systems. A variety of significant benefits for CGF systems and military simulation in general are claimed to follow from automated learning. However, upon closer examination, it seems to be not so obvious that learning by CGF systems would necessarily be beneficial for many uses of CGF systems. This paper takes a respectfully skeptical position regarding CGF learning and provides arguments that CGF learning could compromise and confound the utility of CGF systems for the most common CGF applications.

This paper begins by defining CGF systems and grouping CGF simulation applications into three broad types. Calls in the CGF research literature for automated learning by CGF systems are surveyed. Categories of learning-modified behavior for CGF systems are defined based on what behaviors have been learned. Arguments are given, organized by application and behavior category, explaining how learning could increase and/or reduce the utility of the CGF system for the application. Real and notional examples are provided. Finally, specific applications where learning by CGF systems might be useful are identified.

Background

This section provides background information on CGF systems, types of CGF applications, and automated learning for CGF systems.

Computer Generated Forces

Computer generated forces[1] are automated or semi-automated entities (such as tanks, aircraft, infantry) in a battlefield simulation that are generated and controlled by a computer system (i.e., a CGF system), perhaps assisted by a human operator.[2,3] CGF systems are often used in training simulations to provide both opposing forces and supplemental friendly forces for human trainees. CGF systems are also used to generate many or all of the entities in battlefield simulations being used for non-training purposes, such as analysis or experimentation. CGF systems model both physical phenomena, such as terrain and combat, and behavior, such as tactical maneuvers; the latter is of primary concern here. It is intended that the behavior of the simulation entities generated by a CGF system be both doctrinally accurate, so as to provide a reliable basis for training or analysis, and plausibly human, so as to be realistic and engaging.

One example CGF system is ModSAF, which generates and controls individual battlefield entities, such as tanks or helicopters, in real-time simulation systems. It has been widely distributed and used extensively for training, analysis, and experimentation.[4] (However, in the U. S. ModSAF is being superseded by successor systems.) ModSAF includes user interface, network interface, physical modeling, low (entity) level behavior generation, and high (military unit) level behavior generation capabilities. Each ModSAF system, running on a personal computer or workstation, can generate approximately a battalion-sized force. Behavior generation in ModSAF is based on a library of doctrinal tactical behaviors that can be assigned by an operator to individual entities or groups of entities that compose military units, such as platoons or companies and executed automatically.[5] Once assigned, ModSAF automatically executes the behaviors, controlling the actions of the individual entities.

CGF Applications

For the purposes of this paper, the applications of CGF systems in simulation will be grouped into three classes: training, analysis, and experimentation.

Training. Training simulations, in general, are intended to induce learning in human trainees. The trainees interact with or participate in the simulation, which provides an instructive experience. Flight simulators and command staff exercise drivers are well-known examples of training simulation; the former teaches psychomotor skills via an immersive experience, while the latter teaches cognitive and decision-making skills by providing a realistic battlefield context. CGF systems are often used in training simulations to provide both opposing forces and supplemental friendly forces for human participants in a simulation.

Analysis. CGF systems are also used to generate entities in battlefield simulations being used for non-training purposes, such as analysis and experimentation. Analysis is the use of simulation to answer questions about some aspect of the system or scenario being simulated. Military analysis simulations are often used to assess the effectiveness of new weapons systems, force structures, or doctrine. In analysis applications simulation is used in a carefully controlled way with run-to-run initialization differences restricted to the factors under question (e.g., different weapons performance levels). Desirable aspects of analysis simulations and CGF systems used for analysis are repeatability, determinism, and the capability to isolate the cause of any particular observed effect. As an example of the analysis application, CGF systems were used in the experimental trials testing the design of a new naval surface combatant.[6] Computer generated forces are an important part of that project, providing "…friend and foe entities that make up the simulated battlespace" in which the design concepts were evaluated.

Experimentation. The experimentation application is similar to analysis, in that the simulation and CGF system is being used to answer questions, but in experimentation the questions are more open-ended and exploratory. Strict control of run-to-run differences is less important in experimentation than exploring in simulation a space of possibilities (e.g., a set of different notional weapons systems). The objective of such experimentation is "not to evaluate system effectiveness, but rather, to provide an environment and tools that will allow operators and analysts to discover new insights."[7] CGF systems have been used in large experiments conducted by U. S. military commands.[8]

Automated Learning

Learning, in general, is the acquisition of new knowledge and behaviors, usually as the result of instruction or experience. When the learner is an algorithm running on a computer system, rather than a human, the learning is referred to as machine learning or automated learning. A variety of methods and data structures have been devised for automated learning, with varying degrees of success in different applications.[9] The concern here is with the consequences of learning in CGF systems, not its implementation, so implementation details will not be addressed.

As a side note, it is worth remembering that CGF systems often include human operators. When the operator is considered as part of a CGF system, it is clear that the CGF system can (and often does) learn as the human operator becomes more skilled over time. However, it is assumed here that references in the CGF research literature to learning by CGF systems mean automated learning, i.e., learning by the non-human algorithmic portion of the CGF system.

Calls for Learning in CGF Systems

Calls for learning capabilities in CGF systems appear regularly in the CGF research literature. Some of those calls are surveyed here.

Though it does not mention CGF systems specifically in this context, a frequently cited study of human behavior modeling asserts that "learning within and by the simulations themselves" is "of potential importance" and that "Learning is an essential ability of intelligent systems."[10] Many CGF systems are placed in the "intelligent systems" category by their developers and the theme of learning as a necessary requirement for intelligent systems appears frequently. In an investigation of the application of neuro-fuzzy systems to CGF behavior representation, it is asserted that "Intelligent systems such as CGF must possess humanlike expertise in the military domain. Like a human or group of humans in a military organization they must be able to adapt and learn."[11] In a comparison of adjustable rulesets and neural nets for CGF systems, it is stated that "Intelligent systems must – among other criteria – be able to learn."[12] A list of desired capabilities for military intelligent agent architecture includes the assertion that "learning is clearly a desirable capability for a military [intelligent agent]."[13] An argument for the applicability of the recognition-primed decision-making model of human decision-making posits that "In order to realize the full benefit of a human behavioral model within an intelligent simulator, … the behavior model should incorporate learning."[14]

The perceived need for learning motivates a portion of CGF research. A report of interesting research into implementing learning within stochastic finite-state machines is introduced by statements that "learning and adaptation will become a key issue in future generation [CGF] systems" and that CGF systems lack "needed realism that relates to learning."[15] An outline of a research initiative into behavioral modeling techniques for CGF systems also sees learning as providing a "realism enhancement potential."[16] Another review of CGF research areas concurs: "Successful employment of human behavior models with the [modeling and simulation] synthetic environment requires that the models … possess the ability to integrate learning."[17]

A variety of beneficial effects are attributed to CGF learning, including cost reductions; "To enable rapid and affordable response to operational training requirements, [CGF entities] require a number of capabilities such as learning" and "By developing and inserting a learning capability into [CGF entities], the knowledge base construction expense may be reduced."[18] An examination of domain- and simulation-independent architectures for behavior generation suggests that "the ability to learn … would add to the power of the reasoning capability."[19] Learning is also expected to "… help simulations represent [CGF entities] in a more realistic manner …"[20]

Finally, learning is generally seen as a capability that future CGF systems will require. It has been observed that future CGF systems must be able to "… modify strategies based on observed successes and failures"[21] and must have "new behavior techniques to better support planning, automation, and learning."[22]

Consequences of Learning by CGF Systems

What might a CGF system learn, and what would be the consequences? This section addresses the first question by defining categories of learning-modified CGF behaviors in terms of nested subsets of possible CGF behaviors. It then addresses the second by considering how CGF behaviors in each of those categories would affect each of the three CGF application classes.

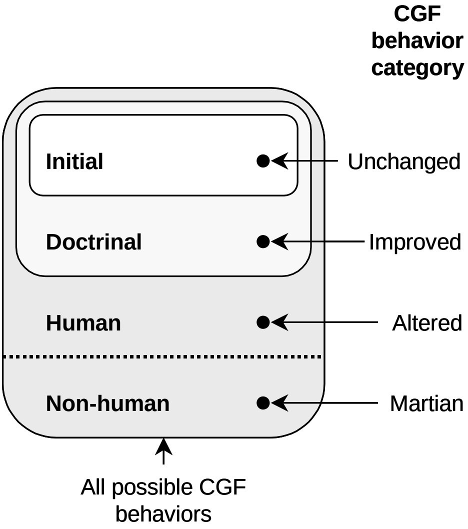

Figure 1: CGF Behavior Categories.

What Might a CGF System Learn?

While the question of what a learning CGF system might learn would be difficult to answer in detail, it can be approached at a more abstract level by considering sets of possible CGF behaviors.[23] Figure 1 illustrates a hierarchy of nested sets of CGF behaviors. The outermost, or most inclusive, set is that of all possible CGF behaviors.[24] It is partitioned into two subsets, those behaviors that are plausibly human behaviors and those that are not.[25] Within the set of human CGF behaviors there is the subset of behaviors that are within the military doctrine(s) of the force(s) represented by the CGF system.[26] Within the doctrinal behavior set there is the subset of behaviors that were initially programmed into some learning CGF system of interest.[27]

Suppose a CGF system with learning capabilities is run so that learning may occur. Its post-learning behavior can not be determined in detail without more information, but at an abstract level it can be classified into one of four CGF behavior categories based on what behaviors it has learned. If the learning CGF system has learned no new behaviors at all, its post-learning behavior is *Unchanged*. If it has learned any new behaviors from the Doctrinal set that are not in the Initial set, but no new behaviors outside the Doctrinal set, its behavior is *Improved*. If it has learned any new behaviors from the Human subset that are not in the Doctrinal set, but no new behaviors from the Non-human subset, its behavior is *Altered*. Altered CGF behavior includes behaviors not within the doctrine of the represented force. This possibility was recognized as the result of an experimental study in implementing learning for automated individual combatants, where it was observed that circumstances in the learning environment "… can cause the [individual combatant] to learn non-doctrinally correct behavior."[28]

If it has learned any new behaviors from the Non-human subset, its behavior is *Martian*.[29] Martian CGF behavior includes behaviors that are not plausibly human. Real examples of arguably Martian behavior exist. One learning algorithm presented with the task of designing a space battle fleet for a game tournament produced designs that were quite unlike any of the human-produced designs; it won the tournament.[30]

Table 1 summarizes these CGF behavior categories.

Effects of CGF Learning by Application

The effects of CGF learning can now be analyzed by considering how behavior from each of the four CGF behavior categories would affect each of the three CGF applications. Table 2 summarizes that analysis.[31] The entries in the table are coded with a "+" or a "-" to indicate an effect of learning that appears to be beneficial or detrimental, respectively, for the application. Note that in several cases there are both beneficial and detrimental effects.

Training. Learning that produces Improved CGF behavior can result in simulated entities that use better, but still doctrinal tactics, making them superior as training opponents or supplemental friendly forces to entities with Unchanged behavior. However, even in the Improved category, there are potentially detrimental consequences. As learning occurs during a sequence of training runs the CGF behaviors become unpredictable to the extent that learning changes them.

Table 1: Summary of CGF behavior categories.

CGF behavior category	Behavior changed?	Includes behavior(s) from				Description
		Initial	**Doctrinal**	**Human**	**Non-human**	
Unchanged	No	Yes	No	No	No	All behavior within doctrine of represented force
Improved	Yes	?	Yes	No	No	All behavior within doctrine of represented force
Altered	Yes	?	?	Yes	No	Some behavior not within doctrine of represented force, but all still plausibly human
Martian	Yes	?	?	?	Yes	Some behavior not plausibly human

This means that the person who has organized the training loses control of the experiences the trainees will have and cannot guarantee that his/her training objectives will be met. It could happen, for example, that the trainer wants the trainees to learn to defend against a hasty attack, but the CGF system learns that hasty attacks are ineffective and dangerous and uses some other tactic. It is also possible that trainees could face CGF opponents that become so proficient due to learning that the trainees are overmatched and become discouraged. Automated learning algorithms generally require numerous trials or executions to present the algorithm with cases from which to learn (e.g., the training sets of neural nets). Those trials can be costly to set up and execute. For example, in some CGF applications Soar's capability for learning "has not been employed because these [computer generated] forces have been expected to perform at an expert level without undergoing a potentially costly training phase."[32]

If the CGF system has learned Altered behavior, the trainees could face behavior that is not within the doctrine of the represented force; such non-doctrinal behavior may provide no training, or even negative training. If Martian behavior has been learned, the trainees could face non-human behavior, with even less expected training value.

Table 2: Summary of effects of CGF learning by category and application.

CGF behavior category	CGF applications		
	Training	Analysis	Experimentation
Unchanged	No effect	No effect	No effect
Improved	+ Improved behavior + Varying experience - Loss of training control - Cost of learning phase	+ Improved behavior - Loss of repeatability - Confounded results - Cost of learning phase	+ Improved behavior + Richer experiment - Confounded results - Cost of learning phase
Altered	+ Varying experience - Loss of training control - Non-doctrinal behavior	- Loss of repeatability - Confounded results - Non-doctrinal behavior	+ Richer experiment - Confounded results - Non-doctrinal behavior
Martian	+ Varying experience - Loss of training control - Unrealistic behavior	- Loss of repeatability - Confounded results - Unrealistic behavior	+ Richer experiment - Confounded results - Unrealistic behavior

Analysis. The consequences of learning for analysis applications seem to be generally more negative than for training. It is true that Improved CGF behavior can make the CGF entities' actions doctrinally better, providing better analytic subjects. But the run-to-run behavior changes due to learning have two significant detrimental effects on analysis. First, repeatability is potentially lost if the CGF system is learning so that its behavior changes between runs. Second, it could be difficult to determine if the run-to-run outcome differences are due to the subject of the analysis (e.g., the performance of a new weapons system) or to new behaviors introduced by the learning algorithm, thereby confounding the results. The cost of the learning phase applies to analysis as well.

The possible problems of loss of repeatability and confounded results apply to Altered and Martian CGF behavior as well. At the Altered level the analysis runs can include non-doctrinal behavior, possibly reducing its value, and at the Martian level they may be against non-human behavior, possibly reducing its value further.

Experimentation. For experimentation, learning has both potential benefits and detriments. The benefits of Improved CGF behavior apply to experimentation. Moreover, learning at any level could support richer experiments that explore a larger solution space, perhaps without requiring human intervention. A report on the use of the JSAF (Joint Semi-Automated Forces) system notes this: "The successful incorporation of learning into the JSAF entities would be a major step forward for experimentation. If the automated entities could adapt to new weapons systems, organizations, and tactics then the entire experimentation process could be performed in closed loop simulations."[33] Learning in a CGF system could be useful in "… the development of new tactics based on changing enemy weapons systems and capabilities."[34]

On the other hand, the risks of confounded results and a costly learning phase apply to experimentation as well. There is the possibility of reducing the value of experiments that include non-doctrinal or non-human behavior. The just-cited JSAF report warned that automated CGF learning could confound the experimentation results with learning that does not correspond to any possible reality, typical of behavior in the Martian category: "there is a danger that automated learning systems will optimize based on simulation anomalies rather than actual real world phenomena."[35] This is not merely a hypothetical possibility; an example of just such an artificial optimization was observed in the author's own early work on automated learning. In an experiment with a simulated robot exploring a hostile terrain, the robot control algorithm learned to avoid damage by remaining motionless as much as allowed by the rules of the simulation, which optimized the robot's survival time but was contrary to the exploratory mission of the robot.[36]

Some Valid Applications of CGF Learning

There are some applications of CGF learning that appear valid. Several of those are noted in Table 2. An application of CGF learning that seems particularly apt is the automated acquisition of knowledge as part of knowledge engineering for CGF behavior.[37]

Conclusions

Many redoubtable experts in CGF have stated their case for the needs of and benefits of CGF learning. These counter-arguments are offered with sincere respect. Nevertheless, there appear to be situations in each of the CGF application classes where CGF learning is detrimental. What can be concluded from this? First, CGF learning is potentially a valuable tool, but like most tools, it must be employed with skill and in the right circumstances to be useful. Second, additional research is needed, not only to develop and improve CGF learning methods, but also to find the applications where learning is truly useful.

Notes

[1] Computer generated forces are also known as semi-automated forces.

[2] Clark R. Karr, Douglas A. Reece, and Robert W. Franceschini, “Synthetic Soldiers,” *IEEE Spectrum* 31, 3 (March 1997): 39-45.

[3] Mikel D. Petty, “Computer Generated Forces in Distributed Interactive Simulation,” in *Distributed Interactive Simulation Systems for Simulation and Training in the Aerospace Environment*, ed. T. L. Clarke, SPIE Critical Reviews of Optical Science and Technology, Volume CR58 (Bellingham WA: SPIE Press, 1995), 251-280.

[4] A. J. Courtemanche and Andy Ceranowicz, “ModSAF Development Status,” in *Proceedings of the Fifth Conference on Computer Generated Forces and Behavioral Representation* (Orlando FL, May 9-11, 1995), 3-13.

[5] R. B. Calder, J. E. Smith, A. J. Courtemanche, J. M. F. Mar, and Andy Z. Ceranowicz, “ModSAF Behavior Simulation and Control,” in *Proceedings of the Third Conference on Computer Generated Forces and Behavioral Representation* (Orlando FL, March 17-19, 1993), 347-356.

[6] D. Ewen, D. P. Dion, T. F. Flynn, and D. M. Miller, “Computer Generated Forces Applications to a Simulation Based Acquisition Smart Product Model for SC-21,” in *Proceedings of the Ninth Conference on Computer Generated Forces and Behavioral Representation* (Orlando FL, May 16-18, 2000), 353-361.

[7] Andy Ceranowicz, M. Torpey, B. Helfinstine, D. Bakeman, J. McCarthy, L. Messerschmidt, S. McGarry, and S. Moore, “J9901: Federation Development for Joint Experimentation,” in *Proceedings of the Fall 1999 Simulation Interoperability Workshop* (Orlando FL, September 12-17, 1999).

[8] Ceranowicz, “J9901, Federation Development for Joint Experimentation.”

[9] Richard S. Forsyth, ed., *Machine Learning* (New York: Chapman & Hall, 1989).

[10] Richard W. Pew and Anne S. Mavor, eds., *Modeling Human and Organizational Behavior: Application to Military Simulations* (Washington, D.C.: National Academy Press, 1998).

[11] Gary R. George and Frank Cardullo, “Application of Neuro-Fuzzy Systems to Behavioral Representation in Computer Generated Forces,” in *Proceedings of the Eighth Conference on Computer Generated Forces and Behavioral Representation* (Orlando FL, May 11-13, 1999), 575-585.

[12] Andreas Tolk, “Adjustable Rulesets versus Neural Nets for Order Generation in Closed Combat Simulation Models,” Paper 8TH-CGF-019, in *Proceedings of the Eighth Conference on Computer Generated Forces and Behavioral Representation* (Orlando FL, May 11-13, 1999), 167-173.

[13] R. B. Calder, J. Drummey, and F. Chamberlain, “Definition of a Military Intelligent Agent Architecture,” in *Proceedings of the Eighth Conference on Computer Generated Forces and Behavioral Representation* (Orlando FL, May 11-13, 1999), 551-562.

[14] Keith O. Hunter, William E. Hart, and Chris Forsythe, “A Naturalistic Decision Making Model for Simulated Human Combatants,” in *Proceedings of the Ninth Conference on Computer Generated Forces and Behavioral Representation* (Orlando FL, May 16-18, 2000), 593-600.

[15] Erol Gelenbe, “Modeling CGF with Learning Stochastic Finite-State Machines,” in *Proceedings of the Eighth Conference on Computer Generated Forces and Behavioral Representation* (Orlando FL, May 11-13, 1999), 113-116.

[16] Denise Lyons and Harold Hawkins, “Cognitive and Behavioral Modeling Techniques for CGFs: A New Initiative,” in *Proceedings of the Eighth Conference on Computer Generated Forces and Behavioral Representation* (Orlando FL, May 11-13, 1999), 3-5.

[17] Sheila B. Banks and Martin R. Stytz, “Advancing the State of Human Behavior Representation for Modeling and Simulation: Technologies and Techniques,” in *Proceedings of the Ninth Conference on Computer Generated Forces and Behavioral Representation* (Orlando FL, May 16-18, 2000), 87-100.

[18] Martin R. Stytz and Sheila B. Banks, “Considerations and Issues for Distributed Mission Training Computer-Generated Actors,” in *Proceedings of the Eighth Conference on Computer Generated Forces and Behavioral Representation* (Orlando FL, May 11-13, 1999), 529-537.

[19] David M. Patrone and Tony Nardo, “A Domain and Simulation-Independent Architecture for Creating Simulated Object Behaviors,” in *Proceedings of the Ninth Conference on Computer Generated Forces and Behavioral Representation* (Orlando FL, May 16-18, 2000), 63-72.

[20] Susan A. Gugel, David R. Pratt, and Rafael A. Smith, “Using Q-Learning to Control Individual Combatants,” in *Proceedings of the Tenth Conference on Computer Generated Forces and Behavioral Representation* (Norfolk VA, May 15-17, 2001), 131-139.

[21] B. D. Lapin, R. L. Kolbe, J. Langworthy, V. Tran, and R. Kang, “Semi-Automated Forces that Learn from Experience,” in *Proceedings of the Tenth Conference on Computer Generated Forces and Behavioral Representation* (Norfolk VA, May 15-17, 2001), 141-146.

[22] Anthony J. Courtemanche, “Design Patterns for Computer Generated Forces,” in *Proceedings of the Eighth Conference on Computer Generated Forces and Behavioral Representation* (Orlando FL, May 11-13, 1999), 25-36.

[23] Here it is assumed that CGF behaviors can be grouped into sets and that individual behaviors can be treated as elements of sets. This assumption seems justifiable only if CGF behaviors that are generated by algorithms are considered, which can be represented unambiguously in forms like computer programs, which certainly can be elements of sets.

[24] Again, if only CGF behaviors that are generated by algorithms are considered, this set is infinite but countable so. Each behavior can be mapped to the integer that is equivalent to the binary representation of the shortest computer program that generates it.

[25] Is there any behavior that is not plausibly human? Even human behavior motivators as basic as self-preservation can be overridden by humans in combat situations. Perhaps behaviors that require cognitive computation beyond the capabilities of humans could exist; they would fall into this subset.

[26] A singular force will be assumed hereinafter to simplify the exposition. The arguments are not materially affected.

[27] Note the assumption that the CGF system was implemented correctly, i.e., that all its initial behaviors are within the Doctrinal set.

[28] Gugel, Pratt, and Smith, “Using Q-Learning to Control Individual Combatants.”

[29] The term “Martian” was deliberately chosen for its double meaning. Mars is both a non-human planet and the Roman god of war.

[30] Douglas B. Lenat, “EURISKO: A Program That Learns New Heuristics and Domain Concepts,” *Artificial Intelligence* 21, 1-2 (March 1983): 61-98.

[31] The analysis of this section, and Table 2, assumes that the training, analysis, or experimentation is occurring over a sequence of runs and that learning is “turned on” and hence can occur during that sequence. An analysis of the similar, but simpler case where learning occurs only before a sequence of runs is omitted for reasons of space.

[32] Scott A. Wallace, John E. Laird, and Karen J. Coulter, “Examining the Resource Requirements of Artificial Intelligence Architectures,” in *Proceedings of the Ninth Conference on Computer Generated Forces and Behavioral Representation* (Orlando FL, May 16-18, 2000), 73-82.

[33] A. Ceranowicz, P. E. Nielsen, and F. V. Koss, “Behavioral Representation in JSAF,” in *Proceedings of the Ninth Conference on Computer Generated Forces and Behavioral Representation* (Orlando FL, May 16-18, 2000), 501-512.

[34] Gugel, Pratt, and Smith, “Using Q-Learning to Control Individual Combatants.”

[35] Ceranowicz, Nielsen, and Koss, “Behavioral Representation in JSAF.”

[36] Mikel D. Petty, Terry J. Frederick, and J. Michael Moshell, “Experiments in Routing an Autonomous Land Vehicle with a Weakly Inductive Learning Algorithm,” in *Proceedings of the Third Florida Artificial Intelligence Research Symposium* (Cocoa Beach FL, April 3-6, 1990), 159-163.

[37] Larry Willis, Lashon B. Booker, Gary N. Bundy, and Paul G. Foley, “Overview of the Advanced Simulation Technology Thrust (ASTT) Program,” in *Proceedings of the Ninth Conference on Computer Generated Forces and Behavioral Representation* (Orlando FL, May 16-18, 2000), 101-112.

MIKEL D. PETTY is Chief Scientist of the Virginia Modeling, Analysis & Simulation Center at Old Dominion University. He received a Ph.D. from the University of Central Florida in 1997, a M.S. in 1988, and a B.Sc. in 1980, all in Computer Science. Dr. Petty has worked in modeling and simulation since 1990 and has interests in simulation interoperability and composability, computer generated forces, and applications of theory to simulation. He was a member of a U. S. National Research Council committee on modeling and simulation and is currently an Associate Editor of the journal *SIMULATION: Transactions of the Society for Modeling and Simulation International*.

Address for correspondence: Virginia Modeling, Analysis & Simulation Center, Old Dominion University, 7000 College Drive, Suffolk VA 23435; *Telephone*: 757-686-6210; *E-mail*: mpetty@vmasc.odu.edu