Proceedings of the 1996 Winter Simulation Conference
ed. J. M. Charnes, D. J. Morrice, D. T. Brunner, and J. J. Swain

MODELING AND SIMULATION ENABLING TECHNOLOGIES FOR MILITARY APPLICATIONS

Alex F. Sisti
Steven D. Farr

US Air Force
Rome Laboratory
32 Hangar Road
Rome, NY 13441, U.S.A.

ABSTRACT

While there has been much attention paid lately to the field of Modeling and Simulation as a general tool to be used in specific and well-founded application areas, little has been done to address those broad technology areas that enable such application-oriented simulations to be more easily constructed, run and analyzed. This paper deals with the wide variety of research issues in simulation science being addressed by Government, academia and industry, and their application to the military domain; specifically, to the problems of the Intelligence analyst

1 INTRODUCTION

In defining the Air Force's New Vector for Modeling and Simulation, the Chief of Staff and Secretary of the Air Force stated that "We need to expand our involvement and investment in advanced simulation technologies to improve our readiness and lower our costs today, and prepare us to dominate the battles of tomorrow" (Air Force Laboratory Ad Hoc Working Group 1996). The rapid growth and expanding role of simulation in the Air Force and throughout DoD attest to the tremendous power and potential that this technology provides to nearly every aspect of the defense enterprise. The Air Force, in parallel with the rest of DoD, is making a substantial investment in the development and use of simulation to support improved decision making, training, systems development, and testing.

Historically, modeling and simulation emphasis within the Air Force has been on the end use product, (i.e., using existing simulation technologies to develop simulations which would be suitable for defined applications). However, as our investment base grows, and as we engage in major long term simulation projects, it becomes imperative that we are aware of emerging or developing simulation technologies that may contribute to or affect these or follow-on programs. Further, if there are significant enabling technologies in development which appear to have unique applicability and high payoff for the Air Force, we should track these developments and consider investing in them to ensure their eventual applicability to our needs.

As the Air Force's Center of Excellence for Research and Development in Command, Control, Communications, Computers and Intelligence (C4I), we have a strong commitment to technology transition and technology transfer, and our Modeling and Simulation programs reflect that commitment. While our charter calls for us to apply Modeling and Simulation tools and support to our operational customers, we also feel that the most successful applications should be, and are, rooted in basic and advanced research into simulation science itself. It is our very strong feeling that as an R&D lab, bent on maximizing efficiency and effectiveness, and on avoiding duplication, such a cradle-to-grave approach to systems development and acquisition is essential (Sisti 96). This paper will describe the foundational technologies in simulation science being addressed by Rome Laboratory basic research programs, and some successful applications of that technology toward solving real-world operational problems.

2 ENABLING TECHNOLOGIES IN SIMULATION SCIENCE

There is a tremendous amount of research being performed, both within and outside of the Department of Defense, in advancing and rethinking the theoretical foundations of simulation science, and in “borrowing” the theoretical foundations from other disciplines and reapplying them to traditional simulation. The technological scope of this body of research varies greatly, as does each area’s level of maturity. The following sections describe some of the more important enabling technologies, based on the subjective metrics of degree of maturity, Air Force relevance and potential benefit to the simulation community in general. The interested reader may also refer to Appendix B in bibliographical reference (Air Force Laboratory Ad Hoc Working Group 1996) for more detailed writeups on these and other technology areas.

2.1 Hierarchical Simulation Construction

Hierarchical Simulation refers to the notion of a validated analytical hierarchy of models, consisting of four levels of modeling fidelity, ranging from most detailed (engineering level modeling) to most aggregated (theater/campaign level modeling). The basic idea of hierarchical simulation construction is to treat an engineering-level model -- say a radar - as being a "software zoom" of its more coarsely-modeled counterpart in a platform-level model; for example, an early warning radar on an aircraft. This platform-level model, in turn, can be viewed as a "zoom" of a rudimentarily modeled aircraft in a mission-level model; perhaps consisting of several aircraft in an air-to-air combat scenario. Finally, the highest level of the hierarchy is the theater- or campaign-level model, in which air-to-air combat is but one facet of the overall campaign (Sisti 1992).

If the objective of a particular simulation experiment is to assess the capability of an existing (or proposed) piece of equipment, or processing technique, obviously a detailed model of that equipment or technique is needed. Ideally (in order to assess its overall "battle worthiness"), it should be evaluated within the context of a theater-level simulation. Again, ideally (since accuracy is key, especially in a pre-Milestone 0 design alternative study), one would like for the entire system to be at the same level of detail as the engineering-level model of the component of interest. Since a simulation involving thousands of entities, each made up of many components, is not inconceivable or even rare, such a simulation is impossible today; or at least unfeasible. Fortunately, it is not even necessary. There are two alternatives being examined now, under the technical umbrella called "mixed fidelity simulation". The first – and this is probably the most prevalent approach taken by researchers -- is what we're calling "model integration". This involves actually replacing a coarsely-modeled function or entity with a more detailed version; the "software zoom" mentioned earlier. Note that this still involves mapping all the input/output associated with each version so that their interconnections to the rest of the system can be resolved. Also, some degree of Verification, Validation and Accreditation needs to be made to ensure that the codes being swapped are both modeling the same thing, albeit at different degrees of fidelity. "Model aggregation", on the other hand, deals with the idea that the level of detail -- complexity, numbers of inputs/outputs, etc. -- should be fairly consistent from component to component, throughout the theater-level simulation. But then how does that ensure that an individual component of interest -- say the radar mentioned earlier -- is modeled to the accuracy needed for a sensitivity study? The answer lies in the field of "model abstraction", described in more detail in the next section.

2.2 Model Abstraction

Model Abstraction is the intelligent capture of the essence of the behavior of a model without all the details (and therefore runtime complexities) of how that behavior is implemented in code (Frantz 1996). As there is no one "best" solution or approach, it is also one of the most diverse areas. It is as old as modeling itself, in that the intent of the modeler has always been to capture the essence of the behavior of the real-world entity or process, to whatever level of detail (complexity) he/she could afford, or was willing to wait. As a discipline, it aligns closely with, and complements, the concept of mixed fidelity simulation within a hierarchical framework. The basic premise is that the appropriate level of resolution and fidelity should be determined by the end-user requirements posed on the simulation, with the goal being to provide a representation of entities and their behaviors that is sufficiently detailed to support the intended end-use, yet lacks any unnecessary complexities. Avoiding such complexities reduces the developmental and computational requirements associated with the simulation, enables the representational focus to remain end-use motivated, and reduces the false security of "more is better".

Abstraction techniques range from lookup tables, where the entries are the outputs of many simulation runs of the detailed code; to performance curves and response surfaces; to mathematical "metamodeling" (Caughlin 1996), which tries to reduce the behavior of a model to some mathematical equation (itself a model), involving the crucial input set -- those factors to which an output of interest is most sensitive. Traditional study in this area has dealt with reduced order polynomials comprised of these input factors and, while more recent research applies another pervasive enabling technology area -- neural nets -- to represent reduced-order models; yet another involves the use of "rational functions" (Cassandras 1993).

One interesting advance in abstraction research has been the application and adaptation of the concept of "qualitative reasoning"; borrowed from the field of Artificial Intelligence. Qualitative Simulation, as this application is called, concerns itself with getting away from the idea of "exactness"; the mindset of traditional (quantitative) simulationists. Some ancillary topics of research in qualitative simulation are: fuzzy modeling, Random Set Theory, Possibility Theory, Rough Sets and Dempster-Shafer Theory, and ordinal optimization. The common factor is that all strive to represent "intermediate degrees of truth" (uncertainty) in such a way as to attain optimal answers, or ranges of answers, as opposed to an optimum answer to 10-decimal place precision.

2.3 Object-Oriented Simulation

Object Oriented programming, as a more natural and intuitive way of representing and coupling software components, is a well-founded software science paradigm. Extending the application of object-oriented techniques to the modeling (i.e., representation) and simulation of complex entities and systems, such as aircraft, sensors, weapons and C4I systems, would greatly facilitate their creation, insertion and interoperability. An object-oriented simulation enables the simulationist to more naturally represent the various types of relationships among these entities and systems, as well as their relationships with their human operators. The semantics of these relationships are particularly important, transcending well beyond what is currently supported in object-oriented programming languages today. Another area of research addresses the attempts of the simulation community to create object-oriented simulations without a clear engineering approach to building models. The emerging area of "model engineering" is as significant method in simulation science as was "software engineering" in the computer science field. Yet another potential area relating to this paradigm relates to the idea of a Virtual Model Repository, discussed later in this paper; specifically, how this repository can benefit from research pursuits in Object-Oriented database technology.

The Object-Oriented paradigm, as a computer science discipline, is quite mature, dating back over 25 years. However, the application of object-oriented principles to modeling and simulation of complex systems is still relatively new. Model Engineering holds much promise as a disciplined approach to simulation construction, and object-oriented database technology is as important and relevant a research area today as general database technology was in the 60s and 70s.

2.4 Simulation Paradigms

An important area of research in simulation science involves investigating new simulation paradigms. Whereas traditional approaches have been adequate for the body of problem-solving applications for which they've been used, the problems confronted by designers, analysts and decision-makers of today imply the need for new ways of representing the behavior of real-world entities/processes and their interactions, new ways of presenting simulation results and data, and new ways of executing a simulation experiment. The reasons are manifold; however, they can be reduced to two basic areas: in some cases, new paradigms are needed to take full advantage of advances in other technology areas (e.g., increased capabilities of hardware now allow problems of enormous complexity to be run), while in other situations, requirements have arisen due to an increased acceptance of simulation as a tool, and the user community's desire to extend its utility.

Traditional modeling paradigms (discrete-event system modeling, continuous system modeling, Monte Carlo simulations, etc.) evolved from the capabilities of the hardware, software and engineering/mathematical principles of their time. They were essentially children born of a bad marriage; that being the union between the domain expert (who didn't know how to program a computer) and the computer modeler (who didn't necessarily know the domain being modeled). However useful they had been, these paradigms are now giving way to new modeling "styles" that are more intuitive and natural (Object-Oriented Simulation, Qualitative/Fuzzy Simulation, Generative Analysis, Multimodeling/Multi-Faceted Modeling and paradigms based on Petri Nets and Neural Nets), quicker (Parallel/Distributed Simulation and Concurrent Simulation), more accurate (Hierarchical

Simulation/Mixed Fidelity Simulation) and more dynamic and interactive (Web-Based Simulation, System Dynamics modeling, Adaptive/Heuristic Simulation and Man/Hardware-in-the-Loop Simulation).

2.5 Distributed Processing/Distributed Simulation

Distributed Processing, as it relates to modeling and simulation, originally was employed as a way of speeding up simulation run time, by distributing the functions of a simulation among separate processors. This approach -- a conscious and deliberate attempt to break apart an existing piece of code and force a distribution -- has since given way to a new tact; that being the networking and interoperation of already distributed models, simulations and simulators. Starting with the SIMNET project in 1983, and continuing on with the ideas of Distributed Interactive Simulation (DIS) and High-Level Architecture (HLA), distributed simulation is an enabling technology that has only begun to show its tremendous potential in effecting the synthetic battlefield environments envisioned by the highest-level military planners and decision-makers.

2.6 Simulation Support Tools

In the classic sense, simulation has long been viewed as comprising three basic phases: simulation setup, simulation execution and post-run analysis/reporting. There has been considerable effort and money expended on the latter two phases; however, simulation design and setup as a technological discipline has gone virtually ignored by the research community (or more specifically, the research funding). As we note regarding most aspects of technology insertion, there have been some piecemeal and ad hoc improvement; e.g., graphical user interfaces and code-generating front-ends. For the most part, however, there is much to be gained by a concerted effort to apply some of the technology areas that are currently being pursued in research circles in the simulation community. The following sections discuss some of the more prevalent enabling technologies as they relate to designing, configuring and managing a simulation experiment.

2.6.1 Model Management Tools

Much has been written lately regarding the need for a "repository of models". There seems to be a general consensus of opinion that what is envisioned is not so much a physically manifested library, but rather a Virtual Model Repository (VMR); given the requirement to reuse as much legacy (read: validated) code as possible, and the fact that these models are independently owned and maintained, at sites distributed around the country. This Virtual Model Repository would consist of validated simulations, models, model components (at varying levels of resolution), data and inter-model coupling relationships. These pieces, either through conformance to model/data format standards or by judicious adherence to "wrapper" constructs, would be selected, instantiated and appropriately coupled to build the desired simulation; all by the Model Management System. In fact, a Model Management System in its most elemental form, is to a Model Base (our VMR) what a database management system is to a database. It eases, and to some extent automates, the user's ultimate model selection.

2.6.2 Resolution/Validation Management Tools

These tools assume the necessity for the proposition, approval and adoption of a definition of discrete levels of fidelity (the simulation community freely uses the term 'fidelity' interchangeably with 'resolution'); which currently are undefined. Basically, this would be a necessary precursory stage to a model's inclusion in the Virtual Model Repository (VMR). In order for the Model Management System to be of any utility, each component in the VMR must have a specific level of resolution associated with it, and it must have been validated as an accurate representation of the behavior of its real-world counterpart, at the prescribed level of fidelity.

There is a run-time aspect to this technology area as well. Earlier in the paper, we introduced the concept of mixed fidelity simulation as it relates to hierarchical construction. In such a framework, we may wish to construct large-scale simulations in which its entities may be modeled at differing levels of resolution. While that section primarily refers to the general paradigm of composing a simulation from building blocks of different (but appropriate) levels of fidelity, most prototype implementations of hierarchical simulation systems do not allow for dynamically changing resolution of entities, once configured. Dynamic Resolution Management represents an excellent research topic whose time has come. The essence of the concept is that at certain stages of the simulation -- either temporal or spatial -- it makes sense to transition from modeling an entity or process at one level of fidelity to representing it at another. The need for such a transition, called a "software zoom", could be either scripted or, more likely, could be dynamically deduced using a new concept called Significant Event Detection (Air Force Laboratory Ad Hoc Working Group 1996).

2.6.3 Experimental Frame

Closely aligned with Model Management Systems, but broader, is the idea of the Experiment Frame (Zeigler 1990). The vital component of the Simulation setup phase, an Experiment Frame enables a simulation designer or constructor to translate the objectives and issues to be addressed by a modeling effort into conditions under which experiments will be run with a model or real system. This technology would allow the user to completely set up an experiment involving multiple executions of a single simulation with parameters changed for each run, or execution of multiple simulations with varying parameters.

2.6.4 Simulation Design

Simulation Design (Fishwick 1995) is the overarching technology area which includes the concepts of experimental frames, Model Management Systems and other sub-phases of the pre-simulation run stage. As does its construction industry counterpart, it speaks to the notion that one can't jump into construction (in our case, a simulation) without a well thought-out blueprint.

2.6.5 Data Acquisition and Insertion

Arguably the "oldest" new idea, this technology area refers to users'/analysts' desire to imbue their simulations with live, or real-time, input data. Some examples of this are: scanning circuitry schematic drawings as a way of setting up appropriate simulations; scanning photographs, documents and other forms of imagery; or driving existing flight simulations with real (or recorded) flight path/waypoint data.

3 MODELING AND SIMULATION FOR THE INTELLIGENCE ANALYST: THE CONCEPTS ARE APPLIED

The scientific and technical intelligence (S&TI) community provides an excellent audience to introduce these developing technologies for several reasons. Intelligence analysts are typically experienced in the use of automated systems and, quite often, modeling and simulation applications. Given these user characteristics, the simulation development community is free to reduce emphasis on building initial prototypes that are significantly robust. For example, input parameter checking, extensive help utilities, and intuitive GUIs need not be implemented to demonstrate the operational suitability of the underlying simulation science. While these factors are necessary for final user acceptance, their development should not be the subject of scarce basic research and development dollars.

The S&TI organization most familiar to the authors is the Air Force's National Air Intelligence Center (NAIC) located at Wright-Patterson Air Force Base in Ohio. NAIC is responsible for analysis of current and projected foreign aerospace technology trends and capabilities. This broad area of responsibility dictates the need for group analysis and assessment. Consider, for example, the common scenario where the intelligence produced by the office responsible for analysis of a missile seeker is passed to the office with responsibility for the air-to-air missile itself. The flow of information continues to the office charged with analysis of an entire aircraft , and from there on to the office responsible for air-to-air engagement analysis. Each stage of analysis requires a different level of modeling fidelity, ranging from the detailed engineering level to the mission (few on few) level. All this must be performed while maintaining traceability back to the original intelligence source. The application of techniques such as hierarchical modeling and simulation support environments seem quite natural to the process described above.

Another interesting aspect of dealing with NAIC is the organization's utilization of support software. Examples include geographic information systems, relational databases, solid geometry modelers (e.g., CAD), automatic text retrieval systems, and desktop publishing packages. Modeling and simulation applications must eventually be seamlessly integrated with these support applications if they are to achieve any level of user acceptance. The opportunity to observe and understand the entire analysis process is critical in order to make decisions on which simulation technologies are mature enough for operational use.

3.1 Case Study: Modeling of an Integrated Air Defense System

In practice, the utilization of just one of the enabling technologies discussed earlier in this paper will rarely occur. Instead, a marriage of these techniques will be necessary in order to provide a simulation capability suitable for operational use. An example of this is found in an integrated air defense system (IADS) modeling effort currently under development for several US Air Force agencies. For the uninitiated, an air defense system consists of the components necessary to detect, track, identify, engage, and disable incoming airborne adversarial weapons systems. That statement alone should scope the complexity of modeling such a system; however, the problem is compounded when one considers active versus passive detection systems, radar jamming, alternate tracking and fusion algorithms, surface-to-air-missiles versus interceptor aircraft for engagement, and the command and control infrastructure. The approach to modeling these systems involves the application of model integration and abstraction techniques, petri nets, and a host of simulation support tools.

Already in existence are simulations capable of performing detection and engagement. Our challenge was to model the C2 components and to integrate all these codes to provide an end-to-end modeling capability. Existing models selected for our use are the FPG flight path generator, the Advanced Low Altitude Radar Model (ALARM), and the Ground Radar Clutter Estimator (GRACE). The Air-to-Air System Performance Evaluation Model (AASPEM) and the Enhanced Surface to Air Missile Simulation (ESAMS) will eventually be used for air to air engagement modeling and surface to air missile effectiveness modeling, respectively.

Model integration was achieved through the Digital Integrated Modeling Environment (DIME). The DIME is a simulation support environment that facilitates the integration of existing models. Inter-model communication in the DIME is achieved through the concept of a cradle. To integrate a model, the developer must develop a cradle that serves as the interface between the model and the DIME environment. The DIME also provides additional support functions such as report generation, plotting functions, and mapping.

The opportunity to introduce emerging technology was seized in the development of the command and control model. The underlying C2 model simulation is based on a Petri Net engine called Modeler. C2 nodes and data links are represented within Modeler. Detailed algorithms for tracking, fusion, and, eventually, weapon control are performed within a set of submodels. Representation of various C2 node types as objects allows for arbitrary configuration of node combinations and therefore, provides a means to represent any air defense system.

To date, DIME only supports non-reactive modeling (i.e., each model executes sequentially without interaction with other models). In our system, the flight path generation is performed first, ALARM executes once for each flightpath/radar pair, then the C2 model executes. Long term goals are to eventually allow dynamic modeling whereby models (including humans) interact during system execution. To this end, the C2 model was designed so as to not prohibit the man in the loop. An open systems man-in-the-loop version of AASPEM is also under development. Dynamic modeling also offers the chance to apply model abstraction and distributed processing techniques; especially where ALARM is concerned. The independent executions for the flightpath/radar pairings present a clear opportunity to distribute the processing load.

Model abstraction opportunities have been identified as applicable to the ALARM ground based radar model (Frantz 1996). For example, ALARM facilitates an *explicit assumption* concerning components of the returned signal strength. The computational complexity of the model is reduced by explicitly assuming that there is no significant contribution to the returned signal from individually selectable elements of the target such as rotor blades, jamming, clutter or atmospheric attenuation. Without consideration of these factors, the element of the code responsible for their calculation is not invoked. *Causal approximation* offers another abstraction opportunity. For example, a simplified model of standoff jamming could substituted for the more detailed model currently in place, particularly if the model is being used to compare alternative radar designs (rather than predict absolute performance).

4 SUMMARY

This paper endeavored to familiarize the reader with current topics of interest to researchers in the simulation community, and to demonstrate some selected Air Force programs where this research has been applied. We feel that of all the enabling technologies, Model Abstraction is probably the most important enabling technology area in simulation science today. The requirements for mixed fidelity composition, for representing models at varying degrees of resolution and for reducing the complexity of monolithic legacy code are also widely stated by all the services. There is much research under way, using a wide variety of approaches; however, it is a fairly immature discipline, with few actual prototype implementations in place. Nonetheless, the potential benefits of such research to the Air Force and the rest of the DoD, and to the simulation community in general, are enormous and far-reaching.

REFERENCES

Air Force Laboratory Ad Hoc Working Group; "Analysis Of Emerging Enabling Technologies With High Potential To Support The Use Of Modeling And Simulation In The Air Force (Interim Report). 1996.

Cassandras, C. G. 1993. "Discrete Event Systems - Modeling and Performance Analysis", Aksen Associates Publishers.

Caughlin, D. 1996. "Metamodeling Techniques and Applications", Rome Laboratory Technical Report.

Fishwick, P. A.. 1995. "Simulation Model Design and Execution - Building Digital Worlds", Prentice Hall Publishers.

Frantz, F. 1996. "Model Abstraction Techniques", Rome Laboratory Technical Report.

Sisti, A. F. 1992. "Large-Scale Battlefield Simulation Using a Multi-Level Model Integration Methodology", Rome Laboratory Technical Report.

Sisti, A. F., and Trott, K. C. 1996. "SYNBAD: A Distributed Interactive Simulation (DIS) Environment for C3I Capability Assessment", Rome Laboratory Technical Report.

Zeigler, B. P. 1990 "Object-Oriented Simulation with Hierarchical, Modular Models", Academic Press/Harcourt Brace Jovanovich.

AUTHOR BIOGRAPHIES

ALEX SISTI is an Electronics Engineer with the Intelligence Technology Branch, Signals Intelligence Division, the Intelligence and Reconnaissance Directorate at Rome Laboratory, Rome, NY. He has been actively applying his modeling and simulation research interests in a variety of Air Force programs since 1982. He received an A.A.S. degree in Accounting in 1975, a B.S. in Computer Science in 1982 and earned his M.S. in Computer Engineering from Syracuse University in 1989.

STEVE FARR is an Electronics Engineer with the Intelligence Technology Branch, Signals Intelligence Division, the Intelligence and Reconnaissance Directorate at Rome Laboratory, Rome, NY. Steve received his B.S. from the University of Connecticut in 1983, an MBA from Rensealear Polytechnic Institute in 1986 and an M.S. from Syracuse University in 1993. His professional responsibilities include the application of emerging technologies to specific modeling and simulation requirements. He has been actively applying his modeling and simulation research interests to numerous Air Force programs over the last 12 years.