

ISSUES IN MODELING AND SIMULATION: POLICIES AND TECHNOLOGIES

David R. Pratt
Drew W. Beasley

Joint Simulation System Joint Program Office
12249 Science Drive, Suite 260
Orlando, Florida 32826, U.S.A.

ABSTRACT

The use of Modeling and Simulation (M&S) within the Department of Defense (DoD) has undergone a major change over the last several years. Unfortunately, the vast majority of the tools have not kept up with these changes. This is partially related to ongoing chasm between the academic and production factions within the M&S community. Due to the diverging agendas of the factions, there is little dialogue between them. This results in the needs of the production community not being met by the academics. Likewise, the advancements made by the academic community are often not included in the production systems. In this paper we will attempt to convey the background and issues facing one such production program, the Joint Simulation System (JSIMS), in an attempt to help bridge the chasm.

1 WARFARE IS EVOLVING

Modern military operations in the twentieth century have gone through three phases of evolution. The first of these started with World War II, the Attrition Warfare era. This phase was characterized by massed armies fighting discrete battles where the ones left standing were declared the victors. In the late 1970's and early 1980's a new doctrine of Maneuver Warfare came into existence. Partially based on the lessons learned in Viet Nam, the determining factor was the ability to mass the forces at the critical time and place to defeat the opponent. While this phase is still with us, we are entering a period where the military has a full spectrum of missions. While the United States military has always been used in non-traditional roles, it is now used in a much broader range. New missions, such as peace keeping, humanitarian assistance, and disaster relief, are stretching the envelope of the training of the armed services.

While the missions and the methods of the military have changed drastically, the simulations to support them have fundamentally remained the same. For the most part, they are large, monolithic, cumbersome, and based

on simplistic attrition algorithms. As we progress in the highly dynamic world of the twenty-first century, the simulations used by the military must be able to reflect the missions the military performs. To accomplish this, the Department of Defense is undertaking the development of a suite of new models. One of which, JSIMS, will be used as an example in the paper.

2 DISTRIBUTED USERS AND DEVELOPERS

While the recent draw downs and base closings have reduced the number of locations the military is based, they still tend to be based by branch of service. This complicates and reduces the services ability to train together. To do so, they have had to travel to central locations capable of hosting the exercise. Very often this has been the driving cost of an exercise. Likewise, due to teaming, skill mix, and economic considerations, the developers of new systems are no longer co-located. For example, JSIMS has developers located in Boston, Orlando, San Diego, Norfolk, Huntsville, and Washington. The challenge for both the military and the developers is to work efficiently and effectively in distributed environments.

Technology has to provide the means to distribute much of the required information during systems development and execution. While this ability to establish a virtual presence has greatly reduced the need for travel, much work still needs to be done to refine and understand the new paradigm.

3 SYSTEM DEVELOPMENT PROCESS

Traditional software development process has used the waterfall type of development process. Each step logically follows the other with the products coming down the waterfall. Experience has shown that this is not an effective way of building systems that are not fully understood on the outset. These types of systems, which include virtually all simulation systems, lend themselves

to an iterative developmental approach. Each iteration is divided into roughly two phases. The first of these is the specification of a conceptual model. This model represents a formalism of the real world. Since the entire world will not be modeled, the system requirements are used to help filter what is included.

Once the conceptual model is built the requirements are again used to determine what is implemented. The implementation is then tested and fielded. While this is going on, the specification of the next phase can be underway. If the parallelism and feed back are managed correctly, several iterations of the system are used to zero in on the true requirements. By building smaller pieces and using the iterations to help, it is possible to develop a system that satisfies the user's needs, even if they were unable to articulate them up front.

4 NEW PRACTICES AND EXISTING POLICIES

Much like the changes in the art of warfighting, software development has undergone a significant change in the last several years. The software has gone from a hand crafted, functionality based, decomposed set of procedures to a tool based, object oriented set of frameworks and classes. While this has been a great help in the development of many systems, the fundamental development policies have not kept up the new technologies.

The two primary Department of Defense Military Standards for software development (MILSTD 2167A and 498) were developed to support the waterfall development paradigm for procedural programs. While some may debate the merits of the standards, they were mandated for production software development efforts. Even with tailoring they do not adequately support the tool based, iterative object oriented development process. This is true for both the style and content of the artifacts required by the standards. What is needed is a new set of process models that more accurately reflect the state of modern software development.

Perhaps the single hardest aspect of a simulation system to quantify is the security portion. The reasons for this is simple: the security policies are not clearly defined, there are new ways to penetrate the newly developed protection mechanism, it is easier to say no than to make something work, and the policies take a long time to create. As a result, the policies are often out of date, arbitrary, and cumbersome. To compound these problems, new data distribution standards can create violations of existing security policies. An example of this is the publish / subscribe mechanisms in the High Level Architecture (HLA). By monitoring the subscriptions, it is possible to determine the capabilities of a system being modeled. This is commonly called a backdoor. What is needed is a way to do modeling of security

policies to determine the impact of the system development and run time capabilities.

Due to the size and complexity of many of the simulation systems, it is not uncommon to have a system in development between three to seven years before it is fully fielded. In this timeframe, several generations of technology have elapsed. The performance perimeters of the original specified system will have changed while the hardware that was originally specified is no longer on the market. At the same time, it is very difficult to risk the program on projected advancements that may not come. What is needed is a better way to manage technology insertion

5 COMPONENTS OF A SIMULATION SYSTEM

A modern simulation system, such as JSIMS, can be divided into eight major components. Each of these can further be divided into subcomponents. It is these subcomponents that can be brought together at compile, run, or execution time to satisfy the needs of the exercise. The first of these components is the Object Services. This serves as the backbone on which the other components communicate. The system representations are made of two subcomponents, the modeling framework and the Mission Space Objects (MSO). The modeling framework provides a standard foundation for the development of the detailed representations of the MSO. The representation of the Synthetic Natural Environment (SNE) is also in the MSO subcomponent. The third major component is the user interface. In an effort to "Train As You Fight," the primary user interface mechanism is the user's actual Command, Control, Communication, Computer and Intelligence (C4I) system. To use these systems an interface set of software must be built to convert between the two message formats. Likewise, since no program can afford to build everything that is needed for all exercises, an External System Gateway (ESG) must be built. The primary purpose of the ESG component is to isolate the changes needed to interoperate with other systems.

The next two components are both databases, but they serve different purposes. The Common Data Infrastructure (CDI) deals with the runtime data generated in an exercise. This data is used to restart the system and to evaluate the exercise. The Common Database is used to store the static data that is needed to help compose the simulation for the exercise and generate the scenario. The exercise generation and evaluation tools are part of the next component the Life Cycle Applications. Every system has a set of tools to aid the users. These are needed to help hide the complexity and details of the implementation of the users. The final component is the Security Gateway.

6 TECHNOLOGY NEEDS

While there are many areas of technology development that could benefit the new simulation systems, this section will focus in on four of them.

6.1 Component Based System

As discussed above, JSIMS is based upon largely separable components that can be composed to create the simulation needed for a given exercise. This represents a fundamentally new technology. As such, there remains a significant amount of work in the determination of how to make this type of system efficient and extensible. Furthermore, the development of new components to represent new and emerging system is an area ripe for research.

6.2 Resource Reduction

Perhaps the biggest technical challenge facing JSIMS is that of resource reduction. Currently, exercises take between six and eighteen months to prepare. A ratio of one support person for every trainee is considered outstanding. The computer resources to run the existing simulations are so large they only exist at a few selected sites. Clearly, if simulations are going to become part of the mainstream training cycle, these resource requirements will have to be severely reduced. Some users have envisioned system that can develop an exercise in less than ninety-six hours without support personnel and run on commonly available hardware.

6.3 Synthetic Natural Environment (SNE)

One of the most resource consuming MSOs and the greatest impediment to interoperability is the representation of the SNE. While the level of detail needs to be consistent with the task being performed, there has to be a consistent representation of the SNE across the entities in the simulation. For example, a simulated Marine walking across the terrain needs a higher fidelity representation of the terrain than the aircraft flying overhead. However, if the Marine can not see the plane, the pilot better not be able to see him. To complicate matters further, there has to be a consistency across the environmental domains. So that when the Marine is getting wet and muddy in the rain, the plane's visual sensors are degraded.

6.4 Computer Generated Forces (CGF)

This paper started out with a very brief discription of how warfare has changed, it ends with the same issue.

However, this time the concern is how to rapidly generate the new and emerging behaviors. The miliariy is going to continue to evolve. If it is going to stay viable, the simulation system must also continue to do so. This can only be done if there are ways to effciently and effective capture and encode the new requirements.

While the modeling of the physical process of the weapon systems in the battlespace can be modeled adequately, the modeling of the cognitive and physiological processes of the humans still does not pass the Turing Test. This deficiency is very noticeable in the system that require automated staff and command echelons modeling.

The modeling of an integrated national infrastructure is also an open issue. This is largely due to the second and third order effects that are hard to capture. For example, a power substation might be destroyed. As a result a communication node ceases operation. This prevents a message from getting to a unit in the field. This unit is then cut off and does not attack at the prescribed time. This results in a hole in the lines that can be exploited for a counter attack.

7 SUMMARY

There are a large number of reasons why the academic and production communities have not worked together efficiently in the past. These include:

- Not enough money
- Not enough time
- Too applied
- Too theoretical
- Success in niche does not translate into universal application
- Too high of expectations
- Too many stove pipes

Most of these are simply excuses to continue the status quo. However, in this era of severally reduced budgets, if the two communities do not work together to help solve the problems facing modeling and simulation as a whole, both are going to suffer. It is to be hoped that, this paper has provided some insight is the structure and problems facing a major simulation development program.

More information on the programatics and the challenges facing JSIMS can be found on the JSIMS homepage at <http://www.jsims.mil>.

AUTHOR BIOGRAPHIES

DAVID R. PRATT is the first Technical Director of the Joint Simulation System (JSIMS) Joint Project Office in Orlando, Florida. He holds this position while on sabbatical from the Department of Computer Science, Naval Postgraduate School (NPS) in Monterey, California. As

the JSIMS Technical Director, his responsibilities include the overall technical direction of the program and the investigation of applicable technologies relevant to large scale constructive simulations. Dr. Pratt holds a Ph.D. and M.S. in Computer Science from NPS and a BSE in Electrical Engineering from Duke University.

DREW W. BEASLEY is the Program Manager for the Joint Simulation System (JSIMS) Joint Project Office in Orlando, Florida. CAPT Beasley, USN has been the Commanding Officer of two ships of the line, the USS Pegasus (PHM 1) and the USS STUMP (DD 978). He has had numerous program management and policy level positions with in the Department of Defense. Captain Beasley's academic achievements include a BS in Oceanography from the United States Naval Academy, graduate of the National Defense University, Industrial College of the Armed Forces, and a Master of Public Administration degree from the George Washington University. He is an inducted member of Pi Alpha Alpha, the National Honor Society for Public Administrators.