Proceedings of the 1990 Winter Simulation Conference
Osman Balci, Randall P. Sadowski, Richard E. Nance (eds.)

DEFENSE SCIENCE BOARD RECOMMENDATIONS: AN EXAMINATION OF DEFENSE POLICY ON THE USE OF MODELING AND SIMULATION

Barry M. Horowitz

The MITRE Corporation
Bedford, Massachusetts 01750

ABSTRACT

Over the last several years, models and simulations have been used increasingly to support the development, test, and evaluation process for Department of Defense systems, a practice that is expected to continue. The last ten years have seen dynamic growth in the capabilities of computers and networks, the underpinnings of digital simulation. As a result, the Defense Science Board Task Force on Improving Test and Evaluation Effectiveness was requested to look at ways of improving the use of modeling and simulation as tools in the test and acquisition of Defense Department systems. This paper discusses the conclusions of the Task Force, along with recommendations on the role of simulation in the test and acquisition process. It should be noted, however, that the paper does not represent an official position of the Defense Science Board, and contains only the opinions of the author.

1. INTRODUCTION

Simulation is becoming a more valuable and more widely used tool throughout the Department of Defense. We can expect this trend to continue and perhaps even accelerate in the future — simulation is an efficient way of performing a number of different tasks, and will become even more valuable in light of declining defense budgets. Political reasons also come into play; for example, as the current changes in Europe continue, there is less and less support for the war games and battle simulations that we used to run in the past. In the future, we will run the games using simulation rather than driving tanks through the German countryside.

Simulation currently is benefiting from the increasing capability of computers. As processing power and memory simultaneously become faster and cheaper, we can begin to simulate complex situations that we could never before address.

As computers become more and more capable, there will be many new roles for them in simulation. These new roles are not, however, the subject of this report — the Secretary of Defense requested us, the Defense Science Board Task Force on Improving Test and Evaluation Effectiveness, to examine the role of simulation in *testing* and *acquisition*.

The panel feels that the amount of testing currently being performed is somewhat less than generous, and that simulation offers a tempting, though inadequate, substitute. The most effective role for simulation is as a supplement to testing, not as a replacement. For example, simulation can highlight areas of sensitivity in a program and indicate where testing ought to be most rigorous. Testing and evaluation, when guided by simulation, provide more significant data earlier in the program, resulting in more efficient development and tighter management.

The overriding issue for the effective use of simulation is credibility. One inaccurate simulation can have a significant adverse effect on a project, and is remembered even though it was preceded by a hundred accurate simulations that were helpful.

2. THE TEST AND ACQUISITION PROCESS

In the course of our study, we discovered that in order to make useful suggestions on modeling and simulation, we had to make corresponding recommendations in the areas of test and acquisition. Accordingly, we broadened the scope of our activity to include all of these processes.

We recognized that over the last ten years there has been a continually increasing emphasis on the use of special government test organizations, independent of acquisition organizations, to define, conduct, and evaluate the results of operational tests on newly developed systems. This has been done to improve the acquisition process by adding confidence to the production decisions — those buy/no-buy decisions that heavily weigh operational test data. A significant side effect of this emphasis on the use of operational testing has been a corresponding shift in the development community's outlook on the overall role of operational testing. This change has been to de-emphasize the use of operational testing as a learning tool during development. (Operational testing often was something the developer did while designing a system to better understand the complicated interrelationship between the specifications for a system, the design for a system, and the ultimate operational utility being sought.)

Why does this correlation between operational testing and credibility occur? The program manager sees that his or her job is simply to develop a system. A user substantiates the utility of the system both before and during its development, and an independent organization evaluates its utility once it is developed; thus, the program manager's role is to be a provider of the system with no specific requirement to consider the utility of the system. This forces us to consider the consequences of this de-emphasis on operational testing as a learning tool. This paper includes several examples indicating that the consequences are negative, and are serious enough for us to make a number of recommendations to increase the use of operational testing as a learning tool during development.

We recognize that operational testing during development is expensive, and that it does not receive funding unless the value is clear — it is a question of knowing which tests are worth running. It is that role we advocate assigning to simulation — namely, conducting simulations to help identify and focus on areas of concern related to the ultimate utility of the product or system being developed. When the concern is sufficient, operational tests should be targeted at the issues raised by simulation. The method of doing this is discussed later in the paper.

Another question arises about simulation credibility. Every developer has had experiences with simulations that produced inaccurate results. If simulations are to be more widely used, what is the relationship between credibility and the production decisions made by the Defense Department?

There are many other very high-value decisions made prior to production decisions that do not have operational tests as their bases, but rely instead on simulation. How does a decision-maker know whether a decision is based on a bad simulation or a good one? Corresponding questions have been asked of our panel: Should a central organization be created in the government to accredit simulations that are permitted to be used in decision making, similar to what has been done with operational testing — that is, bring in an independent team for greater confidence? Should the government manage the distribution and re-use of standard simulations across a wider segment than might otherwise use them? Clearly, the issue of credibility is the most important aspect of simulation.

2.1 The Acquisition Process

The acquisition process consists of the requirements, development, and operational test phases (refer to figure 1).

The requirements phase is the period during which people in the services generally wish to improve their capability to perform a given mission, and are imagining how some new system or new technology can make that possible. To do that, they must extrapolate their past experience in military operations with their expectations of future systems and technology to bring about some vision of how a new

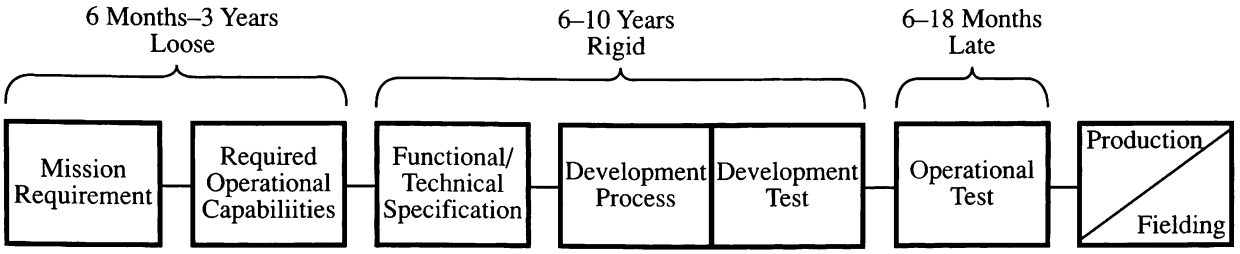

Figure 1. The Standard Defense Acquisition Process

system can make them more capable. This involves making assumptions, such as what the threat will be, what our doctrine (and that of the enemy) will be, what the procedures for using this new capability will be, what the deployment of the new system will be, and so on. These predictions have to be made in conjunction with the extrapolation of system utility.

During the requirements phase, opinions of the user and development communities can fluctuate significantly; sometimes operations research simulations substantiate the level of utility before proceeding with development. In the sense that it requires a great deal of prediction and extrapolation, the process, by its nature, is imprecise. It is generally not the result of poor or inadequate work — there is simply little reliable information available to make predictions about our own forces, let alone those of the enemy, at this stage.

When the decision to proceed with development has been made, the development phase begins with the creation of a technical specification for the product to be developed. The technical specification is created by two groups: an operational group, which had the vision in the beginning, and a more technically oriented group that is going to be responsible for development. Through a process of further extrapolation — extrapolation at lower levels concerning detail of the technology and about what might be the coupling between the specification and operational utility — a specification is born. By its very nature, it is a process prone to error, but, again, not human error. It is simply that the specification generation process involves even greater levels of prediction and extrapolation, adding to the uncertainty of the loose process upon which it was founded.

Once the specification has been created, it becomes the basis for a contract that must be rigorously managed. The project then shifts from the looser, more error-prone stage to a stage that has to be managed rigorously. This is the nature of contracts — they are specific and binding. The development process lasts many years; a program manager who does a good job is one who keeps programs stable through rigorous management.

The program then moves into the final phase, operational testing. At this point, an independent group evaluates the proposed, contracted system to verify its utility and production worthiness. There are, in fact, three evaluations taking place in this phase. One is an evaluation of the equipment itself; the second is an evaluation of the early work done by the planners (who imagined what the utility would be if such a system were built). The third is an evaluation of the process that translated the vision of those operational planners into a specification, which itself is subject to error. We make three evaluations, two of which could have been made years earlier if the equipment had been available. In the sense that two of the three evaluations have had a very long delay without much additional work, they are very late.

The tests are also late in that the cost of discovering a problem in this stage of acquisition is at its highest. If the program is cancelled, the money expended on development is lost. If a decision is made to rehabilitate the system, the cost is extremely high, because, in addition to doing the redesign work for the product, there are also all of the support costs associated with changing drawings, changing support equipment, and so on.

When developers do find a failure in operational testing, this failure is referred to as a surprise, because it is hard to imagine that someone would continue developing a system for so many years in anticipation of failure at the end. As everyone knows, in the Defense Department surprises are very bad — in addition to the surprise associated with the particular program, the credibility of the whole acquisition process is questioned. And when the process itself lacks credibility, it has a gigantic side effect, making everything the Department of Defense does seem negative and inefficient. It is, therefore, extremely important from the panel's point of view to avoid surprises, both for the direct cost to the system in question and the credibility loss to the process as a whole.

2.2 Unanticipated Results from Operational Testing

The first kind of surprise discussed is the *change-in-assumption* surprise. This is where early planners make some assumptions that lead them to believe the system or technology they want to advance will be useful; these assumptions include the threat, the deployment of the system, the environment it would operate in, and so on. By the time ten years of development have gone by, the threat, the deployment plans, the key features, or some other basic assumption has changed. Then the operational test determines that this change is crucial and the utility of the product has decreased dramatically, to the point where production is inadvisable.

A good example of this is the Division Air Defense system (DIVAD). DIVAD was originally conceived by the Army in the early 1970s, and development started in 1977. Its purpose was to protect the moving army from close air support attacks by fixed-wing aircraft (the primary threat) and from stand-off helicopter attack (the secondary threat). When the program was initiated, the stand-off helicopter threat was perceived as a three-kilometer-range stand-off weapon, so the designers of the DIVAD decided on a firing range of four kilometers.

DIVAD development proceeded through 1985, and during that time, the threat changed in two ways: the helicopter became the primary threat, and the range of the helicopter's stand-off weapon increased to six kilometers. The operational test determined that the firing range of the DIVAD was inadequate, given the extended stand-off range of the helicopter threat. The result was that the program was cancelled after a very large investment.

It is not as if the development community did not know the threat was changing. They did, but they argued for many reasons that it was still logical to develop the DIVAD with its four-kilometer firing range. The point is not that we should have produced DIVAD, but that there was no need to wait until the end of the program — and a consequent large expenditure — to decide that the change in assumptions was crucial. Perhaps analysis by simulation could have led to an earlier decision.

The second type of surprise is the *measures-of-effectiveness* surprise. This is where the early planners had some way in which they expressed utility — if the system could perform a specific task, then it would be considered useful. During the development process, however, an independent team runs the test with a different set of measures — different enough so that what seemed a very useful system no longer appears to be useful at all. The result is that the system does not meet the new measures of effectiveness and is cancelled.

One particular version of this situation occurs when one person says, "If the system under evaluation results in a capability that is better than anything in the field today, and I see no other way of getting it in

the near future, then the system will be acceptable to me." A different person says, "No, that's not good enough — I demand a definite level of utility, and unless the system meets this level, it will not be useful at all."

A good example of this case is Aquila, another Army development. Aquila was an unmanned aerial vehicle intended to carry sensors that would enable the Army to take advantage of the extended firing range of the artillery it already owned. These weapons (such as the 155mm howitzer and the Multiple Launch Rocket System) can fire at targets up to 20 kilometers away, yet they have no way of detecting targets at that distance. Aquila was to have television and infrared sensors, and a laser designation system to find and designate targets for those weapons.

In 1974, the original planners said that the system would be useful if the sensors could see half the targets in their area of vision and if, when targets were observed, the weapons could exploit the observation and actually destroy the targets 85 percent of the time. If, in addition, Aquila would not be difficult to use, then the planners would judge the system's capabilities to be of significant utility to the Army. The program went on from 1974 to 1987, when an operational test was run. Aquila, at that time, included only the television sensor, not the full capability; there was confusion during the test resulting from a lack of experience operating unmanned vehicles. The test was run and the system generally satisfied all the original measures of effectiveness. Nevertheless, the decision makers determined that the system was not good enough to warrant production and cancelled it. Large amounts of money already had been spent.

The test report provided no specific indication as to what was good enough. How could we have paid for 13 years of development, with the designers and planners having a vision of what was good enough, and then arrive at a decision point where it was decided that the system was inadequate? Why were there no substantiated measures that the defense community, as a whole, had adopted? Perhaps conducting analyses and simulations early in the development phase could have helped.

In addition, it is interesting to note that many senior military people still believe we should produce Aquila — in fact, there is nothing in development to give extended range to the Army's long-range weapons.

The third type of surprise is the *lack-of-maturity* surprise. There is always a natural tension at the end of the middle stage of development, when the time has come to start operational testing. The very first examples of systems are available for test, but they are not yet mature (certain aspects of systems, such as software bugs and hardware reliability, must have real use experience in order to mature). The developer must determine if it is appropriate to take the time to get that experience, add that maturity, and, therefore, have a better chance of running an operational test with success. The cost of that decision is the expense of leaving a factory idle, a factory that has many workers and machines ready for production. On the other hand, the developer can elect to push the product into test prematurely. In that case, the hope is that the product will get through an operational test, yielding a decision to produce, and getting the factory working as quickly as possible. The maturing can then be accomplished during the long time it takes to get the first production units out. Almost invariably, the Defense Department takes the course of a riskier entry into operational tests to gain the economy of rapid production. This is probably a good thing to do; however, on occasion, a test encounters those maturity considerations, which cause major problems.

A good example occurred with the Joint Tactical Information Distribution System (JTIDS), a tri-service airborne datalink system. In this case, the objective was to produce a system with 400 hours mean time between failures. When JTIDS entered its operational test, the testers noted that the reliability was poor (about 40 hours), actually disrupting the ability to run the tests. They stated, therefore, that the reliability appeared inadequate. This was no surprise to the development community, because they knew that the JTIDS units had not had a chance to mature. But it was a great surprise to members of the three services who were presented with the results of an operational test, and who had no idea about the status of maturity when the test started.

This kind of surprise calls into question the credibility of the organization developing the system and of the management group in the government. The subject then expands from a test of JTIDS to a test of the credibility of the whole acquisition system that created JTIDS. This test does no one any good, and is also very inefficient. Red teams, special panels, and briefings to everyone who has any affiliation with the system are the normal means of recovery, but it is a very long time before credibility is regained. In fact, on JTIDS, during the year or so while all the reviews were going on, the system matured and eventually showed about 80 percent of the specified reliability in its tests. The program is now back in a more normal mode of development, but at the expense (both in time and in dollars) of a long period of credibility loss, credibility that may never be fully regained in the system.

Developers should share the knowledge of the state of maturity with the full set of involved players, rather than have a large number of people be surprised by a demonstration that the system is not mature. This requires analysis and simulation to aid in the determination of the projected growth in capability resulting from maturity.

The last surprise is the *lack-of-usability* surprise. This is when, for whatever reasons, the user — the soldier or pilot or sailor who is going to use the system — has no chance to try the system until the operational test (the development community often uses surrogates to try it during development). The surprise occurs when the users reject the system because it is too difficult to use.

A good example of this is the Strategic Air Command Digital Network (SACDIN). This network was to disseminate emergency action messages to the strategic force structure and to receive status messages back from that force. As befit the task, this had to be a very secure network — SACDIN utilized message-entry terminals comparable to workstations common in offices today, except that these workstations used extensive software measures to ensure security. The system was developed over the normal course of time and the operational test was run.

This was the most secure software system imagined at that time — no one had ever gone this far in building security — so the designers erred on the side of increased security. For example, if an operator entered several incorrect inputs into a terminal, the system viewed this as a potential security breach. The system's response was to freeze the terminal, audit the most recent data, and sound an alarm to bring in a security officer.

During the operational test, the users, of course, had no experience with the system and therefore made errors. These errors satisfied the criteria for a potential security breach, freezing the system. At the end of the test, each user said, essentially, "I cannot use this terminal — every time I make a mistake, it freezes instead of helping me." In response, the designers introduced software changes; however, the process for change was extremely complicated because of the software security requirements. The specification's ability to maintain security had to be mathematically verified; there also had to be manual validation that the real software matched the specification (professional teams attempted to break into the system). This was all done through a regression-test process to accept the new software changes. In the case of SACDIN, it took about a year to redo the system to solve the problem.

To avoid such problems, human-machine interaction must be evaluated early in the cycle by use of rapid prototypes and simulation.

3. SIMULATION

Of course, the Defense Department has been using simulation for years, and in many different ways, ranging from operations research simulations (which help developers to understand the utility of a new product in the very early planning stages), to simulations used by designers and developers (such as microelectronics designers or radar designers), to simulations that deal with human factors (such as cockpit simulators), to simulators that synthesize complicated environments in which systems will operate (such as electronic combat environments), to the ultimate war gaming simulations that are usually the basis for our operational tests.

Over the last ten years we have seen dynamic growth in the computing and networking areas that form the underpinning for digital simulation. There has been a corresponding increase in the use of simulation, which can be both much more elaborate and much lower in cost than in earlier times. It is not a very risky prediction to state that this will probably continue to be true for the next ten years. Since few things get cheaper and better with time, it is important to take this opportunity to take more advantage of simulation.

We are already doing this. The following examples illustrate how simulation is changing, and what kind of additional value can come from these changes. The examples are drawn from government activities in which my company, The MITRE Corporation, is involved.

One of the best examples is the Warrior Preparation Center (WPC), which is designed to give senior European battle commanders and their staffs the opportunity to train for the operational level of war using

interactive computer simulations that replicate, as closely as possible, the real NATO environment. WPC allows staffs from around the world to participate, simultaneously, in some of the most sophisticated and realistic war games.

WPC uses parallel-processing algorithms to satisfy the immense processing requirements of these complex simulations. A multiple user/system interface allows for the networking of simulations from many different locations, and maintains the integrity of the distributed database. The system offers a faster, more realistic simulation to a larger number of staff in more widely separated locations than ever before.

For the Strategic Defense Initiative, the Experimental Version Prototype System (EVPS) is being used to simulate and analyze potential strategic defense systems. It models a set of proposed boost-phase defensive systems, with the threat coming from Soviet intercontinental ballistic missiles launched using several different launch strategies. EVPS can simulate all the sensors, weapons, battle management, and communications functions that will be required of any future strategic defense system.

The most interesting aspect of EVPS is the organization of the simulation itself: EVPS is being developed as a structured prototype, where specific goals are established for a set number of releases. Each release brings new functions into the model, increasing the complexity and fidelity of the simulation, and providing new insights into both the problem being modelled and the model itself.

The National Air Traffic Control Simulation (NAS) has to simulate one of the most complex problems in existence — the entire ATC system of the United States. It has to be linked to other ATC prototypes and cockpit simulators as well as to training rooms at operational facilities, and development has to continue while the system is in use. A distributed architecture is the only possible answer to these requirements, but that raises several problems, such as database contention (several users trying to read or write the same data at the same time) and reconciliation (making sure that everyone is using the same set of data in the face of constant changes from many sources). These problems are currently the subject of some new approaches.

One of these is known as *time warp*. Each simulation runs independently and stores all of its previous states in memory. When the simulations are reconciled, the time warp rolls the simulation back, as necessary, to the point where the databases diverge. It then reconciles them and lets the simulation proceed from that point. This approach is very fast, but requires a large amount of memory and processing capability.

Another approach is the *moving time window*, in which the simulation is run up to a certain point. When the fastest simulation reaches a specified time ahead of the slowest simulation, it is frozen while the slowest catches up. This approach is much less demanding of computer resources, but it is also a great deal slower.

One of the most important recent advances in simulation is called *virtual reality*, in which the user is totally immersed in the simulation. For NASA's space station, planners enter the details of a space-station module into the simulation — how big it is, what shape it is, the lighting, and so on. An engineer can then put on a special helmet with a graphic display faceplate and see the interior of the station as if standing inside it. Whichever way the engineer moves or turns, the station is seen just as it would be from the inside.

These new techniques may well change simulation completely, but even the most sophisticated simulation is of no use if confidence in its accuracy is low.

4. VALIDATING SIMULATION

Confidence, or credibility, is without a doubt the overriding issue. If a developer is absolutely confident that a simulation is correct in every detail — and if this confidence is justified — then no operational tests are necessary at all. On the other hand, if the developer has no confidence in the simulation, everything will have to be tested. Obviously these are the extreme cases, but real life often approaches the extremes rather closely.

The following three examples are typical of the difficulty encountered in validating simulations.

4.1 Extrapolation

The over-the-horizon radar can serve as an example to illustrate the problem of extrapolation. Over-the-horizon radars transmit high-frequency electromagnetic waves that bounce off the ionosphere and illuminate targets at very long range, well beyond the line of sight. These radars were developed to see large targets, such as bombers that might be attacking the United States.

The performance of high-frequency radars is very dependent upon the path of propagation, the time of day, the time of year, and solar activity, all because of their effects on the ionosphere. To account for these variables, the original designers added extra capability to the over-the-horizon radars so that they could maintain performance in the face of abnormal conditions.

While the early systems were being developed, questions arose about the detection of cruise missiles, because the Soviet threat was changing from bombers to stand-off cruise missiles (which have smaller radar cross sections and are, therefore, more difficult to detect). The Air Force asked whether these radars could reliably detect cruise missiles and, if not, what modifications could be made to give them this capability.

Six organizations had developed simulation models that they used regularly to analyze high-frequency radar performance against bomber-size targets. Experts in these organizations gave significantly different answers to the Air Force's question. It soon became apparent that the experts were working under different assumptions, so the government set up a special process in which all the experts were asked to estimate radar performance under identical circumstances.

Figure 2 shows the results; each of the black bars represents one organization's estimate of the target size (in square meters) for which the radar would achieve a 50-percent detection probability. The answers ranged from 80 square meters to a few square meters.

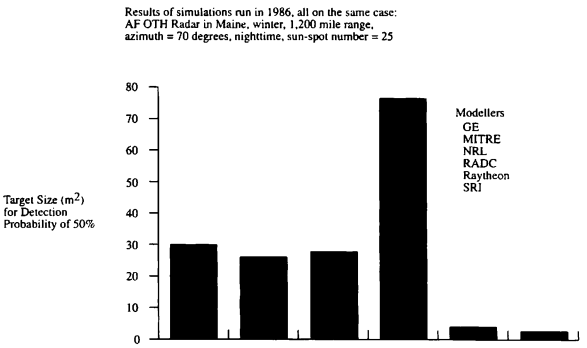

Figure 2. OTH Radar Simulation Results

How could all these experts with their trusted simulation models give such widely different answers? The difficulties arose because, in this application, the models were being extrapolated beyond their regions of validity. Although they gave roughly similar results when the radar target was large, they diverged in their predictions against small targets.

A close examination of the technical details inside the models soon revealed that they all lacked fidelity in accounting for various propagation phenomena such as ionospheric focusing and multipath, and that they contained subtle differences in their representations of these effects. After much study, a community concensus was reached on appropriate algorithms for each of the important phenomena. The Air Force also calibrated the models by conducting field experiments with an over-the-horizon radar against small airborne drones (facsimiles of real cruise missiles). This work eventually resulted in a radar model appropriate for small-target use, and earned the confidence of the community.

The point here is that confidence in a simulation model requires not only knowledge about the model, but information about the extrapolation involved in dealing with the specific problem.

4.2 Marketplace-Validated Models

The next example is the *marketplace-validated model*. These are

models that companies create and sell, and that other companies use all the time; the government uses them quite often as well. The market is the test of validation — if the models work, people buy them and use them. Unfortunately, this mechanism is far from perfect.

One illustration is response-time models for data processing systems. When a developer is building a large distributed data processing system, perhaps a worldwide system with many thousands of users, it is important to know how long it takes for a user requesting data to get a response. One of the key factors in determining the answer is the amount of contention (contention occurs when many users happen to make requests that ask the same processors to function at the same time). If there is substantial contention and the management of the processing and communications resources is inefficient, then responses from the system can take a very long time.

How does a software designer know what the contention will be in a system that has not yet been fielded? There is no way to make measurements, so the designer is forced to make some judgement as to what the contention will be. People with experience in this field know that this is very frequently misjudged; as a result, models often produce very wrong predictions.

How can this problem be solved? It is extremely helpful to get an early version of the system out into the field, where measurements can be taken, even though the system may be far from complete. Actual data, even when approximate, are generally much preferable to no data at all, and provide at least partial validation of the modeling assumptions.

4.3 Reliability Models

Another example is found in *reliability models*. The problem here is one with which electronics developers have to contend: Given the design of an electronic system, the developer must estimate, before it has been implemented in hardware, how reliable it will be before a development decision can be made. Models can provide some information — they account for the quality of the parts, the thermal stresses, and the electrical stresses that the system will have in operation, and through some integrated set of calculations determine what the mean time to failure will be. These models are used all the time, but most of them account only for electronic parts factors — they do not address factors such as workmanship, manufacturing quality, or even larger parts issues, such as the mechanical rigidity of the boards used or the reliability of the connectors.

Even though these models can provide accurate answers for only a portion of the problem, developers find them valuable because at least they can verify that parts-related factors are not limiting the reliability of the system. However, a developer attempting to validate reliability models must also attempt to measure the quality and workmanship standards for each of the factories.

These three examples illustrate that the concept of validating simulations is extremely difficult.

4.4 Simulation in Operational Testing

The Joint Surveillance Target Attack Radar System (Joint STARS) provides an outstanding example of how simulations should be used. Unfortunately, this type of use is not common enough in defense acquisition systems.

Joint STARS is an airborne radar system capable of detecting slow-moving vehicles (for example, tanks on the battlefield), thus serving as a surveillance resource for many Army and Air Force weapons. The radar detects moving targets by directly measuring their velocity, and thereby separating them from the relatively stationary ground clutter. (Therefore, the slower the target, the more difficulty a radar has detecting the target against clutter.) A radar of this sort is capable of measuring only the component of a target's velocity projected along the imaginary line joining the radar to the target; this component is called the radial velocity of the target.

Figure 3 illustrates the basic technical design issue involved in developing the Joint STARS radar: how slow a target should the radar be able to detect? The figure was derived by creating a model of the roads and off-road areas in Europe passable to tanks and trucks. The modelers laid a hypothetical Soviet force of moving vehicles down on this area, and calculated the radial velocity of each target to the Joint STARS radar. They then computed the fraction of the target set that would be visible if the radar could detect targets moving at greater than some specified minimum radial velocity. The two curves show the percent of targets seen by the radar as a function of the minimum detectable target velocity, for fast-moving and slow-moving targets. It can be seen that the percentage of slow-moving targets *declines dramatically* as the minimum detectable velocity for the radar is increased.

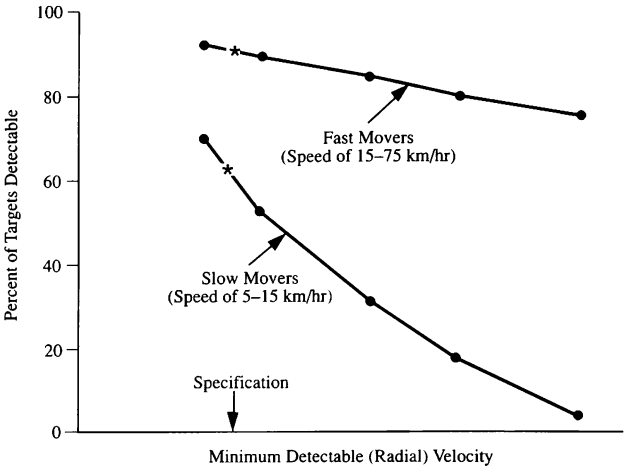

Figure 3. Joint STARS Operational Simulation

On the basis of these curves, the two services demanded that the radar be sensitive to very slow targets, and specified the minimum detectable velocity as shown by the arrow (the actual number is classified). If the radar could detect every target that moved at or above this speed, it would detect 95 percent of the fast-moving targets and 70 to 75 percent of the slow-moving targets.

The radar designers performed a corresponding simulation. For a given set of assumptions about the radar's frequency, antenna length, platform speed, clutter behavior, and so on, they estimated the relative ability of the radar to detect a target as a function of the target's radial velocity. They found, as shown in figure 4, that the radar's detectability moves down a *very steep curve* below a certain target velocity. The specified minimum target velocity shown in the figure was dictated by the state-of-the-art of radar design.

The designers recognized from these simulations that the estimated performance — and therefore the operational utility — of the system was very sensitive to the fidelity of the simulations. Because the curves shown in figures 3 and 4 both have very steep portions, small errors in the simulation could result in large changes in system utility. The simulations strongly suggested that more work to increase confidence would be desirable, rather than waiting for ten years of development to discover whether the system had utility.

Given this focus of attention, an aggressive program of experimentation and early operational testing was established. Component and subsystem tests were conducted to measure antenna performance, oscillator stability, behavior under vibration, and other factors as part of the process of validating many of the internal assumptions made within the simulations. The system developers scheduled operational flight tests as soon as the basic (but incomplete) system was capable of operation, as another step in getting early validation of the simulation results. An operational test in Europe was conducted to verify additional assumptions about the target, clutter, and interference environment to be expected. At the same time, the designers began to show the data to operational users in order to get from them an early indication of whether they thought the system would have utility at the end-point of development.

It is this concept for simulation that should be stressed — namely, that simulation should be a focusing mechanism for running expensive, but very useful, operational tests, as early in the development process as possible. This is infinitely preferable to waiting six to ten years for the

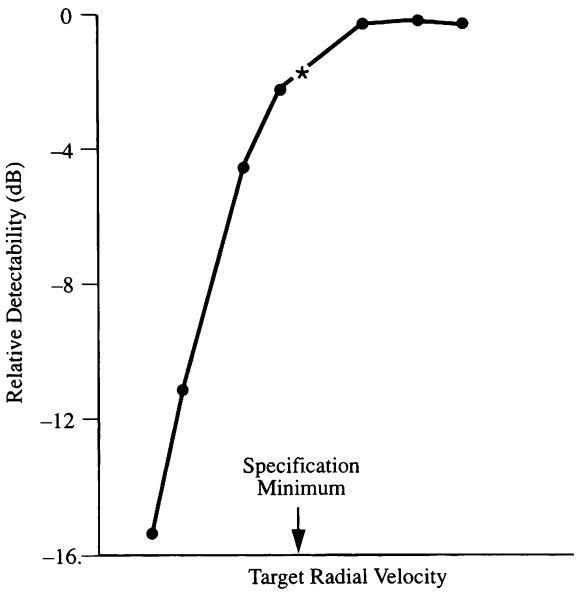

Figure 4. Joint STARS Radar Performance Simulation

operational test, which is otherwise the first opportunity for understanding whether the system has utility or not.

5. CONCLUSIONS AND RECOMMENDATIONS

This section presents the recommendations of the panel, which included well known individuals with a set of experience that covers all of these topics in depth. It also highlights some instructional comments made by individual members of the panel.

The point was made that simulation is generally not something that confirms or rejects a hypothesis, but is instead a mind extender — it makes us think about an area of concern that we would not have focused our attention on otherwise. As a result, simulation can lead into areas of evaluation that may be crucial but that might not have been considered.

It was also noted that central management of the reuse and distribution of simulation ignores a very important point: simulation designers themselves carry all the knowledge about what went into the models, and about what can be extrapolated and what cannot. Transferring the models without the designers would be an error, because many of these programs are so large that it would impossible for another organization to understand all the subtle ingredients.

5.1 The Learning Role

Imagine setting up, at the start of a development program, measures of effectiveness, assumptions (such as threats and environments), and an understanding of how testing and simulation will augment each other over the life of the program. There is no doubt that most of these measures and assumptions will be inaccurate at the outset, but imagine further that they can be continually refined as knowledge is gained throughout the development planning process. We will call this an evaluation framework.

Our first recommendation is that we need to emphasize the learning role for operational testing during development; the panel would like to see this standardized by the setting up and documentation of evaluation frameworks at the beginning of every program. This will also provide assurance that the program will not encounter the change-in-assumptions surprise (as DIVAD did) or the change-in-measures-of-effectiveness surprise (as Aquila did). To do this, we have to involve the independent operational test people from the start — they cannot be brought in at the end of the process.

This raises two concerns. The first is that the testers will not remain independent if they get involved in the programs earlier than they do now. The committee's view is that aloofness should not be confused with independence. The value of independence is that the testers have knowledge to provide and a management chain that gives them the ability to apply it; to keep them aloof is to lose this.

The second is a concern about how these operational testers can properly lay out the evaluation framework, since they are a small group and do not have the resources or knowledge to do it. We believe that the development community (not the testers themselves) should take the lead in laying out an evaluation framework, and that the community of operational people should be involved in agreeing to and continuing to modify the framework as development goes on.

5.2 Simulation to Provide Focus

As noted earlier, simulation cannot, in most cases, prove or disprove hypotheses, but it can isolate high-sensitivity areas, areas that could change the prevailing view of system utility. We want to establish an important role for simulation in performing excursion analyses to focus the early operational tests. The second recommendation of the panel is that the Department of Defense should require sensitivity analyses at the beginning of all development programs. We do not want to see fixed-point simulation results; we want excursion analyses that can be used as the basis for deciding whether or not early operational testing should be done. In other words, simulation should focus, not replace, testing.

5.3 Periodic Re-Evaluation

The third recommendation deals with the rigid period of the development cycle. Simulation is an evaluation tool, and if we have a management process that attempts to keep contracts and specifications fixed, then there will be no room for evaluation (because the purpose of evaluation is to determine whether the specifications are correct or not).

The panel believes that the current acquisition process stifles evaluation. Program managers are not motivated to examine their programs — they are interested simply in stability. The culture in the Defense Department should promote evaluation; the panel recommends that the Office of the Secretary of Defense (OSD) establish policy and provide guidance to the acquisition community for systematically re-evaluating system specifications using modeling, simulation, and test.

This is a very difficult recommendation — we do not have a simple solution. Nonetheless, we think this is crucial if we are going to get any value out of the tools and the capabilities of simulation. This raises another concern — some will say that this will encourage a mode where we are always changing everything, and that we will end up with nothing (in other words, we will lose management control). In response, we want to make clear that we do not advocate giving up configuration management on system developments, or insist that every evaluation should result in a change. There should be two distinct processes: one that is evaluating and one that is changing. We expect the rate of evaluation to be much higher than the rate of change; however, if we do not evaluate at all, there will be no changing, and then we will end up facing one or more of those surprises.

5.4 Human Factors

The fourth recommendation is stimulated by the human-factors problem illustrated by SACDIN. The tools are now available, and the cost is sufficiently low, that every program should build mock-ups of human-machine interfaces as soon as possible, and bring in the real users to understand better the utility of the design. The panel recommends that the service acquisition executives ensure the use of human-in-the-loop simulations in all development programs, beginning with requirements definition and continuing throughout the acquisition process.

There is one set of systems where this is particularly difficult: the large command and control systems used by many geographically separated generals or admirals. It is not easy to bring them to a central facility to test the system. The solution is a new technique called distributed simulation, where many decision makers can play their parts in a global simulation while remaining in or near their own offices. Perhaps through that mechanism, we can get higher-level Defense Department staff to test those portions of the systems that they will ultimately use. We are seeing the technique used today for training, but we think that it can be useful in the area of development as well.

5.5 Credibility

Finally, the panel made several recommendations with regard to the issue of credibility. How do we know whether to trust a simulation? Should we set up a central office to accredit simulations? Should we set up a management process to distribute and reuse simulations?

The panel felt that there are no single-point answers to the problem of trusting simulation. We should be doing excursion analyses and sensitivity analyses; when we find something that makes us nervous, we should run a test — the test, and not the simulation, validates the answer.

We should have professionally documented simulation results. Decision makers should see the whole set of data so that they understand how a simulation was calibrated and to what degree the results are dependent on extrapolation.

Should we set up a central office? The panel believes that we clearly should not — there is no office with the capability to perform this difficult task. Should we set up a distribution process for distributing these simulations? The panel believes that we should not — if we cannot accredit simulations, we certainly cannot have a very logical process for distributing them. But the panel does have some recommendations concerning validity.

There are certain models in the Defense Department that tend to be used and re-used by expert groups — for example, the Defense Nuclear Agency (DNA) models on nuclear effects or the Defense Intelligence Agency threat models. The DNA models have never been fully validated, and they often lead to problems. At the same time, they are the best available, and everyone uses them. Models of this type should have a budget line to reinforce their improvement and keep them current. We recommend that the Joint Chiefs of Staff and OSD allocate money directly to those groups that develop and use the models.

When an acquisition decision depends heavily on simulation, an independent panel may be used to perform several valuable validation tasks. While it is nearly impossible to validate an entire simulation, a panel can validate many aspects of it, including the people who designed the simulation. It can also validate the extrapolation (that is, it can ask specific questions about the historical use of the simulation versus the extrapolation now being used) and the fidelity of the input data. The panel can determine if users are doing a partial evaluation or a full evaluation, and, in the case of a partial evaluation, whether the parts not being evaluated are important to credibility. These are all questions that panels of experts can answer in a fairly short period of time, to add at least that level of confidence to the use of simulation when needed. Such panels, however, should be used only in special cases.

In regard to professional documentation, Defense Acquisition Board documentation has a place for, but does not specifically call for, the data that determine the basis for validating the simulation and the credibility factors. We believe that these data should become part of the documentation.

Finally, as the over-the-horizon radar example points out, strange results sometimes arise when comparing different validated models against the same problem. Such comparisons, when available, can be valuable indicators of the overall credibility of the models.

5.6 Summary

The panel believes that it is extremely important to avoid operational test surprises, for reasons of both cost and credibility. This can be done by performing more operational testing during development, enabling developers to learn about problem areas while they still can be fixed rather than waiting until the system is fielded. It also helps to develop evaluation frameworks at the onset to specify how evaluation is to be performed; as the real program progresses, these must be upgraded so that they are consistent with the state of knowledge. Operational testers must be involved so that there are no unanticipated gaps or changes in viewpoint.

The current acquisition process stifles evaluation, and unless we have a more open attitude to performing evaluation, the problem will remain unsolved. Something must be done at upper management levels to change the process, but this does not include an independent simulation office to accredit or manage the use or distribution of simulations. Such an office cannot add confidence, but it can add confusion.

Finally, the panel believes that simulation should be used to focus testing onto those areas where we do not have confidence.