CROSSTALK

January 2001 **The Journal of Defense Software Engineering** Vol. 14 No. 1

Modeling and Simulation

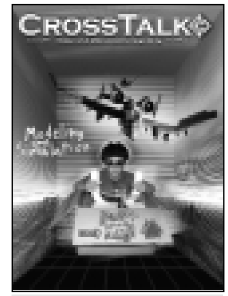
Modeling and Simulation

4 **Evans & Sutherland: A Model in the Simulation Industry**
An interview with Evans & Sutherland vice presidents David C. Janke and Kevin J. McLaughlin.
by Matthew Welker

8 **Computers and M&S – Modeling and Simulation 101**
This educational primer leaves no room for simulation tunnel vision as it explains the inherent inseparability of modeling and simulation in project success.
by David A. Cook, Ph.D.

11 **Leading, Integrating, and Leveraging M&S for the Warfighter**
Through an ambitious set of goals, the Defense Modeling and Simulation Office becomes the warfighters' eyes and ears for technological advancements.
by Col. Wm. Forrest Crain

15 **Integrating Process Simulation and Reliability Models**
If you are wondering how your software reliability will be affected by defect prevention and detection activities, this simulation model can estimate the impact.
by Ioana Rus and James Collofello



On the Cover: Kent Bingham, Digital Illustration and Design, is a self-taught graphic artist/designer and freelances print and Web design projects. Thanks to Dax Bingham for being the model.

Software Engineering Technology

19 **Project Scheduling According to Dr. Goldratt**
A discussion of Goldratt's insight on eliminating or minimizing causes of schedule waste in his book *Critical Chain.*
by Timothy K. Perkins

22 **Event-Driven Learning as a Process Improvement Strategy**
Training experiences designed directly into the operational process optimize learning and improve performance.
by Michael J. Hillelsohn

Open Forum

27 **Intrusion Detection: Implementation and Operational Issues**
Intrusion detection systems are part of the proactive puzzle that protects computer systems and networks from attack.
by John McHugh, Alan Christie, and Julia Allen

Departments

3 From the Publisher

14 Coming Events

21 Quote Marks

26 M&S Web Sites

31 BackTalk

CrossTalk

Sponsor	*Lt. Col. Glenn A. Palmer*
Publisher	*Reuel S. Alder*
Associate Publisher	*Elizabeth Starrett*
Managing Editor	*Pam Bowers*
Associate Editor/Layout	*Matthew Welker*
Associate Editor/Features	*Heather Winward*
Graphic Designer	*Abby Hall*
Voice	801-586-0095
Fax	801-777-5633
E-mail	crosstalk.staff@hill.af.mil
CrossTalk Online	www.stsc.hill.af.mil/Crosstalk/crostalk.html
CRSIP Online	www.crsip.hill.af.mil

Subscriptions: Send correspondence concerning subscriptions and changes of address to the following address. You may use the form on page 31.

Ogden ALC/TISE
5851 F Ave., Bldg 849, Rm B-04
Hill AFB, Utah 84056-5713

Article Submissions: We welcome articles of interest to the defense software community. Articles must be approved by the CrossTalk editorial board prior to publication. Please follow the Author Guidelines, available at www.stsc.hill.af.mil/CrossTalk/xtlkguid.pdf. CrossTalk does not pay for submissions. Articles published in CrossTalk remain the property of the authors and may be submitted to other publications.

Reprints and Permissions: Requests for reprints must be requested from the author or the copyright holder. Please coordinate your request with CrossTalk.

Trademarks and Endorsements: This DoD journal is an authorized publication for members of the Department of Defense. Contents of CrossTalk are not necessarily the official views of, or endorsed by, the government, the Department of Defense, or the Software Technology Support Center. All product names referenced in this issue are trademarks of their companies.

Coming Events: We often list conferences, seminars, symposiums, etc. that are of interest to our readers. There is no fee for this service, but we must receive the information at least 90 days before registration. Send an announcement to the CrossTalk Editorial Department.

STSC Online Services: at www.stsc.hill.af.mil
Call 801-777-7026, e-mail: randy.schreifels@hill.af.mil

Back Issues Available: The STSC sometimes has extra copies of back issues of CrossTalk available free of charge.

The Software Technology Support Center was established at Ogden Air Logistics Center (AFMC) by Headquarters U.S. Air Force to help Air Force software organizations identify, evaluate, and adopt technologies to improve the quality of their software products, efficiency in producing them, and their ability to accurately predict the cost and schedule of their delivery.

Virtual Reality Can Be a Great Value or Merely a Vice

Living in a virtual world may have its advantages. Like the pilot who practiced landing an aircraft in extreme conditions only to discover within a month that he has to do it for real. His practice in the virtual world allowed him to accomplish a seemingly impossible feat in the real world, and lives were saved. Practice makes perfect and practicing in adverse conditions enables us to find the right solution through trial and error and away from danger.

For the Air Force, modeling and simulation reduces cost and enhances warfighter capability through training. While the general public also has access today to simulations through video games, I wonder if the potential for the same positive results in people's daily lives is manifested. Video games can be focused on unproductive themes like sex, violence, and the occult. Why would we want to get more practice with our vices? Do we really need any help to improve our animal cravings? Will violence help us to solve problems in the real world? If we hone our autonomic system to blow away our virtual opponent, will our initial instinctive response be to blow them away in the real world?

I can think of better things to simulate like maintaining an effective budget, developing problem solving skills, composing music, learning languages, and improving comprehension to mention only a few. Surely there is a way to use simulation technology to help us to improve ourselves instead of indulge our vices. The question is, are we intelligent enough to choose self-improvement over self-indulgence? Can we recognize the difference between pleasure and fulfillment?

Then again, why stop with the simulation of physical objects? I can think of some human relationships that could use some practice. For those of you who have teenagers, I wonder if a simulation of parent-teen relationship would help. We spend a lot of time getting educated in technical subjects and leave to chance the most challenging aspects of our lives—human relationships.

Meanwhile, the Air Force forges ahead with modeling and simulation advancements to provide the maximum warfighter training potential within budget. Dave Cook gives a wonderful introduction to this technology. The benefits are enumerated along with a cautionary note: Verify and validate the simulation. While military flight simulation provided fertile ground for computer graphics, the demands of commercial civil airlines pushed its development. So we interviewed top management at Evans & Sutherland because of their expertise in this technology. Their large R&D investment indicates a high level of commitment, but it is backed by their customer focus.

We also feature an article from Col. Wm. Forrest Crain in the Defense Modeling and Simulation Office, which serves as the executive secretariat for the DoD Executive Council on Modeling and Simulation, and acts as a full-time focal point for modeling and simulation. Ioana Rus, Fraunhofer Center for Experimental Software Engineering, and James Collorello, Arizona State University, contribute to this month's theme by proposing a software process simulator for estimating the impact of different software reliability engineering practices.

May you enjoy this issue of CrossTalk and gain a renewed enthusiasm for modeling and simulation possibilities. We hope that your simulations are anchored in reality.

Reuel S. Alder, Publisher
Software Technology Support Center

Evans & Sutherland: A Model in the Simulation Industry

CrossTalk Associate Editor Matthew Welker had the opportunity to interview David C. Janke and Kevin J. McLaughlin, vice presidents at Evans & Sutherland, a graphics pioneer in Salt Lake City, Utah.

David C. Janke is vice president of Strategic Marketing for the Simulation Group. He has been with Evans & Sutherland (E&S) since 1988 and has 25 years of experience in management of high technology systems. His broad experience includes assignments in business development, program management, engineering management, contract administration, and operations management. In recent years, he has presided over the rapid expansion of E&S into international markets, especially Europe and Asia. Janke holds a bachelor's degree in electrical engineering from Stanford University and a master's degree in business administration from Brigham Young University. His academic honors include election to Phi Beta Kappa, Tau Beta Pi, and Beta Gamma Sigma honorary societies.

Kevin J. McLaughlin is vice president of Shared Technology for the Simulation Group. He has been with E&S since 1987 and has worked in the engineering industry for 21 years. His assignments include managing product development, hardware engineering, application-specific integrated circuit (ASIC) development, system exploitation and verification, software development, and database modeling tool development. Products developed by Shared Technology include the Harmony image generator, Integrator synthetic environment software, and EaSIEST modeling tools. Prior to his assignments at E&S, he worked as a senior project integration engineer and senior electrical engineer for Eaton-Kenway. Previously, he was an electronics technician and instructor for the U.S. Army. McLaughlin has a master's degree in business administration from the University of Phoenix and a bachelor's degree in electrical engineering from the University of Utah. He is also licensed as a professional engineer.

CrossTalk: The evolution of modeling and simulation from sketch-pad style interfaces to virtual reality has happened at an astounding rate. What role has Evans & Sutherland (E&S) played?

Janke: Obviously, E&S has been on the forefront … David Evans and Ivan Sutherland, the founders of E&S, were pioneers of computer graphics who produced the first line-draw system, and who developed the very first algorithms for computers to draw pictures. Much of the progress in the industry has come about because of people who either worked at E&S or were students of Evans at the University of Utah. These people include Jim Clark, who started Silicon Graphics, John Warnock of Adobe Systems, and others who have spun off and pursued careers in the graphics industry.

E&S has been a breeding ground for graphics technology, either internally or from people who have some association with one of the founders. Our own progress within the company has been astounding as well. From the early days, our main business has been flight simulation. Although our first systems were for the military, we made our bread and butter in commercial simulation for civil airline pilots. The demands and requirements of the civil airlines pushed development at E&S in such areas as calligraphic capabilities, texture, complex shading algorithms, anti-aliasing, and other graphics technologies that have become common today. Although most of those early advances in computer graphics were driven by our flight simulation customers, our business today is much broader, and many of our newer developments are targeted toward other applications such as ground-based simulation, driving simulation, mission planning and rehearsal, virtual prototyping, and vehicle development.

Even today, as others have entered the industry, I believe we are the only ones for whom simulation is our core business. Our graphics developments are focused toward meeting the requirements of the simulation customer rather than meeting the requirements of the gaming industry, or the server market, or something else. Our chip design, and our software design are today focused toward our core simulation business.

McLaughlin: To put it in perspective, 14 or 15 years ago we could sell a system that had 200 to 250 polygons on a per-channel basis updated at 60 hertz. Today, we are achieving over 15,000 polygons. That is an astronomical amount of computations based on sheer horsepower. Again, 12 to 14 years ago was the advent of the use of texture. Now we have texture that is applied to everything and even gives a higher effective polygon capacity [3x-4x] than what we had in those days. We now also have features such as reflections, Phong-shading, and sophisticated texture algorithms. Realistic sensor (infrared, night vision, radar) simulation has always been something very difficult to achieve in a real-time simulation. We have now developed what we call Sensor Texture™ that allows the assignment of material attributes on a per-texel basis, which is something that is not possible with polygons. This allows more robust sensor simulation for infrared devices because you are able to work with different materials with texture to present a more realistic simulation.

The day-to-day advancements do not seem dramatic, but when you look at them over the past five or six years, it is actually incredible. The very first image generator system that I worked on would fill this conference room with numerous 72-inch cabinets. Now you can do the same thing in one cabinet [approximately 24 inches high] that is essentially PC-based.

Janke: I have seen dramatic progress in the 12 years that I have been with E&S. In 1990, we built a system for the German Air Force that had 20 cabinets and approximately 4,000 cards. The total cost was $6.5 million. Today, a single cabinet replaces all of that at a fraction of the cost.

CrossTalk: Can you compare the effort you spend on hardware as opposed to software? Which requires more effort? What are the biggest challenges in regard to cost, schedule, and quality?

Janke: Graphics have always been very compute-intensive. In order to draw pictures with a computer, lots of mathematical equations need to be done very rapidly. To achieve real-time simulation, a new picture must be computed every sixtieth of a second. In the past, general-purpose processors were not powerful enough to do that, so we developed our own special-purpose ASICs to do all those rapid calculations. As special-purpose hardware was the only way for us to meet the requirements of our simulation customers, we were very hardware-centric with much emphasis placed on hardware design.

Now with the rapidly advancing power of general-purpose hardware, we have much less need for custom-designed chips, and we are able to more broadly utilize commercial off-the-shelf (COTS) hardware. For example, our highest-performing product, the Harmony™ image generator, incorporates standard Intel Pentium processors for the geometry portion of its graphics pipeline. However, the rendering part of the architecture, which processes the pixels that are actually displayed on the screen, still requires special-purpose hardware that we design ourselves. As the general-purpose hardware becomes more powerful, we are able to use it more and more in the total graphics pipeline. Today it is used in a portion of it; tomorrow it may be used more.

Simulated Images from Evans & Sutherland

As we use more and more commercially available hardware, our company is transitioning from hardware- to software-centrism. In fact, we have probably already crossed the line to become more of a software company than a hardware company. If you look at the number of people working on software and the amount we spend on software development, it is more than we spend on hardware.

McLaughlin: Actually, in terms of total dollars spent, we still spend more for hardware development because custom ASIC design has significant non-labor costs associated with it. But in terms of sheer labor, the ratio of effort spent on software compared to hardware is approximately 3:1.

Janke: We, of course, try to use COTS software as much as possible if it will do the job. If it will not do the job, we have to augment it with our own designs.

CrossTalk: Where do you find your COTS, and how big a role does it play? The graphics in simulation are so specialized, it is hard to imagine that [COTS] would meet your needs.

McLaughlin: We use several COTS applications such as: Versant, Multigen, ArcInfo, HTFS, 3-D Studio Max, Photoshop, SocetSet, RoboHelp, Vx Works, and the Windows NT operating system. For some of the operations of our database modeling tools, we use a product called XoX, which is an application that allows us to perform operations for clipping edges and planes of polygons. Utilizing this functionality allows us to access the proper computations without writing all the software ourselves.

CrossTalk: Do you obtain the source code for COTS? If so, are there any cost savings?

McLaughlin: There is definitely a cost savings when we use COTS software. Indeed, that is why we do it. Furthermore, if we do not have to develop a certain portion of the software code ourselves, we can allocate our internal resources more cost-effectively. We can focus our efforts on areas of the design that require our special skill or our unique intellectual property. I should note, however, that the use of COTS software has sometimes not worked out very well. There have been cases where we purchased a COTS package expecting it to have certain functionality and then later discovered that it did not. Or we found the COTS software to be extremely difficult to integrate into our overall product architecture. In those cases we were forced to spend tremendous unplanned effort and cost to make it all work. We have learned that COTS software can be very valuable only if we understand it well before we commit to a design.

CrossTalk: How does your SimFUSION product compare to the high-end Harmony™ product?

Janke: SimFUSION is our PC-based product that we offer at a very low cost. Of course, it also has lower performance. It is basically a PC with an E&S graphics card for the rendering side of the graphics pipeline. SimFusion is targeted at the low-end part of the simulation market. It is ideal for applications such as shiphandling simulators, ground warfare applications, driving simulation, human factor studies, virtual prototyping, engineering visualization, and modeling and simulation laboratories. It is fully OpenGL compliant and will run a wide variety of standard simulation software applications. SimFUSION is our low-end product and Harmony™ is our high-end product. Comparing the two products is difficult because they are targeted at different market segments. Although there are big differences in capability, each is the best in its class.

CrossTalk: How do you balance the high cost of R&D necessary to remain on the leading edge with the bottom line?

Janke: That is a very important question that we ourselves struggle with. It takes a tremendous amount of R&D investment to stay competitive in this business, yet we cannot spend so much that we jeopardize our profitability. When we prepare our operating plans for a coming year, there are many internal debates about how much to spend. Every dollar we spend on R&D affects our short-term profitability but may improve it in the long term. Achieving the right balance between short-term and long-term profitability requires vision and a willingness to accept some risk.

"The U.S. military has always been the world leader in adopting new technology and encouraging new developments in industry. However, on some critical simulation programs the interests of schedule and risk prevail, and more proven technologies are often purchased. Oddly enough, we are currently selling more of our high-end, state-of-the-art products overseas than in the U.S."

Historically our company has spent between 20 and 25 percent of gross revenue on R&D, which is a very large percentage compared to others in the industry. Yet we seem to have done the right thing, and we believe that our focused, market-driven R&D spending has been among the reasons for our success.

So how do we determine the right balance? First of all, we keep our fingers on the pulse of the market so that we understand the changing needs of our customers. What are their needs for the future? Where is simulation going? What kind of deficiencies in today's visual systems need correcting? It all starts with the needs of our customers and where we think they are going in the future. In developing our plan, we involve many groups within the company, including marketing, engineering, and project management. We consider where spending an extra dollar today will bring about greater sales or reduced cost tomorrow. There is much internal debate that ultimately leads to a prioritized list of R&D projects. Upper management then reviews the list and adds their perspective and vision for the future. In the end, we have a plan that usually hits the mark pretty well.

I would like to emphasize the importance of vision in developing the R&D plan. It takes years, not months, to develop new graphics products, and you cannot turn on a dime to react to changing market conditions. Although it is risky, you must commit to a development direction and stay with it. And such vision only comes about by staying close to your customers and listening to them all the time.

CrossTalk: As an aside relating to the market for your products, much of what you sell is to the U.S. military, but you also have a large customer base in Europe. Do you ever encounter a technology gap, or do you apply the same technological standards?

Janke: There is no question that the U.S. military is our single largest customer. And Europe is our second largest market. In most cases, the technology is similar between the two markets. In general, the fidelity of a simulator is determined by two things:

1. The capability of the aircraft or vehicle itself.
2. The customer's training doctrine.

There have been cases recently when our international customer actually demanded more from us technically than the U.S. customer. For example, our U.S. Air Force has much open, unpopulated land available to it for training purposes, and pilots would rather fly airplanes than simulators. Europe, however, is densely populated with little land area for training. Consequently, the European customer is forced to do a larger percentage of his training in a simulator, which puts more demands on the functionality of that simulator.

McLaughlin: The aerospace engineering laboratories, whether in the U.S. or Europe, also place heavy demands on us technically. In that kind of development environment, many what-if questions are explored and evaluated in order to come up with the most optimal aircraft design. Those labs often want to do things with the simulator that push the very limits of technology, that are real challenges for us.

Janke: The U.S. military has always been the world leader in adopting new technology and encouraging new developments in industry. However, on some critical simulation programs the interests of schedule and risk prevail, and more proven technologies are often purchased. Oddly enough, we are are currently selling more of our high-end, state-of-the-art products overseas than in the U.S.

CrossTalk: Rod Rougelot said, "E&S tends to be informal and tries to be less hierarchical, but much of the structure that we've had to establish is a result of dealing with the military." Bob Schumacker said, "With the military, you've got the user, you've got the agency that procures the system for the user, you've got one or two other layers or interpreters, you've got your prime contractor, then us." [from www.es.com/corporate/history.html] To what extent has working with the military aided E&S development? Can you characterize your relationship with the military?

Janke: Let me first speak to our general relationship with the military. It has been our strategy to market ourselves directly to the end user. Indeed, we have found that to be a key to our success. For example, we go directly to the Air Force customer and try to develop close relationships with the people who actually use the simulation systems. We try to understand what their needs are, and we try to show them how our products can meet those needs. Then when they need to buy a visual system, they think of E&S because they know that we can meet their training requirements.

Working with the government is much different than working with a commercial customer because the government tends to be very structured and organized in its procurement processes. And when you get a contract with the U.S. military, there are certain things you have to do to satisfy the demands of those processes. There are requirements for documentation, reporting, technical reviews, schedule forecasts, and so forth. Consequently, we have put into place the internal structure, the processes, and the organization needed to satisfy government requirements. But we have also worked hard to remain flexible and responsive. We do not want to become a big, stodgy, inflexible aerospace company. We intend to stay close to our customers, so we understand what their needs are, how they are changing, and how we should plan our product developments to meet those needs.

McLaughlin: An example is E&S's invention of global texture that was specifically designed to address the intelligence or the special operations market that requires mission rehearsal capability. Now global texture has become almost a universal requirement for military and commercial applications. This is an innovation that we developed because we needed higher database fidelity in a short amount of time. It was an invention that came about in direct response to the needs of our military customer.

Another product that was developed specifically for the military customer is RapidScene, which allows the rapid 3-D visualization of photogrammetric source material. The product evolved over time as our government customer kept asking us, "Can you do this? … Is this possible?" or "Wouldn't it be great if … ?"

Janke: Last June, three technical people and I made a visit to the Air Force Labs in Phoenix, Arizona. Our purpose was to present our technology roadmap for computer graphics and display technology, and to get feedback on our plan for R&D spending for the next year. We wanted to hear from an important customer whether or not we were headed in the right direction to meet their needs for the future. It was a great exchange, and we made some important adjustments to our development budgets as a result of the input. We try to have such meetings on a regular basis with all of our military customers.

CrossTalk: Do you use the Capability Maturity Model® (CMM) or other process models?

McLaughlin: We are ISO certified, and we have a group looking into being CMM-compliant. We have not decided if we are going to go through the entire certification and validation process. What we want to do is improve our entire software development process so we get the most benefit for resources applied to development. Our main goal is to do things smarter and more efficiently. We are also considering some outside tools to help us in our development. A lot of people out there are doing similar things, so we might as well take advantage of it.

CrossTalk: Do you use any of your modeling and simulation tools internally?

McLaughlin: E&S essentially has two modeling tool packages for creating the virtual environment: EaSIEST and Integrator. EaSIEST is used for the ESIG™ family of image generators, and for Liberty™ image generator. Integrator is used for our Harmony™ and Ensemble™ image generators. There are also data conversion processes between them. The tools are used by our own engineers to develop databases for simulation and are also sold as stand-alone products for customers who want to do the development themselves. We augment our database modeling tools with third-party software tools such as Multigen, Photoshop, 3-D Studio Max, Erdas, or anything we believe can actually be of benefit to us, and that we can integrate into the package. I should note that our own development people are probably our best testers, our biggest critics, and our biggest fans.

CrossTalk: What languages and platforms do you use?

McLaughlin: Right now we do development on NT workstations, but we also do some development on Sun workstations. Our primary languages are C, C++, Visual Basic, and then we have some database management tools like Oracle, Ingress, Versant. We use VxWorks as our run-time operating system that resides on specific processors.

CrossTalk: How does E&S recruit and retain good employees?

Janke: We have an active college recruiting program at a number of top colleges and universities. Our retention rate is quite good, and our turnover is fairly low compared to other companies in the industry. E&S is known to have one of the finer work environments in this area. The work is state-of-the-art, exciting, challenging, and makes use of the latest technologies. These are things that are attractive both to recent graduates and to experienced engineers.

As a practical matter, we have slightly better success retaining people who are from the West than from the East. For someone from the East Coast, this is a long way from home, and the cultural adjustment can sometimes be a bit too much. On the other hand, some new recruits come here and see the mountains and say, "I'm not leaving." For a creative software guy who likes to ride his bike to work through some spectacular scenery, it is an ideal environment.

McLaughlin: We also have an open campus so employees can access the buildings 24-hours-a-day. It is not uncommon to come in here at 2 a.m. and find a whole team of people climbing over a problem they cannot let go of.

The work is exciting and some people come here because they cannot find this type of start-to-finish work anywhere else. A software engineer at E&S gets to work on a variety of projects and is able to see the overall impact of what he does. Unlike many companies, our work involves an engineer in all aspects of a complete system. It is very satisfying to make an entire system work rather than spend all of your time focusing on one small piece and not really understand why your work is important. For example, in order to implement light points for calligraphic displays in a simulator, a software engineer might begin with modifying an extension for a third-party tool such as Multigen. He then creates a database of the gaming area with the proper light attributes, modifies a data converter, incorporates changes into the run-time software, and then actually displays it on a calligraphic device to see the results of his work. His responsibility necessarily involves him in each step and each component of the total system. Such work is very satisfying to intelligent and creative technical people.

We, of course, also offer competitive pay and benefits. We try to be competitive with the local market as well as the national market. For all major companies right now, the demand for technical people is far greater than the supply. So we continuously review our salaries and benefits to be sure we can attract the better people.

I think it is interesting that many of our people who have left to join flashy companies developing 3-D games have eventually returned to E&S. They all came to realize that the work here is more challenging, and that our environment is not easily duplicated. Our intelligent, creative, and cohesive workforce is a big attraction that really helps to maintain high morale.◆

® The Capability Maturity Model and CMM are registered trademarks of the Software Engineering Institute and Carnegie Mellon University.

Computers and M&S – Modeling and Simulation 101

David A. Cook, Ph.D.
Shim Enterprise Inc.

Software engineers have long viewed simulation as a tool to help them understand and develop complex systems. Unfortunately, some software practitioners focus solely on simulation failing to understand that modeling is a necessary foundation for a successful simulation. Before a simulation can be created and examined, the system being simulated must be mathematically modeled, then verified and validated. These two concepts, modeling and simulation, are both necessary and inherently inseparable.

One of my favorite definitions of simulation itself incorporates the concept of modeling and simulation as indivisible. Simulation is "the process of designing a computerized model of a system (or process) and conducting experiments with this model for the purpose either of understanding the behavior of the system or of evaluating various strategies for the operation of this system [1]." Simply put, a simulation allows you to develop a logical abstraction (an object), and then examine how an object behaves under differing stimulus.

There are several distinct purposes for simulation. One is to allow you to create a physical object such as a jet engine, as a logical entity in code. It is practical (and faster) to develop a code simulation for testing engine design changes. Changes to the engine can then be implemented, tested, and evaluated in the simulation. This is easier, cheaper, and faster than creating many different physical engines, each with only slightly different attributes.

Other uses of simulation are to create computer-based programs that can model complex systems such as an airport. Since it is impossible (or at least wildly impractical) to construct an actual working airport to test changes in operations, a computer simulation of an airport allows you to view the effect of changes in fueling, landing patterns, or take-off timing.

Unfortunately, before a simulation can be of benefit a model of the system must be developed that allows the simulation developer to construct the computer-based simulation. Modeling is the first step—the very foundation of a good simulation.

The Modeling Phase

You must have a model prior to creating a simulation. Modeling is an attempt to precisely characterize the essential components and interactions of a system. It is a "representation of an object, system, or idea in some form other than that of the entity itself [2]." In a perfect world, the object of a simulation (whether it be a physical object such as a jet engine or a complex system such as an airport) would have precise rules for the attributes, operations, and interactions. These rules could be stated in natural language, or (more preferably) in mathematical rules. In any case, a successful model is based on a concept known as *abstraction*, a technique widely used in object-oriented development. "The art of modeling is enhanced by an ability to abstract the essential features of a problem, to select and modify basic assumptions that characterize the system, and then to enrich and elaborate the model until a useful approximation results [3]."

However, we do not have a perfect world. Parts of the simulation might not have well known interactions. In this case, part of simulation's goal is to determine the real-world interactions. To make sure that only accurate interactions are captured, the best method is to start with a simple model, and ensure that it is correct and representative of the real world. Next, increase the interactions and complexity iteratively, validating the model after each increment. Continue adding interactions until an adequate model is created that meets your needs.

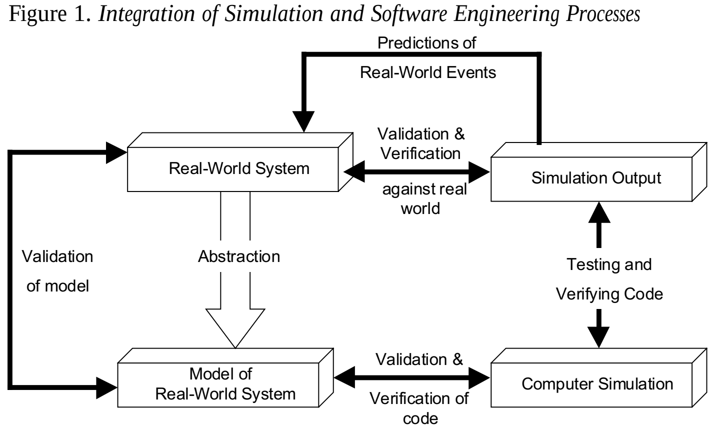
Unfortunately the previous description implies that you have clearly identified needs. This requires valid requirements. It also requires planning for validation of the model. As in creating any software product, requirements and needs must be collected, verified, and validated. These steps are just as important in a simulation as they are in any system. A system that is unvalidated has not been *field-tested* against the real world and could produce invalid results. Abstraction and validation are equally necessary to create a reliable model that correctly reflects the real world, and also contains all attributes necessary to make the model a useful tool for prediction.

The steps of abstraction and validation are in themselves, however, not totally sufficient to create a valid and usable model. Other steps are necessary to create a model that is of sufficient detail to be useful. These steps include planning, data acquisition, and model translation. Figure 1 shows how these steps are related. For a more detailed explanation, the reader may consult [4].

If the model is to be credible and a predictor of future behavior, it is critical that you validate it. This validation must take place at two different times. First of all, you must validate the model against the real world [5]. In addition, you must also validate the model again after the simulation code has been created, as Banks and Carson recommend. At this time, the simulation output will help you revalidate the model against the real world. However, my experience is that this second validation (after coding) tends to focus on validating the code against the model, with little emphasis on revalidating the model against the real world.

It is my opinion that valid software engineering practices dictate an emphasis on reviewing and validating the model prior to and also after coding. Even if sufficient tools and output artifacts

Figure 1. *Integration of Simulation and Software Engineering Processes*



are missing, the advantages of validation on the model prior to coding would outweigh the time and effort involved. A reevaluation of the validity of the model after coding is required. A. Davis looks at studies that show that the most effective way to find errors in software is to inspect it [6]. He further says that one study " … puts to rest the myth that we should not spend time analyzing nonexecutable forms of the software (for example, requirements and design) because the computer can more easily find errors when it executes the program." The model should be reviewed and inspected thoroughly—before *and* after coding.

The creation of a valid model requires domain experts who understand the workings of the physical system. Software engineers should note that domain expertise is a requirement for a valid model. The domain experts must be available not only to help create the model, but also for all phases of verification and validation. If domain expertise is absent during verification and validation, only verification will be performed. This will result in a system that is consistent and internally correct, but also one that might not actually correspond to (and therefore cannot be used as a reliable predictor of) the real world.

The Simulation Phase

The model is next transformed into code once it has been validated. Several methods exist to do this using either a specialized simulation language or a general-purpose language. Specialized simulation languages, such as GASP, SLAM, SIMSCRIPT, or SIMULA offer programming constructs designed to quickly convert a model into coding. Examples of such coding constructs are queues, random-number generation, event scheduling, event generation, built-in statistical analysis collection tools, and time management tools.

Drawbacks of the specialized language approach include limited availability of hardware and trained programmers, and general drawbacks associated with commercial off-the-shelf (COTS) software. In my opinion, the main drawback of specialized simulation languages is the general lack of flexibility in terms of modifying the system to operate on varying (dynamic) combinations of hardware. In general, special-purpose languages operate on a single hardware platform. Distributed simulations are not possible.

General-purpose languages, such as Ada, C, C++, or even Fortran are suitable for implementing simulations. With the use of predefined packages, templates, or subroutine libraries all of the constructs available in the specialized languages can be achieved in general-purpose languages. In addition, developers can take advantage of language features to distribute complex simulation across multiple platforms. Platforms can be homogenous (all similar hardware) or heterogeneous (dissimilar hardware). Distributing a simulation is important, because simulation can be an extremely time-consuming computational process [7]. Five methods exist for speeding up lengthy simulation runs [8]:

1. Using a compiler or compiler-based optimization technique to produce code that can be executed in parallel.
2. Running separate simulations on distinct systems in parallel, and then combining the results.
3. Using different systems for separate subroutines to support a larger distributed simulation.
4. Having a single computer act as a controller, and distributing simulation events to any available processor.
5. Having many processors, each of which only handles certain components (and events) of the overall simulation (object-oriented simulation).

Method one is useful when specialized hardware is available to permit parallel operations. Many specialized simulation hardware platforms contain multiple processors that permit this type of speedup. A different approach is used in method five. This method is useful for object-oriented distributed simulation.

Just as abstraction is used to create a valid model, abstraction is used to create computer-world objects that model real-world objects. Each object can then be distributed to a distinct processor. To complete the simulation, communications network and simulation protocols allow the objects to interact together. While this method of distributed simulation can be complex, the complexity is largely in the protocol and network. Once these have been established, the objects themselves interact relatively easily. Many protocols for distributed simulation exist and can be tailored to specific needs. See [9] for additional information.

In either case, you must select between general-purpose languages (utilizing libraries and subroutines) or specialized simulation languages. Also, you need to choose COTS hardware platforms or specialized simulation hardware. In any case, the following is a checklist for capabilities needed in choosing a simulation language (based on [10]). This list can also be used for comparing potential languages or language and hardware combinations with each other for their adequacy for your simulation needs.

- Well structured (easy to use and understand) data input and output mechanisms.
- Predetermined time-flow mechanisms.
- High quality random number generation routines (discrete and continuous).
- Clock routines to store, sequence, and select simulated events.
- Automatic statistical collection.
- Ease of use.
- Adequate documentation.

Simulation Output

The criteria of well structured data input and output mechanisms is particularly important is selecting a modern simulation platform. Many modern simulation languages and simulation platforms include real-time output using high-quality graphics. For example, having a constantly updated picture of the airflow around a jet engine during a simulated flight might be preferable to simply outputting airflow data. This visual output can make your simulation easier to verify; a picture is probably much easier to understand and evaluate compared to a large amount of numeric data. Because the output of the simulation is a picture of a real-world entity, domain experts can quickly examine and evaluate the picture without having to understand the output of a computer simulation program. In addition, visual data quickly show the effects of modifications to the model.

If graphical representation of output is important to your application, you must then compare hardware requirements (graphical quality and graphical speed) when selecting your simulation hardware and software. Options range from using the built-in display features of off-the-shelf systems (least expensive, but lowest in speed and quality) to using high-end specialized hardware.

Verification and Validation

First the model is created and validated. Next the simulation is coded, verified, and validated. Only then can meaningful results be obtained. The results of the simulation must be carefully examined against reality. If care is not taken, all the hard work and effort and modeling and simulation can create the computer equivalent of a self-licking ice cream cone. The purpose of a simulation according to Shannon is "understanding the behavior of the system or of evaluating various strategies for the operation of this system." The results have to be examined in light of the model and the physical world (see Figure 2). Tests must be performed to ensure that the model and simulation accurately represent the real world before the simulation can be used to predict behavior.

The tools and techniques necessary to validate a model fall in the realm of mathematics and include such methods as the chi-square goodness-of-fit test and the Kolmogorov-Smirnov test. Such phrases as "the results look quite good" [11] are meaningless. Statements such as "90 percent of the predicted values lie within 5 percent of the observed values" are better, but do not imply whether this is good (adequate) or bad (inadequate). Only a statistician working with a domain expert can make such judgements.

Finally, in situations where observed data cannot be used (for example, when making observations about a jet engine that does not actually exist), trained domain experts may have to extrapolate the quality of the simulation based upon data from a differing (but existing) system.

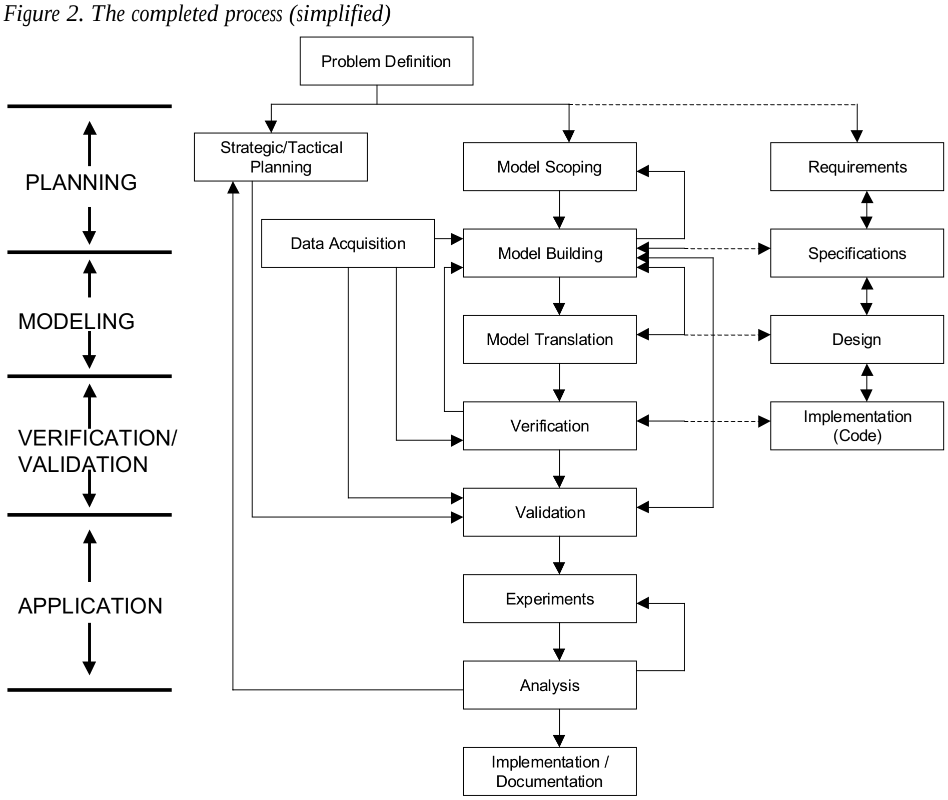
Summary

Simulation can be an effective tool to save money during development of complex systems. Once the problem has been identified, and a system analysis of the overall system has been accomplished, there are only four basic tasks that need to be performed in a simulation [12]:

1. Determine that the problem requires (or would benefit from) simulation. Crucial factors are the cost, feasibility of conducting real-world experiments, variability of the system, and the possibility of mathematical analysis.
2. Build a model of the problem.
3. Write a computer program that converts the model into a computer simulation.
4. Use the computer simulation to resolve the problem.

Although this list is simple, remember mathematical and domain experts must be involved to verify and validate the model. Care must be taken during each step to ensure that the model accurately reflects the real world, and that the code accurately reflects the model. Finally, analysts must examine the output of the simulation to make sure that the results are valid, and can be used to make accurate predictions.

Figure 2. The completed process (simplified)



References

1 Shannon, R.F., Simulation: A Survey with Research Suggestions, *AIIE Transactions*, Vol. 7, No. 3, Sept. 1975.
2. Shannon, R.F., *Systems Simulation*, Prentice-Hall, 1975.
3. Banks, J. and Carson, J., *Discrete-Event System Simulation*, Prentice-Hall, 1984.
4. Hamilton, J., Cook, D., and Pooch, U., A Software Engineering Perspective on Distributed Simulation, CrossTalk, February 1996.
5. Law, A. and Kelton, W., *Simulation Modeling & Analysis*, McGraw Hill, 1991.
6. Davis, A., *Software Requirements: Objects, Functions and States*, Prentice-Hall, 1993
7. Cook, D., Accelerated Time Discrete Event Simulation in a Distributed Environment, *International Journal of System Science*, Vol. 24, No. 3, 1993.
8. Righter, R. and Walrand, J., Distributed Simulation of Discrete Event Systems, *Proceedings of the IEEE*, 1989.
9. Cook, D., Accelerated Time Discrete Event Simulation in a Distributed Environment, Ph.D. Dissertation, Department of Computer Science, Texas A&M University, August 1991.
10. Adkins, G. and Pooch, U., Computer Simulation: A Tutorial, *Computer*, Vol. 10, No. 4, April 1977.
11 Graybeal, W. and Pooch, U., *Simulation, Principles and Methods*, Little, Brown and Company, 1980.
12 Pritsker, A., *The GASP IV Simulation Language*, Wiley and Sons, 1974.

David A. Cook is the chief function officer of Shim Enterprise Inc. He is currently assigned as a software engineering consultant to the Software Technology Support Center, Hill AFB, Utah. He was formerly an associate professor of computer science at the U. S. Air Force Academy (where he was also the department research director), and also a former deputy department head of the Software Engineering Department at the Air Force Institute of Technology. He was a member of the Air Force Ada 9x Government Advisory Group, and has published numerous articles on software process improvement, software engineering, object-oriented software development, and requirements engineering. He has a bachelor's in computer science (University of Central Florida), a master's in teleprocessing from the University of Southern Mississippi, and a doctorate in computer science from Texas A&M University. Cook can be reached at david.cook@hill.af.mil◆

Leading, Integrating, and Leveraging M&S for the Warfighter

Col. Wm. Forrest Crain, U.S. Army
Defense Modeling and Simulation Office

Refocused in 2000, the Department of Defense's modeling and simulation (M&S) efforts have gained direction and magnitude. The Defense Modeling and Simulation Office (DMSO) is leading the implementation of an ambitious set of goals to support the warfighter that has buy-in from the commanders in chief, the services, and the joint staff. The DMSO's bottom line: Lead, integrate and leverage M&S for the warfighter.

The Defense Modeling and Simulation Office (DMSO) is a Department of Defense (DoD) policy and standards office that does not build simulations or, as a rule, develop software. Through a process of collaboration and consensus, the DMSO develops policies, standards, and architectures that allow organizations to build interoperable models and simulations without telling the organizations how to build their products.

Since it was established in 1991 to provide a full-time focal point for DoD modeling and simulation (M&S) activities, the DMSO has been a leader in guiding the community away from the costly practice of building proprietary, *stovepipe* simulations. Until then simulations were not necessarily built to work or communicate with other simulations or systems, nor share or reuse components.

To encourage unity, the DMSO's Focus Call program funded selected M&S projects that could benefit the entire community. Though a step in the right direction, it did not have the desired impact. So in 1994 the DMSO implemented a top-down, strategy-driven investment program aimed at fostering interoperability and reuse through a common technical framework of standards and architectures, and also a set of DoD-provided common M&S services, including help desks, an education program, resource repositories, and supporting software and tools.

The DMSO works for the director, Defense Research and Engineering (DDR&E), in the office of the under secretary of Defense (USD) for Acquisition, Technology and Logistics (AT&L). The deputy DDR&E chairs the DoD Executive Council on Modeling and Simulation (EXCIMS), a general officer-level board of directors.

A New Vector of Direction and Magnitude

In March 2000, the EXCIMS asked the DMSO to look for opportunities to better serve the M&S community. Given the maturity of DoD M&S today and the speed at which technology is advancing, it was time for us to reexamine how we do business, and how we support the warfighter.

Our reexamination was a community effort that spanned several months. We could not have done it without the input of the services, and joint and DoD M&S organizations who are our partners in supporting the warfighter. The EXCIMS approved our plans in August 2000 for moving forward on our new vector. Since then it has evolved and gained direction and magnitude and we—the M&S community collectively—are enthusiastically implementing it.

The DMSO has been talking to the warfighting commanders in chief, and I think it is having a positive effect—they are telling us what they need, what we are doing that is helping, and what we are doing that has not added much value. One of the good things we are hearing from the warfighters, our service counterparts, and other DoD leaders and staffs is that they feel we are on the right track. All the comments above encompass feedback we need to be effective.

We used that feedback to build our plans and budget for fiscal year 2001. A vision emerged in reexamining our role that defines what we think we need to do. That vision is "to lead and integrate the DoD M&S community and leverage M&S science and technology (S&T) advances to ensure that the warfighters of today and tomorrow have superior and affordable M&S tools, products and capabilities to support their missions, and to give them revolutionary war-winning capabilities."

Put more simply, "to lead, integrate, and leverage M&S for the warfighter."

Col. Steve Collier, deputy director of the Army M&S office, said in a discussion leading to our new vector, "You don't know what you don't know, but sometimes you don't know what you *do* know either." Not having a handle on what we, as a community, do know can be expensive. The DoD cannot afford to have organizations investing unnecessarily in redundant capabilities. And the warfighter does not have the time or resources to look any farther into the future than his focus on near- and mid-term operations and planning.

There are things that the warfighter does not know he needs. Once he gets them, he will wonder how he lived without them. For example, in 1975 I did not know I needed a computer in my home. I paid my bills, balanced my checkbook, wrote letters, kept my calendar, played with my investments and did my taxes by hand on paper. Now I do not know what I would do without my computer to do all those things. If you took it away, it would mean a major readjustment for me.

That same thing is true for the warfighter. There are technologies percolating out there that have great potential. The DMSO is well suited to discovering them. We are the warfighter's eyes and ears across the spectrum of time for science and technology developments that have M&S potential—from the present to five or more years in the future. But we need to know what his M&S requirements are to guide our search.

We are task organized to do that, as well as to continue providing the M&S community with those DoD-furnished services and resources that are part of the common technical framework. There is a circular flow in the process of identifying near- (present to three years), mid- (three to five years), and long-term (beyond five years) requirements and finding the scientific and technological solutions for each time frame. The process is continuous. M&S capabilities identified and capitalized on in the mid- and long-term will eventually move into the near-term as time passes, and they reach maturity.

Warfighter experience with those once future, but now present, solutions will likely contribute to new mid- and long-term requirements. At the same time, the common services that the DMSO provides to the community overlap the three time frames. This continuity allows the collective M&S community to share information throughout the life cycle of a product or capability from concept to fielding.

The DMSO's Warfighter Requirements Division, which is responsible for Command, Control, Communications, Computers, and Intelligence (C4I)-to-simulation cognizance; joint activities and programs; service coordination; and verification, validation and accreditation focuses on M&S needs and efforts from the present to three years out.

The Enterprise Division—composed of current DMSO programs like data standards and functional descriptions of the mission space (previously conceptual models of the mission space), integrated natural environment, human behavior representation programs, and the high level architecture for simulation—focuses on M&S needs and efforts three to five years out.

The S&T Division concentrates on finding promising M&S tools, products, and capabilities that are on the drawing boards five or more years out.

The Concepts Application Division provides resources such as the M&S Information Analysis Center, the M&S Resource Repository (MSRR), the M&S education program, and proactive outreach programs that provide a conduit for the M&S community to share information throughout the life cycle of a tool, product, or capability from concept to fielding. The division also serves as the DMSO's liaison with the simulation-based acquisition community.

The DoD Approach to M&S

DoD M&S efforts are mainly focused in three functional areas—training, analysis and acquisition—with more and more attention being paid to experimentation. The EXCIMS oversees three subordinate functional area councils that respectively work function-specific issues with the services and DoD components. The DoD is developing major joint simulations to support users in each area—the Joint Simulation System for training, the Joint Warfare System for analysis, and the Joint M&S System for acquisition.

Training improves performance.

Military forces deployed worldwide in support of U.S. security interests are under increasing pressure to protect the environment, reduce cost, and increase safety. While our new weapons systems are becoming more complex and lethal, our range facilities and live training opportunities are diminishing. Increased use of simulations and simulators is essential to maintaining force readiness. Despite the fact the United States is at peace, the number of simulation-supported training events is steadily growing. Twenty-two simulation-driven joint training exercises were conducted last fiscal year. These exercises train combatant command, joint task force, and service component staffs, many of which participate in more than one exercise each year. It is cheaper, safer, and reduces environmental impact to move electrons, rather than troops, around the globe.

Analysis improves decision-making.

Analysis simulations are workhorses. We have used them for force structure analysis to work the force cap in Bosnia. We use them for course-of-action analysis for operations plan development in real-world actions like Kosovo, and war plans at the commander in chief (CINC) and component command level. We use them for analysis of alternatives in weapon system(s) and organizational development. And we use them for mobilization, deployment, logistics, and sustainment planning. Finally, we are looking at how we can use analysis simulations to support small-scale contingencies and operations other than war, like the de-mining effort in Bosnia-Herzegovenia and resettlement of displaced persons, refugees, and evacuees.

Acquisition improves system(s) design.

The DoD Simulation-Based Acquisition program promotes the use of simulations throughout the life cycle of every weapon system and among all weapon system development programs. There are tremendous time and dollar savings to be had, and perhaps zero environmental impact and safety worries if we can wring out systems and certify their performance using M&S before we ever bend metal in production. The Army's Crusader, a revolutionary cannon artillery system of systems, is using more than 150 simulations. None of them were developed specifically for Crusader, which is an excellent example of how simulation reuse can save time and reduce costs. The program's integrated data environment, which makes extensive use of simulations, allowed the Crusader team to re-baseline the system in a matter of weeks when they had to trim more than 40 percent of its weight in order to preserve its role in the Army's future lighter force structure.

Experimentation helps us explore, develop concepts.

The U.S. Joint Forces Command (JFCOM) was formed in October 1999 with a charter to conduct joint experimentation. This experimentation will help put meat on the bones of the concepts outlined in the joint chiefs' of staff far-looking Joint Vision (JV) 2010, and more recent JV 2020. The Air Force's Joint Expeditionary Force Experiment (JEFX) in September 2000 was its third annual large-scale experiment combining live aircraft, M&S, and technology insertion to explore and evaluate new processes in the development of operational concepts and doctrine for the new Air Expeditionary Forces. Air Force Col. Kevin Dunleavy, JEFX 2000 director, summed it up well when he said, "[Air Force experimentation] provides a means to assess new technologies and operational concepts, allows warfighter involvement early in the acquisition process, and produces better informed investment decisions." JEFX 2000 served as the Air Force portion of Millennium Challenge 2000, the JFCOM's first major joint experiment.

International Activities

U.S. warfighters have to be able to operate in multinational coalitions as well as jointly. So it behooves us to work with our allies to make our M&S systems interoperable. As world events become more complicated, it is essential that we work and train together without the necessity of actually being in the same location—moving troops and materiel costs money; training and maintenance monies often compete with real-world operations costs.

The DMSO helped NATO produce an M&S Master Plan (MSMP) for the alliance. We helped to establish a DMSO-equivalent NATO M&S group and a subordinate M&S Coordination Office to work on alliance M&S issues. NATO has adopted the high level architecture (HLA) as its simulation architecture, and we have assisted in a number of demonstrations and exercises for further understanding of the standard. We also assisted in the development of a NATO M&S education course for staff officers.

> "There are technologies percolating out there that have great potential. The DMSO is well suited to discovering them. We are the warfighter's eyes and ears across the spectrum of time for science and technology developments that have M&S potential—from the present to five or more years in the future. But we need to know what his M&S requirements are to guide our search."

The DMSO has provided M&S education, training, and technical exchanges with our defense counterparts in Australia and Korea. Our foreign outreach efforts include participation in simulation and training conferences in Europe, Australia, and South Korea, where we have had the opportunity to inform military and industry representatives about our programs.

Where Are We Headed in 2001?

Our fiscal year 2001 program was developed with feedback from the CINCs, the services, and the joint staff. It was approved by the EXCIMS. Here is our program for fiscal 2001:

- **Begin HLA technology transition.** The HLA is a general-purpose architecture for simulation. It is the cornerstone of the common technical framework. It supports reuse and interoperability across the large numbers of different types of simulations developed and maintained by the DoD. Besides being the DoD, and now NATO, standard, it has been accepted by the Object Management Group, an international standards organization. In September 2000, it was accepted by another international standards organization, the Institute of Electronic and Electrical Engineers (IEEE). We have begun to transition the HLA to a sustainment phase in 2001. Reducing HLA support costs allows the DMSO to redirect dollars to other activities. We continue to do bug fixes (and performance enhancement as required) types of support, but we are not developing any new tools. However, we still have some work to do to ensure the DoD M&S community's needs are met now that the HLA is an IEEE standard.
- **Improve M&S service to the warfighter.** The M&S needs and priorities of every warfighting CINC have been incorporated into a Web-based, interactive database with keyword search capability. If someone in the Pacific Command wants to know about verification, validation, and authentication (VV&A), the search turns up VV&A efforts across the M&S community. Unlike the MSRR, which is more content or product oriented, this database is requirements based. Having the CINCs' prioritized requirements will be a big help as we revise the DoD M&S Master Plan[1] (MSMP) in 2001.
- **Expand the S&T Initiatives Program.** Thirteen S&T projects were selected and funded in fiscal year 2000 by the DMSO and the service and joint M&S offices. At press time the prospects for fiscal 2001 look even better. We received submissions from more than 50 academic, industry, and government organizations in response to our request for information on advanced M&S technologies. We used that information to lay out the fiscal 2001 M&S S&T Initiatives Program and issue a request for proposals in October 2000.
- **Address challenges and coordinate Integration Task Force (ITF)results.** One of our tasks from the EXCIMS is to develop an integrated M&S strategy for the DoD. Because part of determining what needs to be done is knowing what we have and have not accomplished, we are also assessing how successful we have been at meeting the objectives prescribed in the current DoD MSMP (published in 1995). This task, which falls to an Integration Task Force composed of DMSO, DoD, joint, and service representatives, will provide the foundation for the MSMP revision.
- **Continue investments based upon warfighter needs, ITF results, and S&T initiatives.** One of the things we hear about from CINCs is the void in simulation support for operations other than war. We are already looking at how we can support this requirement in 2001. We are also looking at federating joint simulation systems (JSIMS) and joint warfare simulations (JWARS) to enable CINCs to use JWARS as a course-of-analysis tool with real-world data piped through JSIMS C4I interfaces. The S&T division will look at next-generation simulations that are agent-based vice algorithmic. Agent-based simulations use environment to constrain behavior whereas algorithms do that job in our current simulations.
- **Continue to upgrade M&S education, and mature service academy education partnerships.** We recognize the need to start laying a foundation of M&S knowledge in the officer corps at the earliest opportunity and are working with the services and respective academies to promote M&S education in each school. In 2001 we will implement an M&S intern program for cadets; present a number of M&S courses at all three academies; take a look at how we can sponsor visiting M&S professors at each academy; and lastly, expand the over all effort to the services' ROTC programs in fiscal 2002.

Future Challenges

Looking beyond 2001 I think limited resources and keeping up with technology are two challenges that will continue. But I do not think they will thwart our continued use of an entrenched, value-added tool like M&S.

M&S is a combat multiplier. Our use of it is only going to increase. Our U.S. leaders know the value of M&S in maintaining readiness, and their support is evident in the continued funding provided for programs across the DoD components and for big-ticket systems like JSIMS. Further, as stewards of the taxpayer's dollars; the military will always be driven by the need to find a cheaper, better, safer, and environmentally sounder way of doing things. M&S lets us do that.

Rapid technological advances figure heavily in the challenges facing the warfighter. The array of issues facing our forces continues to expand while the time they have to prepare and respond is shrinking. Information availability for both sides of a conflict is increasing exponentially. The common denominator in all of this is human capacity.

The gap between what we can do and what we have to do is widening. We cannot wait for the latest and greatest technology to emerge before we start building our M&S tools. The warfighter will go to war with the training and equipment he has now, not what is on the drawing board. We have to put a stake in the ground and start building, incorporating newer and improved technology as resources and time permit.

As the DMSO implements its vision in 2001 and beyond, we hope not only to anticipate selected technology leaps, but also to shorten the maturation time of M&S technologies so the DoD can incorporate them into planning and acquisition cycles for current or future programs.

Your Feedback

I invite CrossTalk readers to visit our Web site at www.dmso.mil for more information about our programs and progress to date. The warfighters' return on investment in our efforts will be measured in how well we pursue our goals and implement our programs. If you have comments or an idea that will help us better serve the warfighter let us know. Contact information and the DMSO staff's e-mail addresses are available on our Web site. If you do not know whom to contact, send a note to ask@dmso.mil —we will sort it out and get back to you. Give us your input.◆

Note

1. Department of Defense, DoD 5000.59-P, *Modeling and Simulation (M&S) Master Plan*, October 1995.

Additional Reading

Department of Defense Directive 5000.59, Subject: DoD Modeling and Simulation (M&S) Management, January 4, 1994.

Department of Defense, DoD 5000.59-M, DoD Modeling and Simulation (M&S) Glossary, January 1998.

About the Author

Colonel Wm. Forrest Crain assumed directorship of the Defense Modeling and Simulation Office in March 2000. He received his commission from the United States Military Academy in 1975. He is currently pursuing a doctorate in information technology at George Mason University. He has served in a variety of combat arms positions throughout his career. Before joining the DMSO he served as chief of Strategic Plans for the Multi National Division (North) in Bosnia.

Defense Modeling and Simulation Office
1901 North Beauregard Street, Suite 500
Alexandria, Va. 22311-1705
Voice: 703-998-0660
Fax: 703-998-0667
E-mail: wcrain@dmso.mil
Internet: www.dmso.mil

Coming Events

February 7-9
Network and Distributed System Security Symposium
www.isoc.org/ndss01/call-for-papers.html

February 12-16
Software Management Conference
www.sqe.com/sm

February 12-16
Applications of Software Measurement Conference
www.sqe.com/asm

March 5-8
Software Testing Analysis and Review
www.sqe.com/stareast

March 5-8
Mensch and Computer 2001
http://mc2001.informatik.uni-hamburg.de

March 12-15
Software Engineering Process Group Conference
www.sei.cmu.edu/products/events/sepg

March 31-April 5
Conference on Human Factors in Computing Systems
www.acm.org/sigs/sigchi/chi2001

April 22-26
Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies
www.ieee-infocom.org/2001

April 29-May 3
Software Technology Conference (STC 2001)
www.stc-online.org

May 1-3
2001 IEEE Radar Conference
www.atlaessgrss.org/radarcon2001

May 12-19
23rd International Conference on Software Engineering, and *International Workshop on Program Comprehension*
www.csr.uvic.ca/icse2001

June 11-13
E-Business Quality Applications Conference
http://qaiusa.com/conferences/june2001/index.html

June 18-22
ACM/IEEE Design Automation Conference
www.dac.com

June 25-27
2001 American Control Conference
www.ece.cmu.edu/~acc2001

Does Jerry Weinberg read CrossTalk? See www.geraldmweinberg.com/ReadingStuff/Each_Reading.html/Book%20Reviews/CrossTalk.html

Integrating Process Simulation and Reliability Models

Ioana Rus
Fraunhofer Center for Experimental Software Engineering

James Collofello
Arizona State University

The problem addressed here is predicting how the reliability of a software product, as well as project cost and schedule, will be influenced by adopting different defect prevention and detection activities. The solution we propose using is a software process simulator for estimating the impact of different software reliability engineering practices. We have created a prototype simulation model of the dynamics of defect evolution (introduction, detection, and removal) and of the process factors that influence it throughout the entire development lifecycle. Our simulation model relates defects to failure occurrences by integrating existing reliability estimation and prediction models in the system-testing component of our system dynamics simulator. This article describes our model and presents a hypothetical example illustrating how an organization could use our simulation tool to make decisions regarding reliability strategies. The model is available free of charge to readers willing to work with us on our ongoing research on the impacts of process choices on reliability, cost, and schedule.

A reliability strategy is a set of software engineering practices defined for each project by combining different reliability achievement and assessment activities and methods, according to the software reliability goal and project's characteristics. In [1] is a description of a decision-support system for reliability strategy selection based on a set of product, project, and resources decision factors.

There are two main approaches to achieving high software reliability:

1. Avoiding defects in the final product.
2. Using fault tolerance methods.

Fault avoidance can be achieved by using fault prevention and fault detection and correction methods. Fault tolerance allows the system to continue to operate in the presence of latent faults, enabling the whole system to function as required.

Once the reliability strategy is selected, it must be assessed in terms of the projected product reliability, budget, and schedule constraints. One approach for strategy evaluation is process modeling and simulation, which involves analyzing the software development process, creating a model of the process, and then executing the model and simulating the real process. The simulation model presented here is meant to be a tool that supports reliability prediction as well as cost and schedule estimation. It helps the user to forecast the impact of different reliability practices not only on software reliability (effectiveness), but also on cost and development time (efficiency).

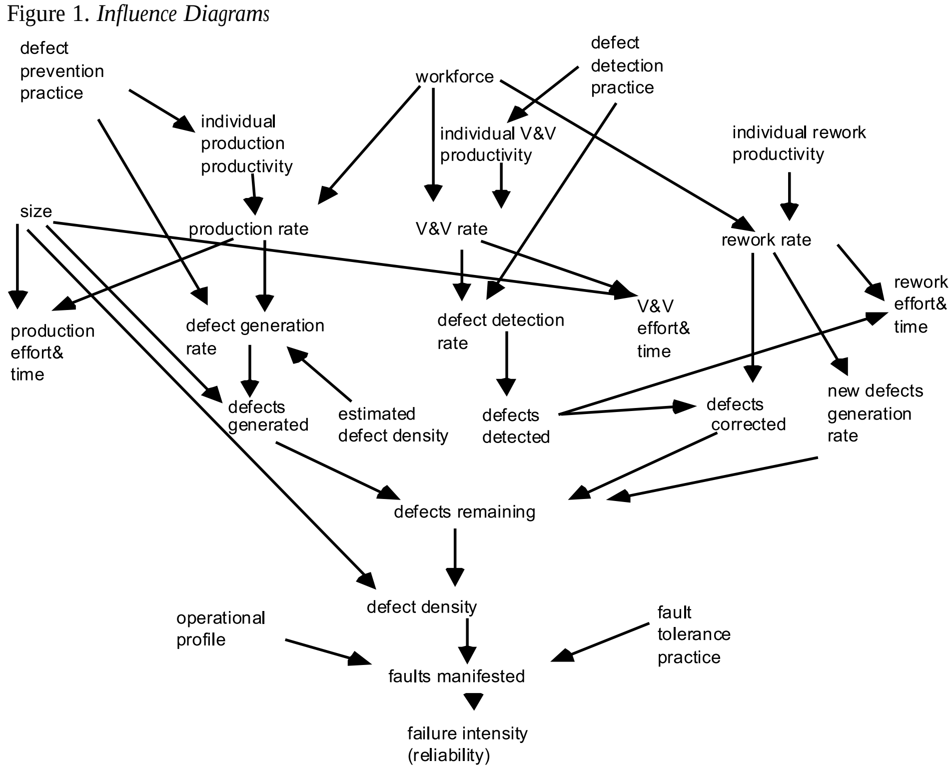
Reliability Practices

A model is an abstraction of a real object or system. Modeling a system means capturing and abstracting the system's components, relationships, and behavior, according to the modeling objective. The goal of developing this model was to capture the impact on quality, cost, and development time and effort of different software reliability practices during the development process of a software product. Currently our prototype model addresses new product development from inception to delivery and does not address maintenance.

The influence of defect prevention, detection, and fault tolerance practices on product and process variables captured in our model are shown in the influence diagram in Figure 1. Defect prevention practices contribute to the reduction of defects injected in the product but also might reduce the development productivity. Therefore, applying these practices could take more time and effort in production, but the time and effort spent later for detecting and reworking defects is reduced. Defect detection practices increase the number of defects detected and do not let them propagate to subsequent phases, where they are more expensive to remove.

Reducing the number of remaining defects in the final product results in increased reliability manifested in decreased failure intensity. Fault tolerance practices mask some of the remaining defects and do not allow them to manifest when the software is executed, so the number of failures is reduced. Testing with operational profiles will uncover the faults that are more likely to be encountered in

Figure 1. *Influence Diagrams*



the operational phase. More details about modeled variables relationships, inputs, and outputs can be found in [2].

Our model was designed to be modular; it consists of components corresponding to the requirements, design, coding, and system testing phases. Each component addresses both managerial and development aspects of each phase. Managerial aspects include effort and staff allocation, activity duration, cost, activity progress, and productivity for production, quality, and rework activities. Development aspects include modeling work product production as well as defect flows.

Figure 2 illustrates our modeling of the system-testing phase. The activities are test case execution, fault identification, rework, and regression testing. Test cases are executed in order to uncover remaining defects given testing equipment constraints (e.g., CPU execution time). When a failure is encountered, the corresponding faults and defects are identified (fault identification) and then corrected (rework). It is assumed that regression testing is performed periodically to detect possible new defects introduced (bad fixes) that are also subsequently corrected. Defects that do not manifest by causing failures and therefore are not detected during testing will remain in the delivered product, affecting the field value of software reliability.

System Dynamics Modeling Simulator

Our model was implemented using the system dynamics modeling (SDM) approach. System dynamics is defined as "the application of feedback control systems principles, and techniques to modeling, analyzing, and understanding the dynamic behavior of complex systems [3]."

The premise of SDM is that the dynamic behavior of a system is a consequence of its structure. SDM models the behavior of a system based on *cause-effect* or *influence* relationships between entities observable in a real system. These relationships constantly manifest while the model is being executed; thus the dynamics of the system are being modeled. One of the most powerful features of SDM is realized when multiple influence relationships are connected forming a circular relationship known as a feedback loop—a concept that reveals that any actor in a system will eventually be affected by its own action. The tool support existing for this modeling approach allows the computer model to be executed, thus simulating a real project [4].

Construction of the simulation model involved representing each of the software development and testing process activities using continuous modeling in simulation environment, along with process metrics for each activity corresponding to productivity and quality. For example, the test cases execution activity from Figure 2 would be represented with metrics corresponding to execution time and effectiveness in terms of identifying failures.

To examine the effect of defect avoidance practices on reliability, we integrated SDM with existing reliability prediction and growth models. To model the evolution of the failure rate and the number of remaining faults during system testing, we used an equation from an existing Poisson-type exponential-class model [5].

FailureRate = InitialFailureRate x (1- Failures / TotalFailures)

In the context of system dynamics simulation, which is continuous simulation (as opposed to discrete event simulation), *rate* means the change in value for a variable (number of failures in this case), from time t to time t+1. The interval [t, t+1] is the simulation time interval. In a continuous simulation, time increases with constant increments. InitialFailureRate is the failure rate at the beginning of system testing. Failures is the number of failures encountered from the start of system testing to time t. TotalFailures is the total number of failures expected.

To account for the influence of specific test process factors on failures' dynamics, we changed the above formula to:

FailureRate = InitialFailureRate x (1- Failures / TotalFailures) x Fct(TestEffect) x Fct(NrTestCasesEx)

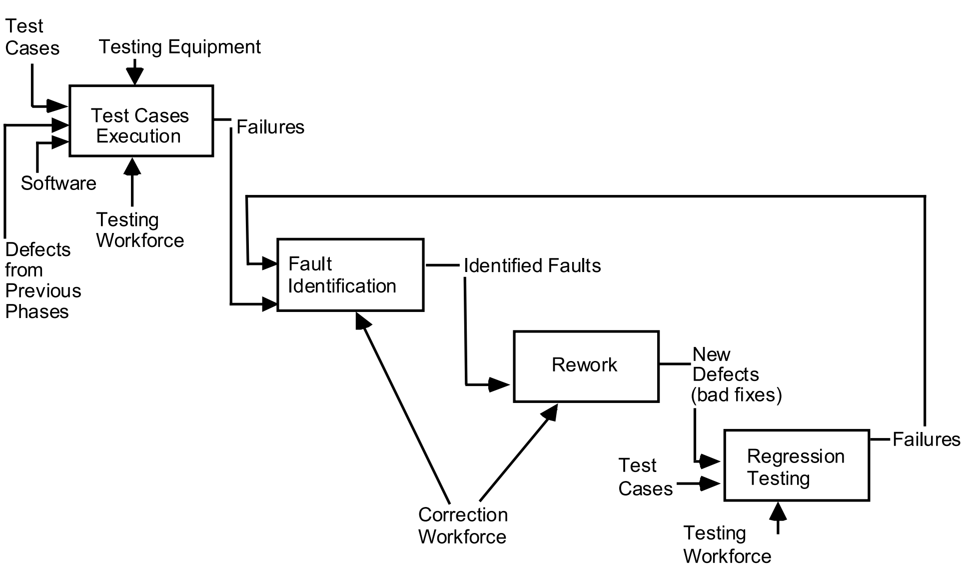
where Fct(TestEffect) is a function of the effectiveness (capability to uncover defects) of the test cases, and Fct(NrTestCasesEx) is a function of the number of test cases already executed. For our model, we assumed that failure intensity decreases with the increase of the number of test cases executed (in time, fewer and fewer failures are detected, and the failure rate decreases until its variation becomes extremely slow). The model's users should define these two functions according to historical data collected for their own projects. This is a part of the model calibration for a specific application, company, or type of projects.

For the InitialFailureRate model, users can input the value, if they know it, or an existing prediction model can be used. We used the Rome Laboratory model [6] that predicts the InitialFailureRate as a function of the remaining defect density (defects per size units, e.g., LOC) at the beginning of system testing.

InitialFailureRate = C x RemainingDefectsDensity

Here C is a constant called transformation ratio, that depends on the application type, and was empirically determined as presented in Table 1.

Figure 2. *Activity based description of the system testing phase*



Application Type	Transformation Ratio
Airborne	6.2
Strategic	1.2
Tactical	13.8
Process control	3.8
Production Center	23
Developmental	Not available
Average	10.6

Table 1. *Fault density to failure rate transformation[6]*

Simulator users can replace the formula and table above with any other model that best describes their project.

Example: Simulator Use

By using the modeling environment [4], we implemented a computer process model that can be executed, simulating the behavior of the real process. To use the simulator, an organization must first customize the model (ensuring that the appropriate processes are represented) and populate it with their own metrics. Our current model follows a generic waterfall life cycle and is populated with default metrics based on industry averages. Once a model has been customized to an organization, specific project data can be entered such as project size, work force, target delivery date, etc.

To exemplify using the simulator, this section presents the execution of what-if scenarios for predicting how varying defect detection and correction activities could impact upon defects dynamics, reliability, effort, and schedule. We present the results of simulating two hypothetical projects that will be called Case1 and Case2 created by using industry benchmark data from [7]. The common characteristics of both projects are shown in Table 2 and serve as inputs to the simulation run.

The values for the other simulator inputs, such as effort and resource allocation, productivity, verification and validation (V&V), and correction rates, and efficiency of V&V activities, are found in [2].

V&V and correction activities are performed in requirements, design (reviews or inspections), and coding (reviews or inspections, and unit testing) for both cases. The difference between Case2 and Case1 is that the effort initially allocated in the design phase for V&V activities and rework is reduced to 20 percent in Case2 from that used in the Case1. The remaining 80 percent of the design V&V and rework effort is instead allocated to the system testing phase for Case2. This simulation is performed to investigate whether allocating more effort in defect detection and correction earlier in the life cycle improves the reliability of the final product and ultimately saves effort and time.

Size of final delivered software	130 KLOC
Programming language	C
Requirements size	1,000 function points
Estimated completion time	3 years
Work force size	5-18 people over phases
Required defect density in final product	at most 1.3 defects / KLOC

Table 2. *Simulation inputs – Common characteristics of Case1 and Case2*

The outputs of the simulation are presented in Table 3. There is a dynamic evolution of defects and failures for Case1 and Case2. Defects generated in different phases (requirement defects, design defects, and coding defects) evolve in time throughout the development life cycle; their number increases or decreases as they are introduced, detected, corrected, and reintroduced by bad fixes. Due to the fact that V&V activities and rework are performed in all phases, a significant number of defects are discovered and corrected before system testing begins. The number of defects detected for Case2 in the design phase is smaller than the same number for Case1 (as expected). Hence, the number of coding defects for Case2 is also higher than for Case1. That is because design defects, if not detected in design, will generate coding defects. Therefore, the total number of defects in the product at the beginning of system testing is around 330 for Case1 and 1,100 for Case2.

These cases present the variation of failures encountered and estimated to remain in the product during the system-testing phase. Although more effort was allocated to system testing for Case2 and the number of failures encountered by the end of system testing (project completion) is higher for Case2 (841) than for Case1 (270), the number of estimated remaining failures is 598 for Case2, much higher than for Case1 (197).

These results show that for the project modeled here, allocating more effort to early defect detection activities will improve reliability. The results of simulating both Case1 and Case2 are summarized in Table 3.

Table 3. *Simulation Outputs: Effort, Time, and Quality Data*

Project Parameter	Case1	Case2
Effort (staff hours)		
Coding	2,670	2,470
V&V in Design	265	175
Coding	830	870
System testing	21,710	23,416
Total V&V effort per project	*22,880*	*24,536*
Rework in Design	2,300	1,000
Coding	4,905	4,962
System testing	3,820	8,495
Total rework effort per project	*11,370*	*14,802*
Total effort per project	**41,035**	**45,923**
Time to complete (days)		
Design	69	57
Coding	113	109
System testing	378	451
Total project duration	***593***	***650***
Remaining defects in the product		
Design defects	22	92
Coding defects	151	430
Total defects delivered	173	522
Defect density in delivered product	1.3	4
Reliability		
Failures encountered	270	841
Estimated failures remaining	197	598

Since the requirements phase is identical for the two projects, only the data that are different for the two cases, corresponding to the design, coding, and testing phases are presented.

From the simulation output data in Table 3, the following observations can be made:

- Although the overall effort initially estimated and allocated is the same for the two projects, due to a different effort distribution to defect detection and correction activities over phases, the overall consumed effort is about 4,900 staff hours greater for Case2 than for Case1.
- Case2 is completed 57 days later than Case1.

These results confirm, for the simulated projects, the hypothesis that allocating more effort earlier in the development will eventually save effort and time.

An unexpected result of the simulation, observable in Table 3, is that for Case2 the actual effort to produce code is smaller than for Case1, although the same value would be expected. This is an example of what is called model's surprise behavior. Revealing these results and getting people to think about the causes of this behavior and to understand the complex relationships of a development project is one of the benefits of a system dynamics simulator.

In this case, there may be several explanations for the reduced coding time. One possibility might be that since V&V and rework during design take longer than planned, in the coding phase people would work under pressure and complete the implementation faster. This might also explain why the number of code defects introduced is higher for Case2 than for Case1, since schedule pressure often could reduce coding quality.

Although we can obtain numerical values (as shown in Table 3) as the output of the simulation, system dynamics models are best if used not for point predictions, but for analyzing trends of behavior, as reflected in the graphs that may be seen on the Web version of this paper [available at www.stsc.hill.af.mil].

Conclusions

We presented a process model and simulator valuable for:

- Increasing *understanding* and *communication* about the software development process structure, relations, and behavior; the influence of the software reliability engineering practices on reliability and process parameters; and more insights into the trade-off between reliability on one side and cost and time on the other side.
- Providing an experimentation tool that helps in decision making to improve *planning* and *tracking* the development process.
- Enabling *behavior estimation* prediction, i.e., trends in the dynamic evolution of a set of project parameters such as defect and reliability prediction.

The model was developed for a project that follows a waterfall life cycle process (but can be adapted to a different life cycle) and is calibrated for a hypothetical project. To utilize the model in a real project, it has to be calibrated to that specific environment. Calibration has several levels, from modifying numerical values of some variables, to changing relationships and equations, or even the structure of the model. The more the model is used and tuned to an organization and application domain, using their own historical data, the more accurate and valuable the model and the simulation will be. The model is available to organizations at no charge. Make inquiries to Dr. Ioana Rus.◆

References

1. Rus, I. and Collofello J.S., A Decision Support System for Software Reliability Engineering Strategy Selection, *Proceedings of COMPSAC*, Scottsdale, Ariz., October 1999.
2. Rus, I., Modeling the Impact on Cost and Schedule of Software Quality Engineering Practices, Ph.D. Dissertation, Arizona State University, 1998.
3. Abdel-Hamid, T. and Madnick, S.E., *Software Project Dynamics An Integrated Approach*, Prentice-Hall, Englewood Cliffs, New Jersey, 1991.
4. *Extend – Performance modeling for decision support, User's Manual*, Imagine That Inc., 1995.
5. Musa, J.D., Iannino, A. and Okumoto, K. *Software Reliability Measurement, Prediction, Application*, McGraw-Hill, 1987.
6. McCall, J., Randall, W., Bowen, C., McKelvey, N. Senn, R., Morris, J., Hecht, H., Fenwick, S. Yates, P. Hecht, M. and Vienneau, R. *Methodology for Software Reliability Prediction*, Rome Air Development Center Technical Report, RADC-TR-87-171, 1987, Vols. 1 and 2.
7. Jones, Capers, *Applied Software Measurement: Assuring Productivity and Quality*, McGraw-Hill, 1997.

Note

1. An *error* that is a result of a human action (misconception) results in a *defect* (a product anomaly), that will lead to a *fault* (a manifestation of a software error) that, if encountered, might cause a *failure* (termination of the ability of a functional unit to perform as required).

About the Authors

Ioana Rus is a scientist for Fraunhofer Center for Experimental Software Engineering, University of Maryland. She holds a doctorate in computer science and engineering from Arizona State University. Her research interests include software process improvement, modeling and simulation, measurement and experimentation in software engineering, and artificial intelligence.

Fraunhofer Center for Experimental Software Engineering
4321 Hartwick Rd., Suite 5000
College Park, Md. 20742-3290
Voice: 301-403-8971
Fax: 301-403-8976
E-mail: irus@fc-md.umd.edu
Internet: fc-md.umd.edu

James S. Collofello is a professor in the Department of Computer Science and Engineering at Arizona State University. He received his doctorate in computer science from Northwestern University. His teaching and research interests are in software quality assurance, software reliability, safety, and maintainability, software testing, software error analysis, and software project management.

Arizona State University, Dept. of Computer Science & Engineering
Box 875406, Tempe, Ariz. 85287
Voice: 480-965-3733
Fax: 480-965-2751
E-mail: collofello@asu.edu
Internet: www.eas.asu.edu:80/~csedept/people/faculty/collo.html

Project Scheduling According to Dr. Goldratt

Timothy K. Perkins
Science Applications International Corp.

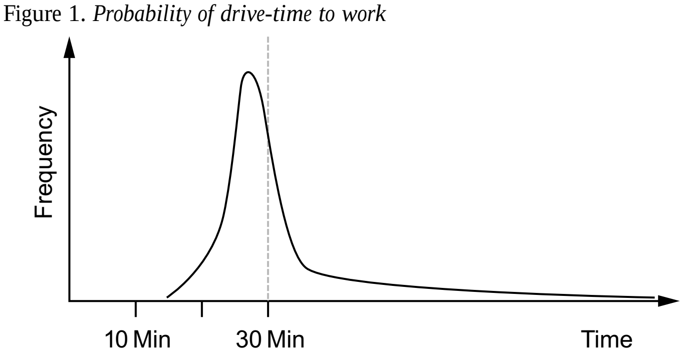
Schedule overruns are more common than meeting projected dates. What can be done to improve? Dr. Eliyahu M. Goldratt provides insight as to how a great amount of schedule safety is incorprated into project plans, yet teams succeed in wasting the safety until they experience schedule overrun. By eliminating or minimizing the causes of schedule waste, the probability of achieving the originally planned project completion date increases.

We are all familiar with the common maladies of cost and schedule overruns that inevitably plague projects, including projects that run long. If they do not, the extended schedule includes enough slack time that considerable waste occurs at each step as idle time. With all the emphasis on process improvement and metrics, why does this still happen? In chapters 6 and 13 of his book *Critical Chain* [1], Eliyahu M. Goldratt offers some ideas regarding the causes of schedule overrun.

Uncertainty regarding one's ability to accomplish a given task within a given time period results in schedule uncertainty. If someone were asked how long it takes for her to drive to work, she may say the average time is 25 minutes. On a good day, perhaps the day after Thanksgiving, she may make it in 15 minutes. If there has been an accident or if it is bad weather, it may take more than an hour. If we were to collect data over a period of time and graph it, the result would be a distribution curve skewed to the right. The area under the curve is the probability of completing the trip within the given time. This is illustrated in Figure 1. The curve shows that the probability of arriving at work in zero time is zero. The probability remains at zero until 15 minutes is reached, increases at a rapid rate, then drops of gradually. Even at the extreme right, the probability is not zero.

Normally when driving to work, we are willing to accept the median travel time unless we are aware of accidents or bad weather. However, when this curve is applied to project scheduling, we are reluctant to agree to a completion estimate near the median, i.e., a probability of 0.5. We usually deal with the uncertainty of achieving the scheduled completion of a project by adding a safety factor that includes the majority of the area under the curve. This increases our probability of completion within the associated time from a probability of 0.5 to 0.9 or higher, much further to the right. (Figure 2) Note that adding this safety factor about doubles the estimated schedule duration. Even with an expanded schedule including a safety margin with a probability of 0.9, organizations seldom deliver on time.

Figure 1. *Probability of drive-time to work*



Safety Mechanisms

Few organizations consciously use the curve below in determining schedule. Goldratt lists three mechanisms they do use to insert safety into time estimates for almost every step of the process. First, time estimates are based on a pessimistic experience, i.e., they select a completion time near the right end of the curve. The reasoning is that there is generally little recognition for early or on-time completion, but there are usually severe repercussions for being late. For self preservation, the right end of the curve is chosen.

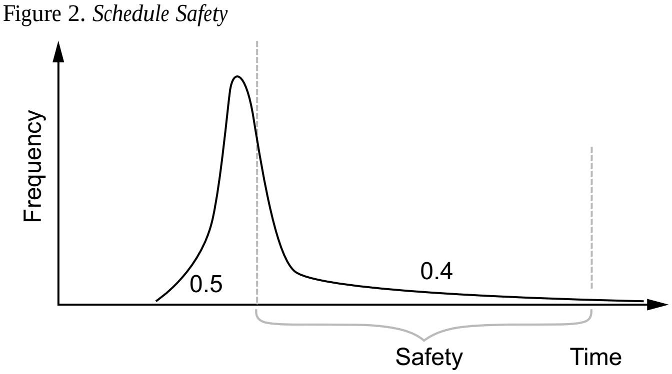
Second, the larger the number of management levels involved, the higher the total estimate. Each level adds its own safety factor for the reasons cited in the first mechanism.

Third, because most individuals have experienced global *peanut butter spread* project cuts, estimators protect their schedule estimations by a large factor. When cuts come, the result is close to the originally desired schedule. But what happens to all this included safety? Why are projects so often late?

There are three basic ways that schedule is lost. First, advantage is seldom taken of early completions. Why? Early finishes are seldom reported because "We will then be expected to finish early on all subsequent projects. Because we cannot guarantee early finishes, we will not report the early finish but use the time to polish the product or add a bell or whistle." However, even if early finishes are reported, the subsequent step may not be prepared to begin. Sometimes the rationale is: "According to the master schedule, we are not to start until a week from Thursday, and we will not be ready until then, even though you have completed the predecessor task today." So the time is wasted. The rule can be stated as: "A delay in one step is passed, in full, to the next step. An advance made in one step is usually wasted."

While the foregoing rule applies to serial processes, parallel processes experience a similar phenomenon. For example, take the case of four parallel steps that are scheduled to end on the same

Figure 2. *Schedule Safety*



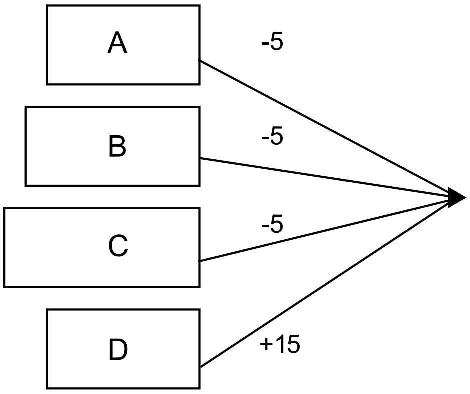


Figure 3. *Parallel Tasks*

date. Three of the four finish five days ahead of time while the fourth is 15 days late. (Figure 3) Statistically, we are on time. In reality, we are 15 days late. The rule for parallel steps is: "The biggest delay will be passed on to the next step. All other early finishes do not count at all."

Second is the *students' syndrome* of lost schedule. Think back to when you were in school. You were given an assignment to hand in a report four weeks hence. When did you begin working on the report? The typical student doesn't get right on it because there is plenty of time. It usually is not until a week before it is due that you start working on it. Only then do you discover that the resource material you counted on is not available, or in our time, the network is down or e-mail is not working. All the safety provided by the instructor has been wasted, and now you are going to be late. This syndrome is alive and well in our projects today.

Third is the multitasking syndrome. Suppose you have tasks A, B, and C to accomplish and each requires 10 days to complete. These tasks do not have to be performed sequentially. You must complete these tasks before others can work on subsequent tasks. These tasks are not necessarily all for the same project. If worked sequentially, task A would complete after 10 days, task B after 20 days, and task C after 30 days. In fact, the schedule that includes each of these tasks has allocated 10 working days for the accomplishment of each of them (see Figure 4). In your effort to please everyone, you decide to work five days on task A, then five days on task B, then five days on task C, and then repeat the entire process to complete. Task A does not finish until 20 days after initiation, task B finishes 25 days into the effort, and task C finishes 30 days into the effort. Did you notice that task A finished 10 days later than if you had worked serially, and task B finished five days later? The entire time from start to finish of each task doubled from 10 days to 20 days. Thus, if subsequent tasks were planning to start 10 days after the initiation of a task, each subsequent task would have experienced a 10-day delay. These delays may be sufficient to cause noncritical paths to become critical paths or to cause a 10-day slip in a critical path.

Multitasking can take the form of meetings, emergencies, other tasks, etc. The impact is that lead time increases for all tasks and completion time is later for all tasks except the final task.

No one wants to finish late. To avoid this, we add considerable safety time to the project. We then succeed in wasting the safety time by not reporting or not taking advantage of early finishes, by embracing the students' syndrome, and by multitasking. This is not to say that bona fide problems do not occur, but such problems rarely account for all the schedule time lost.

All Paths Are Critical

Some may say that they are not concerned because the schedule lost was not on the critical path. However the three ways mentioned above by which schedule is lost occur just as readily on the critical path as on noncritical paths. In fact, given sufficient occurrences of the above ways, a noncritical path may become the critical path. The goal of the project manager should be to maintain or advance the planned schedule rather than to allow schedule float to be consumed to the point that several former noncritical paths now have high likelihood of becoming the critical path.

Our metrics can also mislead us. Goldratt offers two criteria for measurements [metrics]:

- Criterion one: Measurements should induce the parts to do what is good for the system as a whole.
- Criterion two : Measurements should direct managers to the point that needs their attention.

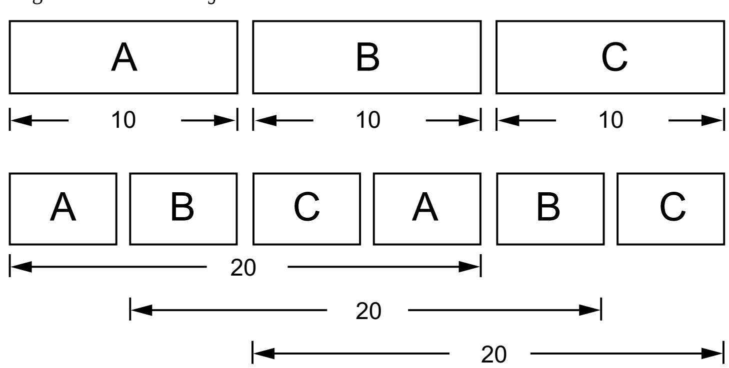
Our project management metrics do not always meet these criteria.

Take the earned value management system (EVMS) for example. We "earn value" as we gain credit for the tasks that were to have been accomplished by a given date, and lose value if we have expended more resources than estimated to complete the task. The mistake made is in grouping all tasks together. If we make significant progress using less than estimated resources on noncritical path tasks, yet are behind on the critical path, the earned value calculation will tell us that all is well. If we are not careful, we will not notice this until it is too late. EVMS reporting should be structured to emphasize the tasks on the critical path while not forgetting the noncritical path tasks. Noncritical path tasks that are late in completion can, by default, become the critical path.

Rethink Project Management

Schedules are difficult to estimate, and more difficult to meet. We have discussed three ways that try to protect product schedule and three behaviors that result in schedule being wasted. Also, we have seen that inappropriate application of project management metrics can lead to a false sense of schedule security. Several of you are probably saying to yourselves, "We already know the problem, what is the solution?" The solution is to rethink how

Figure 4. *Multitasking*



projects have been managed in the past, and to apply the results of that rethinking. Rather than leave the rethinking as an exercise for the student, I recommend reading Goldratt's *Critical Chain* and applying the principles to the project you are managing.

Do Goldratt's principles work? An article published in the January 1999 issue of *Midrange ERP* [2] describes how Harris Semiconductor achieved the design and erection of a building, installation of equipment, hiring and training of employees, and ramp-up to 90 percent of designed production rate in 13 months. The typical industry time is 54 months.

"Applying the management principles suggested by Goldratt can help you achieve your schedule and actually reduce your time to market."

Habitat for Humanity in New Zealand set a new world record for constructing a four bedroom house in 3 hours 44 minutes and 59 seconds, besting the previous record set in June 1998 in Nashville, Tennesee of 4 hours 39 minutes and 8 seconds.

Better On-Line Systems planned a product release for August 1997. Using the principles in *Critical Chain*, they revised their schedule for a May 1 release. Actual release was in early April, five months ahead of the original plan.

Several providers of project scheduling software, including Scitor's PS8 and ProChain Solutions' ProChain [3, 4] have included *Critical Chain* Project Scheduling as a product feature.

Applying the management principles suggested by Goldratt can help you achieve your schedule and actually reduce your time to market. This reduction in time to market results in a more competitive position as well as cost savings from reduced labor expenses. Try it, you just may like it.◆

References

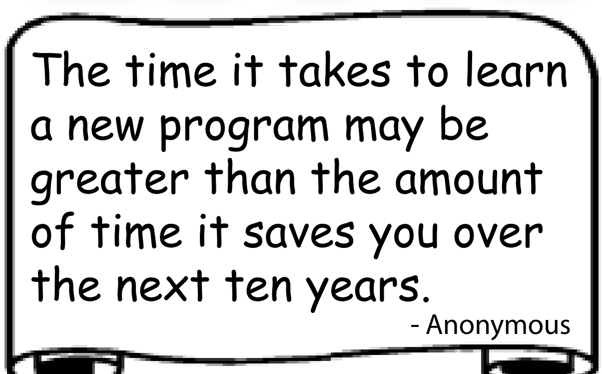
1. Goldratt, Eliyahu M., *Critical Chain*, The North River Press Publishing Corporation, Great Barrington, Mass., 1997.
2. Goldratt Institute Web site (www.goldratt.com), Success Stories.
3. Scitor Web site (www.scitor.com).
4. ProChain Solutions Web site (www.prochain.com).

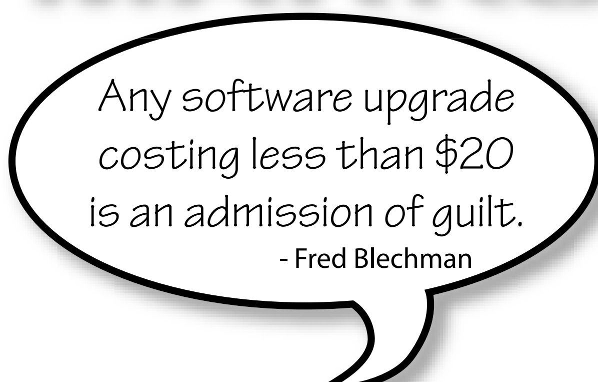
About the Author

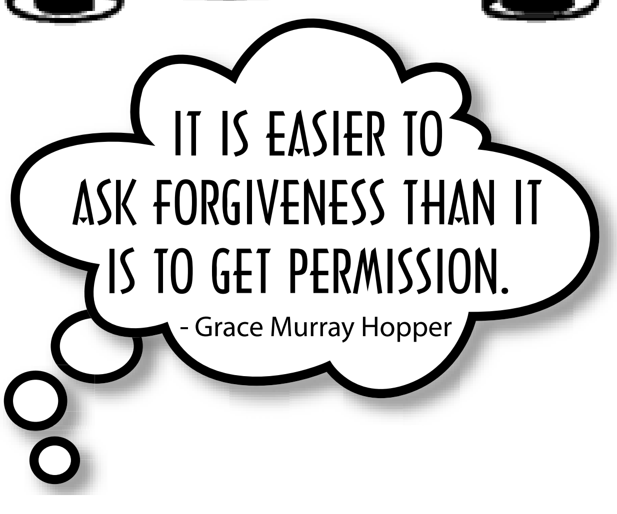
Timothy K. Perkins has been involved in software process improvement for the past 11 years, ever since he led the effort to initiate the software process improvement effort at the (then) five Air Force Air Logistics Centers. As the Software Engineering Process Group leader at the Software Engineering Division at Hill Air Force Base, he led the division in reaching CMM Level 3. The division has gone on to achieve CMM Level 5. Since retiring from the Air Force, Perkins has been employed by Science Applications International Corp. as a process improvement consultant. He currently supports the Software Technology Support Center at Hill Air Force Base in providing consulting services to Air Force and other Department of Defense and government agencies. Perkins holds a bachelor's degree in electrical engineering from Brigham Young University and a master's degree in business administration from the University of Phoenix.

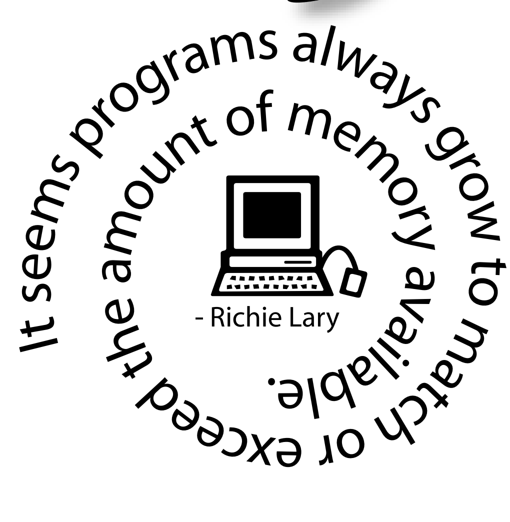
OO-ALC/TISE
7278 4th Street
Hill AFB, Utah 84056
Voice: 801-775-5736
Fax: 801-777-8069
E-mail: tim.perkins@hill.af.mil

QUOTE MARKS









Event-Driven Learning as a Process Improvement Strategy

Michael J. Hillelsohn
Software Performance Systems

Software and systems engineering tasks, with their well defined development life cycles, can incorporate an effective training option—event-driven learning, which improves performance of development teams by providing training just prior to performing a task. The key to event-driven learning is that training is divided into "chunks" that apply only to the next task in the life-cycle phase. Each chunk of courseware is just long enough to teach the new skill, generally one to four hours in length. Since it is critical that the training is part of the development process and on the project time line, the information chunks are discrete and easy to remember and practice. Workers apply the new concepts and skills quickly, thereby retaining the training. A well designed course has the work-team starting on the task during training. Event-driven learning reduces overall training costs by reducing optimizing costs and increasing course development efficiency. It improves job performance through the immediacy of the training that prepares trainees for the specific task, and the task reinforces training that has just occurred. There is no time for forgetting. Event-driven learning capitalizes on two critical learning characteristics: People are motivated to learn because they need to immediately apply the information; and those learning can relate the information to a functional context provided by their jobs.

Event-driven learning is a training delivery approach designed for the process-oriented organization. An organization with well defined processes for all, or even part of its business processes, can increase its practitioners' effectiveness. This improves product quality and optimizes training expenditures by designing training experiences directly into the operational process.

In addition, event-driven learning addresses the organization's need to provide satisfactory training to project staff in a cost-effective manner. In today's organizations the appropriate allocation of overhead dollars is critical to the organization's survival. The modern quality frameworks (CMM®, ISO, Baldrige, SPICE, etc.) all stress the importance of training for staff development and job satisfaction. Rapid technology changes also put pressure on organizations to maintain their staffs' skill levels.

Traditional courses that last several days often use valuable resources and do not provide optimal payback. Course registration and trainee time are both high-ticket items, especially with an increasingly fluid technical environment. Corporate expenditures for training experiences must produce ongoing benefits to both the organization and the staff being trained.

Training *experiences* is intended to be an inclusive term. Event-driven learning capitalizes on the dynamics of just-in-time instruction and uses whatever instructional method is most appropriate for the situation. Technology and instructor-based instruction can be effectively used in an event-driven learning strategy. Computer- and Web-based training are becoming pervasive in organizations with the availability of large libraries of instructional materials acquired from many vendors and combined with desktop delivery capability.

The realism of video-based instruction for modeling task performance and interactions is made more accessible through improved Web delivery of images. The Web itself is a vast encyclopedia of instructional information on just about any topic. It offers the ability to continuously update and disseminate information and performance enhancement tools.

Electronic performance support systems are embedded in many of the products we use and develop to supply information and training opportunities at precisely the moment during task performance when the training is most required. Instructor-led training certainly has not fallen off the radar screen as an effective instructional delivery method, especially as organizations move to more team-oriented management approaches. And document or electronic job aids provide focused instruction for short procedural tasks.

The Times Demand Flexibility

The dynamics of our organizations are changing almost as quickly as the technology that both supports them and is frequently part of their products. People move from job to job more readily than ever before in our history. Increased use of telecommuting and more consultants with fewer full-time staff are pressuring organizations to find new ways to maintain high quality levels in their delivered products and services while being responsive to the needs of a changing workforce. Organizational performance management systems are providing easier visibility into human resource status within the organization. Stability in organizations is based on the way they perform business and produce products. It is the process-oriented organization that will continue to maintain quality. Meanwhile the structure and human resource characteristic of the organization may change. We have all experienced *flavor of the month* approaches to accomplishing business goals. But across all of these approaches, the common themes define and improve organizational processes and satisfy cognitive, skill, and affective staff needs to maintain organizational health.

Event-driven learning is based on the concept that successful organizations have well-defined processes in place. Organizations that specialize in systems and software development generally base their processes on one of the capability maturity models. Manufacturing and segments of service industries are more likely to use ISO as a framework.

Regardless of the structure, once a process is defined it is imperative that practitioners be trained in performing the process. Successful organizations integrate learning and work since rapid organizational and technological changes require that training and production must take place concurrently. In this model training has two primary goals:

- For individuals to acquire the skills and behaviors necessary to effectively execute the process.
- To institutionalize how processes are performed and standards implemented across the organization.

Event-driven learning optimizes goal

achievement by integrating the training and work process model, tying the training episodes to specific events in the process model.

Engineering Framework Integration

Defining an organization's implementation of an engineering framework (software or systems) generally consists of documenting organizational standards, processes, and procedures (work instructions). Entry criteria, inputs, activities, output, and exit criteria are documented, and roles and responsibilities are assigned. This may be a top-down process: A group within the organization defines the processes and pushes them down in the organization. Or it may be bottom up: Best practices are culled from existing processes. Usually, the institutionalization of processes is a combination of approaches where existing processes are refined by a central group and disseminated. The group also fills in gaps where the existing processes are not consistent with the selected framework.

Typically training is offered at the process area level. For example, a *Procedures for Deriving and Allocating Requirements* course may be offered. Event-driven learning breaks this larger body of material into chunks that are directly related to the activities performed in the process area, and offers training experiences that enable practitioners to perform the specific activities in the next process step. The training events build into a complete support system for long-term individual and organizational performance as the workers execute progressive activities in the overall process. Figure 1 is an example of a high-level process diagram for requirements analysis showing several courses being integrated at appropriate points in the process.

Course 1 – Requirements Management Process. This is a facilitated discussion of the organization's process for managing requirements. It presents the underlying importance of requirements management to the program's success and walks participants through the process step by step. As each activity in the process is presented, participants discuss how the activity will be instantiated in their program. This course generally takes one to two hours depending on the participants experience and the program complexity.

Course 2 – How to Facilitate Requirements Gathering Meetings. This is a workshop where participants learn how to extract functional information from customers and end-users in a meeting format, then document the functional requirements and verify them with the user. Interpersonal and documentation techniques are taught. Each participant gets the opportunity to practice the new skills in the workshop setting. Since people are asked to role-play extensively in this course it generally takes about three to four hours for six to eight people. It is administered immediately prior to the requirements analysts going into the requirements gathering meetings. If other requirements gathering techniques are used, such as task analysis or reverse engineering an existing system, then courses appropriate to those techniques would be offered at this point in the process.

Course 3 – How to Write Requirements. This course focuses simply on the format and content of a well written requirement. The importance of well written requirements is explained and rules for the structure and content of the requirements are presented; templates and checklists are provided. The participants are engaged in exercises where they identify well and poorly written requirements and then write and critique requirements. This course lasts about one to two hours and is most effective when the customer is included as a participant.

Course 4 – How to Conduct a Requirements Walk-Through. This is a simulation-based course where walk-through participants practice the dynamics of the walk-through session consistently counting defects according to the rules laid out by quality assurance. The walk-through simulation lasts about half an hour for a five-person team. If a formal inspection technique is used at this point in the process, then the training takes about four hours for up to six participants, as each has the opportunity to practice the roles in the inspection. The checklists and templates from the previous course are used in the walk-through and inspections courses.

Course 5 – Interpreting the Software Requirements Specification (SRS) Standard. This course walks the partici-

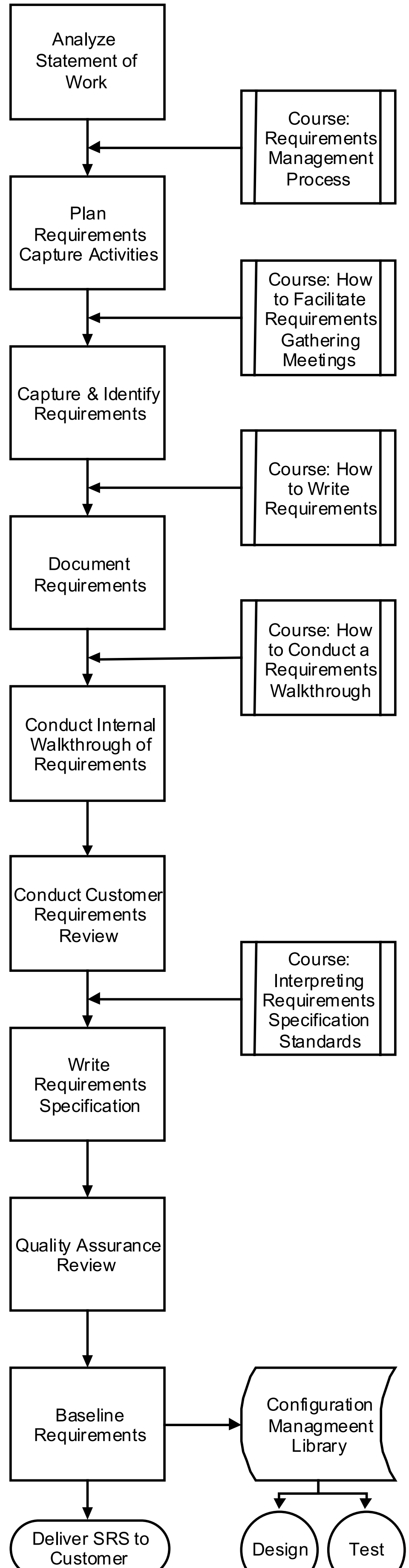


Figure 1. *Event-driven learning courses support specific activities in process models.*

pants through the standard (data item description) for the requirements specification before a word of the document is written. A quality assurance analyst, who will be reviewing the document when it is produced, most effectively teaches this course. Expectations for the content of each (sub)paragraph are discussed so that the author(s) is ready to write a compliant document. This course generally takes about 15 to 30 minutes.

These five courses are examples of applying event-driven learning to the requirements development practice area. Typical requirements courses are advertised as three-day courses and generally cover the same instructional objectives as the five event-driven learning courses cited above. Course objectives for a commercially available three-day *Writing Testable Requirements* course are:

- Techniques to produce requirements documents that are concise, accurate, modular, and highly testable.
- How to identify and correct ambiguities in specifications.
- How to ensure that your requirements documents conform to standard industry guidelines.

These sound like a subset of what is accomplished in the nine to 12 hours of event-driven learning training described above. Because the learner in a multi-day course is often weeks away from putting all the techniques learned in the course into practice, training time needs to be extended to account for additional in-course exercises so that skills are ingrained. Conversely, event-driven learning with its process activity-focused *chunking* approach allows the learner to start practicing the new skills or applying the new knowledge within hours. It also makes it easier to change the instructional delivery method to better suit the content, audience, and logistics of the topic being learned.

Event-driven learning is also a useful approach when a particular concept is frequently misunderstood yet easily clarified with a brief training experience. Work breakdown structure (WBS) is an example of such a topic. When asked to produce a WBS for a program management plan, practitioners frequently produce a Gantt Chart of activities: an hour-and-a-half training session on producing WBS results in a valid WBS supported by activities. The workshop format evolved from trying to solve the problem by publishing job aids. When that did not get the desired results, a classroom setting was used that explains the differences between WBS and Gantt. It gives the participants a chance to derive and critique WBS for some everyday situations. Other activities in performing program planning are addressed by additional event-driven learning courses similar in concept to the courses described for requirements definition.

A commercially available, Web-based management training product also applies the event-driven learning concept. Each of the training experiences in the curriculum address a single management situation. In this case the event is a manager/staff-member interaction on a particular topic, such as excessive tardiness, perception of insufficient raise, correcting bad personal habits, etc. The just-in-time concept used in this event-driven learning experience is that just before the manager has such an interaction with a staff member, the manager would review the guidelines for conducting such an interaction. With the instruction and modeled behavior fresh in the manager's mind, the interaction proceeds more effectively and the desired results are achieved.

The Development Process

Event-driven learning development follows a typical instruction systems development approach. The event-driven learning development process is initiated after the work processes have been defined. Once this is done, the training needs are determined. Training needs should be prioritized so that the process with the greatest impact on the program is the initial candidate for event-driven learning support. Impact may be derived based on cost, criticality, risk, complexity, and other dimensions relevant to the project. Once these initial processes are chosen an event-driven learning training plan is developed that identifies goals, scope, concept of operations, general approaches, target learner audience, resources, schedule, and other information necessary to accomplish the training within the organization or program's operational constraints.

Decomposition of the process model down to the work-instruction/procedural level helps to define the abilities the learners will need to possess when they complete the training. Analysis of standards and output descriptions for the process steps is also needed at this point. The analysis results are translated into instructional objectives that describe the training outcomes by using terms that complete the phrase, "At the conclusion of the course the learner will be able to .…"

With the instructional objectives and training plan in hand, the event-driven learning experience can be designed. Grouping the objectives into meaningful, complete, stand-alone groupings defines the chunks of training to be offered. The decomposed process models define when the training courses are to be inserted into the process area. The nature of the work performed and the logistics of the work environment will help determine the instructional delivery method used. Since event-driven learning is so focused on the next task in the work process, event-driven learning design starts with designing the practice exercises that the learner will perform during the course. The underlying premise is simply that people best learn by doing. These hands-on experiences during training are designed to transfer directly to the work instructions on the job.

Using actual work samples and outputs from previous process steps as examples (of both good and bad instances) makes the training most relevant. Small, focused chunks of training are developed at a time, so it is easier to customize each instance of the course for the target audience. If the program is at a stage where relevant work products are not available, or the class is a mix of students from several programs, current events often provide a rich, timely source of material for exercises.

All event-driven learning courses must be formatively evaluated before being implemented. Peer reviews are a useful technique for an initial evaluation. A dry run is then suggested to verify that the course produces the expected outcomes. For technology-based event-driven learning, both the logic and the content of the course must be evaluated. It is also a good idea to verify the logistics of delivering the course regardless of the delivery mechanism. Once the course is positively evaluated, the process model is updated to

Benefits of Event-Driven Learning

Event-driven learning provides many benefits to the organization. These benefits positively affect customer satisfaction and the organizational bottom line. Some of the direct benefits are:

- **Increased training efficiency** – Because process support training is more focused on steps within the organizational processes rather than being designed to address a broader scope, limited resources are expended to generate the appropriate training. Fewer total training hours are required to achieve the instructional objectives; so if the resources are held constant, more objectives can be achieved by the trainees.
- **Increased training effectiveness** – Trainees remember the things that they practiced the most and learned last in a class. Information overload during training often results in new information never being consolidated by learners during the instructional session. Information presented in small, retainable chunks is quickly applied in a work context that reinforces the learning experience. Because the training session takes place during the implementation of a work process, the learner better perceives the daily relevance of the information and skills being taught and is therefore intrinsically motivated to learn.
- **Positive opportunity costs** – The short courses that are characteristic of event-driven learning are usually taught in-house, rather than sending people off-site. In addition to reduced administrative costs for the delivery of the courses, the trainees are more available for their regular jobs thus improving the opportunity cost picture.
- **Reduced rework/revision costs** – One of the underlying concepts of event-driven learning is that people generally like doing things correctly and hate going back and redoing things. With a training event properly scheduled just before an activity is undertaken, the workers will learn a skill and immediately put it into practice thus increasing the likelihood that they will get it right the first time. Also, the training design often incorporates checklists and other job aids as handouts (electronic or physical) that reinforce training concepts and facilitate correct job performance. Using event-driven learning for team training enables the entire team to better coordinate their activities, assure a common focus, and reduce wasted effort.
- **Modifiable and maintainable courseware** – Because of their relatively small size event-driven learning courses are easy to modify and maintain. Most processes evolve over time rather than undergo major, revolutionary changes. Part of the process change control activity should be modifying any event-driven learning course associated with the process. Event-driven learning is a key component for an organization intent on continuous learning and change because of the relative ease with which it can be modified.
- **Institutionalization of processes** – Using the same course materials to teach people in an organization how to perform specific activities within the organization's processes assures that they are performed similarly across the organization. People will practice performing the activity during training, increasing the likelihood that they will perform it correctly on the job. As each group in the organization takes the training and gets feedback on how well they are applying the new skills in their work, the institutionalization of the process takes place without resistance.
- **Quick fix process improvement strategy** – In the event that particular process steps are consistently not performed correctly, event-driven learning training experiences can be inserted in the process fairly quickly. Assuming 20 hours of development for one hour of instruction, a typical event-driven learning course can be available within a week or two to address the process issue(s).

include the course at the appropriate point in the process. Thus, whenever the work process is performed, the training occurs at a time when it can have the most impact on the successful completion of the process.

Event-driven learning is a flexible approach to improving the performance of processes within an organization. Regardless of the framework used to guide system and software development efforts, small, relevant, integrated training experiences enhance people's ability to correctly perform process steps and adhere to program standards.

Summary

Event-driven learning is an approach for satisfying training needs in a process-based development environment that optimizes job performance and thereby product quality. The technique for defining and developing event-driven learning is a combination of a systematic approach, experientially educated guesses, and analysis of actual performance data. During process definition activities an analyst identifies products (outputs) and activities (process steps) that are new, unique, or complex. Other factors may relate to criticality of the product and frequency of performance of the activity (low frequency activities may need to be retrained prior to performance). These process steps are then candidates for being supported by event-driven learning courses.

Alternatively, data from previous performance of the process may indicate that a particular activity in the process is being poorly performed (takes too long, too many errors, has to be redone several times, etc.), or the products from the activity consistently show a large number of defects. In either of these conditions, event-driven learning is a proactive method for process improvement. Once the need has been established, the process model is changed to include the course at the appropriate step in the process, and an instructional systems development process results in implementing the training.

The primary reasons for using event-driven learning in your organization are:

- Event-driven learning is a cost-effective approach to software and system engineering training that improves organizational process performance.
- Applying newly learned skills immediately after training reinforces the learners' training experience and reduces overall training time and forgetting.
- Event-driven learning capitalizes on adult learning principles regarding focus on real world application and retention of new information and skills.
- Event-driven learning can focus on satisfying job performance needs of the individual or a team.

Everyone benefits from using event-driven learning as a process improvement strategy. The organization improves its process control and maintenance methods and institutionalizes processes and standards. The individual is given the skills to achieve greater job satisfaction and thereby higher morale. Teams achieve greater cooperation and cohesiveness. And the customer acquires a higher quality product because the process that produces the product is better performed and managed.◆

About the Author

Michael J. Hillelsohn is a director of Product Assurance at Software Performance Systems (SPS) in Arlington, Va. SPS builds secure e-commerce, case management, and network solutions for government and industry clients. Hillelsohn is a certified quality professional with more than 30 years of experience doing development, management and performance improvement in software and systems development environments. His multidisciplinary approach combines quality systems and training expertise to improve the performance of organizations and individuals. His process-oriented, performance engineering methods facilitate adoption of external frameworks (CMM®, ISO, Baldrige) to improve the quality of organizational products and services.

Software Performance Systems
2011 Crystal Drive
One Crystal Park, Suite 710
Arlington,Va. 2202
Voice: 703-797-7707
E-mail: mhillelsohn@goSPS.com
Internet: www.goSPS.com

Kudos for CrossTalk

I have used back issues of CrossTalk often in my software process improvement work both at Xerox and Hughes. It is really helpful! – **Delores J. Harralson**

I'm having withdrawal ... not having CrossTalk to read since I left Puget Sound Naval Shipyard! The hard copy will be perfect to read on Metro on my way to and from work! – **Cathy Ricketts**

Some time ago I noticed the change in your cover for CrossTalk, I thought at the time, what a nice change. The colored, more graphic cover adds dimension and class to your publication. I felt at the time and still do the change was an improvement. I also find the articles informative and interesting. I work in the communications industry (Quest) in support of Information Management System (IMS, IBM mainframe software) and find information that relates to my work environment. The publication also gives me a look at other perspectives. Good job! **– Kevin Mauchley**

Modeling and Simulation Web Sites

www.ist.ucf.edu

The Institute for Simulation & Training is an internationally recognized research institute that focuses on advancing modeling and simulation technology and increasing understanding of simulation's role in training and education. Founded in 1982 as a research unit of the University of Central Florida and reporting directly to the vice president for Research, the institute provides a wide range of research and information services for the modeling, simulation, and training community.

www.msiac.dmso.mil

The Modeling and Simulation Information Analysis Center assists the Department of Defense (DoD) in meeting their modeling and simulation needs by providing scientific, technical, and operational support information and services. MSIAC is a DoD Information Analysis Center, sponsored by the Defense Technical Information Center and the Defense Modeling and Simulation Office.

www.scs.org

The Society for Computer Simulation (SCS) International is the principal technical society devoted to the advancement of simulation and allied computer arts in all fields. The purpose of the society is to facilitate communication among professionals in the field of simulation. To this end, the society organizes meetings of regional councils, sponsors and cosponsors national and international conferences, and publishes the monthly journal *Simulation* as well as the quarterly journal *Transactions* of the SCS.

www.ecst.csuchico.edu/~mcleod

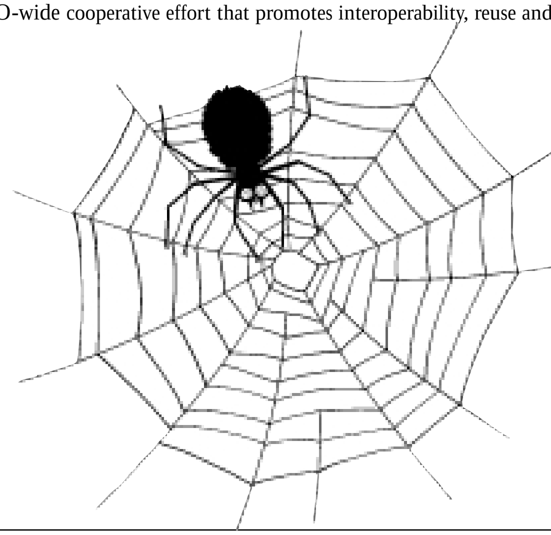
The McLeod Institute of Simulation Sciences was established in 1986 at California University, Chico, to be a center of excellence in computer modeling and simulation as a collaborative effort between the university and the Society for Computer Simulation International. In creating the institute, the university recognized a unique opportunity to develop a widely-recognized center engaging in state-of-the-art simulation research and educational activities by virtue of a critical mass of experienced faculty available to contribute to this effort. The institute provides a mechanism through which faculty from various disciplines and their students and associates can bring their talents to bear in the general area of computer simulation or can seek help with the application of simulation to new areas.

www.vmasc.odu.edu/main.htm

The Virginia Modeling, Analysis and Simulation Center is a cooperative venture between business, government, and academia whose purpose is to promote economic development through application of computer simulation. It is administered by the College of Engineering and Technology, Old Dominion University, the Department of Defense, the state of Virginia, and Suffolk City. The center consists of a consortium of universities working cooperatively to advance business applications of simulation technology , and to transfer new simulation technology developments between defense and civilian organizations.

www.rta.nato.int/msg.htm

The Modeling and Simulation Group provides readily available, flexible and cost-effective means to dramatically enhance NATO operations in the application areas of defense planning, operational planning, training and exercises, and support to operations and modernization. This goal will be accomplished by a NATO-wide cooperative effort that promotes interoperability, reuse and affordability.



Intrusion Detection[1]: Implementation and Operational Issues

John McHugh, Alan Christie and Julia Allen
Software Engineering Institute, Computer Emergency Response Team/Coordination Center

Intrusion detection systems (IDSs) are an important component of defensive measures protecting computer systems and networks from abuse. This article gives an overview of the most commonly used intrusion detection (ID) techniques. It considers the role of IDSs in the overall defensive posture of an organization and provides guidelines for their deployment, operation, and maintenance.

Attacks on the nation's computer infrastructures have become an increasingly serious problem. While government agencies have been common targets, the distributed denial-of-service attacks that materialized last year primarily targeted commercial sites.

"Hackers attacked some of America's most popular Web sites yesterday for the third day in a row, walling off frustrated consumers from companies that provide news and stock trading as law enforcement officials launched a nationwide criminal investigation … The computer attacks earlier this week temporarily blocked access to Web sites that read like a *Who's Who* of the new economy, including Yahoo, eBay, Amazon, CNN.com and Buy.com," reported the *Washington Post* on Feb. 10, 2000.

This story reflects the serious and sophisticated nature of today's cyber-attacks. During the past 12 years, the growth of incidents reported to the Computer Emergency Response Team/Coordination Center (CERT/CC®) has roughly paralleled the growth of the Internet.

As e-commerce sites become attractive targets and the emphasis turns from break-ins to denial of service and widespread virus attacks, the situation will likely worsen. Many early attackers were motivated by the challenges of breaking into systems, but there is an increasing trend toward attacks motivated by financial, political, and military objectives.

In the 1980s, most intruders were experts who discovered vulnerabilities and developed methods for breaking into systems. The use of automated tools and exploit scripts was rare. Now, anyone can attack a network using readily available intrusion tools and exploit scripts that take advantage of widely known vulnerabilities.

Today damaging intrusions can occur in a matter of seconds. Intruders hide their presence by installing modified versions of system monitoring and administration commands, and by erasing their tracks in audit and log files. In the 1980s and early 1990s, denial-of-service attacks were infrequent and not considered serious. Now, these successful attacks can put e-commerce-based organizations such as online stock brokers and retail sites out of business as evidenced in February of 2000.

ID has been an active field of research for about two decades. One of the earliest papers in the field is James Anderson's *Computer Security Threat Monitoring and Surveillance* [1] published in 1980. Dorothy Denning's seminal paper *An Intrusion Detection Model* [2], published in 1987, provided a methodological framework that inspired many researchers and laid the groundwork for commercial products. Despite substantial research and commercial investments, ID technology is immature, and its effectiveness is limited [3]. Within its limitations, it is useful as one portion of a defensive posture, but should not be relied upon as a sole means of protection.

The Intrusion Perspective

Defining what constitutes an attack is difficult because multiple perspectives are involved. Attacks may involve any number of attackers and victims. The attacker's viewpoint is typically characterized by intent and risk of exposure. From a victim's perspective, intrusions are characterized by their manifestations, which may or may not include damage. Some attacks may produce no manifestations, and some apparent manifestations may result from system/network malfunctions. Some attacks involve the (involuntary) participation of additional machines, usually victims of earlier attacks. For an intrusion to occur, it requires both an overt act by an attacker and a manifestation, observable by the intended victim, which results from that act.

A victim's view of an attack is usually focused on these manifestations:

- What happened?
- Who is affected and what were the consequences?
- Who is the intruder?
- Where and when did the intrusion originate?
- How and why did the intrusion happen?

Meanwhile, the attacker may have quite a different view:

- What is my objective?
- What vulnerabilities exist in the target system?
- What damage or other consequences are likely?
- What exploit scripts or other attack tools are available?
- What is my risk of exposure?

The goal of ID is to characterize attack manifestations so as to positively identify all true attacks without improperly identifying false attacks. The motivation for using ID technology may vary. Some users may be interested in collecting forensic information to locate and prosecute intruders. Others may use ID to trigger actions to protect computing resources. Still others may use ID to identify and correct vulnerabilities.

Dimensions of Intrusion Detection

IDS can be characterized in a variety of ways. Here, we choose system structure, sensed phenomenology, and a detection approach. The viewpoints are somewhat generalized, and a given IDS may combine structural components, sense multiple phenomenologies, or use multiple detection approaches.

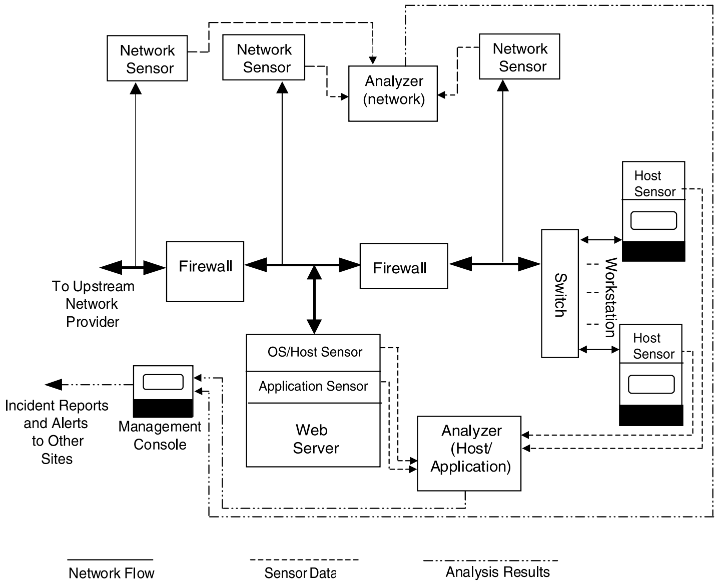


Figure 1. *IDS System Structure*

Figure 1 illustrates system structure and sensed phenomenology. The figure shows a small enterprise configured with firewalls to isolate its Web server. Computers configured as network sensors extract suspicious packets from the three main network segments and forward them to a network specific analysis station. The Web server and workstations are equipped with software to monitor suspicious interactions with the operating system and report them to a host-specific analysis station.

In addition, the Web server looks for abuses such as common gateway interface bin exploits that are specific to hypertext transfer protocol servers. The analyzers report to a management console that serves as the user interface for the IDS. The management console alerts the enterprise administrator who may, in turn, report intrusions to incident response organizations.

More elaborate configurations are possible. An analyzer may use inputs from any or all sensed phenomenologies in deciding whether or not an attack has taken place. Analyzer outputs may also be used as sensed data for other analyzers.

Intrusion Detection Approaches

ID can be viewed as an instance of the signal detection problem [4]. In this case, intrusion manifestations are viewed as the signal to be detected while manifestations of *normal* operations are considered to be noise. In classical signal detection approaches, both the signal and noise distributions are known. A decision process must determine if a given observation belongs to the signal-plus-noise distribution or to the noise distribution. Classical detectors use knowledge of both distributions in making a decision; but intrusion detectors typically base their decisions either on signal (signature-based) or noise (anomaly-based) detector characterizations. Each approach has strengths and weaknesses. Both suffer from the difficulty of characterizing the distributions.

In order for a signature-based IDS to detect attacks, it must possess an attack description that can be matched to sensed attack manifestations. This can be as simple as a specific pattern that matches a portion of a network packet, or as complex as a state machine or neural network description that maps multiple sensor outputs to abstract attack representations. If an appropriate abstraction can be found, signature-based systems can identify previously unseen attacks that are abstractly equivalent to known patterns. They are inherently unable to detect truly novel attacks and suffer from false alarms when signatures match both intrusive and nonintrusive sensor outputs.

Signatures can be developed in a variety of ways, from hand translation of attack manifestations to automatic training or learning using labeled sensor data. Because a given signature is associated with a known attack abstraction, it is relatively easy for a signature-based detector to assign names to attacks.

Anomaly-based detectors equate unusual or *abnormal* with intrusions. Given a complete characterization of the noise distribution, an anomaly-based detector recognizes as an intrusion any observation that does not appear to be noise alone.

Characterizing the noise distribution so as to support detection is nontrivial. Characterization approaches have ranged from statistical models of component/system behavior to neural networks and other artificial intelligence techniques to approaches inspired by the human immune system. The primary strength of anomaly detection is its ability to recognize novel attacks.

Its drawbacks include the need to train the system on noise plus the difficulties attendant in tracking natural changes in the noise distribution. Changes may cause false alarms while intrusive activities that appear to be normal may cause missed detections. It is difficult for anomaly-based systems to classify or name attacks.

Organizational Issues

Installing and effectively using intrusion detection systems on networks and hosts require a broad understanding of computer security. The complexity of information technology infrastructures is increasing beyond any one person's ability to understand them, let alone administer them in a way that is operationally secure.

An organization needs to fully appreciate the commitment required before deploying an IDS. Otherwise, the project runs the risk of wasting time, money, and staff resources in the initial phases of the IDS life cycle.

Before an organization makes an investment in security technologies, it must understand what assets require protection and the real and perceived threats against those assets. Threats can be characterized by the likely type of attack and attacker capabilities (i.e., resources and goals), and the organization's tolerance for loss of, damage to, or disclosure of protected assets.

Attacker motives can be arbitrary, based on curiosity or vandalism, or targeted to meet a specific objective such as revenge or gaining competitive advantage. Motives may make some forms of attack more likely than others. Gaining a competitive advantage may require compromising specific information such as a marketing plan. Each form of attack requires diverse detection strategies. For example, information retrieval is likely to be performed using a stealthy attack, while information corruption may require speed. Determining whether the potential attacker is inside or outside of the organization's infrastructure has a bearing on the type and placement of an IDS.

Often the most significant obstacle to the success of an information security improvement initiative is lack of management support.[2] In surveys conducted by security trade magazines during 1999 [5, 6], lack of management support was cited as one of the principle barriers to effective information security.

This is consistent with our experience at the SEI in implementing security improvement initiatives. Managers have many goals to meet, and they must often make tradeoffs. Security only becomes important when it impinges on the organization's high priority interests and reputation.

The deployment and operation of an IDS requires significant management support at the level of the corporate chief information officer and information security manager. Without this, the successful operation and use of this technology will be short-lived, sustained only by the interest of those internal champions who believe in its benefit. This is likely to last only until another high-priority item requires their attention.

Defense in Depth

ID is only one aspect of a layered defensive posture or *defense in depth*, which begins with the establishment of appropriate and effective security policies. Effective policies help ensure that threats to critical assets are understood, managers and users are adequately trained, and intrusion response actions are defined. A good security policy places ID into its proper perspective and context.

Establishing a layered security architecture is advantageous whether an IDS is deployed or not. In addition to formulating a security policy, the essential steps are:

- Implementing user authentication and access controls.
- Eliminating unnecessary services.
- Applying patches to eliminate known vulnerabilities.
- Deploying firewalls.
- Using file integrity checking tools such as Tripwire.[3]

Since most real-time commercial ID systems base their detection approach on known attempts to exploit known vulnerabilities, an administrator's time is often better spent minimizing vulnerability by applying patches or other security measures. Detecting and responding to penetration attempts that cannot succeed (such as UNIX-specific attempts against a network of Windows machines) is not an effective use of resources except as an indication of threat level.

The IDS Life Cycle

Vendors frequently release new IDS products and aggressively compete for market share. Evaluating these new systems is crucial, yet there is a lack of credible, comprehensive product evaluation information. Hiring and retaining personnel to competently administer security in general, and intrusion detection in particular, are increasingly challenging. Rapid changes in information technology make it difficult for an organization to implement an effective, long-term security strategy.

- **Evaluation and Selection:** If an organization plans to acquire an ID system, it should consider the resources available for its operation and maintenance and choose one that meets its needs within these constraints. This is difficult because there are no industry standards against which to compare ID systems. The new product cycle for commercial IDSs is rapid, and information and systems quickly become obsolete. Northcutt recommends use of product guides that are updated at least monthly [7].

  Marketing literature rarely describes how well a given IDS finds intruders. Neither does it tell how much work is required to use and maintain that system in a fully functioning network with significant daily traffic. IDS vendors usually specify which prototypical attacks can be found by their systems. However, without access to deployment environments, they cannot describe how well their systems detect real attacks while avoiding false alarms.

  Topics to consider include detection and response characteristics, use of signature- and/or anomaly-based approaches, accuracy of diagnosis (false alarm rate), ease of use, effectiveness of user interface, quality of vendor support, etc. A paper by Amoroso and Kwapniewski [8] provides guidance in selecting an IDS. The Computer Security Institute[4] has a number of relevant Web pages, including a list of questions for IDS vendors. Setting up a facility to objectively compare IDSs will be prohibitively expensive for all but the largest potential users, and some third-party or industry-sponsored effort is needed.
- **Deployment:** Issues to address here include placement of sensors to maximize protection for the most critical assets, configuring the IDS to reflect security policy, installing appropriate signatures and other initial conditions, establishing forensic procedures to preserve evidence for possible prosecutions, and determining when (if ever) and what automatic responses are allowed. Procedures must be developed to handle IDS alerts and to consider how alerts are to be correlated with other information such as system or application logs.
- **Operations and Use:** Once an IDS is deployed, it is necessary to monitor the system and to respond to the reported alerts. This means establishing roles and responsibilities for analyzing and acting on alerts, monitoring the outcomes of both manual and automatic responses, etc.

  IDSs themselves are logical targets for attack [9]. Smart intruders who realize that an IDS has been deployed on a network they are attacking will likely attack the IDS first, disabling it or forcing it to provide false information (distracting security personnel from the actual attack in progress). In addition, many commercial and research ID tools have security weaknesses resulting from flawed design assumptions. These may include failing to encrypt log files, omitting access control, and failing to perform integrity checks on IDS files.
- **Maintenance:** Activities include installing new signatures as they become available, as well as installing periodic IDS upgrades. Sensor placement should be revisited periodically to ensure that system or network changes have not reduced the effectiveness of the IDS.

Use of technology alone is not sufficient to maintain network security. An organization needs to attract, train, and retain qualified technical staff to operate and maintain ID technologies. In today's market, there is a decreasing availability of qualified intrusion analysts and system/network administrators who are knowledgeable about and experienced in computer security.

ID Technology

Commercial ID technology (such as ISS RealSecure[5] and Tripwire) is evolving and is often dynamic to the point of instability. New vendors appear only to be absorbed by others. One consequence of this rapid change is that product lists, surveys, and reviews quickly become outdated. Because of the volatility of the market, we advise using a Web search to locate current products, reviews, etc.

Commercial product literature is generally weighted towards marketing, which often makes it difficult to determine the product's functionality and detection approach. Virtually no commercial literature addresses issues such as frequencies of false alarms, missed detections, or the system's sensitivity to traffic loads.

Public domain systems, such as Shadow[6] and Snort[7], are unlikely to have the same level of support as commercial systems, so a higher level of technical expertise is required to install and manage them. However, the effort required results in the payoff of a better understanding of ID and its strengths and limitations.

Based on several limited experiments, we found that commercial ID tools are easier to install than public domain tools. None of the tools had an understandable, easy-to-use configuration interface. However, the commercial tools did employ graphical interfaces while the public domain tools did not. All of the tools required labor-intensive signature tuning. We found no indication of any integration between vulnerability scanners and configuration interfaces despite the fact that most IDS vendors sell vulnerability analyzers.[8] The configuration process could be made simpler if signatures associated with detected vulnerabilities could be loaded automatically.

The commercial products that we installed did not provide sufficient supporting data (such as raw packets) to verify events they claimed to detect. The use of proprietary algorithms and signatures made it difficult to determine why an alert occured. Distinguishing between intrusions and false alarms required manual investigation. In most cases, the analyst had to examine logs for supporting evidence.

IDS products based on current signature-based analysis do not provide a complete ID solution, but do produce useful results in specific situations and configurations. The majority of intrusion detection systems we examined appeared to provide good capabilities for enhanced network monitoring and might be more useful in this capacity than for intrusion detection.

Conclusion

ID technology is immature and should not be considered as a complete defense. However it can play a significant role in an overall security architecture. If an organization chooses to deploy an IDS, there is a range of commercial and public domain products to choose from with varying effectiveness and deployment costs. Since any deployment will incur ongoing operation and maintenance costs, the choice should be made considering the full IDS life cycle.

When an IDS is properly deployed, it can provide warnings indicating that a system is under attack, even if the system is not vulnerable to the specific attack. This warning can be used to alter the defensive posture of the installation to accomplish greater resistance to attack. In addition, an IDS may be used to confirm secure configuration and operation of other security mechanisms such as firewalls.

Acknowledgements

The authors wish to thank the members of the CERT/CC staff who contributed to *State of the Practice of Intrusion Detection Technologies*, on which this is based: William Fithen, Jed Pickel, Ed Stoner, James Ellis, Eric Hayes, Jerome Marella, and Bradford Willke. Stefan Axelsson of Chalmers University provided insight into the characterization of ID as a signal detection problem. Many members of the ID community have influenced our thinking on this topic. Among them are Ed Amoroso, Dick Kemmerer, Gene Kim, Jay Larrew, Teresa Lunt, Roy Maxion, Phil Porras, and Mark Woods.

References

1. Anderson, James P., *Computer Security Threat Monitoring and Surveillance*. James P. Anderson Co., Fort Washington, Pa., 1980.
2. Denning, Dorothy E., An Intrusion Detection Model. *IEEE Transactions on Software Engineering*, SE-13(2):222–232, February 1987.
3. Allen, J., Christie, A., Fithen, W., McHugh, J., Pickel, J., and Stoner, E., *State of the Practice of Intrusion Detection Technologies*. CMU/SEI-99-TR-028, Carnegie Mellon University, Software Engineering Institute, January 2000. Available at www.sei.cmu.edu/publications/documents/99.reports/99tr028/99tr028abstract.html
4. Egan, James P., *Signal Detection Theory and ROC Analysis*. Academic Press, 1975.
5. Larson, Amy K., Global Security Survey: Virus Attack. *Information Week*, July 12, 1999.
6. Briney, Andy, Got Security? *Information Security Magazine*, July 1999.
7. Northcutt, Steven, *Network Intrusion Detection*. New Riders, Indianapolis, Ind., 1999.
8. Amoroso, Edward and Kwapniewski, Richard, A Selection Criteria for Intrusion Detection Systems. *Proceedings of the 14th Annual Computer Security Applications Conference*. IEEE Computer Society Press, December 1998.
9. Ptacek, Thomas H. and Newsham, Timothy N., *Insertion, Evasion, and Denial of Service: Eluding Network Intrusion Detection*, 1998. Available from www.snort.org/IDSpaper.pdf

Notes

1. For a more complete discussion of this subject, see reference 1.
2. One individual told an author of this article that he obtained management sponsorship by demonstrating how easy it was to break into his manager's confidential computer files. This approach is not necessarily recommended, but at least in this case, appears to have been effective.
3. See www.tripwire.com—both commercial and public domain versions are available.
4. See www.gocsi.com
5. See www.iss.net
6. See www.nswc.navy.mil/ISSEC/CID
7. See www.snort.org
8. ISS is integrating their IDS and vulnerability scanner.

About the Authors

Julia Allen is a senior member of the technical staff (MTS) working in the CERT/CC area at CMU's Software Engineering Institute. She has an master's degree in electrical engineering from the University of Southern California.

Voice: 412-268-6760
E-mail: jha@cert.org

Alan Christie has spent the last ten years at the Software Engineering Institute focusing much of his attention on process, technology, and security issues. He is a senior MTS and has a doctorate degree in nuclear engineering from Carnegie Mellon University.

Voice: 412-268-6324
E-mail: amc@cert.org

John McHugh is a senior MTS at the Software Engineering Institute (CERT/CC). He has a doctorate in computer science from the University of Texas at Austin and is a past chair of the IEEE Technical Committee on Security and Privacy.

Voice: 412-268-7737
E-mail: jmchugh@cert.org

CMU, SEI
4500 Fifth Ave.
Pittsburgh, Pa. 15213
Fax: 412-268-6989
Internet: www.sei.cmu.edu, www.cert.org

Politics in the Red Zone

It is 1:00 a.m. Wednesday morning, Nov. 8, 2000. I am lying in a St. Louis hotel room glued to the television set watching the presidential election turn into a soap opera. For a minute I thought the networks hired software engineers to make their projections. Then I reasoned, no, they got more than half of the states right.

By the time I logged in a couple of hours of sleep, a short run, and a refreshing shower the politicians had spun the results up tighter than a NASCAR tachometer approaching the red zone. That diverted my concerns to more insipid issues like what does the red zone on a tachometer really mean? How do automobile manufacturers determine it? Why do engineers disdain their managers?

The red zone theoretically is the engine speed at which you are in real danger of having your engine fall apart. However, we all know that definition is not entirely true. I've ventured into the red zone a time or two and all I got was a citation from a local law enforcement officer asking for a donation to the policeman's ball.

Obviously engine manufacturers realized whimsical people, like myself, touch red zones before believing them to be hot. So they built a buffer between the advertised red zone and the real point of no return.

In the early days of the automobile industry, the red zone was determined by trial and error. When a blown engine was brought into the shop, the service manager would ask the driver, if still coherent, "Did you get a look at the RPMs just before the piston ripped through the hood?"

As the industry flourished, the red zone was measured in the laboratory using sophisticated monitoring equipment. Bottom line is you really should not spend much time in the red zone. Your Yugo is more reliable, effective, and economic if you stay out of the red zone.

Wouldn't it be nice if software engineers came equipped with a stress meter and red zone indicating when the engineer might fall apart (like a mother who discovers her daughter didn't make the cheerleading squad)?

As long as we are wiring engineers—as if we are not wired enough—we could add a green zone indicating peak performance, a yellow zone for idleness, and an orange zone for when they have stalled.

Unfortunately colleges and universities are not rolling out engineers with factory-installed stress meters. No dials, no knobs, no zones, just a handful of offers and more bravado than Pavarotti on opening night.

It appears that some managers, left over from theory X, are toying with the auto industry's early trial and error approach. "Excuse me, what was Gilbert's workload just before his head ripped through the monitor? I don't know but police suspect Mountain Dew may have been involved."

Somehow I doubt this approach will last. With competition for talented engineers fierce, a manager employing this crash and burn tactic will be disenfranchised faster than a senior citizen voting in Palm Beach County, Fla.

On the other side of the scale, cautious theory Y managers rev up their engineers but forget to put them in gear. More concerned about being accepted as a manager than producing a product, they are long on stories, jokes, and antidotes, and short on direction and resolve to lead à la Rush Limbaugh.

So does that leave us with a theory Z management approach? No way! Theory Z managers waffle more than Al Gore with his campaign strategy. It's too firm, too soft; I'll decide, no you decide … this waffling is an engineer's nightmare. Software engineers have a hard enough time with requirements. Adding a manager that flops on decisions daily is like asking Robin Williams to teach yoga.

What can a manager do? How about trying something completely radical, foreign to engineer and manager alike—something that makes your management upper lip sweat when mentioned? How about talking with the engineers? Not talking at engineers but holding a meaningful dialogue where you listen and try to understand. Spend a day working with the engineer. You do remember engineering don't you? You do know how to listen don't you? You do know the difference between discussion and dialogue, don't you? Can you fill out a butterfly ballot? Never mind.

—Gary Petersen, Shim Enterprise Inc.

Sponsored by the Computer Resources Support Improvement Program (CRSIP)

CrossTalk / TISE
5851 F Avenue
Building 849, Room B04
Hill AFB, UT 84056-5713

PRSRT STD
U.S. POSTAGE PAID
Kansas City, MO
Permit 34