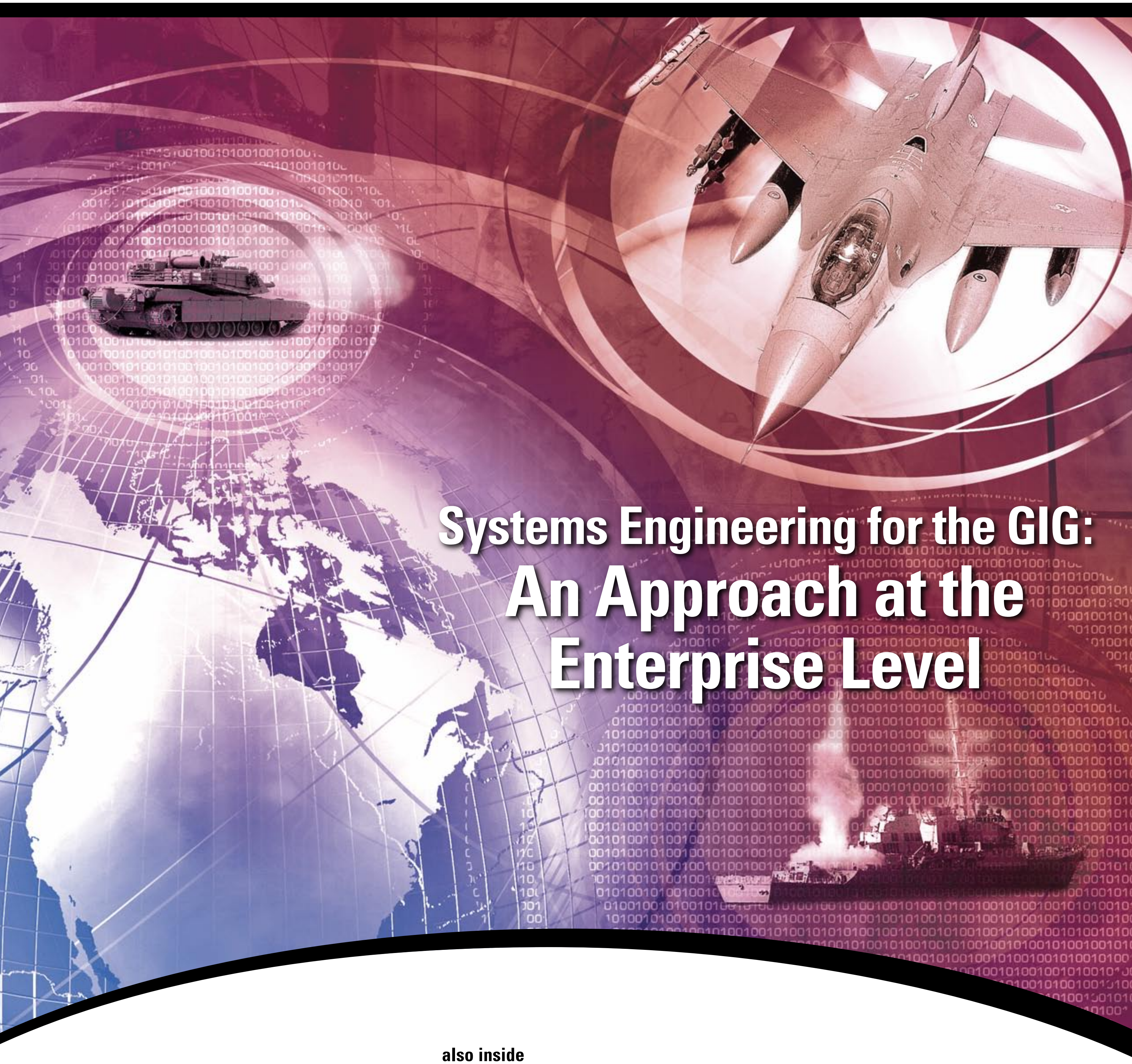
10/2

Volume 10 Number 2 • Summer 2007

IAnewsletter

The Newsletter for Information Assurance Technology Professionals



Systems Engineering for the GIG: An Approach at the Enterprise Level

also inside

Software Agent Technology

Enabling Mission Critical Operations Through Mature Implementation

CyberCIEGE: An Information Assurance Training and Awareness Video Game

DISA Partnership Conference

IATAC Spotlight on Research

IATAC Spotlight on Education

Ask the Expert

IATAC

contents

Systems Engineering for the GIG: An Approach at the Enterprise Level
The GIG is an ambitious undertaking that is fundamental to network-centric warfare. We have established an enterprise process to apply systems engineering discipline to the decisions that need to be made to make the GIG a reality.

8 Software Agents (SA): A New Programming Paradigm
Software agent technology defines a new programming paradigm that is conceptually vastly different and potentially superior to conventional network programming models such as client-server.

16 Enabling Mission Critical Operations Through Mature Implementation
Today's world calls for organizations to deal with complex connectivity, increased immediate security risks, and interoperability requirements. Organizations are responsible for meeting warfighters' unique demands, sharing knowledge reliably and securely, and responding to threats efficiently. Organizations must strengthen their own infrastructure to be effective, reliable GIG members.

22 CyberCIEGE: The Information Assurance Training and Awareness Video Game
The high degree of flexibility built into CyberCIEGE permits scenarios to be created that illustrate virtually any security topic in a range of environments, generic and organization specific. Future work might include scenarios on topics such as configuration and patch management, security in wireless networks, and using public key cryptography to support security objectives.

26 Conferences: DISA Partnership Conference
This conference is key to DISA because it involves networking and enables relationships to be renewed and built on, which are critical to the warfighter.

27 IATAC Spotlight on Research
Dr. XinYuan (Frank) Wang

28 IATAC Spotlight on Education
George Mason University

30 Ask the Expert: Security Metrics
If the security profession is to meet credibility and accountability standards applied to other functional areas (*e.g.*, finance, sales, operations, IT), security teams must be willing to share their historical data for inclusion in a database that compares results across industries. For now, IT security teams within the commercial sector are using metrics increasingly to indicate progress and tie their activities into the larger business mission.

in every issue

3 IATAC Chat
29 Letter to the Editor
31 Product Order Form
32 Calendar

DDRE

About IATAC and the *IAnewsletter*
The *IAnewsletter* is published quarterly by the Information Assurance Technology Analysis Center (IATAC). IATAC is a Department of Defense (DoD) sponsored Information Analysis Center, administratively managed by the Defense Technical Information Center (DTIC), and Director, Defense Research and Engineering (DDR&E).

Contents of the *IAnewsletter* are not necessarily the official views of or endorsed by the US Government, DoD, DTIC, or DDR&E. The mention of commercial products and/or does not imply endorsement by DoD or DDR&E.

Inquiries about IATAC capabilities, products, and services may be addressed to—

IATAC Director: Gene Tyler
Inquiry Services: Peggy O'Connor

***IAnewsletter* Staff**
Promotional Director: Christina P. McNemar
Creative Director: Ahnie Jenkins
Art Director: Don Rowe
Copy Editor: Louise Price
Designer: Kacy Cummings
Editorial Board: Ronald Ritchey
Tara Shea
Gene Tyler
Buzz Walsh

***IAnewsletter* Article Submissions**
To submit your articles, notices, programs, or ideas for future issues, please visit http://iac.dtic.mil/iatac/IA_newsletter.html and download an "Article Instructions" packet.

***IAnewsletter* Address Changes/Additions/Deletions**
To change, add, or delete your mailing or email address (soft-copy receipt), please contact us at—

IATAC
Attn: Peggy O'Connor
13200 Woodland Park Road
Suite 6031
Herndon, VA 20171

Phone: 703/984-0775
Fax: 703/984-0773

email: iatac@dtic.mil
URL: http://iac.dtic.mil/iatac

Deadlines for future Issues
Winter 2007 September 14th, 2007

Cover design: Kacy Cummings
Newsletter design: Bryn Farrar
Donald Rowe

Distribution Statement A:
Approved for public release;
distribution is unlimited.

IATAC Chat

Gene Tyler, IATAC Director

Software assurance is the "justifiable confidence"—or trust—that software will consistently demonstrate its required properties.

Summer 2007 will continue to be fast paced and interesting for the Information Assurance Technology Analysis Center (IATAC). Two State-of-the-Art Reports (SOAR) will be published. SOARs are in-depth analyses and comprehensive assessments of information assurance (IA)/defense information operations (DIO) technologies. I encourage you to be on the lookout for these interesting and important publications. We have numerous SOARs in our library such as *Data Embedding, Malicious Code, Modeling and Simulation, Visualization Technologies*, and *Global Information Grid (GIG) Network Defense (GND) Gap Analysis*.

By mid-July of this year, our *Software Security Assurance* SOAR should be available. Software assurance is the "justifiable confidence," or trust, that software will consistently demonstrate its required properties. The software security assurance community is interested in policies, activities, practices, methods, standards, technologies, and tools that can contribute to achieving that high level of confidence, regardless of whether the software performs security functions. This SOAR addresses what the software security assurance community has accomplished, is accomplishing, and is planning to accomplish to further the cause of software security assurance.

By mid-September, our second SOAR dealing with the insider threat should be available. The term "insider threat" is in itself a broad concept covering overlapping protect, detect, and react needs. Depending on the technology user's perspective, a solution addressing one area of concern often is of little use in addressing other areas of concern. The real issue with detecting true malicious insiders is that it involves examining human characteristics, individual and group; psychological profiling; examination of motivations and intentions; and standards of ethics. This SOAR examines the following questions:

- ▶ What are possible insider threat venues?
- ▶ What properties must be exhibited for a solution to be considered secure?
- ▶ What parts of the insider threat problem can be partially solved?
- ▶ Where is the current research focused?
- ▶ What are the relationships and differences between those who monitor at the network level for misuse and those who perform in-depth examinations for potential insider activity?
- ▶ What are the legal requirements for insider threat monitoring? IATAC goes after only sharp ideas and sharp individuals.

Additionally, we are pleased to welcome Ms. Laurie Ann Lakatosh as IATAC's new librarian. Ms. Lakatosh, who holds a Masters in Library Information Science (MLIS) from Pittsburg University, comes to IATAC with extensive library and research experience. She is quickly becoming a key asset to the IATAC team. In the short time that she has been on our staff, she is already offering innovative ideas, tremendous skills, and vast amounts of knowledge and insight. We are pleased to have Ms. Lakatosh on the IATAC team, knowing that IATAC can only benefit more from her extensive research and library experience.

In this edition of the *IAnewsletter*, you will find several thought-provoking, well written articles. Dr. Cynthia Irvine and Mr. Michael F. Thompson from the Department of Computer Science at the Naval Postgraduate School (NPS) have written a very interesting article about an interactive IA education, training, and awareness tool, called CyberCIEGE. NPS, in cooperation with Rivermind, Inc., developed this interactive video game as a powerful IA teaching tool. Another article featured in this edition is "Software Agents (SA) A New Programming Paradigm." This in-depth article, which centers on software agent technology and programming misconceptions, is well worth reading. As always, you will find several other intriguing articles of interests as along with our recurring features. ■

Gene Tyler

Systems Engineering for the GIG: An Approach at the Enterprise Level

by Patrick M. Kern, Deputy to the ASD (NII) / DoD CIO

The Global Information Grid (GIG) is a complex, ongoing effort for integrating all information systems, services, and applications within the US Department of Defense (DoD) and the Intelligence Community (IC) into a seamless, reliable, and secure network that will support horizontal information flows and network-centric warfare.

The GIG represents a different way of thinking about delivering capabilities, one that can cope with the uncertainties we face in the world today. In the past, missions focused on narrow objectives against known adversaries, and we were organized with tightly managed organizational responsibilities across DoD and IC constituencies. Today, adversaries are shadowy and shifting, objectives are far reaching, and new responsibilities link our organizations at all levels. DoD and IC networks built in the past evolved into stovepipes, tied to missions and organizations that now are forced to adapt to a more fluid world. The GIG confronts uncertainty, inherent in today's world, with the agility that comes from interconnected, interoperable solutions that can be tailored to today's missions and objectives. Making the GIG a reality requires breaking out of stovepipes and solving interoperability and performance issues at the enterprise level. We have approached the problem of building, populating, operating, and protecting the GIG by applying systems engineering discipline to the complex set of communications systems, information systems, services, and applications that make up the GIG. Systems engineering as a discipline provides us with techniques to manage complexity of systems.

Enterprise-wide systems engineering (EW SE), as applying systems engineering to the GIG at this level is known, can only succeed by properly focusing the effort. EW SE uses interoperability and end-to-end performance as the criteria for determining what is within scope. Enterprise decisions for these requirements are then documented and enforced in the design of GIG component systems, laying the groundwork for the GIG Technical Foundation (GTF), the set of requirements on which the design of all future GIG component systems will be based.

Background

With origins in a wide range of component systems procured to support autonomous agencies and services, the GIG is more accurately an organizing construct than an actual system. Its legacy components vary in terms of performance, storage, and process and must continue to support their existing user communities, even as they become part of the GIG. Although many individual component systems are unknown at the enterprise level, the GIG's component set, and the components themselves, will evolve to reflect participant groups' capabilities and financial priorities. The challenge is to establish a process that brings together these disparate components into a single entity that meets all users' needs.

As GIG component systems are designed, built, and funded by member organizations, it is necessary to deductively establish the functions, protocols, and data models required for their interoperability and performance. Such an investment will benefit all GIG users.

Scope of the Effort

The Office of the Assistant Secretary of Defense for Network and Information Integration/DoD Chief Information Officer [ASD(NII)/DoD CIO] tasked the Defense Information Systems Agency (DISA) to lead an Enterprise Documentation Framework Working Group that would apply systems engineering practices to create what we now know as the GTF. The GTF provides structure and traceability for all GIG documentation in a manner similar to that of a document tree. DISA was also tasked to populate this framework after it was established.

Applying systems engineering at the enterprise level to support the development of the GTF must start with the GIG's vision as outlined in the Net-Centric Operations Environment Joint Capability Document (NCOE JCD) *http://www.jcs.mil/j6/netcentric.html*. Once top-level requirements are defined to identify the necessary functionality,

this functionality can be decomposed into system segments and subsegments.

Top-level requirements have been decomposed into three areas—General, Enterprise Management, and Information Assurance—and flow down to requirements at the segment level: Transport, Services, Applications, Computing Infrastructure, and Enterprise Operations.

Segment and subsegment requirements are specified as needed for interoperability and performance according to the top-level requirements, which can be traced from GIG capabilities and requirements to segment and subsegment requirements. Subsegment requirements, needed to achieve interoperability and end-to-end performance, are often the specification of protocols or mechanisms.

As an example of a Transport segment, Figure 1 illustrates the relationship among top-level requirements, segment-level requirements, and subsegment requirements.

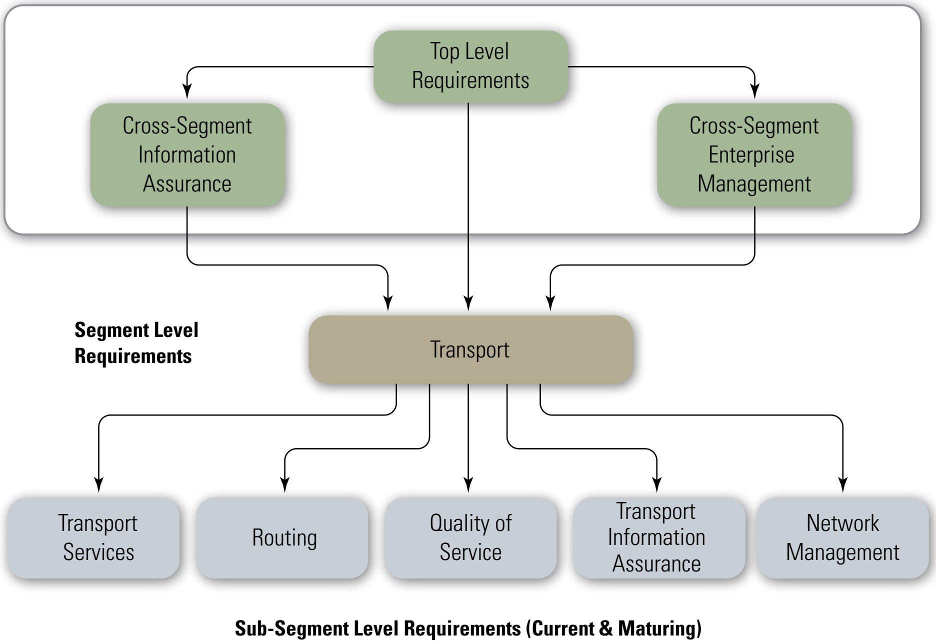

Top Level Requirements
Cross-Segment Information Assurance
Cross-Segment Enterprise Management
Segment Level Requirements
Transport
Transport Services
Routing
Quality of Service
Transport Information Assurance
Network Management
Sub-Segment Level Requirements (Current & Maturing)

Figure 1 Transport Segment

Systems Engineering Challenges

In addition to scope, the GTF addresses numerous systems engineering challenges involving focus, evolution, coverage, and applicability:

- **Focus**—Requirements for achieving the GIG capabilities must be specified by the GTF to ensure that they are not limited to what is feasible. Programs, services, and agencies responsible for existing GIG component systems will need to determine transition plans that reflect the requirements of the GTF.
- **Evolution**—Many aspects of the GIG's long-term vision, including pervasive mobility, ad hoc network connection, efficient resource use, and dynamic resource allocation/management, are not achievable by using current technologies. Long-term GIG design must not be limited to requirements that are dependent on current technology; they also must include provisions for emerging and future technologies.
- **Coverage**—The GIG is composed of a wide variety of components, many of which are unknown at the enterprise level. Components will be added and removed as organizational needs evolve, and the components themselves also will evolve. Consequently, GIG requirements must be specified in terms of component type rather than for specific components. Requirements must also be defined for the set of systems needed to meet GIG capabilities rather than for those

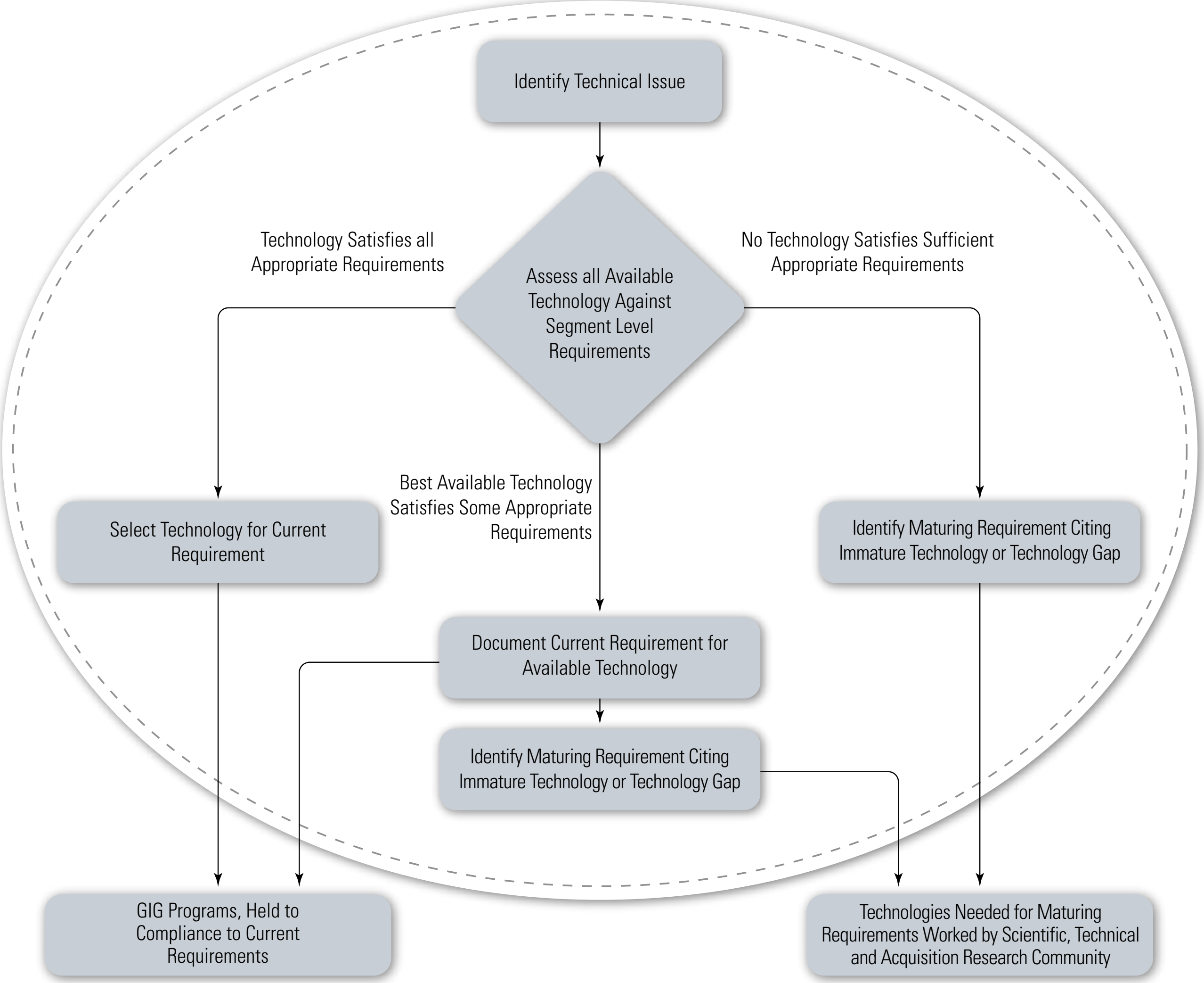

Identify Technical Issue
Assess all Available Technology Against Segment Level Requirements
Technology Satisfies all Appropriate Requirements
No Technology Satisfies Sufficient Appropriate Requirements
Best Available Technology Satisfies Some Appropriate Requirements
Select Technology for Current Requirement
Identify Maturing Requirement Citing Immature Technology or Technology Gap
Document Current Requirement for Available Technology
Identify Maturing Requirement Citing Immature Technology or Technology Gap
GIG Programs, Held to Compliance to Current Requirements
Technologies Needed for Maturing Requirements Worked by Scientific, Technical and Acquisition Research Community

Figure 2 Process for assessing technologies for inclusion in the GIG Technical Foundation.

appropriate only for existing and planned systems.

- **Applicability**—Because GIG component systems will operate in various environments, requirements need not apply to all environment or modes. Specific domains of applicability must be defined, which will work in concert to provide overall enterprise capabilities. For example, fixed users are well connected and can reliably reach centralized data centers. The fixed users are not severely constrained in power, memory, storage, and processing. Examples of fixed user modes are camps, posts, stations and bases served by the Defense Information System Network. Advantaged Tactical users operate in a slowly changing environment subject to high latency and limitations on bandwidth that may constrain reach-back to centralized data centers. The advantaged tactical users are not severely constrained in power, memory, storage, and processing. Examples of advantaged tactical user modes are Tactical Operations Centers and Navy ships. Disadvantaged Tactical users operate in a highly dynamic topology, with limited and sometimes no fixed infrastructure, subject to disruption in communications and with severe constrains on one or more of power, memory, storage, and processing. An example of disadvantaged tactical user mode is a Mobile Ad Hoc Network formed by vehicles and dismounted soldiers.

Assembling the GTF

The GTF will address all requirements relating to the GIG's long-term vision, even those not achievable by using available technologies, protocols, and mechanisms. Subsegment requirements are divided into two categories: current requirements, which are achievable using current technology; and maturing requirements, which rely on emerging and future technologies.

Current requirements are testable and will be enforced in the design of

GIG component systems. By contrast, maturing requirements are used to document technologies needed for achieving GIG capabilities, verify the feasibility of achieving GIG capabilities, and provide insight regarding research needed to meet the GIG vision.

Occasionally, use of a technology, mechanism, or protocol that does not satisfy GIG requirements is sanctioned if no other resource is available. In these instances, a current requirement is defined for the existing technology, and a maturing requirement is defined for the needed technology. For example, inter-domain routing today would use border gateway protocol version 4 (BPGv4) as a current requirement. A new protocol to support pervasive mobility is defined as a maturing requirement.

At all phases of the process of assembling the GTF, stakeholders and subject matter experts participate in working groups to assess technologies and determine the appropriate match for current and maturing requirements. Figure 2 illustrates the process used to assess technologies for inclusion in the GTF.

Community Role in GTF Development

Before the establishment of the GTF, different organizations attempted to define the GIG in separately developed technical, policy, and guidance documents. This effort resulted in more than 7,000 pages of documentation. Although well written, this documentation contained gaps, overlaps, and inconsistencies that reflected the GIG's fragmented origins in component systems originally intended to function independently.

Today's GTF consists of a set of source documents drawn across the GIG community, along with governing statements for GIG development, providing portfolio and program managers with clear guidance regarding how to implement Net-Centricity and end-to-end interoperability throughout the acquisition life cycle. It includes authoritative source documents that define the strategic guidance, operational context, operational capabilities, GIG capabilities, GIG activities, and technical direction needed for taking the GIG through the following time frames: near (0–2 years), mid (3–7 years), and far (8+ years).

The GTF also contains governing statements extracted from these source documents that describe the GIG more concisely and are traceable throughout the GIG's Enterprise Document Framework. All content is stored and managed in a Dynamic Object-Oriented Requirements System (DOORS ™) requirements database to facilitate requirements management and configuration control.

Compliance

By developing this integrated approach to compliance assessment that aligns current processes and provides an entry point to the Net-Ready Key Performance Parameter (NR-KPP) evolution, the GTF—

- ▶ Allows program managers to self-assess individual programs
- ▶ Can be applied consistently to all programs at all oversight levels
- ▶ Ensures high confidence in end-to-end interoperability and performance compliance at the enterprise level

Policy also has been revised to direct all compliance to the GTF.

Conclusion

The GIG is an ambitious undertaking that is fundamental to network-centric warfare. We have established an Enterprise process to apply systems engineering discipline to the decisions that need to be made to make the GIG a reality. The product of the enterprise approach is a GTF, a new approach to GIG policies and a set of processes for compliance to the GTF. While the GTF is still an evolving effort, the requirements in the GTF have been flowed into program requirements documents, ensuring more robust interoperability and performance as those programs come on-line as part of the GIG. The approach we are putting in place at the enterprise level will allow us to build, populate, operate and protect the GIG to meet the challenges of today's world. ■

About the Authors

Patrick M. Kern | is the NII Net-Centric Systems Engineer leading the integration of transformational programs for OSD /NII. He is responsible for end-to-end system engineering for the Global Information Grid (GIG). As the senior system engineer for net-centric initiatives, he—

- ▶ Establishes and manages the end-to-end systems engineering function across GIG programs
- ▶ Evaluates program satisfaction of established net centric criteria
- ▶ Supports the services and agencies as they develop and deploy GIG initiatives.

Before joining NII, Mr. Kern held business development leadership positions for the next generation military communication systems at Boeing Space Systems and Lockheed Martin. From 1994–2004, he led the initial development of DoD, International satellite communications (SATCOM) programs, and integrated telecommunications architecture analysis for military and commercial SATCOM, end-to-end Network Operations and related classified new business initiatives.

After serving 25 years in the USAF, Mr. Kern retired as a Colonel in 1994.

Mr. Kern earned a BS in Aerospace Engineering from the University of Michigan in 1969 and a Master of Business Administration in Engineering Management from the University of Colorado in 1976. He attended Air Command and Staff College at Maxwell AFB and the Industrial College of the Armed Forces at the National Defense University.

He may be reached at the Office of the Assistant Secretary of Defense, Networks and Information Integration, by telephone at 703/695-2855, or by email at Systems.Engineering@osd.mil.

Software Agents (SA): A New Programming Paradigm

by Giorgio Bertoli

Software agent technology constitutes a new programming paradigm with the potential to revolutionize the way software is developed. Unfortunately, the underlying concepts pertaining to this new programming methodology are nebulous and heavily dependent on the frame of reference and context in which they are utilized. This issue has resulted in the abuse of the "software agent," or "intelligent software agent" moniker, effectively turning it into a buzzword that is prominently displayed in product marketing literature, with little regard for correctness of usage.

This paper explains software agent technology beyond the conventional "agent" specific attributes, which are normally offered as a pseudo-definition for this new programming construct. The article further discusses the current issues and misconceptions associated with software agent programming and, through a simple example implementation, illustrates the fundamental differences between it and the conventional client-server model. The paper then concludes by discussing the key advantages and disadvantages of this new programming paradigm and its potential future.

Introduction

In the past decade, along with the continued increase in information technology (IT) and Internet services, a new programming concept known as software agents [1] has emerged within the academic, commercial, and government communities. This new software development methodology makes many promises. Among them is the utopian vision of autonomous mobile code that can exist within "the network," sensing and interacting with this virtual environment to achieve a complex goal with minimal human supervision. Current software agent technology is, however, still relatively immature [22] and has only recently begun a concerted standardization effort. [15]

Of key concern is that conflicting frames of reference [18], coupled with a general lack of understanding of basic software agent concepts in the software and business communities, have caused the "software agent" moniker to be proliferated with little regard for consistency or accuracy. If left unchecked, software agent technology is doomed to suffer the same disillusioned fate that artificial intelligence suffered in the 1980s, when it was demonstrated that it could not come close to meeting its touted expectations. [30] This document is a guide for software engineers and program managers who wish to understand the true intent and capabilities of software agent technology, its potential benefits, and associated implementation issues.

Brief Description of the Problem

Before defining software agent technology, it is important to first make a very clear distinction between a "software agent" and "software agent programming". [2, 24] A software agent constitutes a particular implementation of software agent technology as applied to some domain-specific problem. In the physical world, this is analogous to a Ford Taurus being a specific implementation of automotive technology, or even better, Internet Explorer being a specific implementation of web browsing technology. The key point is that you cannot have the former without the latter. A web client would be of little use without the supporting infrastructure of the World Wide Web and its associated protocols. Similarly, a software agent alone cannot simply be; it must have a supporting architecture of protocols and services to sustain its existence and allow it to function as intended.

Though subtle, this concept is at the core of most misunderstandings related to software agents. Unlike automotive or web technology, software agent technology is still in its infancy; incomplete in its definition, with limited software engineering techniques; and even more problematic, not standardized. [20, 22, 23] Yet, over the past several years, an increasing number of software products, claiming

to use software agents, from intrusion detection systems to steel production optimization software [16, 29] are being advertised by academia and industry alike. Consequently, a would-be consumer or technology manager is left with the arduous task of having to differentiate between products that do indeed apply appropriate software agent programming principles (not many exist) with those that do not.

This new software development methodology makes many promises. Among them is the utopian vision of autonomous mobile code that can exist within "the network," sensing and interacting with this virtual environment to achieve a complex goal with minimal human supervision.

Common questions that most managers ask when reviewing products claiming to use software agents are, What is a software agent? [31, 36], and How do you distinguish between a software agent and a regular program? [18] These, however, are not the best questions to ask. It is much more important to focus on whether a developer defines and uses an appropriate framework [32] for their software agent implementation, and does not instead simply repackage standard programming models and then lets marketing take them the rest of the way. [3] The following questions are much more probative and revealing: why was software agent programming used for this application? How is software agent technology, as implemented in this application, providing benefits over the use of conventional programming methodologies? And finally, what software agent framework is this developer using or providing to support this implementation?

Comparable confusion and turmoil has occurred repeatedly in all industries as companies battle to institute their developed technology solution as the standard. Unfortunately, software agent technology is still mainly in the realm of academia and research. As such, there is no true driving force, such as near-term realizable profits, or market share to be gained, to fuel the investment that would be necessary for the advanced development and standardization of this technology. The reason for this is simple. Though software agent technology has the potential to outperform current conventional programming methods, the commercial world has no immediate need (no "killer" application) for this advancement [29, 33] because current development practices are still more than adequate for performing needed software functions. Until software agent technology proves itself to be a significant improvement over present programming practices, no industrial impetus will be present to drive the standardization of a universal software agent framework and the development of an accompanying software agent application programming interface (SA–API), both of which are essential if software agent technology is to ever achieve its full potential and live up to expectations. Until then, we will continue to have numerous, nonstandardized, application specific software claiming "software agent" capabilities with no hope of ever being able to achieve other then rudimentary collaboration, interoperability, or the true benefits that software agent technology was designed to provide.

Software Agent Technology

Having defined the difference between software agents and software agent technology and some of the associated issues, what exactly is this technology, and what is its intended purpose? This question is also inconsistently answered in available software agent literature. [17, 18, 33, 35] Asking someone to define software agent technology usually yields the same ambiguous response: "A software agent is software code that is autonomous, mobile, can react and adapt to its environment, can cooperate with other Agents, is long lived, can learn, *etc.*". [17, 20, 21] A key problem with this definition is that it describes a software agent, not the software agent technology itself. This is analogous to asking someone to define automotive technology and having that person respond with a list of generic automobile attributes (*e.g.,* power steering, six cylinders, bucket leather seats). Although true, these properties do not do a good job of explaining what a car is, how it works, and what its intended function is. Similarly, this standard definition, although it does provide a glimpse of some of the attributes we might wish software agents to possess, does not explain the reason that these properties are useful or why these capabilities are now assigned to a software agent rather than to more complicated software code. Consequently, it is essential to explain what software agent technology is intended to provide.

Software agent technology is a new way of programming. [20, 33] It is a new programming paradigm, just as structured programming was in the 1970s and object-oriented and network programming were in the 1980s and 1990s. This new programming construct tries to redefine the way we write software and formulate our problem space by making a software agent the new basic encapsulation structure. In this paradigm, instead of objects with associated data and methods, [4] we now have all the standard attributes commonly associated with software agents. [24] Furthermore, software agent programming is network based. A fundamental attribute associated with software agent technology is its inherently distributed nature. [20] As such, attempting to make a stand-alone software agent application does not make much sense.

The basic philosophy supporting the use of software agent programming is to transition away from the conventional, bandwidth intensive, and static client-server programming model to a more dynamic and truly distributed construct. With software agent technology, computation is no longer resident on a specific central node; rather, it is fully abstracted from any specific network or hardware component and is free to move among diverse processing resource to increase efficiency, reliability, performance, and ease of implementation. A recent trend accentuating this fact is the current merging of software agent programming and grid computing principles. [19] Software agent technology also aspires to promote cooperation between software agents. [5] This is very much like the object-oriented programming concepts of code reuse but in a dynamic, not static, form in which cooperation occurs among running processes, not offline during development between programmers sharing class structures. This goal, however, has yet to be realized (in a generally applicable form) in available software agent frameworks. [23, 24, 30, 35]

Example Application

The following example (see Figure 1) explains how software agent programming differs from conventional network programming models such as client-server.

A few assumptions made are that issues associated with information assurance (IA) [6] and time synchronization [7] are already solved and need not be considered for this example. We will also assume that each sensor "knows" and on request (via some function call) can provide its position and/or distance relative to its neighbors, either through the Global Positioning System (GPS) or manual configuration during emplacement. Although perhaps a bit contrived, these assumptions will enable us to focus on the intrinsic differences between these programming methodologies, rather then on issues which are not directly related to this problem space. As Figure 2 illustrates, using conventional programming practices, a likely implementation would use the client-server model. In this implementation, each sensor (client), when triggered by a passing vehicle, will send a message back to the remote console (server) indicating that it sensed a vehicle. It is then the responsibility of the server application to correlate and process all sensor messages and extract from them which sensor events correspond to a triggered sequence of signals identifying a specific vehicle traversing the sensor field.

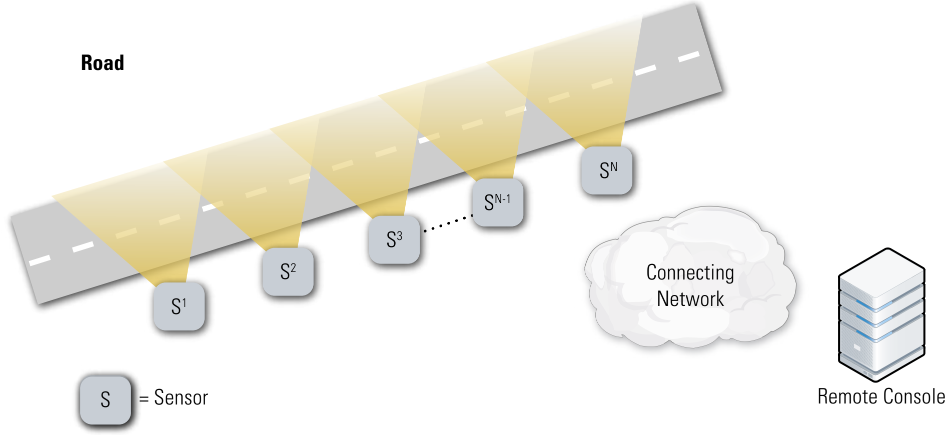


Figure 1 We wish to monitor a road for the presence of vehicle traffic. The goal is to sense when vehicles, traveling in either direction, pass through this road section by monitoring several sensors, numbered S^1 to S^N, located at the edge of the road. Then, based on sensor feedback, compute the direction and speed of the vehicle and display the derived information to a remote console connected to the sensors via some network.

The advantage of this approach is that the problem's solution is kept conceptually simple. This solution is easy to visualize and architecturally implement. In addition, sensors can be rendered minimally complex. At the extreme, they need not have any processing power at all. These simplifications, however, are mortgaged on the added complexity required at the server. Being the central hub of computation, the server will need to be capable of handling the worst case sensor message load and must have enough processing power to sustain providing vehicle alerts and associated metrics to the operator in a timely manner. Furthermore, the lack of any computation or data reduction at the sensors requires that all detection events be transmitted to the server. This places the full burden of supporting this requirement on the connecting network, which is not a particularly desirable trait in bandwidth sparse environments. There also is the issue of providing sensors a means of locating the server. Configuring each sensor with a static server Internet Protocol (IP) address might be too restrictive, whereas the alternative of creating a supporting discovery protocol mechanism, to advertise the server's location, places added overhead on the network and substantially increases the complexity of this solution.

From the computer programmer's point of view, although the client is trivial to develop, the server application requires additional overhead, which does not directly relate to the problem at hand. For instance, provisions must be made to account for instances in which multiple vehicles cross the sensor field from the same or opposing direction at close intervals in time. In circumstances of heavy traffic, the correlation of all sensor data in an attempt to determine which triggered events belong to the same vehicle could become intricate and error prone. Lastly, a key weakness that is inherent with the client-server model is that it inevitably creates a single point of failure in the system unless expensive steps are taken to implement redundancy and automatic switchover mechanisms.

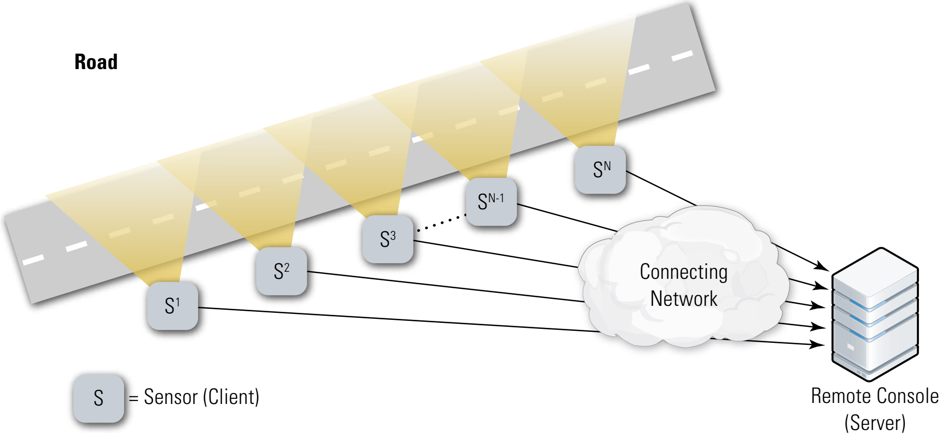


Figure 2 Software architecture for the described example problem as implemented using the client server model.

Implementing a solution for this same problem using software agent programming would instead yield an architecture similar to the one displayed in Figure 3. Using the software agent programming paradigm, each node in the system (sensor and console) would be equipped with a small [8] agent server (AS). [9] This is simply a running program that can understand what an agent is and allows it to gain access to local processing resources. [10] Furthermore, in this example implementation, the two outermost sensors would have a software agent (henceforth referred to as vehicle detection agent [VDA]) passively executing on them at all times.

When a vehicle passes by either of these outer sensors, the VDA would trigger, record the time of the event, duplicate itself, and then send its clone to the next neighboring sensor. When the vehicle then passed that sensor, the process would be repeated until the agent arrived at the final sensor node, accumulating time stamps and calculating vehicle speed along the way. In addition, the user console would be implemented as a software agent. This user console agent (UCA) would simply need to display vehicle detection events for the operator as reported by the VDAs. Lastly, most software agent frameworks also include a form of global management console (MC), which allows for the holistic view and control of all agents in a particular

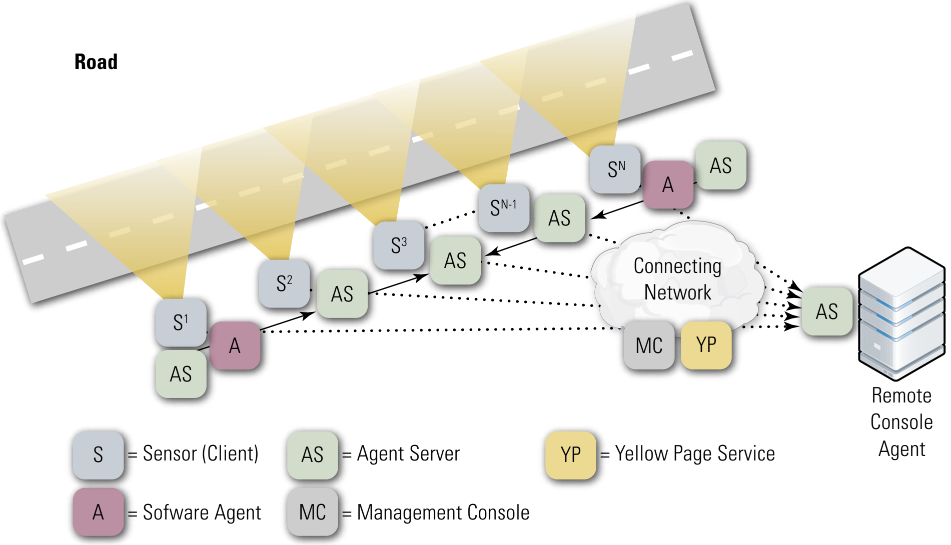


Figure 3 Software architecture for the described example problem as implemented using the software agent programming principles.

domain, and a Yellow Page (YP) or directory service, which enables agents to advertise their capabilities and locate each other. [38]

When implemented in this manner, the software agent—not the remote console—is performing all the work. As it moves from sensor to sensor, the cloned agent would collect the time the vehicle passed by each sensor; from that (along with the distance between sensors), it could compute the vehicle's speed and direction easily. Once the agent reached the final sensor, it would then need to send back only one very simple consolidated message (*e.g.,* vehicle direction = northeast; speed = 47 mph) to the user console agent for display.

Another very important advantage that software agent programming possesses over conventional network programming is that redundancy (hence, increased reliability) is inherent in the architecture. With a software agent implementation, a single point of failure no longer exists.

The concept is straightforward, but radically different from conventional programming practices. The computation for this application is not in the server, nor is it performed consistently on any particular sensor node. Rather, it is implemented as an "agent" entity that does not have a fixed processing resource. This effect can be amplified further by a slight modification of our previous implementation. If the goal is simply to determine vehicle direction and speed, then the VDA may be programmed to report back to the UCA before it reaches the final sensor node. For instance, if the calculated vehicle speed remains consistent after several sensor hops, then it seems reasonable to assume that the vehicle is traveling at a constant velocity and no additional measurements are required.

Another important benefit that this implementation provides is that the vehicle detection problem is naturally segregated to a per-vehicle occurrence basis. The programmer need not worry about correlation of sensor signals or the determination of which triggering event belongs to which signal set. Instead, each initial sensor event inherently assigns a dedicated VDA to track that specific vehicle. If another vehicle comes along shortly thereafter, a second agent is dispatched, and so on. Vehicles traveling in opposite directions also can be easily mediated via intra-agent communication. For instance, if two vehicles were passing the sensor net from both directions almost simultaneously, there would come a time (somewhere in the middle) when the corresponding VDAs would pass each other (be on the same sensor node). At this time, they could easily acknowledge each other's presence if only to say, "Hi. I'm VDA Bob. I'm investigating vehicle 1 going to the left—you stick with your vehicle 2 going to the right, and don't be confused by duplicate sensor events."

System expandability is also greatly improved. In the client-server model, if capability augmentations are desired, the server would need to be amended, recompiled, regression tested; undergo configuration management; and then be reinstalled, reconfigured, and executed. On the other hand, augmenting capabilities in a software agent system requires only the development of new agents. For example, assume that the same sensors we employed to detect motion also had vibration detectors (which are currently unused by our VDAs), and that another developer wanted to use this feature to determine the approximate weight of a vehicle as it passed by. Also assume that to perform this calculation, we would need to know how fast the vehicle is traveling. One option could simply be to rewrite our VDA to include this new feature; however, this would not be much better then modifying the server code. Instead, if we had a standard software agent application programming interface (SA–API), the programmer could simply design a new "vehicle weight detection agent" with an interface that would allow it to follow a VDA as it moved to track a vehicle and to query it for speed information as required. In this manner, all previous code is completely unchanged and never has to stop executing for this new functionality to be added.

Another very important advantage that software agent programming possesses over conventional network programming is that redundancy (hence, increased reliability) is inherent in the architecture. With a software agent implementation, a single point of failure no longer exists. As mentioned previously, the user console itself is also an agent that can be cloned and dispatched as desired. Multiple UCAs can easily be deployed on different network nodes and registered with the YP service to receive any required agent information. The YP service within the software agent framework will automatically ensure that all registered agents within the environment can continuously locate each other and interoperate. [11] The inherently distributed nature of software agent technology is a critical requirement to what truly constitutes software agent programming.

Academia and industry have already conducted considerable work in implementing software agent programming frameworks and APIs. Cougaar [28], JADE [27], CoABS [38], Aglets [37], ZEUS [39] and Jack™ [25, 26] are only a few of the public domain and commercial software agent architectures available today that could be used for the realization of this sample problem and more. Each of the software Agent architectures listed provides the infrastructure, protocols, and resources required for easily developing agent software. At a minimum, these include the AS, agent communication protocols, YP services, MC services, support for agent mobility, and security mechanisms. [12] All of this build in functionality fully segregates a developer from the complexities associated with the underworkings of the software agent architectures and their associated components, freeing them to focus entirely on the problem domain at hand.

The following pseudocode is a potential software agent (framework and SA–API nonspecific) implementation of the VDA.

Agent: *Vehicle Detection Agent*

```
1.  SPEED = 0, AVG_SPEED = 0,
    DIRECTION = 0;
2.  DISTANCE = 0, HOP_COUNT = 0;
3.  CUR_LOCATION = this.location();
4.  Wait: Sensor Event;
5.  Upon: Sensor Event;
6.  INIT_TIME = Time.now();
7.  TIME = INIT_TIME;
8.  Clone Agent;
9.  If (Clone) {
10. Dispatch to neighbor Sensor;
11. Upon: execution at new node;
12. HOP_COUNT = HOP_COUNT + 1;
13. If (another agent is present here) {
14. Communicate: resolve multiple
    vehicles events
15. } //end if at line 13;
16. Wait: Sensor Event;
17. Upon: Sensor Event ;
18. DISTANCE =
    (CUR_LOCATION – this.location());
19. DIRECTION = sign(DISTANCE) [13];
20. CUR_LOCATION = this.location();
21. SPEED =
    |DISTANCE| / (TIME - Time.now());
22. TIME = Time.now();
23. AVG_SPEED = running average of
    SPEED;
24. If (more unvisited neighbors exist)
25. Loop to 10;
26. } //end if at line 22;
27. else
28. Communicate: to Console Agent
    (INIT_TIME, AVG_SPEED, DIRECTION)
29. Terminate();
30. } //end else at line 25
31. } //end if at line 9
32. start else {
33. Loop to step 4
34. } //end else at line 28
End AGENT
```

Key agent capabilities, such as the ones displayed in bold italic (Clone, Dispatch, and Communicate), are intrinsically provided by the SA–API. As can be seen, a minimal amount of code could be categorized as nonproblem-specific software overhead. The actual amount of code required to implement this agent would vary depending on the software agent framework used; however, even with a relatively rudimentary SA–API, no more than a few dozen lines of code and minimal effort should be required on the part of the programmer.

All is not perfect. Agent programming does have disadvantages. The sensors must now have enough processing power to run the AS. [14] Furthermore, the sensors not only need to communicate back to the server but also must be able to communicate with each other. Aside from these augmented hardware requirements, some of the advantages previously discussed relating to software agent programming come at the expense of some performance. Agent frameworks, available now, have a moderate to potentially high amount of associated overhead. How much of a performance penalty is incurred is problem and implementation specific and not well documented, but it will likely need to be accounted for.

Another interesting software agent performance consideration, noticeable from this sample implementation, arises from having VCAs (which run directly on the sensors) perform all the processing. If not properly mitigated, this has the potential to introduce an unacceptable processing delay error in the performed calculations. For instance, referring to the pseudocode provided, the time at which line 17 is triggered should be the time used for calculating the vehicle's speed. However, in this implementation, a time reference is not performed until line 21. This time delay would result in noticeably less accurate results. To prevent this from occurring, it would have been better to introduce a new variable (say TRIGGER_TIME) immediately after line 17 to record the time the vehicle passed by the sensor as accurately as possible; this value could then be used for all subsequent calculations.

Some of the previous issues presented in the client-server implementation example are also not entirely alleviated. For instance, in that implementation, clients needed to know how to locate the server, which could be accomplished by use of a static IP address, or dynamically by implementing some type of discovery protocol. Now instead, VDAs (as seen on line 10) need to know how to locate their immediate neighbor sensors. This, however, is not as large of a limitation as it was in the client-server case. Using static IP addressing is still not ideal; however, because the IPs would be preconfigured within the agents, updating them would be relatively simple and would not require reconfiguring each sensor node, but only the two initial VDAs.

Alternatively, implementation of a discovery protocol, which would be a daunting task in the client-server example, is highly simplified by using the inherent capabilities available in software agent programming. For this example, we could create a trivially simple SENSOR AGENT, whose sole function would be to register the location and IP address of a particular sensor with the YP service on initiation or on any change in address or location. Then, by deploying one such agent on each sensor, VDAs would be able to determine which sensor node to dispatch to next by simply querying the YP service. This modified implementation not only alleviated the original sensor discovery issues, but additionally allows for sensors to be moved, replaced, added, or removed, and the system would be able to automatically reconstitute itself and continue to provide vehicle detection capabilities.

Further, software agent limitations reside with the maturity of available agent frameworks. I do not believe any SA–API currently offered is mature enough to meet everyone's software agent needs, particularly in areas related to information assurance (IA) and security. Although some type of security mechanism is included with most software agent frameworks, it is doubtful that any would meet commercial requirements [35]. Furthermore, with current SA–APIs, software agent programming is not quite as simple as presented [32], especially when requiring events to occur in a synchronized manner. Though not necessary for the simple example presented, requiring interactions and execution of multiple agents to follow a specific timing sequence or progression is not without challenges.

The key disadvantage to software agent programming is, however, not directly related to the technology itself. As demonstrated by this simplistic application, although the utilization of software agent technology is advantageous, most of the benefits gained are negated because conventional programming methods, with which a programmer is likely to be considerably more familiar, are effective at implementing a satisfactory solution. In turn, this does not instill a sense of urgency for adopting this new programming paradigm. Much like the slow transition from IPv4 to IPv6, a technology must clearly outlive its operational effectiveness before industry is willing to invest in the costs associated with migrating to the next generation solution.

Conclusion

Software agent technology defines a new programming paradigm that is conceptually vastly different and potentially superior to conventional network programming models such as client-server. When determining if an application is using software agent technology, the standard attributes applied in defining a software agent (*e.g.,* autonomous, mobile, reactive, long lived) are to ambiguous. It is instead more important to focus on the implemented software agent framework then the agent itself and to then evaluate if the application in question is indeed applying sound software agent programming principles.

By implementation of a simple example problem, the major differences between software agent programming and the client-server model are prominently visualized. The key feature, which imparts software agent technology the greatest advantage and potential is its inherent ability to simplify the implementation of truly distributed and decentralized software solutions, thus allowing programmers to focus on the problem at hand rather than the required supporting protocols or data correlation overhead.

Unfortunately, software agent technology has not yet demonstrated a clear application that would be infeasible to implement using current programming methodologies. Much potential has been shown by applying this technology to difficult industrial and commercial applications. [29, 33, 35] The intrinsically complex nature of these problem domains has, however, currently prevented the development of a truly revolutionary solution. Until this occurs, software agent technology will not become widespread in the programming and industrial community. In the meantime, beware of aggressive marketing. ■

About the Author

Mr. Giorgio Bertoli | is an Electronics Engineer in the Intelligence and Information Warfare Directorate (I2WD), Communications-Electronics Research Development and Engineering Center (CERDEC), US Army Research Development and Engineering Command (RDECOM), where he is currently serving as Chief of the Offensive Information Operations (OIO) Branch.

With 14 years of federal service, Mr. Bertoli has extensive government experience in information operations and military tactics as a civilian and as a former active duty soldier. His primary research areas include the development of advanced electronic warfare (EW), computer network operations (CNO) and quick reaction capability (QRC) technologies. Mr. Bertoli, a highly proficient programmer in several computer languages, is a subject matter expert in genetic algorithms and software agent technology.

Mr. Bertoli holds bachelors and masters degrees in Electrical Engineering from the New Jersey Institute of Technology, as well as a second masters degree in Computer Science from the University of Massachusetts. He is also a Certified Information Systems Security Professional (CISSP). Furthermore, he is the recipient of the "2002 AFCEA Young Engineer of the Year" award and holds several decorations of achievement for his active duty military service. Mr. Bertoli can be contacted at giorgio.bertoli@us.army.mil

References

1. It is also common for this new programming construct to be referred to as "intelligent software agents (ISA)." I purposely omitted the "intelligent" descriptor because it is subjective. Such an adjective can be applied to any software application, based solely on your definition of what constitutes intelligence. As related to software agents, this attribute does not add much in the development of an intuitive understanding of this technology and thusly is omitted from this discussion.
2. The terms "software agent technology," "software agent programming," and "agent oriented programming" are used interchangeably in this document as done in most software agent literature.
3. I once was discussing with a company a proposal for a new software tool. After only a few slides into their presentation, I realized that they were simply reversing the client-server model. When I asked them about this, they stated that the client-server model has one server to many clients while instead they had one client to many servers, and this was their definition of a software agent. Although perhaps imaginative, this is most definitively not the case.
4. As related to object-oriented programming.
5. Commonly referred to as multiagent systems (MAS).
6. Such as the encryption of communication links and authentication mechanisms.
7. Ensuring a consistent network time reference.
8. There is an obvious tradeoff in the size of the AS and the amount of functionality that is desired to reside within it. At its simplest, the server needs only to recognize, validate, and allow the agent to execute.
9. Each of the various software agent frameworks available has its own name for this architectural construct. However, the functionality is practically identical; I find "agent server" to be the most intuitive of the naming conventions.
10. I find that many technology managers have poor understanding of the required existence of this agent server. For an agent to migrate between network nodes, there must be a running process resident and listening on the receiving host who understands what a software agent is, and on transfer allows it to execute within the local processing environment. Without such a mechanism, agents cannot exist (or at least they cannot move—a quality I think is essential of true software agent

programming). You cannot just "send" an agent to a random machine and hope that somehow it will be properly interpreted and allowed to execute.

11. The YPs and MC services are also usually implemented as agents and inherently support redundancy and/or migration.
12. Comparing the differences between these software agent architectures and APIs is beyond the scope of this document. Note that even though the basic concepts remain unchanged, significant implementation differences do exist.
13. DIRECTION is simply deduced from the sign (positive or negative) associated with the calculated distance between sensors. This calculation can be performed only once; for the sake of brevity, it is repeated in this sample implementation.
14. Furthermore, because all available agent frameworks are written in a high-level programming language, the sensor will likely need to be able to support at least a minimal operating system and possibly a runtime environment if an interpreted language (*e.g.,* Java™) is used.
15. IEEE computer society standards organization, Foundation for Intelligence Physical Agents; *http://www.fipa.org.*

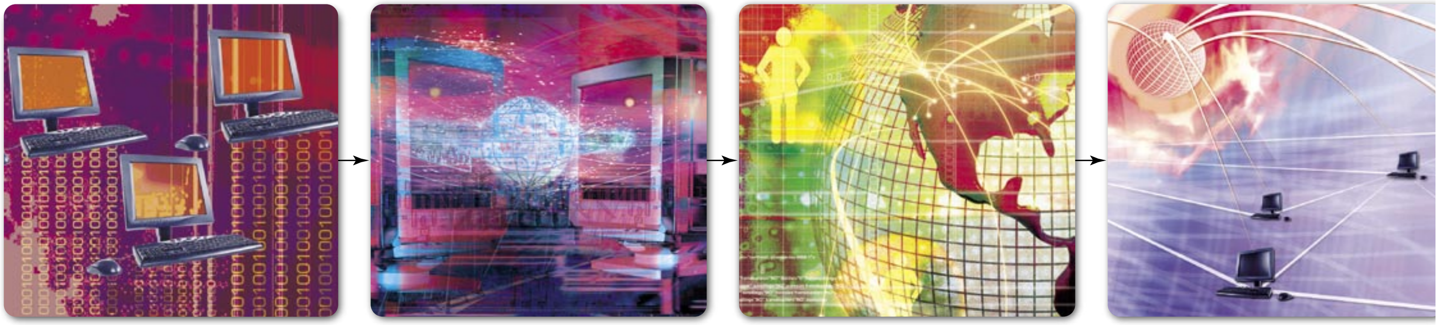
16. H. Helaakoski, S.C. Feng, K.K. Jurrens (USA), K. Ojala, and J. Kipinä (Finland), *Collaborative Software Agents in Steel Product Industry*, ACTA Press, From Proceeding (411) Artificial Intelligence and Applications, 2004.
17. Bjorn Hermans, *Intelligent Software Agents on the Internet*, First Monday journal, 1997.
18. Stan Franklin and Art Graesser, *Is it an Agent or Just a Program? A Taxonomy for Autonomous Agents*, Proceedings of the Third International Workshop on Agent Theories, Architectures, and Languages, Springer-Verlag, 1996.
19. Ian Foster, Nicholas R. Jennings, and Carl Kesselman, *Brain Meets Brawn: Why Grid and Agents Need Each Othe*r, In Proceedings of 3rd Int. Conf. on Autonomous Agents and Multi-Agent Systems (AAMAS 2004), New York, USA, 2004.
20. Peter Leong, Chunyan Miao. and Bu-Sung Lee, *Agent Oriented Software Engineering for Grid Computing*, Proceedings of the Sixth IEEE International Symposium on Cluster Computing and the Grid Workshops, 2006.
21. M. Wooldridge and N. R. Jennings, I*ntelligent Agents: Theory and Practice*, Knowledge Engineering Review, 1995.
22. F. Zambonelli and A.Omicini, *Challenges and Research Directions in Agent-Oriented Software Engineering*, Autonomous Agents and Multi-Agent System, September 2004 , pp. 253–283.
23. Xue Xiao Zeng Junfang Liu Liding , *Towards an Engineering Change in Agent Oriented Software Engineering*, Proceedings of the First International Conference on Innovative Computing, Information and Control, 2006.
24. Francesco Pagliarecci, Luca Spalazzi, Gianluca Capuzzi, *Formal Definition of an Agent-Object Programming Language*, IEEE, 2006.
25. Howden, N., R. Ronnquist, A. Hodgson, and A. Lucas, *JACK™—Summary of an Agent Infrastructure*, 5th International Conference on Autonomous Agents, 2001.
26. *JACK™ Intelligent Agents: Agent Manual*, Release 5.2, June 10, 2005, Copyright © 1999–2006, Agent Oriented Software Pty. Ltd.
27. F. Bellifemine, G. Caire, A. Poggi, G. Rimassa, JADE–*A White Paper*, exp -Volume 3 - n. 3, September 2003.
28. Aaron Helsinger, Michael Thome, Todd Wright, Cougaar: *A Scalable, Distributed Multi-Agent Architecture*, IEEE 2004.
29. H. Van Dyke Parunak, *Practical and Industrial Applications of Agent-Based Systems*, Copyright © 1998, Industrial Technology Institute.
30. Hyacinth S. Nwana and Divine T. Ndumu, *A Perspective on Software Agents Research*, 1999.
31. Hyacinth S. Nwana, *Software Agents: An Overview, Knowledge Engineering* Review, Vol. 11, No. 3, pp. 205–244, October/November 1996.
32. Michael Wooldridge and Nicholas R. Jennings, *Pitfalls of Agent-Oriented Development*, 1999.
33. N. R. Jennings and M. Wooldridge, Applications of Intelligent Agents, 1998.
34. Y. Shoham, *Agent Oriented Programming*, in reading in Agents, M.N. Huhns and M.P. Singh (ed.), Morgan Kaufmann, 1998.
35. Michael Luck, Peter McBurney, Onn Shehory, Steve Willmott, and the AgentLink Community, *Agent Technology: Computing as Interaction: A Roadmap for Agent Based Computing*, AgentLink report, September 2005, available from www.agentlink.org/roadmap.
36. Jeffrey M. Bradshaw, A*n Introduction to Software Agents*, in Software Agents, Bradshaw, J.M. (ed.), Cambridge, MA: MIT Press, 1997.
37. Bill Venners, *The Architecture of Aglets*, JavaWorld.com, 04/01/97.
38. Martha L. Kahn and Cynthia Della Torre Cicalese, *The CoABS Grid*, Presented at Goddard/JPL Workshop on Radical Agent Concepts, Tysons Corner, VA, January 16–18, 2002.
39. Hyacinth S. Nwana, Divine T. Ndumu, Lyndon C. Lee, and Jaron C. Collis, *ZEUS: A Toolkit for Building Distributed Multi-Agent Systems*, Applied Research and Technology, BT Laboratories, 1999.

Enabling Mission Critical Operations Through Mature Implementation

by Nadya Bartol, Eric White, Stephanie Shankles, and Michelle Moss

Operations environments are growing increasingly complex as companies and agencies join the net-centric community, where architecture is collaborative and information access instantaneous and global. Led by the U.S. Department of Defense's (DoD) vision of Net-Centric Operations and the demands of modern warfare, a community has formed in which warfighters and business and intelligence users can share knowledge on a secure and reliable network anywhere worldwide. [1] The dilemma that weighs heavily on the minds of information technology (IT) managers and technology leaders is how they can support the community's warfighters and still respond consistently, with accuracy and speed, to mitigate the risks their systems face. These risks come not only from their own network but also from others with whom they interface. For example, to provide warfighters access to real-time information on the ground, their vehicles were networked. With the increased functionality comes increased risk. Now, the enemy capture of a US Army Humvee represents more than simply loss of transportation; it may also be a potential threat to the greater tactical network. The Army vehicle, through its on-board computer, is linked to the Marine Corps' ground network, which is part of the Navy's tactical networks. If that opportunity is exploited, all connecting information and networks share the potential risk. [2]

To mitigate those risks, an increased number of industry leaders are seeking to protect their missions and systems by using a powerful combination of information assurance (IA) management tools and processes that strengthen security of core business operations and help them interact with external organizations seamlessly and securely. These IA tools center on implementing maturity model principles, coupled with other accepted industry specifications and standards. Organizations implementing these IA tools are improving the responsiveness and robustness of IA operations and are facilitating an increase in their effectiveness, support repeatable execution, and ability to respond to sudden events with confidence. As Figure 1 illustrates, teams are learning that repeatable execution takes the guesswork out of response and frees up time and resources for solving real problems. Operational IA standards and controls are helping leaders move forward and identify other areas that could benefit and improve using the same standards and controls. Defined and well managed activities not only lead to better managed and lower IT costs but also support increased collaboration between the IT and business teams. Organizations are pleased to realize that



Stove Piped
Disparate Systems

Networked
Connected Systems

Integrated
Enterprise Systems

Seamless and Dynamic
Net-Centric

Figure 1 Impacts of Creating an Effective Organization

as they learn to master effectively running their teams and projects using effective standards and controls, they are able to build and manage more reliable systems and handle greater complexity.

The need to work across the IA community to build stronger systems for net-centric operations is apparent. Yet, many organizations struggle with internal divisions operating independently with disparate IA, policies, and operational processes. This means missions are inadequately defined, resources are duplicated, and debates persist over methods for IA implementation, which translates into increased cost and can waste valuable response time. Organizations need to be able to work across functions and respond to threats and challenges to increase success and protect assets. Employing standardized measures and procedures, built into the systems and operations from inception, exponentially decreases the risk and cost incurred because teams can act without guesswork when action is needed.

Protecting and enabling the warfighter in the field means that an organization's core business practices need to be structured, secure, and interoperable. This does not imply bureaucracy; on the contrary, structured, secure, and interoperable processes provide a foundation for making complex decisions timely and effectively, which impacts the number of benefits. Team members actively participate in creating standards and roles ensuring a relevant structure. Leaders experience a productivity increase because only traceable, authorized work is performed by teams that, by following the defined structure and processes, minimize overlapping efforts. Service quality increases as irregularities decrease, eliminating distractions and allowing team members to focus their energy on work directly relating to the operational goal. Corresponding costs and levels of IT service are better understood, permitting informed business decisions and better relationships between business and IT partners. Built-in continuous improvement processes ensure that business applications operate efficiently throughout the life cycle, making the decision and action repeatable so that responses are complete and reliable. Consequently, complex, critical missions receive the support they need when they need it.

Success Factors

Management commitment and patience are keys to creating an effective organization. Mistakes and setbacks should be expected, along with initial resistance from stakeholders. Preparing unified processes and imparting a streamlined structure constitute a major change; as such, they require team members to fundamentally change the ways of performing the mission, which is neither easy nor consuming. Leadership must continuously communicate to all stakeholders that improvement is key to success and that the change is inevitable. Leadership must also lead by example in simple tasks such as following new processes and attending training efforts.

As organizations begin to strengthen their core, they realize that processes cannot be improved without a means to measure a desired outcome. Tracking activities and results and turning data into information will free up resources and money, enabling teams to respond to new challenges and maintain acceptable security posture. Measurement of activity allows leaders to view data captures from work efforts and match them with mission objectives. The bidirectionality of the data also helps refine mission objectives by highlighting where the teams' largest impact is being made. Measures provide the data needed to make the right decisions and meet requirements on schedule.

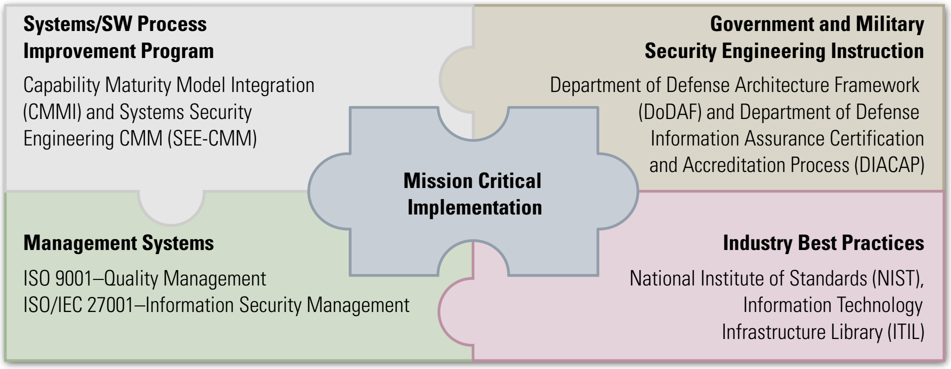
Unified teams operating with clearly defined behaviors and actions experience simplified IT change management. Because they share a common point of reference for internal communications, the right groups understand the information being communicated. Change management produces these benefits because all IT approaches and developments are standardized. Teams can interact and share information efficiently and securely because integration points and handoffs are well documented and understood, ensuring interoperability and effectiveness in the global environment. Through repeatable actions, performance improvement opportunities are identified, enabling teams to leverage previously defined actions from other teams and apply to and improve their operations. Organizations can respond to additional opportunities because their processes, guides and training, and knowledgeable team members become interchangeable as more teams begin to use similar processes and procedures. Having a strong core also allows measurement to be applied to other project areas for easier tracking and faster results. A strong core throughout the organization enables a structured enterprise view, making it easier to see and maintain various service levels in a complex net-centric environment.

Enabling Tools and Techniques

Various models and tools are available to facilitate increased effectiveness of operations in support of the mission. Governments and industry organization have created standards, frameworks, and maturity models to help organize activities for increased effectiveness (see Figure 2). Although many are focused on technology implementation, they can be easily adopted for increasing effectiveness of operations in support of the mission.

Models are typically composed of processes that are sets of practices performed to achieve a goal. Processes include procedures, methods, tasks, tools, equipment, and people. The quality of a system is governed by the quality of the processes used for developing and maintaining that system. Standards exist for nearly every field of work and are typically documents established by consensus and approved by a recognized body. They provide rules or characteristics for activities and their results. Standards are guidelines and considered voluntary; however, they can become mandatory if they are adopted or referenced by laws or regulations.

The common thread among most of these standards, frameworks, and maturity models is that they mention management commitment, measurement, and change control as key components for successful implementation.


Systems/SW Process Improvement Program
Capability Maturity Model Integration (CMMI) and Systems Security Engineering CMM (SEE-CMM)
Government and Military Security Engineering Instruction
Department of Defense Architecture Framework (DoDAF) and Department of Defense Information Assurance Certification and Accreditation Process (DIACAP)
Mission Critical Implementation
Management Systems
ISO 9001–Quality Management
ISO/IEC 27001–Information Security Management
Industry Best Practices
National Institute of Standards (NIST), Information Technology Infrastructure Library (ITIL)

CMMI= Capability Maturity Model Integration ISO= Internationial Organization Standardization

Figure 2 Mission-Critical Implementation

Examples of such models, standards, and frameworks include ISO 90001, *Quality Management System*; ISO/IEC 27001, *Information Security Management System Requirements*, *Capability Maturity Mode Integration* (CMMI); and ISO/IEC 21827, System *Security Engineering Capability Maturity Model* (SSE CMM), and Information Technology Infrastructure Library (ITIL). The US Government also uses its own series of policies, standards, frameworks, and requirements, such as DoD IA Certification and Accreditation Process (DIACAP), the National Institute for Standards and Technology standards and guidance, DoD Architecture Framework (DoDAF), and Federal Enterprise Architecture (FEA).

Blurring the boundaries between government and industry, caused by increasing interconnectedness and interoperability of networked systems, outsourcing of services, and the fact that more than 85 percent of national critical infrastructure are owned by the industry, necessitates government and industry to ensure that the requirements and the involved domain are interoperable and compatible. Increasingly, government procurements are requiring adherence to government and industry standards, models, and frameworks. By proving compliance with these requirements, vendors can provide a level of assurance that their products and services will withstand the pressure of the operational environment and will continue supporting the mission in adverse circumstances.

Meeting these standards is often a qualifier for customers to select providers because most mature teams prefer to work with other mature groups. [6] As more organizations realize that they must identify ways for improving their processes and practices, they recognize that working with less standardized organizations wastes resources. That situation effectively requires them to teach the other organization better methods and subjects themselves to greater risks because the less mature group may cut corners or worse and not have necessary IA controls in place to protect their fighters and information.

Any of these methods can be used as a means for provider organizations to evaluate their own behaviors and identify areas of improvement. For example, Lockheed Martin was able to use a combination of methods, including CMM, ISO standards, and a process library, all while achieving their CMMI rating. The team achieved an overall 72-percent increase in productivity from SW-CMM maturity Level 3 as a result of process improvement. [4]

Compliance or Assurance?

Networked systems and organizations must trust each other so that responses are automatic and timely for effective information sharing and to minimize damage and loss when security is

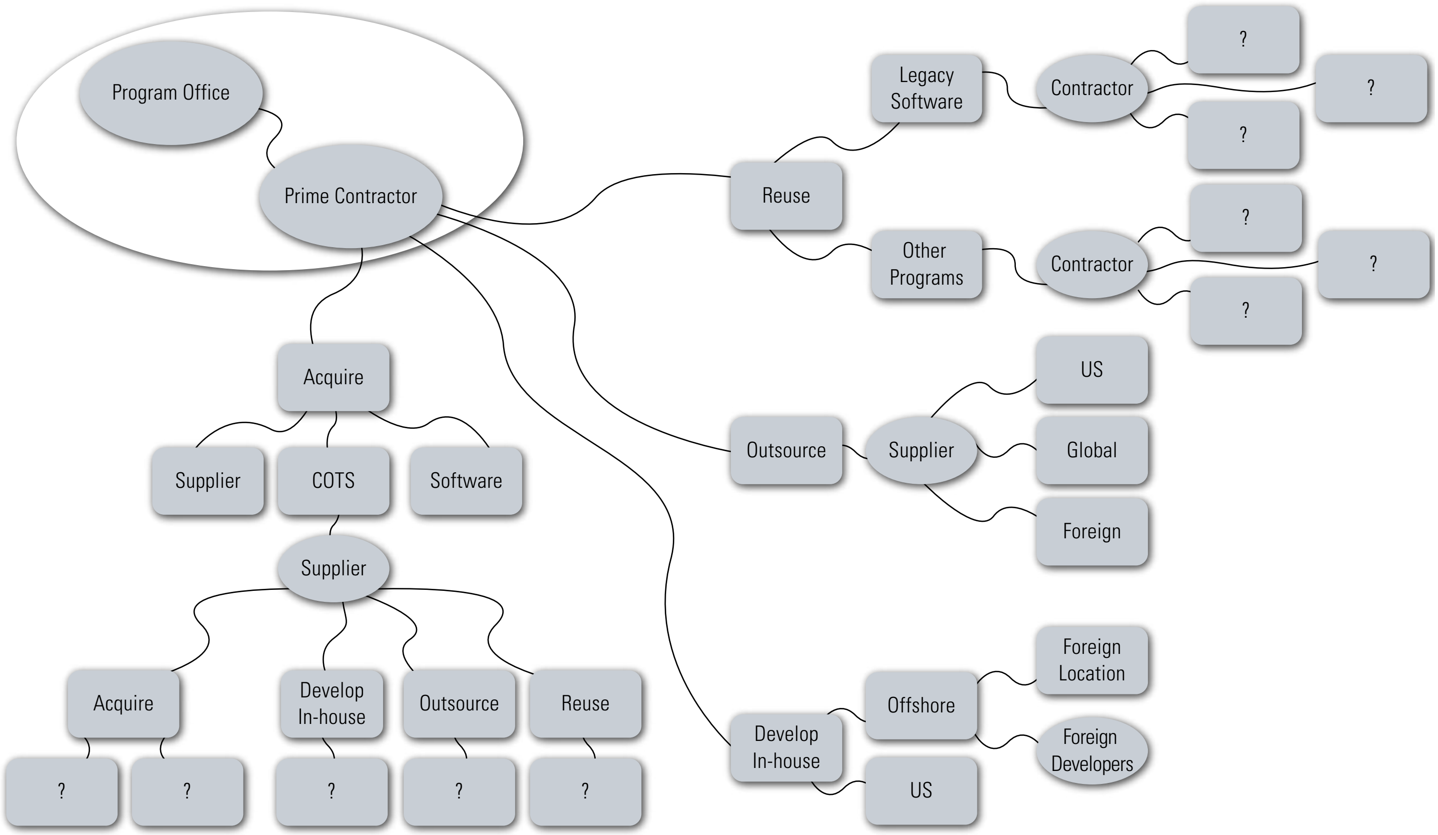


Figure 3 Outsourcing requires sophisticated assurance strategy. [12]

compromised. Establishing this trust among an ever-increasing network of partners and allies poses a great challenge for government agencies and their contractors. In the interconnected and outsourced world, it becomes extremely challenging to provide assurance that the product was developed by trusted developers who used mature processes and procedures. Therefore, having assurance that the system does what it is supposed to do and does not do what it is not supposed to do is virtually impossible.

Industry and government standards, frameworks, and maturity models can help. Buyers can require suppliers to certify how they conduct business and develop their products to provide needed assurance. Although it does not fully protect from malicious acts, it reduces the risk that vulnerabilities were inadvertently introduced due to lax process and procedures. Furthermore, use of standards, frameworks, and maturity models increases probability that vulnerabilities are found *before* **the product is implemented, regardless of whether they were introduced accidentally or on purpose**.

Measuring, assessing, and reporting interoperability, as a part of an overall assurance strategy, provides direction that is critical for setting the right priorities. Using an interoperable and compatible set of requirements is key to ensuring interoperability. Several existing efforts are facilitating interoperability of requirements, including the DoDAF and DIACAP.

DoD has developed the DoDAF to provide an outline for developing a systems architecture or enterprise architecture (EA). All major DoD weapons and IT system procurements are required for developing and documenting their EA architecture using the set of views detailed in the DoDAF. The benefit of DoDAF is that it provides completeness and consistency across systems—a critical component for interoperability and security. [5] The framework separates statements of operation from descriptions of system mechanism, as well as from the statement of applicable technical standards, which makes it easier to compare different solutions. The reduced effort spent on translating systems simplifies the task of integrating systems and increases the detection of incompatible approaches while it is least consuming and expensive to resolve them. DoDAF also shifted the DoD's focus from simply collecting documents to a more efficient process of capturing the knowledge and data items pulled from documents and putting them in accessible repositories. This architecture of what an organization knows reduces redundant effort, eliminates opportunities for inconsistency, and guides the way to more streamlined processes. [6]

The DIACAP is DoD's largest movement for securing Net-Centric operations using repeatable processes to facilitate risk management and apply it to all Information Systems. It provides visibility and control during the implementation of IA capabilities and services, as well as the certification and accreditation (C&A) process for DoD information systems from core enterprise services (CES) to applications. [7] DIACAP is a great resource because it provides a formal standard set of activities, general tasks, and management structure processes. This allows for the C&A of DoD information systems that will maintain the IA approach throughout the system's life cycle. Those seeking more information can locate it at the DIACAP knowledge base hosted online for users who meet the requirements at *https://diacap.iaportal.navy.mil*. The site hosts a DIACAP Instruction guide, DIACAP training and information about recent DIACAP developments, and DIACAP community forums.

Getting the Right Balance

It is understandable that most organizations are seeking a balance between having a reliable, secure, and interoperable infrastructure without spending a fortune on IA, tying up resources, or subjecting their information assets to unacceptable risk. Standardizing business processes helps manage the risks of outsourcing and, if implemented well, can ensure availability of assurance evidence that the requirements have been implemented as stated. Implementing a standard enterprise-level process to replace many similar processes can yield productivity improvements and cost savings.

In a recent article in *IT Business Edge*, DoD was featured because it effectively implemented a standard procurement system (SPS). The SPS is an automated contracting system that standardized procurement processes across DoD, replacing more than 70 separate purchasing and contract management applications used within the department. SPS facilitates ordering and delivery materials, supplies, and services for America's warfighters. DoD created a web-based version of its procurement system that has more than 43,000 users in 800 locations. In a DoD statement, the effort had made operations 70 percent more efficient and saved more than $1 billion simply by reducing accounting errors, system failures, and processing time. [8]

Just like "putting the cart before the horse," the same principle of delivering a product and then testing it makes little sense. Consequently, incorporating standards and best practices should not come after delivery; rather, it should become a part of the initial program or system development. Creating and using improvement processes and procedures saves time and money as opposed to patching systems or working around issues. Northrop Grumman achieved a 13:1 return on investment (ROI), calculated as defects avoided per hour spent in training and defect prevention because they were able to move to CMMI maturity level 5. [9]

Having standards and processes in place avoids many challenges associated with modifying applications or systems at the end of a cycle. Avoidable challenges include systems or applications that can become too slow at transmitting information, or may simply fail to deliver information because of inefficient coding. The potential for security violations increases because the software now suffers from an inability to run specific programs at specified times resulting from a poorly functioning system caused by inefficient testing and integration during later stages of development. Similarly, having appropriate processes in place simplifies creating new agreements with vendors all over the globe and provides consensus that they will be followed. Procedures can be improved or modified over time as needed to accommodate new demands and requirements.

Companies are getting greater value out of incorporating best practices and repeatable processes into their business and operations models than just meeting requirements or standards—they are getting meaningful results, cost savings, and risk reduction. A Raytheon Corporation site was able to reduce its rework by more than 42 percent over a several year period after it became a CMMI maturity Level 3 organization. Results from Northrop Grumman Information Technology, Defense Enterprise Solutions, achieved similar results. Figure 4 shows changes over a 3.5-year period. In the first build, the project underestimated its costs; however, by build 6, the organization was able to complete the work for less than initially estimated. [10]

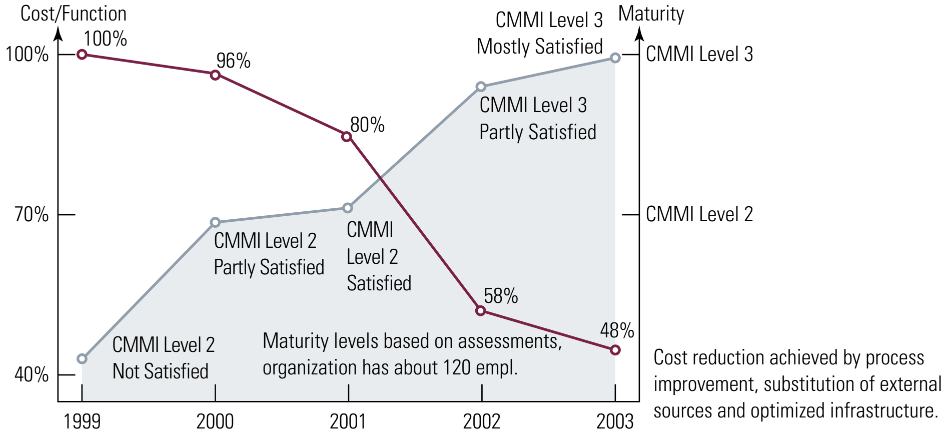


Figure 4 Result of Incorporating Best Practices

Summary

Today's world calls for organizations to deal with complex connectivity, increased immediate security risks, and interoperability requirements. Organizations are responsible for meeting warfighters' unique demands, sharing knowledge reliably and securely, and responding to threats efficiently. Organizations must strengthen their own infrastructure to be effective, reliable GIG members. Incorporating standardized processes and procedures allows government and industry organizations to leverage their resources and respond to challenges and threats with reliable speed and accuracy. Standard processes and architecture allow interoperability and provide improved security and response because teams know what threats exist and how to react to any situation. Measures can be used to evaluate systems and teams and allow managers to make adjustments to facilitate improvement when needed. Repeatable processes and procedures will streamline operations by eliminating redundant actions and rework.

Numerous industry and government standards, frameworks, and maturity models provide guidance on improving processes to achieve cost and productivity improvements and to increase assurance that the IT infrastructure will provide appropriate support to the mission. Implementing them requires long-term management commitment, stakeholder involvement, and dedication from the organizations that embark on improvement efforts. These efforts aim at changing the fabric of the organization—and they therefore constitute a major change. Successful implementation will enable the organizations to handle the increasing complexity of the world around them, respond to new demands, and create better solutions for the challenges of tomorrow. ■

References

1. GlobalSecurity.org, Global Information Grid (GIG). Retrieved from: *http://www.globalsecurity.org/intell/systems/gig.htm*
2. Managing Technology: The weakest link: Keeping your data secure in a collaborative business environment. Knowledge @ W.P Carey, Published: October 25, 2006 Retrieved from: *http://knowledge.wpcarey.asu.edu/index.cfm?fa=viewArticle&id=1320*
3. Secure Systems Engineering- Capability Maturity Model. Retrieved from: *http://www.sse-cmm.org/index.html*
4. CMMI Performance Results, SEI online. Retrieved from: *http://www.sei.cmu.edu/cmmi/results/state_27.html*
5. DOD Architecture Framework v 1.0. February 9, 2004. Retrieved from: *http://www.dod.mil/cio-nii/docs/DoDAF_v1_Volume_I.pdf*
6. Coffee, Peter. Mastering DODAF Will Reap Dividends, eWeek.com. Retrieved from: *http://www.eweek.com/article2/0,1895,1747325,00.asp*
7. Interim Department of Defense Certification and Accreditation Process Guidence. July 6, 2006. Retrieved from: *http://iase.disa.mil/ditscap/interim-ca-guidance.pdf*
8. Flynn, Erin. Military Information Technology Online Archives: Procurement Standard. August 14, 2006, V.10, I. 7. Retrieved from: *http://www.itbusinessedge.com/item/?ci=19593*
9. Secure Systems Engineering- Capability Maturity Model. Retrieved from *http://www.sei.cmu.edu/cmmi/results/state_6.html*
10. Gibson, D., Goldstein, D., Host, K. Performance Results of CMMI- Based Process Improvement, Software Engineering Institute. August 2006. Retrieved from: *http://www.sei.cmu.edu/pub/documents/06.reports/pdf/06tr004.pdf*
11. National Strategy for Homeland Security. Office of Homeland Security. July, 2002. Retrieved from: *http://www.whitehouse.gov/homeland/book/nat_strat_hls.pdf*
12. "Scope of Supplier Expansion and Foreign Involvement" graphic in DACS *www.softwaretech-news.com* Secure Software Engineering, July 2005 article "Software Development Security: A Risk Management Perspective" synopsis of May 2004 GAO-04-678 report "Defense Acquisition: Knowledge of Software Suppliers Needed to Manage Risks."

About the Authors

Nadya Bartol, CISSP, | has more than 13 years of information technology (IT) and information assurance (IA) experience, including IT security and IA performance measurement; security policy development; security architecture design; IT security requirements analysis and traceability; IT security configuration documentation development; risk assessments; certification and accreditation (C&A); project management; process analysis; strategic planning; database management; configuration control; and system analysis, design, development, implementation and maintenance.

In the past 10 years, Ms. Bartol has focused on developing and implementing information security performance management service offering and has established herself as an internationally known authority on information security performance management. She as co-authored National Institute of Standards and Technology (NIST) Special Publication (SP) 800-80, Information Security Performance Measures Guidance; NIST SP 800-65, Integrating Security into the Capital Planning and Investment Control Process; and NIST SP 800-55, Security Metrics Guide for Information Technology Systems. She led and advised multiple information security performance management engagements with government and commercial clients.

Ms. Bartol serves as United States delegate to the International Organization for Standardization (ISO)/International Electrotechnical Commission (IEC) JTC1 SC27 where she is US technical expert working on the ISO/IEC 27000 series standards, Information Security Management System, and a US Head of Delegation (HOD) for Working Group 1. She also chairs the International System Security Engineering Association (ISSEA) metrics working group and is an ISSEA board member. Ms. Bartol is a member of the US International Committee for Information Technology Standards (INCITS) Cyber Security 1 (CS1) technical committee.

Eric White | a member of IATAC, provides program management support to the Joint Task Force–Global Network Operation (JTF-GNO) and to the Defense Components, Law Enforcement and Counterintelligence Center (LECIC), co-located at the JTF-GNO. He has extensive experience supporting operational information assurance (IA) and computer network defense (CND) analysis. Mr. White holds a BA in Criminology from Saint Leo University and an MS in Information Systems and Telecommunications from the Johns Hopkins University.

Stephanie Shankles | supports process improvement efforts across the civilian and government arena. She has previous experience in government public key infrastructure (PKI) operations environments and the customer support arena. She has assisted in implementing CMMI level 3 on the project level. Her background includes a BS in Aviation Management and Flight from Florida Institute of Technology and an MBA in Information Technology from the University of Phoenix.

Michele Moss, CISSP, | is a security engineer with more than 12 years of experience in process improvement. She has assisted numerous organizations with maturing their information technology, information assurance, project management, and support practices through the use of the capability maturity models including the CMMI, and the SSE-CMM. She specializes in integrating security processes and practices into project lifecycles. Ms. Moss is an active member of the Systems Security Engineering Community. She is a Certified Information System Security Professional (CISSP), is an active member of the International Systems Security Engineering Association (ISSEA) and is the Co-Chair of the DHS Software Assurance Working Group on Processes & Practices and Practices.

CyberCIEGE: The Information Assurance Training and Awareness Video Game

by Dr. Cynthia Irvine and Michael Thompson

Information assurance (IA) is multifaceted, and its components include policies, programs, and people. The importance of people as a key component of any IA strategy was emphasized with the promulgation of Department of Defense (DoD) Directive 8570.1, *Information Assurance Training, Certification, and Workforce Management*. This directive outlines objectives and requirements for IA education, training, and awareness (ETA). The responsibility for ensuring that this training is carried out is assigned to DoD components: "All authorized users of DoD information systems (IS) shall receive initial IA awareness orientation as a condition of access and thereafter must complete annual IA refresher awareness."

Effective ETA programs for IA are essential to the security of DoD systems. The many thousands of individuals associated with DoD must make IA a part of their daily activities. The challenge for security educators is to help DoD personnel understand that each person's IA activities *really do matter*. As in so many disciplines, effective IA ETA requires the learner to internalize IA concepts so that security becomes second nature. Thus, IA training and education can benefit from an engaging presentation format that captures the imagination and puts the learner in a stimulating environment in which the participant has a stake in the outcome.

Games and simulations have become increasingly accepted as having tremendous potential as powerful teaching tools, possibly resulting in a revolution in instruction. By using virtual worlds, games provide a concrete experience within which students can internalize domain-specific concepts. Within this virtual world, a student's critical thinking skills are honed. In addition, the game format often appeals to students who have varying attention spans.

> Games and simulations have become increasingly accepted as having tremendous potential as powerful teaching tools, possibly resulting in a revolution in instruction. By using virtual worlds, games provide a concrete experience within which students can internalize domain-specific concepts.

The Center for Information Systems Security Studies and Research at the Naval Postgraduate School (NPS), in cooperation with Rivermind, Inc., has developed a flexible, highly interactive commercial-quality video game for security training and awareness. Called CyberCIEGE, it can support organizational security ETA objectives while immersing players in an engaging security adventure.

In the gaming world, CyberCIEGE is considered a "god-game." Unlike first-person shooter games, god-games place the player in control of a virtual world. Games such as Electronic Arts' The Sims™ and Atari's RollerCoaster Tycoon®, are typical examples of this genre, formally known as resource-simulation tools. Both illustrate the potential of such games to capture the user's attention. By controlling the planning and construction of aspects of the virtual world, players can observe results of their choices and better understand how to manage the virtual world.

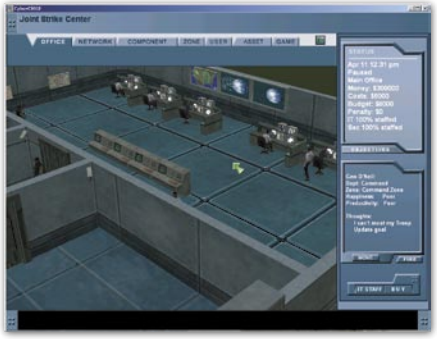
In CyberCIEGE, players are presented with a virtual enterprise, which could be as simple as a small office or as complex as a joint network operations center. Here, players train their virtual users, hire IT staff, provide physical security, and build

and configure networks of computers. Choices that players select have visible effects on virtual users' ability to perform productive work and on attackers' ability to compromise assets. As the assets within the virtual world become more valuable, the attacks become increasingly sophisticated and aggressive.

What makes CyberCIEGE unique relative to other tools that have been developed for IA training and awareness is its dynamicity. Unlike other training vehicles that present the user with a limited and static set of scenarios, CyberCIEGE is a highly extensible game for teaching IA concepts. It may be applied to a wide range of audiences with varying levels of technical sophistication. CyberCIEGE has its own language for creating new training and awareness scenarios. Tools and tutorials also are included to help instructors develop customized scenarios. In addition, an encyclopedia that includes short movies allows students to learn more about various topics.

The tool includes various scenarios, each of which is run separately. Each scenario begins with a briefing that describes an enterprise (*e.g.,* a business that manufactures bowling balls or a shipboard command) and gives the player information about what must be done to make the enterprise successful. Within each scenario, the enterprise has a defined set of users and assets. Users are typically employees of the enterprise whose productive work makes money and advances the objectives of the enterprise.

Assets are various kinds of information that users must access to be productive. Examples of assets are secret formulas, financial information, battle plans, expense statements, and personnel records. Each enterprise has numerous different virtual users, each of whom needs access to different assets in various ways to become productive for the enterprise. These are *user goals*. Occasionally, assets must be shared among users who may also need to simultaneously access multiple different assets. Some of these assets may be classified. Different assets have various secrecy, integrity, and availability values, and different users have different authorizations to access assets as defined by the enterprise security policy.



Apr 11 12:31 pm
Paused
Main Office
Money: $300000
Costs: $6000
Budget: $6000
Penalty: $0
IT 100% staffed
Sec 100% staffed

Cee O'Neil
Dept: Command
Zone: Command Zone
Happiness: Poor
Productivity: Poor

Thoughts:
I can't meet my Treep
Update goal

Figure 1 A Network Operations Center in CyberCIEGE

Each scenario is characterized by predefined users, assets, user goals, and an enterprise security policy. Once established, they are not subject to change by the learner. CyberCIEGE is distinguished by the limitless number of possible scenarios that can be created to teach IA. As shown in Figure 1, which illustrates a Network Operations Center, graphics enhance the ambiance of each scenario.

Elements of CyberCIEGE

CyberCIEGE consists of several elements: a unique simulation engine, a scenario definition language, a scenario development tool, and a video-enhanced encyclopedia. CyberCIEGE is extensible in that new CyberCIEGE scenarios, tailored to specific audiences and topics, are easily created. Scenario-based event triggers are used to introduce new problems for the player to solve and to generate log entries for subsequent student assessment.

The cornerstone of CyberCIEGE is Rivermind's console-based Tybolt *game engine*, which is designed for games and simulations. The engine contains an artificial intelligence system, video-playback library, sound library, memory-management system, resource-management system, and real-time economic engine designed to support resource management simulations.

CyberCIEGE uses a *Scenario Definition Language* through which scenario designers can express security-related risk management tradeoffs, which the simulation engine interprets and presents as a simulation. Players' experiences and the consequences of their choices are functions of the scenario as expressed through the scenario-definition language. The language consists of five major elements that allow security policies to be actualized in realistic networked environments: assets, users, physical zones, conditions and triggers, and objectives and phases.

As Figure 2 illustrates, a form-based *Scenario Development Tool* frees the scenario designer from contending with the scenario-definition language's sophisticated and demanding syntax. It supports reusable libraries of scenario elements (*e.g.,* groups of users or assets), and the development environment includes tools for compiling, validating, and running newly constructed scenarios as simulations.

At any time, players can invoke the *CyberCIEGE Encyclopedia*. Context-sensitive encyclopedia entries explain game play. Other entries describe a broad range of IA topics (*e.g.,* policies, passwords, network security devices, malicious software, and access control mechanisms). To complement the written material in the encyclopedia, CyberCIEGE includes movies covering various topics, including how to use the game. These movies are designed to be understandable and entertaining to all audiences.

CyberCIEGE is a tool for which a large number of scenarios can be developed. This development was motivated by two factors. The first factor is that IA is an enormous field. Many scenarios with various points of focus and depth of detail are needed to begin to cover the large number of IA topics. Some scenarios are lengthy and take hours to run, whereas others are short and focus on specific security concepts (*e.g.,* password management). This feature enables IA educators to tailor scenarios for particular teaching objectives. A log of student play is generated that allows educators to assess student performance.

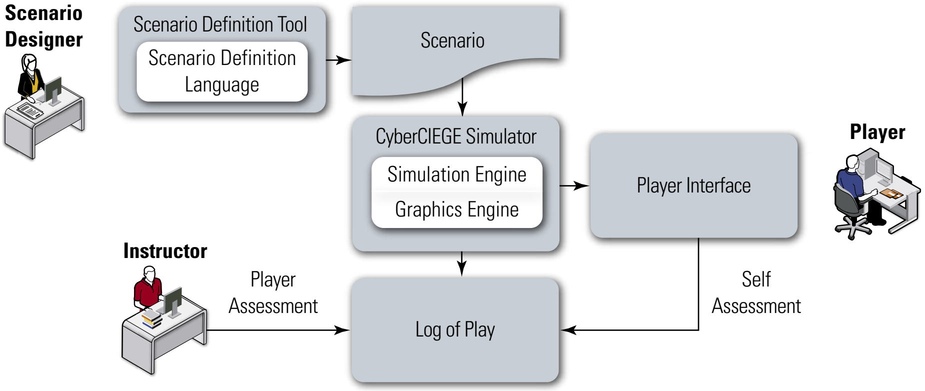


Figure 2 CyberCIEGE components to it's extensibility

The second factor driving the creation of an extensible tool is to allow educators and students to create their own scenarios. Here, the scenario designer must make up an information security policy from whole cloth and imagine the kinds of tensions that might develop from trying to enforce the policy while letting users achieve their goals.

Using CyberCIEGE

At the start of each scenario, the player is presented with a briefing that describes the scenario and the enterprise for which the student must manage computer resources. In some scenarios, the player is responsible for configuring existing computer components: making connections to networks, making choices related to physical security and procedural security, and hiring IT support staff. In other scenarios, the player is also responsible for purchasing specific computer components and connecting them with networks. Players are advised of their limited budget for buying and maintaining equipment and hiring support staff.

The player's objective is to make money for the enterprise by efficiently and securely managing the enterprise computer networks. To succeed in a particular scenario, the player must understand each virtual user's needs to access different assets (*i.e.,* the user goals). The player must then ensure that users have suitable computer components, software, network interconnections, and technical support personnel for achieving their goals of accessing assets.

The player must create and maintain an environment in which the assets are protected in accordance with the enterprise security policy. The enterprise security policy is defined in terms of which virtual users are authorized to access which assets. Failure to adequately protect these assets results in losses to the enterprise attributed to direct loss (*e.g.,* stolen secret formulas), as well as lost user productivity (*e.g.,* time lost reconstructing destroyed assets). The following choices affect the protection of assets in accordance with the security policy:

- Selection of components that enforce selected security policies and deploy the components in suitable topologies
- Configuration of components to aid enforcement of the policies (*e.g.,* automatic logoff after inactivity)

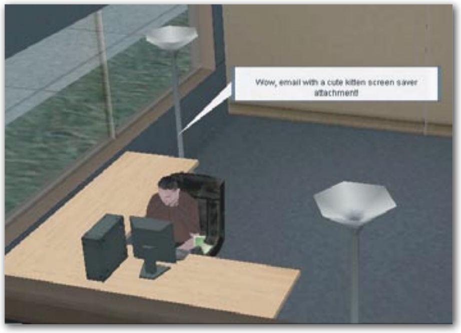

Wow, email with a cute kitten screen saver attachment!

Figure 3 Active triggers can result in pop-up images such as this virtual user's comments.

- Interconnection of components using networks (or chose to not interconnect certain components)
- Instruction of users to follow certain procedures (*e.g.,* discourage them from picking dumb passwords) and provide users with adequate training
- Application of physical security by limiting which users can enter a physical zone (*e.g.,* secure office area) and enforcing these limitations (*e.g.,* armed guards, surveillance cameras)
- Performance of selected degrees of background checks (*e.g.,* criminal records, work history) on various users.

These security choices affect the protections provided to the enterprise assets, which are subject to attack from vandals, disgruntled employees, professional attackers, incompetent users, and acts of nature. The most challenging attacks to protect against are those from sophisticated, nation-state-level adversaries who target specific assets. The means employed by these professionals to compromise assets depend on the attacker's motive (*i.e.,* the value of the asset to the attacker).

Players can start and pause the simulation at any time. Typically, players are encouraged to construct networks and make policy enforcement decisions before starting the simulation. This is analogous to configuring and assessing a deployed system before taking it operational. After the player starts the simulation, virtual users may start creating and accessing their assets; without due care, this action may occur in ways that make the assets vulnerable to attack.

During the simulation, players can select and observe the status of a user's productivity and happiness. For example, users who cannot achieve their goals become agitated and pound on the keyboard. Scenario designers can trigger feedback to the player via message tickers at the bottom of the screen, pop-up messages and, as Figure 3 illustrates, users speaking through cartoon bubbles to inform students of their progress.

Basic Information Assurance Awareness Scenario	
Topic	**Player Interface**
Introductory Information Assurance briefing	Definitions and descriptions of important IA elements and how they interact are introduced in this scenario briefing.
Information value	The user is tasked with the protection of high value information and must answer questions about information dissemination.
Access control mechanisms	In this stage of the game the player is introduced to the concepts both mandatory and discretionary access control. The role of discretionary controls as a supplement to controls on classified information is illustrated.
Social engineering	The player must take preemptive action to prevent a social engineering attack.
Password management	The player must prevent a game character from revealing his password to an outside contractor.
Malicious software and the basics of safe computing	The player must determine what to do and expend resources to procure three procedural settings that will prevent malicious software propagation.
Data protection	This scenario presents a situation where it appears that a game character is leaving the premises with sensitive information. Actions that must be taken by the player allow the importance of secure storage of backups to be understood.
Physical security	The player must select cost-effective physical security measures to prevent unauthorized entry into sensitive areas.

Table 1 Basic Information Assurance scenario

Example Scenarios

Two CyberCIEGE scenarios, designed by LT Benjamin Cone, fulfill Navy IA training requirements. The first scenario makes the player aware of basic IA problems and principles, whereas the second scenario trains more sophisticated users of computer-based assets.

The basic user scenario focuses on computer security fundamentals. The player is placed in the role of a security decision maker, aboard a ship, who must complete objectives that raise the security posture of the organization. If objectives are not completed within a specified time, the game engine triggers appropriate attacks, and the player is penalized. After completing each objective, the player is presented with an awareness message that relates the action taken in the game with real-life circumstances and provides feedback regarding the player's choices. The player wins by completing all objectives without incurring ruinous penalties.

For each topic identified in the requirements analysis, a scenario element was created that requires the player to do something to convey the concept to be learned. Table 1 lists some topics and activities. Among the features that made this scenario of particular use in a DoD context are the protection of classified information and cultural aspects of organizational security associated with the DoD's hierarchical command structure.

CyberCIEGE Availability

One of the most attractive aspects of CyberCIEGE is that it is available at no cost to the DoD and federal agencies. Rivermind has allowed accredited educational institutions to have no-cost access to CyberCIEGE. The latest release of the game can be obtained from NPS at the CyberCIEGE website: *http://cisr.nps.edu/cyberciege.html*. This website contains information about the game and provides contact information. The CyberCIEGE email address is cyberciege@nps.edu.

With a model based on the open source community, the CyberCIEGE website will provide a resource in which those involved in IA education and training can share with others scenarios and other CyberCIEGE developments.

The high degree of flexibility built into CyberCIEGE permits scenarios to be created that illustrate virtually any security topic in a range of environments, generic and organization specific. The future of CyberCIEGE as a training and awareness tool is limited only by one's imagination. Future work might include scenarios on topics such as configuration and patch management, security in wireless networks, and using public key cryptography to support security objectives. In addition, CyberCIEGE could be extended into a multiplayer game.

In a multiplayer version, players are assumed to be concerned about coalition partners with whom they might conduct cyber-based operations. To determine the qualifications of other systems for interconnection and ultimately the protection of information assets, a player would conduct various tests on these foreign systems. The game would consist of a scenario-specific number of rounds of preparation and testing by all coalition partners. As with existing single-player scenarios, tests could be focused on a particular IA issue, such as passwords or firewall configuration, or could cover a broad range of topics. ■

About the Authors

Cynthia Irvine | is a professor in the Department of Computer Science at the Naval Postgraduate School in Monterey, California. She holds a BA in physics from Rice University and a PhD in astronomy from Case Western Reserve University, Cleveland, Ohio. She is the Director of the Center for Information Systems Security Studies and Research, a post she has held from 1996. She served as the Director of the Cebrowski Institute at the Naval Postgraduate School from 2001 to 2003. Dr. Irvine is currently the Vice Chair of the IEEE Technical Committee on Security and Privacy. She has directed the graduate research of more than 100 students and is the author of more than 150 papers, reports, and articles. Her fields of interest are inherently trustworthy systems, security architectures, and security education.

Michael Thompson | is a research associate in the Center for Information Systems Security Studies and Research at the Naval Postgraduate School in Monterey, California. He holds a BS in electrical engineering from Marquette University. Mr. Thompson has been responsible for the design and development of major components of CyberCIEGE. He has guided students in the design of scenarios and tools for CyberCIEGE. His research interests are security engineering and highly secure systems.

CONFERENCES

DISA Partnership Conference

This year's Defense Information Systems Agency (DISA) Partnership Conference was held in Nashville, TN, from April 30–May 3, 2007. Once again, the conference was a tremendous success, thanks to attendees' participation and support. This yearly conference offers DISA an opportunity to focus its attention on customers. As always, attendees discussed critical issues and requirements. This year, Computing Services, Defense Message System, Defense Spectrum Organization, Information Assurance–Peace Enforcement Operations (PEO)-IAN, Joint Interoperability Test Command, and others presented briefings.

This conference is key to DISA because it involves networking and enables relationships to be renewed and built on, which are critical to the warfighter.

Any general conference inquiries or feedback should be directed to customerconference@disa.mil. ■

DISA

Dr. Xinyuan (Frank) Wang

by Ron Ritchey

This article continues our series of profiling members of the IATAC Subject Matter Expert (SME) program. The SME profiled in this article is Dr. Xinyuan (Frank) Wang. Since 2004, Dr. Wang has been Assistant Professor at the Department of Information and Software Engineering at the George Mason University Volgenau School of Information Technology and Engineering (IT&E). His primary research focuses on network-based intrusion source tracing. He also researches intrusion detection and response, viruses and worms, information hiding, and privacy and anonymity. [1]

Based on his previous work in network-based intrusion source tracing, Dr. Wang has developed a new watermarking technique that can be applied to any packet flow on the Internet.

Dr. Wang received his PhD from North Carolina State University in 2004. Earlier, he worked in industry at several companies, including Nortel and Cisco. He believes that his experience in industry has led him to pursue research projects that have direct practical applications.

Based on his previous work in network-based intrusion source tracing, Dr. Wang has developed a new watermarking technique that can be applied to any packet flow on the Internet. By basing the watermarking on timing, his method has proven to work over stepping stones, encryption, and even anonymizing services.

Traditionally, researchers have believed that Voice over Internet Protocol (VoIP) traffic could not be tracked because it could be encrypted, anonymized, or mixed with other data flows. Dr. Wang applied his watermarking techniques to a popular proprietary peer-to-peer encrypted VoIP, demonstrating that it is possible to embed a unique watermark into VoIP traffic, even if the content is unreadable. [2]

Another of Dr. Wang's projects involves watermarking Internet traffic. He has used watermarking techniques to break anonymous communication systems available on the Internet. One example that he used was Anonymizer [3], which provides one of the most popular anonymizing services on the Internet. These services offer secure web proxies and other mechanisms to protect the identities of web users from profiling. Anonymizing services offer significant challenges over VoIP systems: the traffic flow is mixed, split, and merged, and packets can be dropped. Using a different watermarking technique, he successfully broke the strongest service that Anonymizer offered in 10 minutes. Dr. Wang presented his paper, "Network Flow Watermarking Attack on Low-Latency Anonymous Communication Systems," at the 2007 IEEE Symposium on Security and Privacy (S&P 2007).

If you have a technical question for Dr. Wang or other IATAC SMEs, please contact iatac@dtic.mil. The IATAC staff will assist you in reaching the SME best suited to helping you solve the challenge at hand. If you have any questions about the SME program or are interested in joining the SME database and providing technical support to others in your domains of expertise, please contact iatac@dtic.mil, and the URL for the SME application will be sent to you. ■

References

1. A selected list of Dr. Wang's publications can be found at *http://ise.gmu.edu/~xwangc*
2. Dr. Wang's paper on VoIP tracing is available at *http://ise.gmu.edu/~xwangc/Publications/CCS05-VoIPTracking.pdf, http://ise.gmu.edu/~xwangc/Publications/IEEENet06-Anonymity.pdf*
3. Information about Anonymizer is available at *http://www.anonymizer.com*

IATAC SPOTLIGHT ON EDUCATION

Center for Secure Information Systems (CSIS)

George Mason University

by Ron Ritchey

The Center for Secure Information Systems (CSIS) within the George Mason University Volgenau School of Information Technology and Engineering (IT&E) was established in 1990 as one of the first academic centers focused on security at a United States university. CSIS provides a dedicated environment to encourage development of expertise in the theoretical and applied aspects of information systems security. CSIS offers a BS in Information Technology with a concentration in Information Security and Network Administration. The center also offers several MS and PhD programs in information security. [1]

CSIS, directed by Dr. Sushil Jajodia, is one of the nation's leading information security research centers. Since 1992, it has been recognized as one of the original seven National Security Agency (NSA) Centers of Academic Excellence in Information Assurance Education. In 2001, CSIS was selected as a participant in the Department of Defense (DoD) IA Scholarship Program.

With nine full-time research scientists and more than 80 faculty within IT&E across multiple departments, CSIS researches a wide variety of topics [2]:

- Vulnerability assessment and analysis
- Automated penetration testing
- Intrusion detection and prevention
- Auditing, audit log analysis, and data mining
- Flexible authorization management system
- Role-based access control
- Trust management
- Secure key management
- Digital rights management
- Secure information sharing
- Critical infrastructure protection
- High-assurance security architectures
- Steganography and digital watermarking
- Protection from malicious code
- Ad hoc, wireless networks
- Sensor networks.

> CSIS provides a dedicated environment to encourage development of expertise in the theoretical and applied aspects of information systems security.

CSIS research projects are sponsored by several government organizations, including:

- NSA
- Homeland Security Advanced Research Projects Agency
- Air Force Research Laboratory
- Defense Advanced Research Projects Agency
- National Institute of Standards and Technology
- Federal Aviation Administration
- Army Research Office
- National Science Foundation
- Disruptive Technology Office
- Air Force Office of Scientific Research.

One project on which CSIS is now involved is Topological Vulnerability Analysis (TVA). CSIS has been researching this technology for the past 5 years. The technology models all possible attacks through a network based on the network's configuration. TVA simulates incremental network penetration methods that hackers commonly use to build complete maps of the potential attack paths a hacker could use to compromise a network. Using these models, it is easy to calculate the effect of combined vulnerabilities on overall security. CSIS is working with industry to commercialize the TVA technology and has applied for four patents related to this work.

In March 2007, CSIS was awarded $4.8 million from the DoD Multidisciplinary University Research

Initiative (MURI) competition for Autonomic Recovery of Enterprise-Wide Systems After Attack or Failure with Forward Correction. Working with Columbia University and Penn State, CSIS' Anup Ghosh is improving methods for incident response. Based on the health care model for computing systems, in which failing systems are restored to health while others provide their services, this project will allow enterprise networks to easily recover from attack. [3]

In April 2007, two professors and a graduate student from CSIS patented a method for fingerprinting and recognizing images. Dr. Jajodia, Dr. Zoran Duric, and Neil Johnson's method improve on current image identification techniques (*e.g.*, digital watermarking) by generating fingerprints that can survive image distortion and some tools designed to remove watermarking, reducing the effort necessary to detect illegal copies of copyrighted information. [4] ■

References

1. More information about the Center for Secure Information Systems can be found at *http://csis.gmu.edu*
2. More information about CSIS research projects can be found at *http://csis.gmu.edu/publication.html*
3. More information about MURI is available at *http://condor.gmu.edu/newsroom/display.phtml?rid=594*
4. More information about CSIS image recognition patent is available at *http://condor.gmu.edu/newsroom/display.phtml?rid=600*

Letter to the Editor

Q ***I understand that IATAC is putting out another State-of-the-Art Report, this one on software security assurance. Could you tell me more about it?***

A The term "software assurance" (SwA or SA) has slightly varying definitions for numerous organizations. For example—

- For the Department of Homeland Security (DHS), SwA encompasses trustworthiness, predictable execution, and conformance.
- For the National Institute of Standards and Technology (NIST), SA is "the planned and systematic set of activities that ensures that software processes and products conform to requirements, standards, and procedures to help achieve trustworthiness and predictable execution."
- For the National Aeronautics and Space Administration (NASA), SwA is the planned and systematic set of activities that ensures that software processes and products conform to requirements, standards, and procedures.
- For Department of Defense (DoD), SwA relates to "the level of confidence that software functions as intended and is free of vulnerabilities, either intentionally or unintentionally designed or inserted as part of the software."

The objective of SwA via secure software engineering is to report on the current state-of-the-art in understanding SwA and the methodologies, best practices, technologies, and tools that are in use or emerging to help software developers specify, design, implement, configure, update, and sustain software. The IATAC Software Security Assurance SOAR should be available in mid-July. For more information about SwA, please contact us at iatac@dtic.mil. ■

Security Metrics

by Jack Phillips

Security is obvious and important to government and commercial entities alike, but impossible to measure.

That statement was the sentiment at the first two Information Security Forums of the year that we held in Washington, DC, and Dallas, TX. Measurement, metrics, and accountability are currently popular among commercial organizations because the security profession is beginning to mature and dollars spent on security are being more heavily scrutinized.

Based on feedback from counterintelligence support officers and computer security officers (CISO/CSO) who gathered in those two cities to compare notes, the time has now come when computer security is finally being compared against other investments being made within information technology (IT) and beyond, and an increasing amount of accountability is being sought from senior management from IT security leaders. This environment has led to considerable focus on reliable metrics and measurements that can be used to rationalize security spending.

Two questions arise among IT security leaders on the topic:

- ▶ Which are the most appropriate metrics to keep track over time?
- ▶ What reliable sources may I turn to to get a sense of where I stand relative to my peer group?

I will review what we have been hearing from the field on both important questions. The answer to the first question is relatively straightforward; the second, more complicated.

Most CISOs from the commercial sector view two kinds of metrics as important: tactical and strategic. Tactical metrics are used for measuring the weekly, monthly, and quarterly efficiency of a security program at a tactical level. These kinds of metrics include spam and viruses blocked, number of incidents requiring a given level of response, help desk calls, or patching cycle time. Teams measure their own internal efficiencies, and behaviors use these metrics. [1] Commonly, teams are measuring the absence rather than the presence of events or activity. This action often leads to the perverse conclusion by management—if nothing is happening, why do we need to be spending so much on security?

Increasingly, security leaders are developing strategic metrics based on their unique knowledge of their business. Strategic metrics are being tied into large enterprise initiatives or priorities such as increased sales, lower costs, or operational efficiencies. Examples of strategic metrics that have been shared during discussions at our forums include the number of customer wins made possible as a result of stronger internal and product security, costs savings experienced as a result of tighter and more efficient security standards, and overall lower liability insurance costs attributed to a stronger security posture.

The inherent value to the organization of progress against strategic metrics is obvious. The difficulty comes when IT security teams attempt to answer senior management's question, How are we doing against our peers? Time series data measuring internal metrics are useful to show progress over time. However, senior management expects to know how its organization is performing relative to competitors in their industry and across industries.

Currently, no definitive database of tactical or security metrics exists that would allow these kinds of comparisons. A few industry Information Sharing Analysis Centers (ISAC)2 have begun tracking security metrics, but no entity has tried to pull together metrics across industries.

If the security profession is to meet credibility and accountability standards applied to other functional areas (*e.g.*, finance, sales, operations, IT), security teams must be willing to share their historical data for inclusion in a database that compares results across industries. For now, IT security teams within the commercial sector are using metrics increasingly to indicate progress and tie their activities into the larger business mission. ■

References

1. By far the most authoritative review of "tactical" security metrics can be found in Security Metrics: *Replacing Fear, Uncertainty and Doubt*, recently published by Andrew Jacquith.
2. Information Sharing and Analysis Center

FREE Products

Order Form

Instructions: All IATAC LIMITED DISTRIBUTION reports are distributed through DTIC. If you are not a registered DTIC user, you must do so prior to ordering any IATAC products (unless you are DoD or Government personnel). To register online: *http://www.dtic.mil/dtic/registration*. The *IAnewsletter* is UNLIMITED DISTRIBUTION and may be requested directly from IATAC.

Name ____________________

Organization ____________________

Address ____________________

____

____

DTIC User Code ____________________

Ofc. Symbol ____________________

Phone ____________________

Email ____________________

Fax ____________________

Please check one:

□ USA	□ USMC	□ USN	□ USAF	□ DoD
□ Industry	□ Academia	□ Government	□ Other	

Please list the Government Program(s)/Project(s) that the product(s) will be used to support: ____________________

____

LIMITED DISTRIBUTION

IA Tools Reports (softcopy only)

□ Firewalls

□ Intrusion Detection

□ Vulnerability Analysis

Critical Review and Technology Assessment (CR/TA) Reports

□ Biometrics (soft copy only)

□ Data Mining (soft copy only)

□ Wireless Wide Area Network (WWAN) Security

□ Computer Forensics* (soft copy only. DTIC user code MUST be supplied before these reports will be shipped)

□ Configuration Management

□ IA Metrics (soft copy only)

□ Defense in Depth (soft copy only)

□ Network Centric Warfare (soft copy only)

□ Exploring Biotechnology (soft copy only)

State-of-the-Art Reports (SOARs)

□ Data Embedding for IA (soft copy only)

□ Modeling & Simulation for IA (soft copy only)

□ Software Security Assurance

□ IO/IA Visualization Technologies (soft copy only)

□ Malicious Code (soft copy only)

□ A Comprehensive Review of Common Needs and Capability Gaps

UNLIMITED DISTRIBUTION

IAnewsletters Hardcopies are available to order. The list below represents current stock.
Softcopy back issues are available for download at *http://iac.dtic.mil/iatac/IA_newsletter.html*

Volumes 4		□ No. 2	□ No. 3	□ No. 4
Volumes 5	□ No. 1	□ No. 2	□ No. 3	□ No. 4
Volumes 6	□ No. 1	□ No. 2	□ No. 3	□ No. 4
Volumes 7	□ No. 1	□ No. 2	□ No. 3	□ No. 4
Volumes 8	□ No. 1	□ No. 2	□ No. 3	□ No. 4
Volumes 9	□ No. 1	□ No. 2	□ No. 3	□ No. 4
Volumes 10	□ No. 1	□ No. 2		

**Fax completed form
to IATAC at 703/984-0773**

Calendar

August

Federal Information Security Conference
1–2 August
Colorado Springs, CO
http://www.fbcinc.com/fisc

SANS Redondo Beach 2007
18–24 August
Redondo Beach, CA
http://www.sans.org/redondo07/?portal=c202b64d350665e7592b29c43039968a

27th Annual International Cryptology Conference (CRYPTO 2007)
19–23 August
Santa Barbara, CA
http://www.iacr.org/conferences/crypto2007/index.html

LandWarNet 2007
August 21–23
Ft. Lauderdale, FL
http://events.jspargo.com/landwarnet07/Public/Content.aspx?ID=910&sortMenu=105000&exp=3%2f9%2f2007+9%3a24%3a30+AM <http://events.jspargo.com/landwarnet07/Public/Content.aspx?ID=910&amp;sortMenu=105000&amp;exp=3%2f9%2f2007+9%3a24%3a30+AM>

Department of Homeland Security (DHS) Security Conference
28–29 August
Baltimore, MD
http://www.fbcinc.com/event.aspx?eventid=Q6UJ9A00E06F

September

Network Security Conference
10–12 September
Redondo Beach, CA
http://www.sans.org/redondo07/?portal=c202b64d350665e7592b29c43039968a

New England Information Security Forum
September 17–18
Boston, MA
http://www.ianetsec.com/forums/splash.html?forum_id=34

C4ISR Symposium
17–20 September
Atlantic City, NJ
http://www.acceleration07symposium.com

October

25th Annual Communications, Computers Communications, Computers & Intelligence Systems Technology (C4IST)
2–4 October
Ft. Huachuca, AZ
http://www.afceac4ist.com

7th Annual Federal Information Assurance Conference (FIAC)
23–24 October
College Park, MD
http://www.fbcinc.com/fiac/

Infotech 2007
23–25 October
Dayton, OH
http://www.afcea-infotech.org/pages/overview.html

4th Annual Military Information Assurance Summit
24–26 October
Washington, DC
http://www.idga.org/cgi-bin/templates/genevent.html?topic=329&event=13523& <http://www.idga.org/cgi-bin/templates/genevent.html?topic=329&amp;event=13523&amp;>

MILCOM 2007
29–31 October
Orlando, FL
http://www.milcom.org/index.asp

IATAC
Information Assurance Technology Analysis Center
13200 Woodland Park Road, Suite 6031
Herndon, VA 20171

To change, add, or delete your mailing or email address (soft copy receipt), please contact us at the address above or call us at: 703/984-0775, fax us at: 703/984-0773 , or send us a message at: iatac@dtic.mil