Advanced Software Technologies for Protecting America

Gregory S. Shelton, Randy Case, Louis P. DiPalma, and Dan Nash
Raytheon Company

Systems & Software Technology Conference
Thursday, 22 April 2004
Speaker Luncheon: 12:20 - 1:20
Marriott Grand Ballroom

Advanced software technologies are required for the success of homeland security, missile defense, intelligence/surveillance/reconnaissance, and precision engagement. State-of-the-art software technologies for system architecture development such as model-driven computing, reference architectures, and supporting technology enablers are needed for these critical systems.

The events of 9-11 in America and the ongoing actions throughout the world have keenly focused our thoughts on issues of protection and homeland security. Portions of the solutions to these problems will be in better human intelligence, greater diligence, and resources applied to traditional security. However, technology-driven solutions are needed to better increase the use of in-place resources and meet newer threats.

Four system areas are vital to protecting America: homeland security, missile defense, intelligence/surveillance/reconnaissance (ISR), and precision engagement. These all require advanced software technologies that will enable the development of integrated mission systems. These technologies go beyond existing software technologies traditionally focused on *stovepipe* software component or platform solutions. Technologies supporting system architecture development are important for mission success.

Homeland Security

The threat of terrorist attacks in the United States brings into vivid focus the need to harness technology to detect threats and protect against and respond to them. Table 1 presents a list of some recent initiatives directly related to homeland security; the applicable enabling advanced software technologies are also listed. In some *stand-alone* activities such as bomb detection or airline missile protection, no new software technologies are needed. More work in domain-specific algorithms may be required, but fundamental software techniques are adequate for these programs to succeed.

Common to many homeland security programs is the need for searching, mining, and analyzing large databases (for example, visa tracking, biometric pattern matching, and analysis of foreign language materials). The fundamentals of these types of database technologies exist and upgrades in technologies are ongoing, particularly in enhancements to speed and accuracy.

New needs to integrate communication systems from agencies that formerly did not use common equipment (police, fire, etc.) and the need to fuse information such as weather data and models of chemical/biological agents requires the integration of existing system architectures. Tools and techniques to develop these software-intensive system architectures such as using ontology for information definition/retrieval and using reference architectures are needed for the successful development of these systems.

Table 1: *Software Technologies Needed for Homeland Security*

Detection Systems	Advanced Software Technologies Needed
Airport Bomb Detection	• Software Technologies Are Adequate
Open Source Analysis	• Search Engines • Automatic Language Translation • Data Mining
Entry/Exit Visa Tracking	• Data Mining • Predictive Analysis
Protection Systems	
Biometrics	• Intelligent Database Searching
Commercial Airline Missile Protection	• Software Technologies Are Adequate
Response Systems	
Integrated Communications Systems (Fire, Police, National Guard, etc.)	• Information Organization/Retrieval Using Ontology • Context-Sensitive Reference Architectures
Chemical/Biological Agent Response	• Data Fusion for Virtual Weather Modeling

Missile Defense

Recent developments in world events and national policy have renewed the dialogue on missile defense. The mission of missile defense is to defend successfully against missiles of all ranges (short, intermediate, and long) in all phases of flight (boost, midcourse, and terminal). All components must be fully networked to assure coordinated operations with very short timelines. An operational missile defense system must have three fundamental technical capabilities and associated software technologies: sensors, interceptors, and battle management, command, and control (BMC2), as shown in Table 2.

The sensor components (radar, infrared, and electro-optical) have been developed and will continue to be matured. We are seeing model-based software techniques used to support the definition of architectures and generation of executable code for some of these applications. The interceptor components of these systems require software data fusion approaches and system architectures to better enable the data fusion. The most software-intensive portion of missile defense is the BMC2 component. The need for handling large volumes of information accurately and within very short timelines places demands on the development of effective system architectures. This area requires a host of advanced software techniques to develop effective system architectures, as used in software techniques to aid human decision making (intelligent agents, cognitive computing techniques, etc.).

ISR

The ISR programs cover the full spectrum of information management, providing the ability to task, collect, process, exploit, and disseminate national and tactical target data (see Table 3). These abilities are crucial for warfighters to achieve information dominance throughout the entire battlespace. The ISR activities are typically composed of tasking, collection, and activities related to processing/exploitation/dissemination.

A key attribute of ISR is the system integration of multiple sensors, platforms, and networks. This *system of systems* is characterized by the need for well-defined system architectures to support the needed interoperability and integration. New software technologies common to all tasks in ISR include ontology for information management, reference architectures, and model-driven computing architectures. Advances in data mining and intelligent agents will expedite handling of large information volumes in real time. Interoperability and information dissemination to various users will require new techniques to handle multi-level security issues.

Missile Defense Component	Advanced Software Technologies Needed
Sensors – Detect, acquire, and track target missiles; predict their path; identify a threat among decoys; and direct the interceptor to destroy the missile.	• Context-Sensitive Software Reference Architectures • Model-Driven Software Architectures
Interceptors – Seek, discriminate, and destroy targets.	• Data Fusion
BMC2 – Provides the commander with threat and tracking data from sensors, suggests the most effective response, directs interceptors to the target, and measures damage and effectiveness.	• Context-Sensitive Software Reference Architectures • Intelligent Software Agents • Human Factors Interactions With Complex Software Systems • Model-Driven Software Architectures • Cognitive Computing Techniques

Table 2: *Software Technologies for Missile Defense*

ISR Activity	Advanced Software Technologies Needed
Example ISR Tasking Systems • UAV Tactical Control System • Global Hawk Mission Control Element • Intelligence Satellites Control Element • Space-Based Infrared Systems (SBIRS) Control Element	• Information Organization/Retrieval Using Ontology • Context-Sensitive Reference Architectures • Intelligent Software Agents • Model-Driven Computing • Human Factors Interactions With Complex Systems
Example ISR Collection Systems • Global Hawk Integrated Sensor Suite • U-2 Advanced Synthetic Aperture Radar • Multi-Platform Radar Technology Insertion Program (MP-RTIP) • Rivet Joint Aircraft Sensors	• Information Organization/Retrieval Using Ontology • Context-Sensitive Reference Architectures • Intelligent Software Agents • Model-Driven Computing • Human Factors Interactions With Complex Systems • Data Mining • Multi-Level Security
Example ISR Process/Exploit/ Disseminate Systems • Cooperative Engagement Capability (CEC) • Global Broadcast Service (GBS) • National Polar-Orbiting Operational • Environmental Satellite System (NPOESS)	• Information Organization/Retrieval Using Ontology • Context-Sensitive Reference Architectures • Intelligent Software Agents • Model-Driven Computing • Human Factors Interactions With Complex Systems • Data Mining • Cognitive Computing Techniques • Multi-Level Security

Table 3: *ISR Systems Software Technologies*

Precision Engagement

Precision engagement systems enhance America's defense by providing warfighters with highly accurate, adverse weather, rapid sensor-to-shooter capabilities required on today's battlefields (see Table 4, next page). Precision engagement works in conjunction with ISR to provide a wide range of capabilities.

The information from ISR that is needed to provide targeting for precision munitions requires using software techniques that support the development of system architectures (ontology, reference architectures, and model-driven architecture development). In particular, shorter sensor-to-shooter timelines require a system architecture construction optimized for time sensitivity.

Software Technologies for System Architecture Development – A Common Theme

Systems being deployed and developed for protecting America require advanced software technologies. In some cases, where the particular system architecture is stand-alone or composed of mostly point-to-point connections and limited broadcasting, the software approaches of today are sufficient. There will still be needed development of more capable algorithms and processors to support those algorithms, but the underlying software tools, paradigms, and enablers do not require further extensive research and development to be successful.

In many of the other above cases, we find as a common theme the need for existing software capabilities to be extended so that large-scale systems/platforms can work together to achieve the required missions. We believe that success in the new *system-of-systems* environment is enhanced by using software that will be more *intelligent* and developed as a direct offspring of modeling and simulation activities within the context of executable enterprise reference architectures. These technologies are being developed today at Raytheon, other defense contractors, and university/research organizations.

The left column of Table 5 (see next page) shows mature deployed software technologies used in defense applications today. The right column summarizes the software advances needed for the system types previously described. While these technologies are in various states of maturity (including some such as data mining, which are fairly robust), they have not been widely deployed in key systems. Technologies for the development of system architectures are common to many of the systems needed for protecting America.

Looking at the key areas for defending and protecting America, we find that support for development of large, integrated mission systems is needed. The need for well-defined context-sensitive architectures is paramount for achieving these systems of systems such as Common Operating Picture (COP), DDX Destroyer, Future Combat System, or Joint Strike Fighter. The semantics of these large amounts of information are captured using ontological tools. The reference architectures are defined within the contexts of architecture frameworks. Finally, the architectures themselves are actually executable models supported by model-based, architecture-driven software development. Other enabling technologies such as cognitive computing and intelligent agents are all focused toward the software system development. Figure 1 illustrates the relationships of several key software technologies that will help realize the system architectures needed in the future.

Information Organization/ Retrieval Using Ontology

The initial step in developing large-scale system architectures is managing large-scale information semantics. Military knowledge workers are immersed in data smog. We have far more capability to create information than to find and retrieve relevant information. The result is huge amounts of amorphous, unstructured data that overwhelm us when we need pertinent, actionable data for informed decisions.

Technologies to help manage, search, and retrieve data include metadata for data descriptions, taxonomies for data categories, and ontology for data relationships (see Figure 2). Applications have been driven by commercial needs to identify information on the semantic Web and to provide Web services that deliver the right information to consumers. The value of such technologies to military applications is recognized by the Defense Advanced Research Projects Agency (DARPA), who sponsored development and deployment of a machine-processable ontology description language called the DARPA Agent Markup Language (DAML)[1].

Military information users must make life-critical decisions based on large amounts of time-sensitive, rapidly changing inputs from multiple sensors and sources. Having a single, consistently applied meaning for concepts, categories, and relationships reduces confusion, misinterpretation, and mistakes. Cognitive overload is reduced by supplying users with information that is relevant to their location, situation, and responsibilities. Ontology can be used to support both improvements. An example of where this applies is the Common Operating Picture (COP), which is a distributed database. Currently it is packed with disparate and incompatible data. In the future, human operators and software agents marking up information from sensors or sources in accordance with military standardization will generate it.

The Common Relevant Operating Picture is obtained by consumers (humans or software agents) subscribing to relevant information specified in accordance with the same ontology used in the creation of the COP.

Table 4: *Precision Engagement Software Technologies*

Precision Engagement Activity	Advanced Software Technologies Needed
Information Dominance and Enhanced Situational Awareness	• Information Organization/Retrieval Using Ontology • Context-Sensitive Reference Architectures • Intelligent Software Agents • Model-Driven Computing • Human Factors Interactions With Complex Systems • Data Mining • Multi-Level Security
Precision Geo-Location of Time-Sensitive Targets	• Intelligent Software Agents
Shorter Sensor-to-Shooter Engagement Chain	• Context-Sensitive Reference Architectures • Model-Driven Computing
Wide Range of Precision Effects in Any Weather	• Software Technologies Are Adequate

Table 5: *Mature and New Software Technologies*

Mature, Deployed, Software Technologies	Advanced Software Technologies Supporting Protection of America
• High-Level Programming Languages • Compilers • Operating Systems • Object-Oriented Technologies • Relational Databases • Internet • Transmission Control Protocol/ Internet Protocol-Based Layered Networking • XML (Extensible Mark-Up Language)	**Technologies for Development of System Architectures** • Information Organization/Retrieval Using Ontology • Context-Sensitive Reference Architectures • Model-Driven Architecture Development • Reference Architecture Frameworks and Associated Development Processes **Other Advanced Software Technology Needs** • Human Factors Interactions With Complex Systems • Data Mining • Intelligent Software Agents • Cognitive Computing Techniques • Better Collaboration Tools

Context-Sensitive Reference Architectures

Reference architectures (see Figure 3) bridge the gap between processes addressing the development of contingency operations for future systems and the implementation of domain-specific architectures that build on legacy systems while incorporating new technologies and capabilities. Modeling and simulation is a key tool to support evaluating the effectiveness of the reference architectures and the resulting domain-specific architectures.

The results of modeling and simulation analysis provide metrics that can be used to eliminate, aggregate, or validate the key components and relationships with the family of architectures, using information organized via taxonomies and associated ontology. The reference architecture is continually updated and refined based on this feedback loop. The reference architecture is not the final blueprint for implementing systems-specific design and integration, but rather a reference of concepts providing the enabling cornerstone upon which systems can be empowered with large-scale mission capabilities. It is up to the organization accomplishing a software systems task to engineer and build an instance of the reference architecture to suit the needs of a particular domain, while maintaining compatibility with the overall standard reference architecture.

Reference architecture can be considered to have four abstract aspects: social, cognitive, information, and physical. Each aspect provides the context upon which to view system instances. Collections of systems instances change over time. The dynamics of a real-world environment necessitate the flexibility inherent in reference architecture to take into account changing elements over time.

The combination of the reference architecture and the four domain aspects provides the basis for examining mission systems in three dimensions instead of the traditional two as presented by the Department of Defense Architecture Framework (DoDAF)[2]. This three-dimensional view provides the basis for systems interoperability in a logical and meaningful way. Further analysis makes apparent the relationships between data, information, knowledge, and understanding required for combined systems operations and efficient management of available communications resources.

The mission-system reference architecture has the following properties:

- Provides the conceptual framework for specifying the four aspects (social, cognitive, information, and physical) of systems within the bounds of operational, system, and technical views prescribed by DoDAF.
- Acts as a template to guide domain-specific implementations of distributed network-centric systems while allowing a variety of design solutions.
- Defines the ontology for discussion and analysis purposes.
- Defines a complete set of architectural elements with well-defined interactions, functionality, and relationships with themselves and the external context.
- Defines how the elements communicate with each other, the basic operations associated with each element, and the nature of the communication.

Enterprise Reference Architecture Processes

The U.S. government has established direction and expectation for how complex systems of the future will be developed and integrated – via an ever-increasing emphasis on the importance of formalized architecture and enterprise architecture. Many aerospace and information technology companies are now developing and maturing their architecting processes to meet their business needs.

Lockheed Martin deploys its Architecture-Based Design and ARQuest Blueprint. Northrop Grumman has its Information Systems Architecture Analysis Continuum. IBM has the Enterprise Architecture Method. Boeing and General Dynamics promote their open systems architecture frameworks,

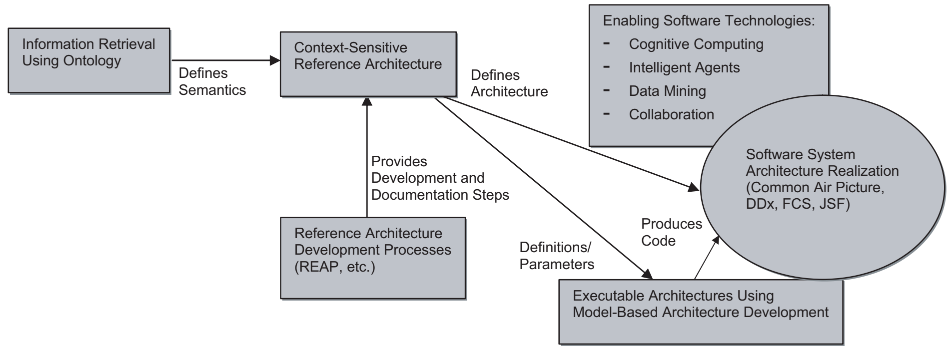


Figure 1: *Key Software Architecture Technologies Interact to Support Large Mission Systems*

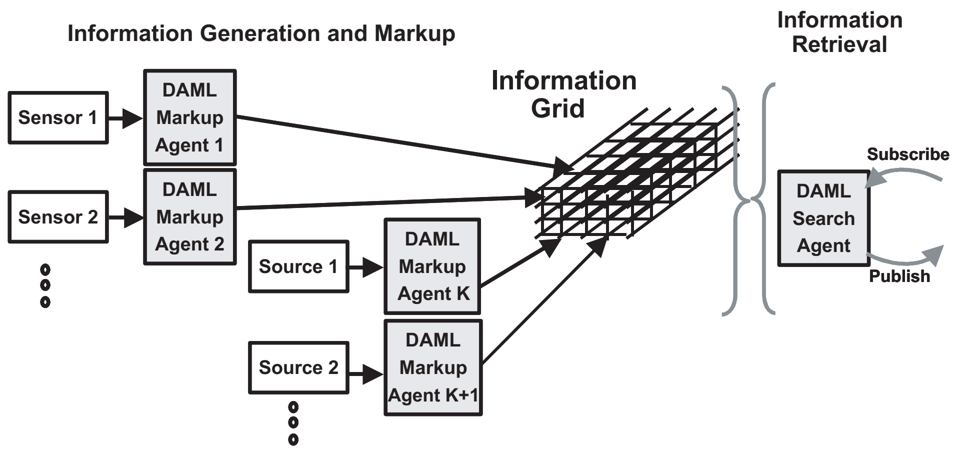
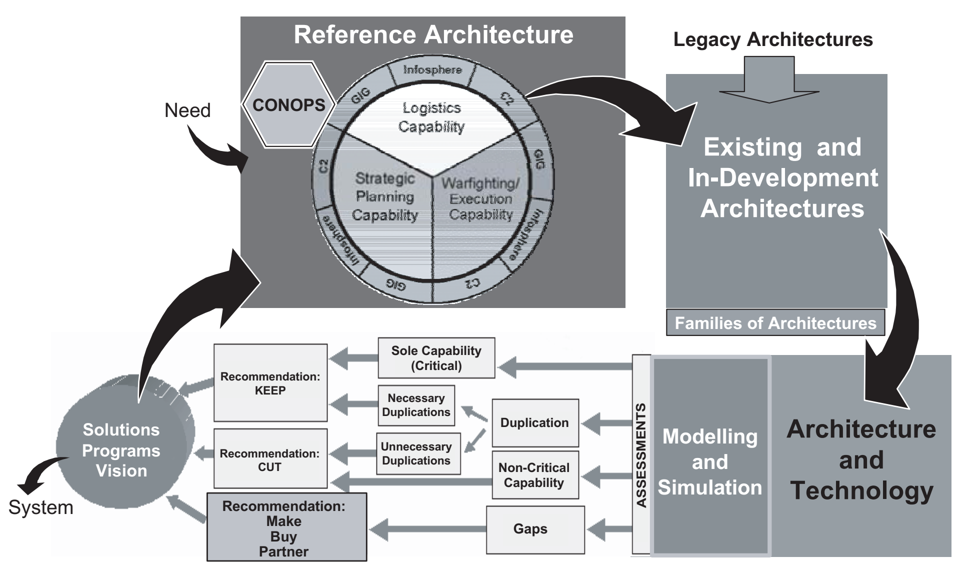


Figure 2: *Ontology-Based Information Retrieval*

Figure 3: *Reference Architecture Application to Domain-Specific Instances*



Bold Stroke and OpenWings, respectively. Government, industry, and academia are establishing consortia, certification programs, and graduate curriculum to address the educational needs of this new discipline.

The system architecting process that Raytheon uses is known as Raytheon Enterprise Architecture Process (REAP) [1]. It extends a traditional focus on technical architecture to include business architecture, providing a comprehensive view across the enterprise. The REAP defines an end-to-end architecture process based on industry and government standards, including The Open Group Architecture Framework[3] Command, Control, Communications, Computers, Intelligence, Surveillance, and Reconnaissance/Department of Defense Architecture Framework[4], Zachman Framework for Enterprise Architecture[5], and the Software Engineering Institute's Architecture Trade-off Analysis Method℠ (ATAM℠)[6].

Components

There are established industry and government standards to help address enterprise-wide architectural alignment among customer mission, business rules, data, application systems, organization, and technology. The primary standards unified within Raytheon's architecture process and other architecture processes to fulfill the components noted above are the following:

- **Methodology:** The Open Group Architecture Framework (TOGAF), Enterprise Edition.
- **Products:** DoDAF, final draft Zachman Framework for Enterprise Architecture.
- **Formats:** Unified Modeling Language[7], Integrated Computer-Aided Manufacturing Definition[8], DoDAF templates.
- **Validation:** ATAM.

It is important to note that although there are several integrated frameworks, they each address very different elements of the overall architecting process and their interrelation is both necessary and complementary.

Activities

Architecture processes are comprised of five primary activities: enterprise understanding, architecture planning, business architecting, technical architecting, and architecture validation. These activities are iterative in nature, internally and externally to the other.

In Raytheon's case, the five activities act as a wrapper around the phases of TOGAF's Architecture Development Method (ADM), providing supplemental guidance and describing its relationships to other standards. These subprocesses extending the TOGAF ADM include those for customer-focused architecting, quality attribute analysis, architecture concordance/configuration/consolidation, DoDAF product generation, ATAM, and quality attribute assessments. The completion of these activities results in a validated architecture package describing the enterprise from a variety of viewpoints or perspectives.

"When proved successful, model-driven computing has the potential to revolutionize the current means of systems specification, development, testing, and maintenance."

Model-Based Computing

Model-based computing is the term for system and software development that is driven and centered on models. These models are used to specify systems and software architecture, and low-level system design details. The models provide the means to translate the specified systems architectural artifacts defined via system architecture development processes into constituent platform-specific and platform-independent components. The concept of developing platform-independent models, followed by platform-specific models is quite powerful and allows our programs to migrate models to new computing hardware with minimal impact. Platform-independent models can also be used in multiple environments such as simulations, using the same system model.

This concept has been standardized via the Object Management Group (OMG)[9] in the Model-Driven Architecture initiative. The OMG is working to standardize these concepts in order to promote tool development and interoperability. Recently, the OMG has also formed an interest group specifically focused on standards for model-driven development of embedded software. This interest group will leverage recent significant advances made possible in large part via the leadership, insight, and funding support from DARPA. These new tools and technologies are laying the necessary foundations upon which the systems of the future will be specified, developed, tested, and maintained.

DARPA has been advancing the state-of-the-art application of model-driven computing to distributed, real-time, and embedded (DRE) systems. DARPA, via the Model-Based Integration of Embedded Systems (MoBIES)[10] program, is establishing an open-source, standards-based tool suite needed to accomplish the program's objectives. One MoBIES technology developer is the Institute for Software Integrated Systems (ISIS) at Vanderbilt University[11]. ISIS, as well as being a major contributor to the MoBIES program, is working to see that DARPA-funded efforts migrate into the mainstream. They are working to migrate DARPA-funded tools to the Eclipse Open-Tool Integration Framework via sponsorship from IBM.

Raytheon and the aerospace industry are actively involved with the development of standards that impact the future of model-driven computing within the OMG. These standards may be impacted by the further evolution of DARPA-developed tools and technologies from MoBIES and other DARPA programs. The maturation of those tools is being supported via membership in the newly formed Embedded Systems Consortium for Hybrid and Embedded Research.

Model-driven computing has had some noteworthy successes despite being used in limited domains. Two popular examples are The Mathwork Company's Matlab/Simulink®[12] and National Instruments' LabVIEW[13]. These pioneering tool suites demonstrate that model-driven computing is effective in limited application domains. Until recently, modeling of the entire system, middleware, and application, needed to be accomplished for each system. This made it cumbersome, time-consuming, and expensive to develop effective models. It was not until the separation of the application from the middleware, and models of the middleware could be shared and leveraged, that model-driven computing has come into its own.

Additional advances in model-driven computing are necessary before it can become commonplace in DRE systems development. Scalability in both breadth and depth of model-based computing must be addressed. When proved successful, model-driven computing has the potential to revolutionize the current means of systems specification, development, testing, and maintenance. We expect that the most significant impact will be realized in system verification. With complete and executable system models that are independent of the hardware platform, system verification will move forward in the development process, reducing the cost and risk of errors, and facilitating the final system verification effort.

Conclusion

William Gibson once stated, "The future is already here; it is just unevenly distributed" [2]. The successful implementation of the large systems of systems needed for America's protection will be expedited by using emerging, but not yet widely deployed, software approaches that support the development of robust system architectures. The key technologies of ontology, context-sensitive reference architectures, architecture definition processes, and model-based computing are beginning to be integrated to develop robust systems that are key for America's defense. More research is required to make these approaches scalable and capable of integrating with existing systems, but the foundations exist today.◆

Acknowledgements

We would like to acknowledge the contributions of Steve Ignace, Rolf Siegers, Mike DaBose, Chris Grounds, Ralph Woods, Tom Flynn, Bryan Lail, Edwin Lee, Bhatra Patel, Doris Tamanaha, Ron Williamson, and Don Wilson.

References

1. Siegers, R. "The Raytheon Enterprise Architecture Process." INCOSE 2003, Crystal City, VA., July 2003.
2. DeLong, J. Bradford. "The Real Shopping-Cart Revolution." Wired Mar. 2003 <www.wired.com/archive/11.03/view.html?pg=5>.

Notes

1. See <www.daml.org>.
2. See <http://deskbook.dau.mil/software/gen/comparch-def.html>.
3. See <www.opengroup.org/architecture/togaf>.
4. See <www.dod.mil/comptroller/bmmp/pages/arch_arch_home.html>.
5. See <www.zifa.com>.
6. See <www.sei.cmu.edu/ata/ata_ method.html>.
7. See <www-306.ibm.com/software/rational/uml>.
8. See <www.idef.com/default.html>.
9. See <www.omg.org>.
10. See <www.rl.af.mil/tech/programs/MoBIES>.
11. See <www.isis.vanderbilt.edu>.
12. See <www.mathworks.com>.
13. See <www.ni.com/labview>.

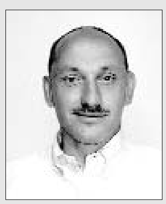
About the Authors

Gregory S. Shelton is vice president of Engineering, Technology, Manufacturing, and Quality for Raytheon Company. He is responsible for developing and implementing enterprise engineering, quality and program management processes and tools, and integrating technology strategies, road maps, and competitive assessments. In 2002, he was elected associate fellow for the American Institute of Aeronautics and Astronautics. Shelton has a bachelor's degree in electrical engineering from California Polytechnic University and a master's degree in engineering and management from the University of California, Los Angeles.

Raytheon Global Headquarters
870 Winter ST
Waltham, MA 02451
Phone: (781) 522-3000
Fax: (781) 522-3001
E-mail: gshelton@raytheon.com

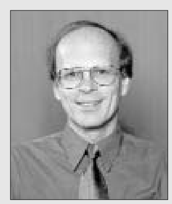
Randy Case is technical area director at Raytheon Company for Architectures and Systems Integration, Garland, Texas. He was the architect of the Raytheon Integrated Product Development System. Case has worked on projects that span the entire life cycle from independent research and development to operational support. He is co-chair of the International Council on Systems Engineering Standards Technical Committee, and has contributed to a number of systems-related standards. Case has a Bachelor of Science in electrical engineering from the University of Texas at Arlington.

Raytheon/Intelligence and Information Systems
1200 South Jupiter RD
Garland, TX 75042
Phone: (972) 205-5306
Fax: (972) 205-8083
E-mail: randy_r_case@raytheon.com



Louis P. DiPalma is the manager of the Integrated Warfare and Sensor Systems Software Department of the Raytheon Integrated Defense System Integrated Software Development and Human Systems Interface Engineering Center. DiPalma has been involved in the design and development of submarine combat control systems and weapon launching programs, as well as several fire control systems for surface combatants. His focus has been centered on the infusion of new technology into the Raytheon integrated decision-support product line, with a primary focus on naval combat systems.

Raytheon Integrated Defense Systems
1847 West Main RD
Portsmouth, RI 02871
Phone: (401) 842-5592
Fax: (401) 842-5232
E-mail: louis_p_dipalma@raytheon.com



Dan Nash is director of Software Engineering, Raytheon Corporate Engineering, Waltham, Mass. He is responsible for coordinating mission-critical software engineering activities across the various business units, as well as other assignments such as Red Teams. His recent work has been in the areas of the Capability Maturity Model®, Integration and software supplier management. He holds a Bachelor of Science in electrical engineering from North Carolina State University.

Raytheon Company
870 Winter ST
Waltham, MA 02451-1499
Phone: (781) 522-3362
Fax: (781) 522-6434
E-mail: j_dan_nash@raytheon.com