Development of Safety-Critical Software Systems

Since the development of the digital computer, software continues to play an important and evolutionary role in the operation and control of hazardous, safety-critical functions. The reluctance of the engineering community to relinquish human control of hazardous operations has diminished dramatically in the last 15 years. Today, digital computer systems have autonomous control over safety-critical functions in nearly every major technology, both commercially and within government systems. This revolution is due primarily to the ability of software to reliably perform critical control tasks at speeds unmatched by its human counterpart. Other factors influencing this transition include our ever-growing need and desire for increased versatility, greater performance capability, higher efficiency, and a decreased life-cycle cost.

In most instances, software can meet all of the above attributes of the system's performance when properly designed. The logic of the software allows for decisions to be implemented without emotion and with speed and accuracy. This has forced the human operator out of the control loop because they can no longer keep pace with the speed, cost effectiveness, and decision making process of the system.

According to the MIL-STD-882D, the main objective (or definition) of system safety engineering, which includes safety-critical software systems, is "the application of engineering and management principles, criteria, and techniques to optimize all aspects of safety within the constraints of operations effectiveness, time, and cost throughout all phases of the system life cycle."

The ultimate responsibility for the development of a "safe system" rests with program management. The commitment to provide qualified people and an adequate budget and schedule for a software development program begins with the program director or manager. Top management must be a strong voice of safety advocacy and must communicate this personal commitment to each level of program and technical management. Project directors or managers must support the integrated safety process between systems engineering, software engineering, and safety engineering in the design, development, testing, and operation of the system software.

This issue of CrossTalk provides an in-depth look at the implementation and development of safety-critical software systems. It also explores how these systems will likely face unplanned challenges during long-term development, requiring developers to build flexibility into their approaches.

Authors Dr. Victor Basili, Kathleen Dangle, Linda Esker, Frank Marotta, and Ioana Rus guide readers through their methodology for developing early safety measures on safety-critical software system projects in *Measures and Risk Indicators for Early Insight Into Software Safety*.

In *Safety and Security: Certification Issues and Technologies*, Dr. Benjamin M. Brosgol analyzes the two primary safety and security standards—DO-178B and the Common Criteria—and gives software professionals the tools to avoid hazards and vulnerabilities.

First responders who need a secure and mobile coordination and communication infrastructure during crisis will take special interest in Sugih Jamin's *WebBee: A Platform for Secure Mobile Coordination and Communication in Crisis Scenarios*.

In *Constructing Change-Tolerant Systems Using Capability-Based Design*, Dr. James D. Arthur and Ramya Ravichandar recognize the need for flexibility and provide readers with thought-provoking ideas on how a capability-based approach may be the answer to complex, large-scale systems that are hostile to change.

And, finally, don't miss *DoD Business Mission Area Service-Oriented Architecture to Support Business Transformation* by Dennis E. Wisnosky, Dimitry Feldshteyn, Wil Mancuso, Al (Edward) Gough, Eric J. Riutort, and Paul Strassman. They examine whether a service-oriented architecture (SOA) is the best fit for the Department of Defense's (DoD's) Business Mission Area (accounting for roughly half of the DoD Information Technology budget) and examine the DoD's SOA vision.

I hope you enjoy reading CrossTalk's variety of articles on how to better approach the development of safety-critical software systems. I certainly did.

Ken Chirkis
Naval Air Systems Command