

# **Keynote Address**

## **Fixing Software Before It Breaks**

### **S. Tucker Taft (AverCom Corporation – A Titan Company)**

How far can we go in finding bugs in software using static analysis? What is practical? Ada already catches an order of magnitude more bugs at compile-time than C-based languages. What kinds of annotations or new language features would take Ada to the next level of support for static error detection, getting another order of magnitude of reduction in undetected errors?

S. Tucker Taft is Chief Technologist for the Commercial Information Technology Solutions division of AverStar, Inc. (a subsidiary of Titan, Inc.) He is also technical director for AverStar's Web Portal Integration services.

Mr. Taft graduated from Harvard College in 1975 with a bachelor's in Chemistry, Summa Cum Laude, and then worked four years for Harvard in the student computer center, managing the first Unix system that was installed outside of AT&T. Thereafter he worked one year as a private consultant, and then in 1980 joined AverStar (then known as "Intermetrics"). From 1980 to 1990, he participated in the development of the Ada Integrated Environment for the Air Force, a commercial C cross-compiler, the Common APSE Interface Set (CAIS), and an Ada binding to SQL (SAME). From 1990 to 1995, Mr. Taft led the Ada 9X language design team, culminating in the February 1995 approval of Ada 95 as the first ISO standardized object-oriented programming language. More recently, Mr. Taft led the development of AverStar's Ada 95 compiler technology, "AdaMagic," AverStar's Ada/Java compiler, "AppletMagic," and AverStar's Java/XML-based Web Portal technology.

