

# Military & Aerospace

## E L E C T R O N I C S<sup>®</sup>

A PENNELL PUBLICATION

### ANALYSIS

## Ada business rebounds in post-mandate software world

By John Keller

**P**ractitioners of the Ada computer language are enjoying a renaissance of sorts after enduring years of flat markets, failed expectations, industry derision, and outright abandonment from its earliest and strongest benefactor, the U.S. Department of Defense (DOD).

The road has been long and difficult since Ada first thrust on the scene in the early 1980s as DOD's official and mandated computer language for real-time mission-critical applications. For years, Ada was widely seen as a military-only language that had been shoved down the throats of unwilling software engineers.

Ada engineers have probably endured the worst — including widespread resentment and disillusionment, as well as a brutal industry shakeout. Now they appear to be on the upswing. Some Ada company leaders today are reporting double-digit revenue growth, stable-if-not-growing interest for military and aerospace applications, and a growing use of Ada in commercial and industrial applications.

"Our Ada business is steadily increasing by 40 to 50 percent a year. It's been robust and healthy, to say the least," says John Carbone, vice president of marketing at Green Hills Software Inc. in Santa Barbara, Calif.

Green Hills Ada programming tools are involved in projects to build the future U.S. Joint Strike Fighter, as well

as programs to upgrade the B-1B strategic bomber, B-52 bomber, AV-8B Harrier jump jet, and C-17 airlifter, Carbone says.

Green Hills officials are not alone in their optimism about Ada. "We are seeing the language popping up in places far afield from the military, such as in yachts, video systems, and video encoding for images stored on tapes and disk," says Joyce Tokar, vice president of technology at Ada tools vendor DDC-I in Phoenix.

"The market is recognizing there is an advantage to a programming language that supports early detection of errors at compile time," Tokar says. DDC-I provides Ada compilers and tools for the U.S. Army RAH-66 scout-attack helicopter computer systems.

Things did not always look so good for the Ada business. As a matter of fact, only three years ago it looked as if Ada had hit bottom and was on the verge of being forgotten.

In 1987 the DOD dropped its long-standing mandate to use Ada. Perhaps more significantly, Pentagon leaders a year later not only disbanded the DOD Ada Joint Program Office (AJPO), which was the premiere government and industry clearinghouse of information and reusable software libraries, but also cut off virtually all funding for Ada development.

At the time, many in the software business saw those developments as the death knell for Ada. Today, however, some experts say the DOD's withdrawal of support for Ada was a bless-

ing in disguise. The reason: prospective customers now judge Ada purely on its technological and financial merits, with little political baggage.

"On a whole it is a good thing that Ada must stand on its own two feet," says Ben Brosgol, Ada technologist at Aonix of San Diego. "With the mandate no longer operative, and with the AJPO no longer in business, the interest is coming from commercial business rather than in DOD."

Freedom from government mandates is giving Ada a second chance, which is starting to pay off, Carbone says. "Now that there is no longer a mandate, people are even more receptive to it. Maybe the Ada mandate had a depressing effect on the use of Ada outside of government."

The demise of DOD's Ada mandate "has not hurt us at all, and may have helped," Tokar says. "Today the software community recognizes the value of process and rules of developing software. People are much more familiar with packages, and components of software."

It is no secret that DOD's involvement in the language gave Ada a huge boost in the early 1980s. Yet many longtime industry experts now believe that DOD attention may have hurt Ada perhaps as much or more than it helped.

Software pioneers crafted Ada, named for 19th century mathematician Ada Byron Lovelace, two decades ago under DOD sponsorship in an attempt to create a standard and

validated programming language that lent itself to rigid software engineering discipline, high reliability, and easy code reuse.

At that time, DOD leaders envisioned Ada as their key to reining in runaway software costs and centralizing control over a fast-growing number of software projects using different programming approaches and languages. DOD leaders moved quickly — some say too quickly and too zealously — to mandate Ada for all of the department's new projects.

Not only did DOD officials provide seed money to push Ada development along, but they also established the AJPO, which served as a nucleus that held much of the Ada community together.

Despite a promising beginning, trouble for Ada began almost instantly. "Early on, Ada was over-hyped, and was considered to be too much of the silver bullet," Brosgol explains. "People ended up with an initial bad taste in their mouths."

The specification for the first version of Ada, called Ada 83, hit the streets before reliable Ada compilers were available. While the fledgling Ada industry struggled to develop usable compilers, DOD officials tried to develop a compiler and tools called the Ada Language System, which Brosgol says was a technological disaster.

"When Ada 83 came out there were no compilers, and when they did come out they were very poor," Brosgol says. Critics not only resented the DOD's forcing them to use a new and unproven language, but they also complained that Ada code was too large to run efficiently on the microprocessors of the day. In addition, the Ada and tools and compilers available then were relatively expensive.

To complicate matters further, many early proponents viewed Ada as a one-size-fits-all language that software engineers could apply to a wide variety of programming tasks, from databases and financial-management software, to real-time embedded applications.

Now much of that is changed. "People are using Ada where it is appropriate," Carbone says. "You don't have to use Ada for anything other than what you want to use it for — large bodies of code where structure and error detection is important. People like the language for its inherent security features. It is a rigorously typed language that keeps the programmer well within defined bounds."

Ada vendors also have learned from past mistakes, and try to put the language's compromised reputation behind them. The latest version of the language, Ada 95, has rectified many of the previous version's technological shortcomings.

More importantly, Brosgol says Ada

industry leaders made sure they had reliable compilers available when Ada 95 became widely available. In addition, the prices of Ada tools have come down, and free open-source Ada tools are available from Ada Core Technologies Inc. (ACT) in New York.

Members of the Ada community also are much more realistic than they used to be. "Ada is no longer at the hottest edges of development," Brosgol concedes.

Even Carbone of Green Hills admits the demand for Ada overall is not growing quickly, and has settled into serving niche markets. His company's Ada business is growing by picking up opportunities that other companies have abandoned. "Our competitors are de-emphasizing their support for Ada, and we are getting more and more of the available business."

Looking back, Ada experts now realize that they faced some problems in the early days that were outside of their control.

"Ada was ahead of its time, and behind its time," Brosgol says. Ada, he points out, came to maturity in the middle of the computer industry's transition from mainframes to PCs. Ada traces its lineage from mainframe languages such as FORTAN, COBOL, and Pascal. It never made a smooth transition to PCs as the C language did. **M&AE**

The AdaMULTI® Integrated Development Environment is a complete software toolkit for building embedded applications in Ada 95 or mixed languages (Ada 95, C, C++, and FORTRAN). AdaMULTI runs on Windows 95/98/NT or Unix hosts and supports remote debugging to a variety of target environments. AdaMULTI provides a direct graphical interface with all Green Hills compilers, and supports multi-language development and debugging.



2708 Alternate 19 North • Suite 708 • Palm Harbor, FL 34683  
Tel: 727.781.4909 • Fax: 727.781.3915 • Email: sales@ghs.com