Ada catches on in the commercial market

Ware Myers, Contributing Editor

After 11 years of development by the US Department of Defense, Ada appears to be on the verge of widespread commercial acceptance. In September, Ada was sanctioned by IBM when the company unveiled the IBM Development System for the Ada Language, a series of compilers and development tools provided to IBM by Telesoft under an OEM agreement. "We are seeing a real upturn in the demand for Ada products," reports Bruce Sherman, director of marketing at Telesoft, a San Diego vendor of Ada compilers and development systems. Other vendors agree.

It has been a long wait. DoD established the High Order Language Working Group in January 1975. It took six years to set requirements, design the language, and evaluate more than 5000 comments. It took two more years to reach ANSI standardization. But to be practically useful, a language must have compilers and a set of programming support tools.

Compilers. The work involved in developing Ada compilers appears to have been underestimated by many. "Writing a practical compiler that exploits Ada's good features is one of the toughest jobs in computer science," said Win Royce of Lockheed's Software Technology Center in Austin, Texas.

Perhaps because of this difficulty, the first three compilers were not validated until late 1984. Even then there was general agreement that they had not attained production quality. By June 1986, however, 47 compilers had been validated and 22 more had been scheduled for validation.

More importantly, users were finding products that they deemed to be of production quality. Vendors of Ada products include Alsys of Waltham, Massachusetts; Digital Equipment Corp. of Maynard, Massachusetts; Gould Computer Systems Division of Fort Lauderdale, Florida; Intermetrics of Cambridge, Massachusetts; Rational of Mountain View, California; Telesoft; and VerdiX of Chantilly, Virginia.

Embedded systems. "About 80 percent of the procurements I see, especially the larger ones, are calling for Ada," Royce said. "I don't think very many people are actually building in Ada yet, but we are on the verge — the next 18 months will see a massive shift into Ada."

That possibility is bothering a sizable portion of the embedded systems community, according to a panel discussion at the Ada-Jovial Users Group in July. All five panel members expressed apprehension and the audience response indicated concurrence. Speaking from the floor toward the end of the session, Air Force Maj. Allan Kopp, deputy director of DoD's Ada Joint Program Office confided that he had made six pages of notes of matters to look into when he returned to Washington.

The problem lies not in Ada's promise. The language was designed to contribute to the more effective management of large, complex projects, and to increase programmer productivity, reduce errors, improve performance, and facilitate maintenance. Those who have used Ada on large projects, principally the organizations that are developing compilers and support environments in Ada, agree that Ada fulfills those promises.

Dudrey Smith of Lear Siegler of Grand Rapids, Michigan, and moderator of the Ada-Jovial panel, acknowledged in an interview that Ada would probably be more efficient in the long run. "In the embedded systems field, however, it is one more uncertainty on top of a number of other uncertainties, beginning with the great difficulty of getting a firm definition of the task in the first place," he said.

Commercial applications. Although DoD originally targeted Ada at embedded systems, it now appears that the new language will also find many commercial applications. "Many companies in the commercial sector whose products or services are highly dependent on software are aware that they, too, must find new software approaches if they are to maintain their competitive edge," a Rational press background reader says. "Typically, these companies' software systems share many of the same characteristics as those in the defense and aerospace communities, including complexity and size."

For instance, the Swedish telecommunications group, Televerket, announced in July a $4 million equity investment in Telesoft. Televerket intends to use Ada in a new switching application. Also, Softech is rehosting its Ada Language System to the DIPS computer used by Nippon Telegraph and Telephone of Japan.

CRI, of Santa Clara, California, has a 180,000-source-line database management system programmed in Ada. Interfaced to another Ada system, CRI's DBMS retrieves data about three times faster than it does when interfaced to a system in some other third-generation language.

Computer Corp. of America, of Cambridge, Massachusetts, has implemented a 400,000-line database management system in Ada with funding from the DoD Advanced Research Projects Agency and the Navy.

Last July, the first meeting of the Commercial Ada Working Group was held with ACM's SIGAda. About 40 percent of the attendees were interested in Ada for classic management information systems applications, according to Intellimac's David M. Dikel, chairman of the working group. Dikel said the first commercial application of Ada was Intellimac's payroll and inventory system, completed in 1982 using an early compiler from Telesoft. Some of the aerospace companies using Ada on military contracts are planning to use it for business applications, he added. The next meeting of his group will be held with SIGAda the week of November 17, in Charleston, West Virginia.

At an August meeting of IBM users, IBM indicated it would support a compiler for the 370 MVS and VM/CMS operating systems, opening the way for Ada to invade the IBM mainframe environment. Then, on September 2, it announced the IBM Development System for the Ada Language, consisting of compilers and productivity tools. Containing about 300,000 lines of Ada code, the compilers have been validated under ACVC Version 1.7. "In the long run, Ada is the right language," observed Grady Booch, author of a textbook on the language. "Projects are becoming more and more complex. Unless managers employ a language suited to this complexity, their work will become so complex that they will hardly be able to do anything."
Architecture, simulation key to SDI success

Galen Gruman, Assistant Editor

A year after a group of computer scientists made their recommendations, the Strategic Defense Initiative Organization approved a “Star Wars” software architecture that incorporates those recommendations.

“We have an architecture now for battle management that incorporates all of the Eastport Group’s concerns,” said Air Force Lt. Col. David Audley in an October 1 interview in Los Angeles. Audley is an assistant director of the SDI Organization and heads the battle management and command, control, and communications (BM/CC3) program there.

The Eastport Group, as the scientists called themselves, told the SDI Organization in a December 1985 report that usual military software development and procurement tactics should not be followed in the SDI program. It recommended an open, decentralized architecture that favored independent nodes and the use of novel approaches to system design. (See January 1986 IEEE Software Soft News.)

Eastport recommendations. The advisory group approved the architecture in late September. Its concepts — some called “radical for government” by Audley — were formally accepted in February by the SDI’s director, Air Force Lt. Gen. James A. Abrahamson.

The Eastport Group and its report attracted scrutiny and criticism soon after the group was formed in spring 1985. David Parnas, a long-time defense computing researcher, quit the panel, saying SDI’s goal was “impossible.”

However, most seem to agree with the general thrust of the Eastport Group’s recommendations, whether or not they believe the recommendations will ultimately result in a successful system. “It’s been regarded highly by friend and foe alike. We’re really serious about doing it this way,” Audley said.

“It’s often said you can’t build an Astrodome over America. Well, what can you do? We envision a program that will look at defensive technologies,” Audley told a Los Angeles ACM meeting October 1. “We can’t build all this stuff and put it in space and scratch our heads and ask ‘What are we going to about it?’ We have to think ahead.”

In the initial SDI concept, “nobody ever said much about the battle management, command, and control software — what is really the heart of the system,” he said. In fact, the only unclassified part of the 1983 report that led to the SDI program dealt with software. It was as if, Audley joked, “the software is so impossible that we can tell them what we’re going to do and it still won’t help.”

Architecture payoffs. The biggest payoff since the recommendations’ acceptance, Audley said, was that a database distributed among stations in space and on the ground is no longer considered necessary. Many researchers feared such a system would be cumbersome, difficult to develop and implement, and easy to disable.

Instead, each space-based platform in the envisaged defense system will create all data required by it. This avoids multiplatform processing and synchronization between the stations, which would have only intermittent contact because of different orbits.

The platforms — both sensors and weapons — will act instead as individual soldiers: each will take orders from a higher level authority and communicate and coordinate with neighboring platforms, but will rely on its own resources and engagement models if necessary.

Another payoff has been the realization that “we can use existing multiprocessors for distributed operating system needs,” Audley said.

One key aspect of the architecture approach is to avoid complexity. “If it’s too complex then we don’t want it.”

The architecture approach also requires that the development of SDI software be flexible, Audley said. “We’re trying different ways.” The object is not to get locked into an unworkable or needlessly complex approach, he explained.

Language. One area of flexibility is in the choice of language. Ada is the preferred language, and will be required for communications applications in 1987 research projects. But computer languages, like human ones, express different things differently. Some concepts are better expressed in one language than another, so requiring that all work be done in one language would needlessly shape the programming approach, Audley said.

Audley is also uncertain about Ada’s runtime performance in SDI applications. Still, he said, “I think Ada is very expressive, a rich language. If I had a dictionary of only 1000 words then I wouldn’t get ‘Tolstoy.’”

Simulation. Until — and if — such a system is developed and implemented, SDI remains a research project to conform with US treaty obligations. The most important part of that research effort, Audley said, was the simulation and prototyping efforts that will be undertaken in the National Testbed, a planned simulation facility distributed across hundreds of sites.

“The simulation [effort] now is uncoordinated. We need to tie it all together,” Audley said. The distributed nature of the testbed will help develop the actual SDI control software, he added, because it will teach program researchers how to implement such a distributed system before developing any SDI code. “A VAX here, a VAX there, a Cray over there — we’ll learn about distributed systems,” Audley explained. “It’s true synergy.”

“The software’s going to think it’s really fighting a war,” Audley said. The simulation may even include space-based computers. “We want to make believe we’re doing it for real. We would like to, in a sense, deploy this prototype. We are not ready for the bricklayers to come in yet.”

Testbed. Work on the testbed began this year. It will begin as the coordination of current simulation facilities and expand as the technology and budget allows. The development of programs under the framework of the adopted architecture also continues. Prototyping and testing will come later.

Key to the National Testbed’s success will be the creation of a networking system. Now called Internet, this system will be more capable than the Defense Department’s ARPAnet system. ARPAnet has a 56-kilobit packet-switching capability. Audley said Internet must handle packet-switching in the megabit range.

Audley envisions the testbed to contain many multiprocessor-based, event-driven, message-passing systems, perhaps large vector machines such as Crays and ETAs.

The testbed will allow substantial prototyping, another key area in the SDI research plans. “You learn a lot with prototypes,” Audley said, inviting University of North Carolina professor Fred Brooks’s famous advice, “Plan to throw one away, you will anyhow.”

Once the prototyping stage is reached, the relatively open, unclassified research approach followed by the SDI Organization will change, Audley said. “We will need to ensure that the implementation and building are not bugged. The concern is physical access. We will use red teams [SDI personnel acting as enemy agents] to break into the system.”
Outside pressures. While Audley insisted SDI research is not shaped by political decisions, he acknowledged that many development and implementation problems will be addressed politically. “It’s like a thermostat, controlled by humans,” he said.

One pressure is the decision by many scientists to not work on SDI research. Audley told of some code used in an astronomy project that “had just what we needed, but the guy who owned the code restricted it so it couldn’t be applied for SDI. . . . It hurts. We need all the talent that we have.”

On the other hand, the large SDI budget — $2.8 billion in 1986, compared to the National Aeronautics and Space Administration’s $5 billion and the National Science Foundation’s $1.3 billion — has prompted some pressure from congressmen for immediate results. Recent footage of hardware tests released to the press is an example of a limited response to this pressure.

Audley stressed that the software effort must not be railroaded by such pressure. His major concern is introducing bugs. “We prevent bugs by design. We don’t like to preplan the removal of bugs. The issue is not to get things going [now] because we [will] have lots of debugging to do [later],” he stressed.

SEI gets new director

The Software Engineering Institute has named Larry E. Druffel, vice president for business development at Rational, as its new director. Druffel replaces John H. Manley, effective September 15.

The two-year-old SEI is a federally funded research center whose mission is to accelerate the implementation of those software technologies that reduce costs and improve quality, particularly in defense software applications. The institute is headquartered at Carnegie Mellon University in Pittsburgh.

Before joining Rational, a Mountain View, California-based software engineering firm that specializes in aerospace and defense work, Druffel was director of computer systems and software at the Office of the Deputy Undersecretary of Defense for Research and Advanced Technology. He was the principal architect of the Defense Department’s Software Technology for Adaptable, Reliable Systems (STARS) program.

Inman leaves MCC for startup firm

Adm. Bobby Ray Inman has announced he will leave his chief executive officer post at the Microelectronics and Computer Technology Corp. research consortium in Austin, Texas. The resignation, announced September 3, is effective December 31. He will become CEO and president of Westmark Systems, a startup defense electronics firm backed by a Dallas holding company.

Inman took the job as the 21-company consortium’s founding CEO in 1982 and agreed to a three-year term, but extended his contract to a fourth year.

He is leaving the consortium because he feels it is now stable, said William Stotesbery, MCC’s public affairs director. “He loves the thrills of building organizations,” Stotesbery said, and MCC has been built.

Before joining MCC, Inman served as director of naval intelligence, director of the National Security Agency, and deputy director of the Central Intelligence Agency.

A search committee is looking for a replacement, which it hopes to find before Inman leaves December 31.

CLASSIFIED ADS

RATES: $4.50 per line, $45 minimum charge (up to 10 lines). Average six typeset words per line, nine lines per column inch. Add $4 for box number. Send copy at least six weeks prior to month of publication to: Carol E. Porter, Classified Advertising, IEEE Software Magazine, 10862 Los Vaqueros Circle, Los Alamitos, CA 90720.

WANG INSTITUTE OF
GRADUATE STUDIES
School of Information Technology
Software Engineering Faculty

Wang Institute provides a unique educational environment that combines the best features of academic and industry. Faculty members teach one graduate course per semester, interact closely with small numbers of graduate students who have industrial experience, supervise production quality software projects, and pursue their individual research interests. Positions are available for full-time faculty members and one-year visitors who have experience in using and teaching the technical and managerial principles of software engineering. Faculty members receive industrial-level salaries and are provided with sufficient equipment and support staff to achieve their professional goals. Each faculty member is expected to maintain a reasonable balance between research in software engineering and exposition of that material in the classroom. Teaching ability and an interest in graduate-level education are essential.

The campus of the School of Information Technology is located near the Boston hi-tech and university areas which provide rich environments of academic, cultural, and professional activities. Instructional computing facilities at Wang Institute include a VAX 11/760 (VMS), a VAX 11/780 (UNIX), a Prime 750, a Wang VS100, Apollo DOMAINS, and IBM, Apple, Wang, and Hewlett-Packard personal/professional computers. A growing collection of software tools for software engineering is maintained by the staff of the Wang Institute Software Environment project.

Applicants should submit the names of three references and a resume that indicates professional specialty areas, teaching experience, and experience in the practice of software development and software management to: Richard E. Fairley, Faculty Chair, Wang Institute of Graduate Studies, 870 Harvard Street, Cambridge, Massachusetts 02138.

November 1986

MODULAR LANGUAGES ARE NOT ENOUGH

For the production of high-quality reusable software components, you need all the power of true object-oriented design and programming.

Effel is the first object-oriented programming system designed exclusively for demanding software designers in practical development environments. Its simple yet powerful design offers a unique combination of features for producing high-quality software.

- classes as the basic system structuring mechanism
- multiple and repeated inheritance
- static type checking and information hiding for security
- assertions and invariants for expressing and checking software correctness
- efficient compilation and execution
- library of highly reusable software components
- automatic run-time storage management, with parallel garbage collection
- automatic configuration management
- supporting tools for documentation and debugging.

Effel is available now on most versions of the Unix* operating system. For other systems, please contact us.

Don't let your software development be hindered by obsolete technology. For reliable, reusable, extendible software, Effel makes object-oriented design a reality in today's industrial environments.

For more information on how Effel can improve the quality of your software, contact Interactive Software Engineering, Inc., 270 Storke Road, Suite 7, Goleta, CA 93117, or call us at (805) 685-1006. We also offer training and consulting in object-oriented methods, and other advanced software engineering tools.

November 1986

Interactive Software Engineering, Inc.

Professional tools for the development of high-quality software

Effel is a trademark of Interactive Software Engineering, Inc. Unix is a trademark of AT&T.