Is Ada Dead or Alive Within the Weapons System World?

Donald Reifer
Reifer Consultants, Inc.

Jeff Craver
U.S. Army

Mike Ellis and Dan Strickland
Dynetics, Inc.

The Theater High Altitude Area Defense program commissioned a study on the long-term viability and staying power of Ada after the demise of its mandate in 1997. The study would help decision makers determine if Ada had the staying power to support future systems, or whether an alternative should be sought. This article puts the results of that study in the public domain.

Since the demise of the Ada mandate on April 29, 1997, little has been said about the debate that raged in the mid-1990s comparing the merits of C++ versus Ada [1] within the weapons system world. Many questions remain such as: "Has the defense community made a wholesale move to competing languages like C++ and technologies like Java?" "Is the migration pattern the same for embedded software as it is for application software?" "Are high quality compilers, bindings, tools and libraries still readily available?" "Is Ada still making converts in academic, industrial and government circles?" "Are vendors making a profit?" "Will vendors exist in the future to satisfy the community's continuing demands for training, compilers, tools and support?" "Will organizations be able to find and hire programmers skilled in Ada language and associated toolsets now and in the future?"

To determine if Ada is still viable, proponents must answer these and many other similar questions.

Like many weapons systems organizations, the Army missile defense community embraced Ada two decades ago because it was the best available alternative to reduce the risk—cost, schedule, and technical—in developing safety critical systems. The language directly supported real-time development needs and risk reduction through standard enforcement. It provided the tools contractors needed to develop highly complex, distributed, real-time systems. If given the chance to make the decision again, under similar circumstances, the Army would still choose Ada.

However, times change along with decision criteria. Today, the missile defense community has millions of lines of Ada software that must be maintained, sustained, and supported up to a 20-year time period. As programs enter full scale development, decision makers cannot help but wonder if Ada will retain the staying power needed for cost effective systems support. Whether to continue using the language or switch to an alternative is a very real question in light of Ada's current status.

To make an informed decision, the THAAD program commissioned a study to address questions relative to the long term viability and staying power of Ada. As part of the study, the program developed a wealth of information that could prove useful to other members of the Department of Defense (DoD) weapons system community relative to Ada's viability. This article shares the information by putting it into the public domain. Hopefully, others will use this information to make informed decisions when answering the question: Is Ada dead or alive?

Viability Assessment

Previous studies have reported that the viability of a programming language is a function of many variables. The accompanying tables 1, 2 and 3 were developed via a Delphi exercise by Reifer Consultants, Inc. (RCI) by having a group of software managers rank items per the criteria listed using a scale of one to five. In these tables Ada scores well in language evaluation when the richness of the language and its degree of standardization are taken into account. Its support for real-time development and reuse features provides facilities that users who work within the weapons systems community always deem important.

Table 1. *Language Evaluation (rating scale 1 to 5 [highest])*

Factors\Language	Ada	C/C++
Core Features • Strong Typing • Exception Handling	5	4
Degree of Standardization	5	4
Object-Oriented Support	5	5
Reuse Facilities	5	4
Real-Time Programming Support	5	3
Subtotal	25	20

Table 2. *Compiler/Tool Availability (rating scale 1 to 5 [highest])*

Factors\Language	Ada	C/C++
Optimizing compilers available for current host/target platforms	5	5
Optimizing compilers planned for future host/target platforms	3	5
Bindings to existing systems software available (POSIX, Windows 98, etc.)	5	5
Bindings to future systems software available (Linux, Windows 2000, etc.)	4	5
Bindings to GUIs and generators available (Fresco, etc.)	4	5
Rich libraries available (run-time, math, class, building blocks, etc.)	4	5
Compiler support tools available (syntax-directed editor, symbolic debugger, etc.)	4	5
Inexpensive visual toolset available	2	5
Subtotal	31	40

Table 3. *Education and Training Support (rating scale 1 to 5 [highest])*

Factors\Language	Ada	C/C++
Popularity	2	5
Public training offerings available	2	5
Literature and textbooks readily available	4	5
Consultants and subcontractors with skills in language available for hire	2	5
Contractor core competency with language and toolset	5	3
Subtotal	15	23

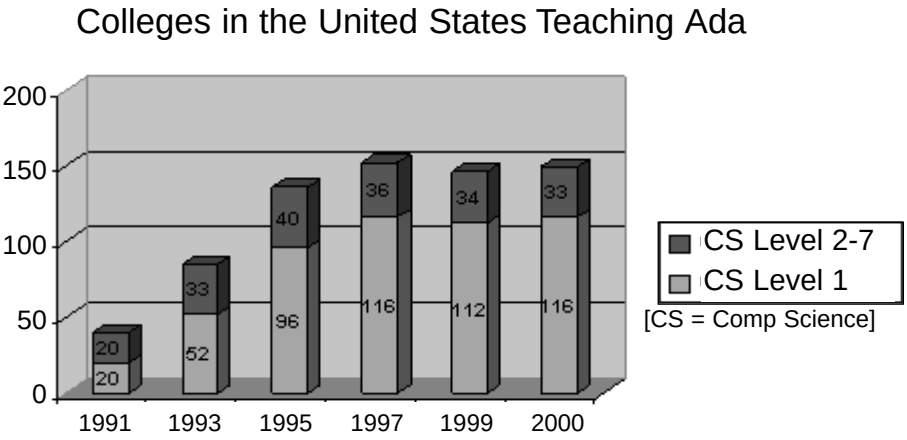
However, its lack of bindings, tools, libraries and inexpensive compilers has been a weakness that has caused users to shy away from selecting Ada in the past, especially for command and control projects. Most importantly, a lack of popularity and education and training shortfalls has detracted from Ada's use in new programs. Today's projects demand a language whose products are stimulated by market forces, not edicts. Such forces stimulate product developments along with language learning and use.

These factors can have a large impact on a program like THAAD. For example, the cost of a compiler for a new target machine is prohibitive if the project, not the market, has to stimulate product development. The cost to produce just a new code generator for THAAD would exceed $1 million.

Additionally, it would take 18 months to field this compiler; and the activity would be on the critical path due to impacts from potential schedule delays. As a result, practical concerns epitomized by the following trends also play an important part in assessing the viability of language alternatives:

• **Vendor/Tool Availability—**Figure 1 summarizes our findings relative to the availability of vendors, compilers and tools. This chart and Figures 2 and 3 were developed using public data available on Ada's Web page (www.adahome.com) by RCI for the THAAD program office. As the figures illustrate, the number of vendors generating Ada products has been cut in half since 1994. Of course, some of the firms that disappeared were acquired. Others just went out of business. In addition, the number of compilers has decreased dramatically as users of Ada 83 have migrated to Ada 95.

On the upside, tools supplied with the compilation systems (debuggers, editors, etc.) that have survived are rich and capable, especially those that are part of a Multi-Language Support System (MLSS). But the cost of these compilers and tools is high compared with the alternatives. In addition, their availability for new platforms is questionable. To ensure options, the program would have to fund the compiler developments and maintenance. As shown, this alternative adds cost to the program, and because compilers are on the critical path, increases programmatic risks.



- Number of colleges teaching Ada peaks in the year (1997) when the government mandate (subsidy) stops.
- Trend from 1997 is relatively flat – academia is slow to change.

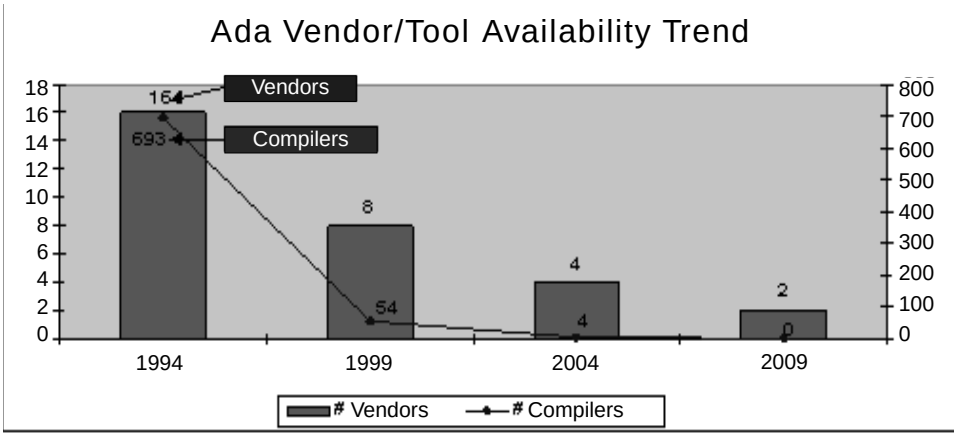
Figure 2 Trends in Academia

• **New Starts/Language Use Trends**—Most vendors interviewed agreed that Ada would continue like Jovial and other DoD programming languages as a niche market. Most of their business was concentrated in maintenance. Very few new projects were selecting Ada. The reasons for this lack of popularity are highlighted in Tables 1 through 3 above. Our primary concern is that without a large market to fuel future compiler and tool developments, firms will rely on projects like THAAD to fund innovations and compilers for new platforms and targets. The costs for this are prohibitive relative to available options.

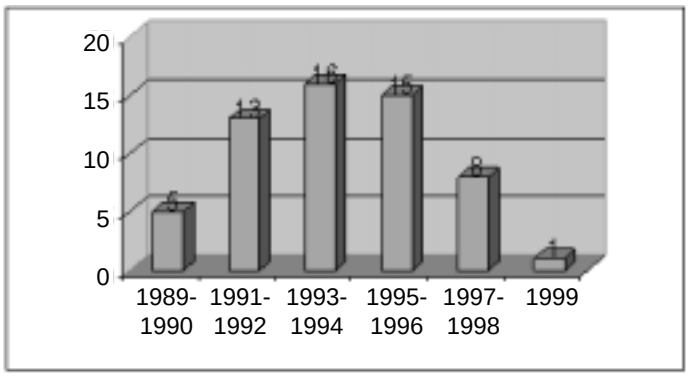
• **European Use Trends**—THAAD kept hearing from Ada advocates that development was stronger in Europe. In response, our team surveyed the overseas marketplace to see if things were any different outside of the United States. This survey confirmed that the European marketplace mirrors the U.S market. Like the United States, there were few new starts for Ada efforts in Europe. Similarly, most Ada efforts identified in the European weapons systems community were focused on maintenance and upgrades. Again, as in the United States, the government Ada supporters were neither funding R&D nor urging their contractors to use the language.

• **Academic Trends**—Figure 2 summarizes our findings relative to Ada's academic trends. As this figure illustrates, Ada use by colleges and universities seems to have peaked in 1997. While Ada is recognized as an excellent teaching lan-

Figure 1. *Tool/Vendor Availability*



- From 1994 to 1999, vendors cut to half (16 to 8), compilers to 8%.
- Continuing trend indicates no pure Ada compilers by 2009.



- 1999 book was not new, but a revision of a 1997 work.
- Trend indicates that interest in Ada is diminishing rapidly.

Figure 3 Publication Trends

guage for introduction to computer science, its use trend from 1997 to present is relatively flat. In addition, the number of Ada seminars offered by firms specializing in educating and training professionals working in the field has fallen off sharply. If these trends continue, it may become increasingly difficult to find programmers skilled in using Ada programming language to staff projects in the near future.

• **Publication Trends**—To indicate popularity, the THAAD team surveyed publications about Ada. Figure 3 summarizes the results, which indicate that interest in Ada is diminishing rapidly. This trend confirms that Ada is increasingly becoming a niche market inherently characterized by high costs, low demand, and lack of innovation; i.e., limited competition forces a degree of complacency.

As a final indicator of popularity, the THAAD team investigated conferences and professional publications. The decline of the Tri-Ada conference, seeming lack of interest for Ada at conferences such as the Software Technology Conference, and diminishing number of sponsors for Ada Letters does not paint a good picture for the future staying power of the language.

Productivity Assessment

Ada supporters would argue that its cost-benefits alleviate these and other concerns. However, the productivity data that is summarized in Table 4 shows that Ada no longer has an edge over other object-oriented languages like C/C++ within the weapons systems domain. RCI developed the information in this table by analyzing cost and productivity data for more than 1,500 projects within their historical databases. The trends indicate that the productivity gap between languages has narrowed, and the competition has caught up with Ada (i.e., see [2] for a 1995 snapshot of the RCI databases).

Table 4. *Cost Per Delivered SLOC by Language/Application Domain*

Application Domain	Ada83	Ada95	C/C++	3GL	Norm
Command and control	70	*	50	100	75
Information systems	25	25	20	35	30
Telecommunications	50	35	40	80	60
Weapons – Airborne	150	125	125	225	175
Weapons – Missile	150	*	*	250	200
Weapons – Spaceborne	150	*	150	200	175
Weapons – Ground	75	75	50	90	75
* Not enough data available					

Conclusions

The results summarized so far should not surprise anyone. Based upon the evaluation factors and identified trends, the verdict on Ada use is in: Lack of DoD institutional support and popularity has weakened its position relative to the competition. Not surprisingly vendors report only one in 10 projects within the weapons system community seem to be selecting Ada for new developments.

The issues involved are pervasive even when you have millions of lines of code to support while looking at developing/upgrading code for new platforms. While the Ada language provides superior support for weapons systems development, the investments needed to continue and sustain its use are large, and for the most part, not budgeted. Neither are the costs needed to convert millions of lines of code from Ada to C/C++.

While the THAAD project supports Ada's continued use, it must do what is in the best interests of the project. However, blanket approval to change to another language is in nobody's best interests. In response, THAAD has elected to permit its contractors to change to another language only when it makes both economic and technical sense. They must justify the conversion technically and in terms of the lifecycle costs before being given permission to change. In addition, they must also develop a transition plan that details how the transition will take place as part of the approval process. Then, projects like THAAD can figure out how to manage the transition and amortize the costs as part of an existing budget.

The approach THAAD has taken is consistent with current DoD policy regarding Ada, which calls for doing what makes sense in the long term for the program. THAAD recommends that other programs critically examine their situation before abandoning Ada because of its technical strengths as a real-time programming language.

THAAD is also investigating using MLSS. Such tool systems permit the vendors to reuse their existing language front-ends (syntax analyzers, etc.) with common back-end tools (code generators, editors, debuggers, etc.). This reduces problems associated with learning different toolsets and increases availability of bindings, tools, and libraries. Yet caution must be exercised to select compilers that enforce and implement published language standards in keeping with required real-time, safety critical systems.

In conclusion, Ada is not dead. It is alive and providing quality support to programs like THAAD. However, its future is not assured. Trends indicate that Ada is following the direction of Jovial and other DoD programming languages. In response, the project must continuously address the risks, and do what makes economic sense for the program. Because others will probably elect to follow suit, we have put the results of our study in the public domain.◆

References

1. U.S. Air Force, *Ada and C++: A Business Case Analysis*, 1991.
2. Donald J. Reifer, Quantifying the Debate: Ada versus C++, CROSSTALK, July 1996.

About the Authors

Donald J. Reifer is a consultant specializing in change management at Reifer Consultants, Inc. in Torrance, Calif. He has more than 30 years of experience managing large software projects and putting software technology to work in Fortune 500 firms. From 1993 to 1995, he was chief of the Ada Joint Project Office, technical advisor to the Center for Software, and chief of the Departmetn of Defense Software Reuse Initiative under an Intergovernmental Personnel Act assignment with the Defense Information Systems Agency. Reifer currently helps clients insert product line and component-based software engineering technologies into their software operations.

Reifer Consultants Inc.
P.O. Box 4046, Torrance, Calif. 90510
E-mail: d.reifer@ieee.org
Voice: 310-530-4493

Jeff Craver is the THAAD System Software Engineering division chief. He has more than eight years experience in software acquisition and development process improvement of Department of Defense systems.

U.S. Army Space & Missile Defense Command
SFAE-AMD-THA-W-SW, P.O. Box 1500
Huntsville, Ala. 35807
E-mail: Jeff.craver@thaad.army.mil
Voice: 256-955-1828

Mike Ellis works for Dynetics Inc. as the System Software Engineering branch chief. He has more than 24 years experience in large system software development, quality assurance, and test.

Dynetics, Inc.
990 Explorer Blvd.
Huntsville, Ala. 35806
E-mail: mike.ellis@dynetics.com
Voice: 256-964-4614

Dan Strickland works for Dynetics Inc. as a software engineer. He has specialized in the software metrics and software cost estimation fields of software engineering.

Dynetics Inc.
990 Explorer Blvd.
Huntsville, Ala. 35806
E-mail: daniel.strickland@dynetics.com
Voice: 256-964-4619

Ed. Note:

Ada has been surrounded by controversy almost since its inception. In this issue we offer one perspective on the current state of Ada and how this affects technology decisions for weapons systems. An upcoming issue will provide an opposing point of view … stay tuned.

Project Management (PM) Web Sites

www.pmforum.org **Project Management World Today**
This is an on-line publication that contains notices, reports, news and information related to project management from around the world. Each issue features editorials and presentations by some of the world's leading project management experts, on leading-edge issues and concepts.

www.ipma.ch **International Project Management Association**
The IPMA is a nonprofit organization based in the U.K. It promotes PM through its global membership network of national PM associations. Currently IPMA comprises 28 national associations representing approximately 20,000 members primarily in Europe but also in Africa and Asia. Moreover, institutional contacts to National Associations in North America, Australia and South Africa ensure a true global dimension to the work of IPMA. The site features listings of conferences, seminars, research, global forum, training courses and quarterly newsletter.

www.infogoal.com/pmc/pmchome.htm **Project Management Center**
This site is dedicated to those with an investment in PM. The site brings information together to cut back on Web cruising and provides practical help. It offers the latest news, articles, software, case studies, links, etc.

www.pmboulevard.com/home.jsp **PMBoulevard**
This PM Community connects you to others facing the same PM challenges you do every day. You can share ideas, compare notes, ask for help, find out the latest news, purchase materials and access other relevant project management sites. There is a calendar of events, a PM Bulletin Board that gives you access to project managers everywhere. Guests can view the ongoing discussion. Members and subscribers can take part in the action by posting messages and replies. It also features a news link, bookstore and PM links.

www.projectzone.com **ProjectZone**
The ProjectZone is a community of technology project leaders discovering, learning, inventing, and teaching each other better ways to lead and manage teams and projects. The material on this site is written by project leaders for project leaders as a volunteer effort, and reflects their experiences and opinions. You will find a variety of different, perhaps even conflicting, points of view here. The site is divided into four distinct zones—strategy, people, structure, and process. Articles on project leadership and management topics can be found in each zone.

www.newgrange.org **NewGrange Center for Project Management**
The NewGrange Center for Project Management was started in early 1997 as a nonprofit professional organization. Its mission is to further the discussion of project management as a professional discipline. The focus is a hands-on, practical approach to project management to determine what really works, why it works, and how to replicate it consistently.

www.allpm.com **The Project Manager's Resource Center**
ALLPM is a site for Information Technology (IT) project management information and resources. It is a clearinghouse of discussion forums, resource links, conference and seminar listings, news releases, contract awards and more.

www.project-manager.com **Project-Manager**
This site is an online guide for anyone who must plan, implement and complete a commercial project. Project-Manager is a learning experience. Here you can get professional advice. Upgrade personal skills. Exchange ideas. Above all, get your project up and running on time and within budget. The site also has links to various Product Depots where you may source equipment, machinery and supplies.