

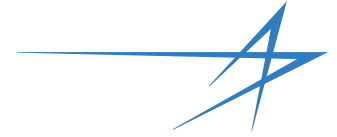
Can Ada Stand Up to the Challenges of C/C++ and Java?

Pam Thompson

***Lockheed Martin
Aeronautics Company***

November 16, 2004

Presentation Objective



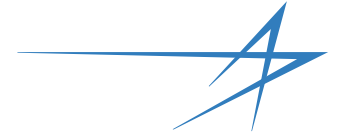
- **Share some technical and business facts on Ada**
- **Key Topics of Most Interest to Industry, Corporate and Customer:**
 - **Technical Suitability**
 - Software Intensive Systems developed as Real Time Embedded Systems
 - Safety Critical Design Solution Applications
 - Language Requirements and Characteristics
 - Compatibility with Modeling Languages
 - **Obsolescence**
 - Certified Compiler availability
 - Toolset support (including cost and availability)
 - Critical mass of Ada-specific Businesses developing new applications
 - **Staffing/Experience**
 - Staffing attraction and retention
 - Interest in Ada as a Niche market

Topic Research Approach



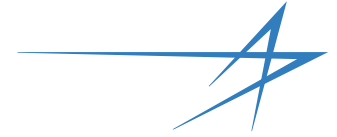
- **Review of several existing Lockheed Martin companies where Ada is currently in use as part of a Real Time Embedded System**
- **Discussions held with LM Aeronautics' Technical Fellow for Software System Safety, Barry Hendrix**
- **Review of Three LM trade studies on aspects of Ada**
- **Review of Related Documentation sited in Several Papers and Presentations from Internal and External Sources**
- **Discussions Held with On Board Software, a supplier of choice for the US Air Force as part of their maintenance strategy for older, unsupported avionics systems and software where the Ada and Jovial language was used**

Findings



- **Review of several existing Lockheed Martin companies where Ada is currently in use:**
 - **Received 26 opinions and views based on program experiences in response to questions requesting their opinion of the future of Ada from a Industry, Corporate and Customer perspective. Overall results:**
 - ***52% of the technical/software engineering respondents recommend considering that NEW projects select a language other than Ada (C, C++, Eiffel, Java) due to staffing and tool availability issues.***
 - ***48% believe Ada is the best choice for the development of long-lifetime, high integrity, real time systems.***
 - **Received much feedback from the business management community about the high cost of Green Hills and Wind River product costs versus more cost effective products like AdaCast**

Findings (Continued)



- **Discussions held with LM Air Traffic Management Software technical experts:**
 - **Ada remains the best choice for FAA systems due to the high integrity nature of the system and the amount of legacy code**
 - **C and C++ are not as safe as Ada, and therefore aren't adequate replacements without additional tool development and coding guidelines**
 - **Good programmers learn Ada quickly – Ada experience not a requirement**
 - **No evidence of Ada tools/support going away in the near term (niche market)**
 - **Worthwhile to pursue some of the real-time Java work going on and to explore transitions as appropriate, while maintaining existing base**
- **LM Aero History with Ada**
 - **No known staffing ramp up issues where the decision was made to use Ada – Software Engineers with Ada skill sets are available**
 - **Software Engineering skills far outweigh the language skills when assessing staff**
 - **Risk of outsourcing Ada projects to other countries very low**

Findings

LM Aero Tech Fellow Perspectives



- **LM Aero Tech Fellow for Software System Safety. Barry Hendrix, was asked to provide his perspective on Ada and its future.**
- **Barry referenced a paper presented at the 22nd International System Safety Conference – 2004 presented by A. J. Kornecki and J. Erwin, both of Embry Riddle Aeronautical University, Daytona Beach, FL titled “Characteristics of Safety Critical Software.”**
- **Excerpts from the paper as they relate to Ada are as follows:**
 - **The selection of a programming language may impact the programming errors ranging from mistyping variable name to misunderstanding and thus incorrectly encoding an algorithm....Hatton (L. Hatton, Safer C, McGraw-Hill, 1995) quotes 195 items of the C language that the ISO standard committee never agreed upon, leaving the interpretation to the compiler writers.**
 - **The expressiveness and style of selected programming language may impact the programmer’s ability to avoid mistakes. Specific language may help (a) programmer to deal with failures or unexpected inputs, and to structure and test whole programs and modules. No language is ideal in all of these respects.**
 - **The developers of Boeing 777 software systems have enjoyed the Ada’s portability, code reuse features, built-in safety features that help reduce development time, expense and concern for debugging the software.**

Findings

LM Aero Tech Fellow Perspectives



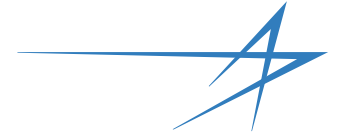
- **Basic language characteristics to be considered for safety critical applications:**

Language Characteristics	Rationale
Strong typing	Help reduce errors in programs at compile-time, enhances the integrity and security
No side effects	Prevent programs to behave in an ambiguous, or possibly unpredictable way
Modularity and structure	Assure that the complexity of software becomes manageable
Formal semantics	Help to produce quality software, often cost-effectively
Well-understood semantics and syntax	Easy to adopt and to implement
Domain specific	Include robust mechanisms for controlling memory, I/O devices or other hardware
Concurrency	Language-level support for multitasking or multithreading, control over scheduling policy and straightforward communication and synchronization mechanism
Predictability	Functional and temporal behavior can be predicted
Run-time environments	Trusted/certified translators are used
Handling errors	Graceful degradation and recovery mechanism outweigh overheads and possible unpredictable behaviors
Model of mathematics	Integer and floating point arithmetic
User documentation	Improve program readability and maintainability
Enumeration types	Help reduce errors
Coding style	Reduce gap between well-established software engineering principles and the actual practice of programming
Abstraction or information hiding	Decrease software complexity and support modularization
User specified assertions	Supports analyses capability and design by contract thus defect reduction
Expressive power	Ability to solve problems and implement all algorithms
Language subset	Enforce the rules/characteristics and simplify the code
Certified analysis tools	Support development, check for errors, such as race conditions and deadlocks
Interface	Facilitates development
Code initialization	Improve the efficiency of programs
Portability	Reduce analyses when porting to another platform

NOTE: A language should not simply be evaluated on how “safe” it is, but on how safe it can be made.

Findings (cont)

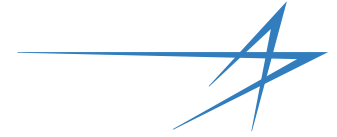
LM Space Systems Trade Study



- 1997 Study by LM Space Systems recommended use of Ada for these reasons:
 - Increased Quality per Dollar Spent
 - Commercial Industry and DoD have different objectives
 - ***DoD Perspective – buy small quantities of a weapons platform, limited market, long-term planning precludes quick reaction, failures cost lives***
 - ***Commercial Sector – sell many, quick reaction to market forces, large profit potential, failures cost money, but rarely lives***

Findings (cont)

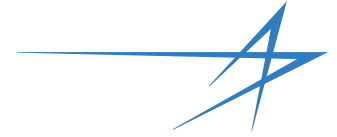
LM Space Systems Trade Study



- Reliability and Safety of Ada over C++: Ada error rate and cost-to-fix rates are lower than C and C++
 - Ada market appears stable and increasing (USA, Europe, Far East)
 - Software Development vs Software Maintenance
 - Look at Systems from total lifecycle perspective
 - 60-80+% of the lifecycle costs of software occur in maintenance
 - Arguments based on lower development costs total only 20-40% of the story
 - Study shows Maintenance cost advantage of Ada over other languages (Source: Ada Technology: Current Status and Cost Impact, IEEE Vol. 79 No. 1)

Findings (cont)

LM Space Systems Trade Study



- 1997 Study by LM Space Systems recommended use of Ada for these reasons:

Availability of Ada programmers

- If you pay them, they will come. Niche technology, good programmers cost top dollar – regardless of language.
 - Many programmers are interesting in working on projects that will not migrate overseas
- Summary:
- Ada is technically superior
 - Demonstrably lower in cost through entire lifecycle
 - Successfully used today and growing for products with long life times

Findings (cont)

LM JSF Trade Study



- **Review of Key JSF Trade Study Drivers on the topic**
 - **2002 JSF (Safety Critical Software) trade study recommended use of C;**
 - **For JSF Mission Systems Software (not safety critical), C++ was chosen.**
 - **Original decision that only 2 of the 38 Vehicle Systems CSCIs (4 %) are planning to use Ada is always a candidate to be revisited. The basis for the selection at that time was:**
 - JSF is a **New** weapon system with a 30-40 year future.
 - Program requires a large hiring effort. Major concern -- getting new people, especially college hires, who were willing to code in Ada.
 - Plan is to use modeling tools which can automatically generate code. Most of those tools generate C/C++.
 - Most language experts agreed that Ada was best from a technical standpoint, but the points raised above were more of a concern.
 - Key research came from an article in Crosstalk, “Is Ada Dead or Alive Within the Weapons System World?”, Donald Reifer, et al – see references. Information tables follow.

Findings (cont)

Crosstalk Data used in JSF Trade Study



- Significantly greater support for C/C++ according to Crosstalk* article which described a study done for the Theater High Altitude Area Defense (THAAD) program:

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

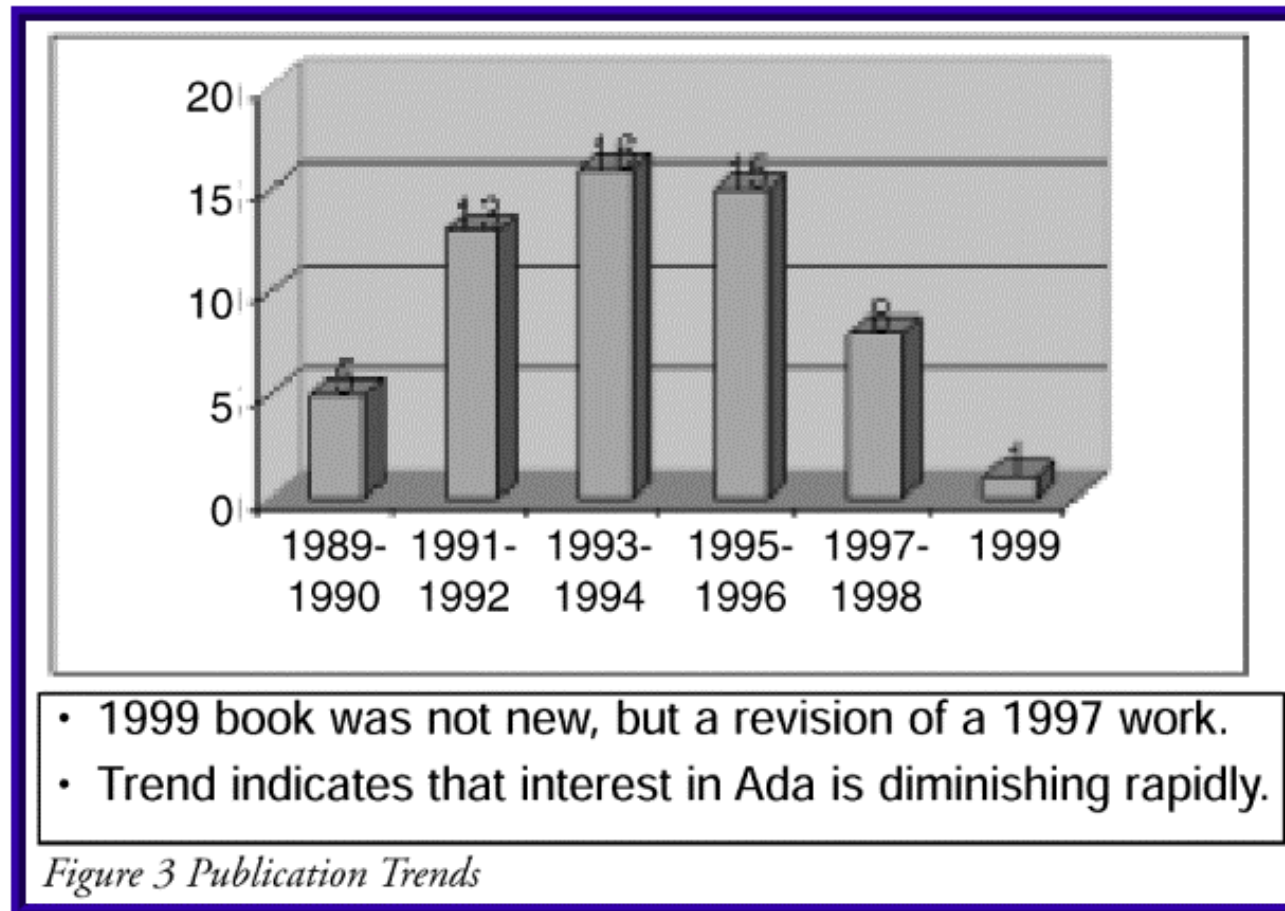
- Reference: “Is Ada Dead or Alive Within the Weapons System World?”, Donald Reifer, Jeff Craver, Mike Ellis, and Dan Strickland, Crosstalk, The Journal of Defense Software Engineering, December 2000, Vol 13, No 12. On-line at: <http://www.stsc.hill.af.mil/CrossTalk/2000/dec/reifer.asp>

Findings (cont)

Cross Talk Data Used in JSF Trade Study



- Lack of interest in Ada conclusion made because a lack of publications on Ada were noted, according to Crosstalk article*



•Reference: “Is Ada Dead or Alive Within the Weapons System World?”, Donald Reifer, et al

Findings (cont)

Cross Talk Data Used in JSF Trade Study



- **C/C++ offer higher compiler/tool availability according to Crosstalk article:**

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

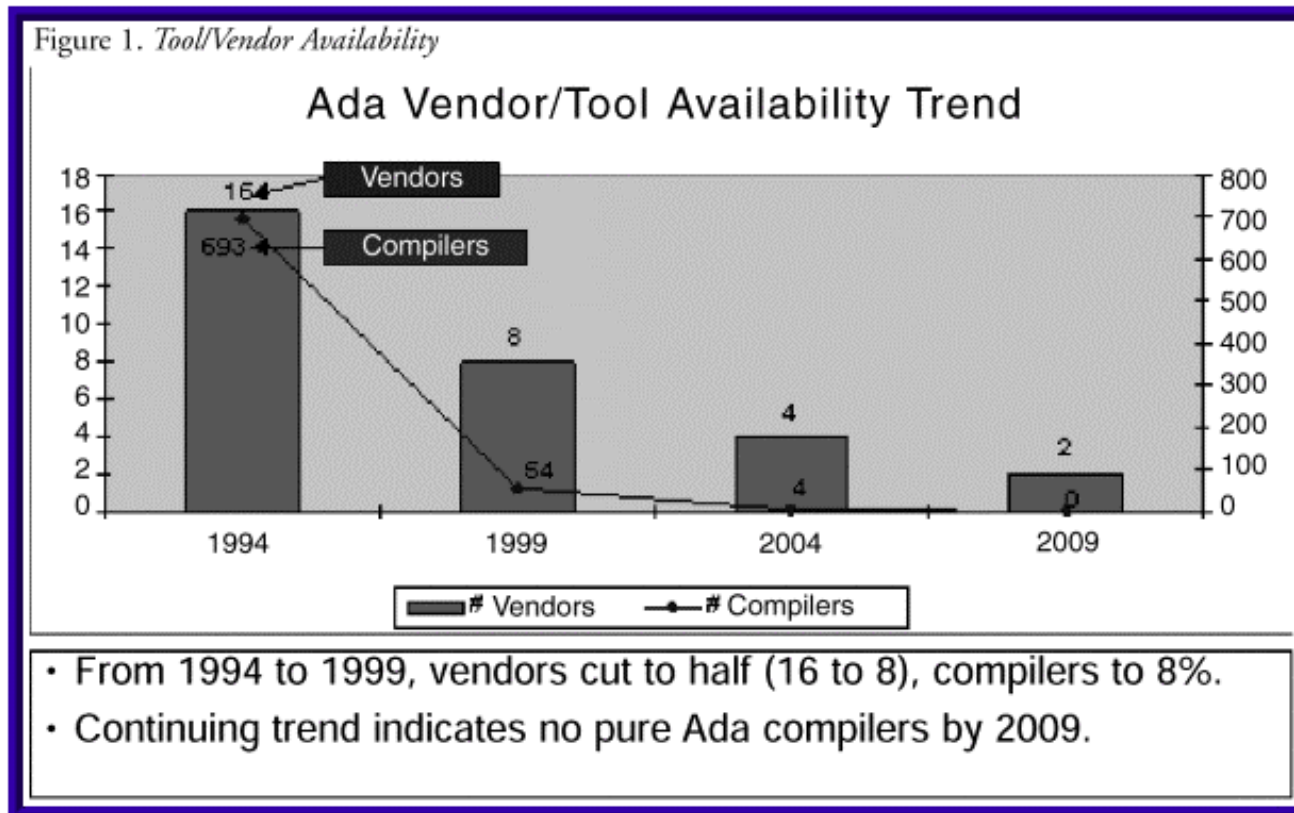
- Reference: “Is Ada Dead or Alive Within the Weapons System World?”, Donald Reifer, et al

Findings (cont)

Cross Talk Data Used in JSF Trade Study



- **Compiler Availability**



•Reference: “Is Ada Dead or Alive Within the Weapons System World?”, Donald Reifer, Jeff Craver, Mike Ellis, and Dan Strickland, Crosstalk, The Journal of Defense Software Engineering, December 2000, Vol 13, No 12. On-line at: <http://www.stsc.hill.af.mil/CrossTalk/2000/dec/reifer.asp>

•Note: LM Air Traffic Management and my research show that the current level of vendor/tool availability appears to be stable.

Findings

Lockheed Martin Tactical Systems – Eagan, MN Language Study



- **In April 2003 LM Tactical Systems issued the results from an internal Ada Lean Team Language study. The report evaluated five general-purpose programming languages and considered emerging technologies like Model Driven Architecture (MDA).**
- **LM has several large software systems implemented mainly in Ada and questions have been raised about possible rationale and approaches to migrating these systems to other languages.**
- **Summary: Java currently offers the strongest support for distributed systems. C++ with a selected middleware approach or networking facilities is less straight forward, but somewhat stronger than C or Ada.**

Findings

Lockheed Martin Tactical Systems – Eagan, MN Language Study

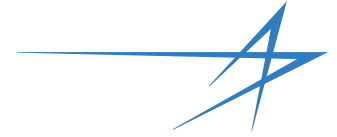


Language Evaluation Spreadsheet						
Criterion	Weight	Scores for Each Language				
		Ada	C	C++	C#	Java
Portability		4	4	3	1	5
Tools availability		3	5	5	3	5
Component availability		3	4	4	4	5
Distributed systems support		2	2	3	4	5
Real-time support		5	3	3	1	2
Runtime performance		4	5	5	2	2
Development productivity		5	3	4	5	5
Defect avoidance		5	1	1	3	4
External expertise		2	5	4	3	5
Internal expertise						
Normalized Overall Weighted Scores:		4.00	4.00	3.00	1.00	5.00

Many different language evaluation criteria are possible, and recent articles provide lists of 15 to 20 criteria. For the LM Tactical Systems study, a shorter list of somewhat broader criteria ranging from technical properties of the languages to social and economic factors was derived:

- Portability
- Tools Availability
- Component Availability
- Distributed Systems Support
- Real Time Support
- Runtime performance
- Development Productivity
- Defect Avoidance
- External Expertise
- Internal Expertise

Findings (cont)



Interview Results from Key Ada vendors (Barnes, Brosgol)*

- **John Barnes – one of the original designers of Ada, author of several Ada textbooks, past chair of Ada Europe, and a member of the working group defining Ada 2005.**

Comments:

- **Ada is still the most technically suitable language for the development of high integrity systems.**
- **US perception of the status of Ada tends to be driven by its use in the US defense industry – where its decline is most marked.**
- **Use in Europe and Asia on new projects is stable or increasing.**
- **US civil aeronautics/aerospace applications (e.g. Boeing 777) are still using Ada as the primary implementation language.**
- **Ada 2005 revision is being actively developed. This will not be a major language overhaul on the scale of Ada83->Ada95 revision that added full object oriented features, but will make small but significant improvements in a number of areas including tasking and packages.**

* discussions held with Mark Dowson, LM Air Traffic Management Rockville

Findings (cont)



Discussions held with key Ada vendors (Barnes, Brosgol)*

- **Ben Brosgol, an original language designer, chair of SIDAda, and a senior member of the technical staff at Ada Core, an Ada compiler/tool vendor. Ben offers a rebuttal article to Reifer's article, "Ada in the 21st Century. More information to follow.**

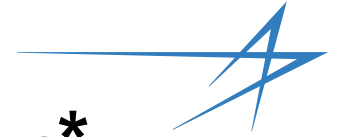
Comments:

- **Ada Core recently held its annual meeting. Growth prospects are good in the Ada market, their revenue increased 25% in 2001-2002.**
- **Barnes and Brosgol agree that the reduction in the number of Ada vendors to the current ~10 is a healthy sign of industry consolidation/stability.**
- **Tucker-Taft (ex-Intermetrics/Averstar) has just launched a new Ada start-up company.**
- **The GNAT compiler, which Ada Core market/support and we use, is open source, reducing dependence on a specific vendor.**
- **Boeing is writing all DO-178B level A (safety critical) software in Ada (including all 777 software).**
- **Ada is at the forefront for safety critical real time systems, although other languages are making an attempt to catch up.**

* discussions held with Mark Dowson, ATM Rockville

Findings (cont)

Discussions held with key Ada vendors (Barnes, Brosgol)*



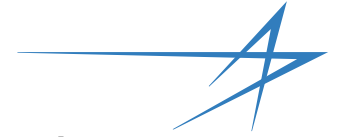
Comments (Ben Brosgol, cont):

- Two active efforts to add real time features to Java are Sun Microsystems and the J Consortium. Although a reference implementation of the Sun proposals is available, it will be some time before commercial implementations are available (although the 2009 JPL Mars Lander project is committed to real time Java).
- Mixed language systems make perfect sense (e.g. Ada for real time/high integrity system components and Java for GUI components. Ada offers good support for mixed language systems.
- C++ provides no support for real time.
- A DoD program (can't be named) converting an existing Ada system to C was cancelled at the point of \$150M overrun.
- The issue of Ada programmer availability is much exaggerated. Ada is easy to learn, for programmers with experience in other languages and a good software engineering background. He notes that his company has many young staff, including recent college hires.

* discussions held with Mark Dowson, ATM Rockville

Findings (cont)

Discussions held with key Ada vendors (Barnes, Brosgol)



Excerpt from Brosgol’s Article in Crosstalk, “Ada in the 21st Century”:

It is difficult to conduct comparative quantitative studies of programming languages in an objective manner, especially for large projects.

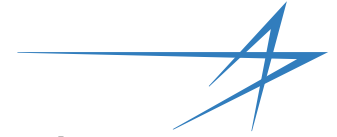
However, an impartial and thorough analysis [see references for “Comparing Development Costs of C and Ada”, Stephen Ziegler, Ph D.] documented a significant difference in productivity between Ada and C on the components of the Verdix VADS product line.

This development comprised roughly the same amount of code in the two languages.

Zeigler’s study considered all relevant factors (such as the effect of programmer skills) and concluded that Ada performed approximately twice as well as C (i.e., costs for Ada were half that for C).

Findings (cont)

Discussions held with key Ada vendors (Barnes, Brosgol)



Excerpt from Brosgol's Article in Crosstalk, "Ada in the 21st Century":

Some of the reasons cited:

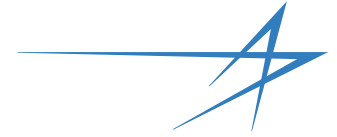
- 1. Ada's additional compile-time checking**
- 2. Its higher-level features,**
- 3. the Ada culture that encouraged up-front design**
- 4. the study observed that using C++ rather than C would not change the underlying result, since bug rates in C++ were higher than in C. It concluded:**

"Our data indicates that Ada has saved us millions of development dollars. For every development dollar, we could make a case for another three dollars for customer support, sales, marketing, and administration costs"

This result should not be surprising because Ada was specifically designed to save life-cycle costs through software engineering support. Quantitative data provides evidence that this goal has been met, even if percentage of improvement varies per project.

Findings (cont)

Discussions held with OnBoard Software



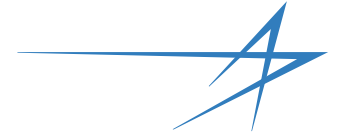
- OnBoard Software is a San Antonio, TX based business that provides engineering services and logistics support for design, development, production and maintenance of embedded software systems, data acquisition systems and information management systems.

Comments:

- Ada continues to exist because it is the most technically suitable language for the development and maintenance of high integrity systems with long shelf lives.
- OnBoard, as part of their college/new hire training process, provide formal instructor-led classes, On-the Job training and in-house training for both Ada and Jovial programming languages.
- OnBoard has done many Jovial code conversions to C++ and updates to Ada code for the United States Air Force.
- The ability of software engineers and programmers to learn and quickly be productive in either language has been demonstrated consistently.

* discussions held with Mark Dowson, ATM Rockville

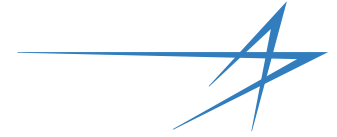
Summary



- **Lockheed Martin should continue to consider Ada where real time, embedded systems are being proposed and safety is an issue. Use of other languages, as appropriate, for related system components should be regarded as acceptable where there are technical or strong business environment reasons for their use.**
- **Lockheed Martin will continue to monitor technical support for Ada as it relates to Ada tools and platforms.**
- **Lockheed Martin will continue to follow events pertinent to the Java Real Time extensions.**

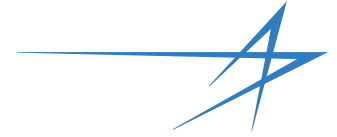
CONCLUSION: Yes, Ada can stand up to the challenges of C, C++ and Java when the desire for high quality real time embedded systems is the goal.

References



- ***Guidelines for Choosing A Computer Language: Support For The Visionary Organization*** PATRICIA K. LAWLIS, August 1997
<http://archive.adaic.com/docs/reports/lawlis/content.htm>
 - ***Is Ada Dead or Alive Within the Weapon Systems World***, Donald Reifer et al, CrossTalk, December 2000.
<http://www.stsc.hill.af.mil/crosstalk/2000/dec/reifer.asp>
 - ***Ada in the 21st Century***, Benjamin Brosgol, CrossTalk, March 2001
<http://www.stsc.hill.af.mil/crosstalk/2001/mar/brosgol.asp>
 - ***Comparing Development Costs of C and Ada***, Stephen F. Zeigler, Ph.D.,
<http://www.rational.com/products/whitepapers/337.jsp>
- Characteristics of Safety Critical software***, A. J. Kornecki, J. Erwin,
Proceedings of the 22nd International System Safety Conference (used with permission)
- ***For information on SIGAda conferences, see***
<http://www.acm.org/sigada/conf/sigada2004/>
 - ***For news of new Ada projects,*** <http://www.adaic.org/whatsnew.html> and <http://www.adapower.com> and ***Ada at Work,*** <http://adaic.org/atwork/>
 - ***For DOD guidance for program language selection, see news of new Ada projects,*** <http://sw-eng.falls-church.va.us/ajpofaq.html#guidance>

Reference Data



Time line - Introduction of Major Computer Languages

Fortran	1954 (Fortran 2000 scheduled for 2002)
Cobol	1960
Jovial	early 1960s
CMS-2	early 1970s
C	1972
Ada	1979
C++	1983
Java	early 1990s

Note: At the time of a major survey in 1995, it was found that over 14M SLoC of CMS-2 had been written for US military systems. (This is compared to 32.5M SLoC for C and 43M for Ada).