Ada
Past, Present, and Future

Robert Dewar
New York University
Ada Core Technologies
Apr 26th, 2004

History of Ada

- Time Frame: 1970's
- Problem
  - DoD spending billions on software
  - Having a heck of a time maintaining old software
  - Identified problem: too many languages
    - Over 450 languages in use
    - Many specialized and idiosyncratic
    - Single supplier and maintainer, no general knowledge base
- Solution
  - Standardize on one language
  - But what language?
  - Enter the HOLWG (Higher Order Language Working Group)

The HOLWG in Action

- Some unusual ideas put into action
- Involve wide variety of inputs and technical resources
  - Universities
  - Industry
  - Defense establishment
  - International standards and scientific groups (ISO, IFIP)
- Write down specifications for the language before design
  - Strawman/Woodman/Tinman/Ironman/Steelman
  - These are not language designs but rather requirements documents
  - Steelman is the final blue print for the new single language

Steelman in Action

- First step. Does any existing language meet specs?
  - Systematic effort looks at C, Pascal, COBOL etc
  - Answer: no, no existing language comes close enough
    - COBOL probably comes closest ☺
  - But note that C++, Java, C#, Eiffel not on the scene
- OK, Let's Make a New Language
- International Competition
  - Four entrants
  - Green (from Honeywell, Jean Ichbiah, based on LIS (sp?))
  - Red (from Intermetrics)
  - Blue (from Softech)
  - Yellow (from Wirth and Pascal crew)

The Competition

- The field is narrowed down
  - Blue, seemed too strange (but interesting, worth a look now)
  - Yellow, warmed over Pascal, not sufficiently responsive to Steelman
- Green vs Red
  - Relative merits discussed in many forums
  - For example, IFIP WG2.2
    - International working group on system programming languages
- And the winner …
  - A win for “Old Europe” ☺
  - Green is announced as the winner
  - And is renamed Ada

Ada Development

- Now there is a winner
- Subjected to intensive (and well funded) scrutiny
  - University research projects (e.g. at NYU)
  - Distinguished reviewers committee
  - Extensive tests and example programs by contractors
  - Formal test suite developed (ACVC tests)
  - An early version is completed Ada 79
- Finally, the result is the proposed Ada 83 standard
- ANSI standard issued in 1983
- First validated compiler a couple of months later
  - NYU Ada/Ed
  - Other commercial compilers appear soon after (ROLM)

Ada in the 80’s

- About a dozen Ada compiler companies appear
- System houses invest large amounts
- DoD mandates Ada for all mission critical programs
- Ada is in fact used for many military programs
  - But by no means all
  - The mandate was widely ignored from the start
- Ada does become the dominant weapons system PL
  - And competition is C, C++, Smalltalk etc
  - Not the 400 junk languages
  - So the mission is achieved at least partially

Languages Influencing One Another

- PL’s always borrow ideas from one another
- So many PL’s exist, that there are very few new ideas
- More like combinations of old ideas
- Ada drew from many sources
  - Elaboration, and modules from Algol-68
  - Tasking from theoretical work based on CSP
  - Typing system from Pascal
  - Etc
  - But NOT Simula 67!.

Languages Influencing One Another

- Many Language Ideas drawn from Ada
  - Ada-83 depended heavily on generics
  - C++ dependend heavily on OOP and dynamic dispatching
  - But C++ eventually adds templates
  - C++ also adds exceptions with a model similar to that of Ada
  - Java, originally designed as a safe language for embedded systems, follows a number of principles inspired by Ada
  - Borland Pascal borrows modules from Ada (which had borrowed them from Algol-68.
  - Not much new under the PL skies

Ada-83 Retrospective

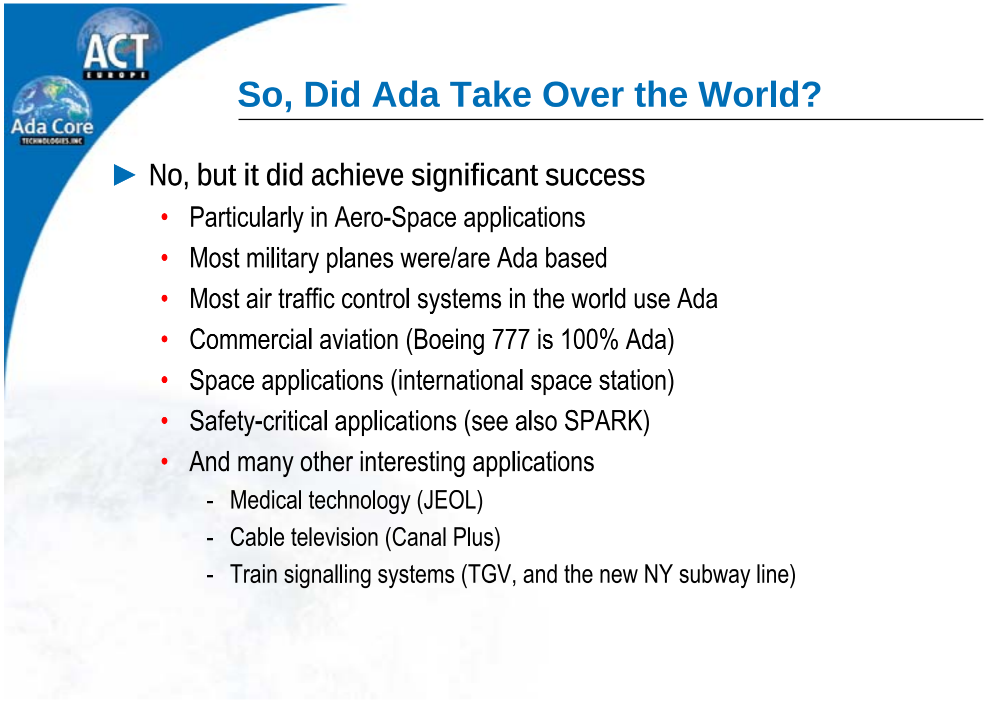
- Ada-83 proves remarkably effective
  - Improving reliability, by finding problems earlier
  - Encouraging a disiplined programming style
  - With a concentration on quality
- What are the most important conclusions
  - The package model is very attractive
  - Separation of declaration and implementation is a key point
  - Type system (and particularly separation of scalar types) is effective in preventing silly errors.
  - It's definitely easier to get programs running correctly first time
  - Ada programs are indeed easier to port
  - But …..

Avoiding Ada Hype!

- Ada programs are guaranteed to be portable
  - Nonsense, it is perfectly possible to write non-portable Ada code
  - And it is possible to write portable C (and even C++)
  - But it tends to be easier in practice in Ada
- Ada programs are guaranteed to be high quality
  - Nonsense, it is perfectly possible to write junk Ada code
  - And it is possible to write high quality C or C++
  - But Ada does tend to encourage and promote quality
- Ada programs are guaranteed to be easier to maintain
  - Nonsense, it is perfectly possible to write unmaintainable Ada code
  - And it is possible to write maintainable C and C++
  - But Ada does tend to promote a style that eases maintainability

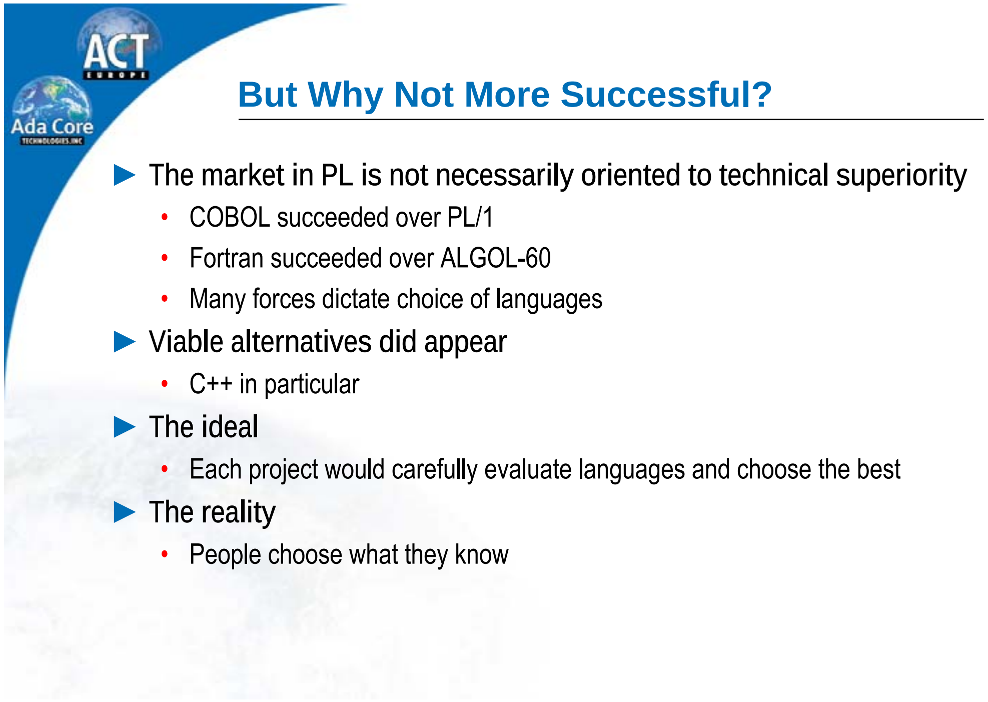
Ada 83: The Bottom Line

- A real step forward
  - Not some huge leap forward
  - But programming is enormously expensive
  - And life cycle maintenance of large programs even more so
  - Even a relatively small help can be very valuable
  - And many many cases have shown that Ada can provide that help



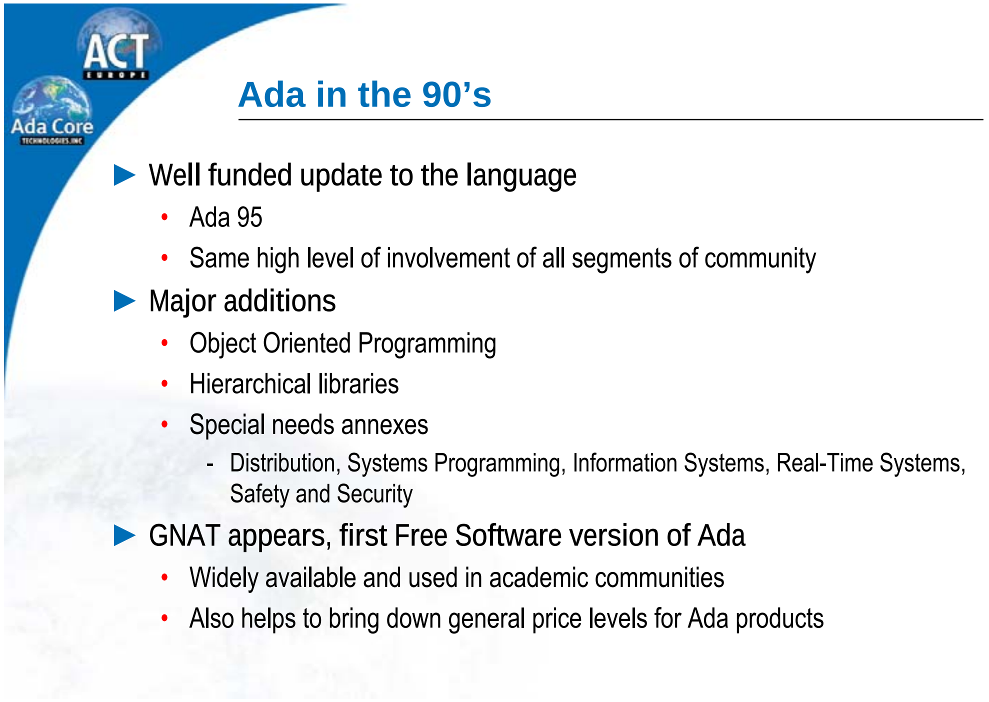
So, Did Ada Take Over the World?

- No, but it did achieve significant success
  - Particularly in Aero-Space applications
  - Most military planes were/are Ada based
  - Most air traffic control systems in the world use Ada
  - Commercial aviation (Boeing 777 is 100% Ada)
  - Space applications (international space station)
  - Safety-critical applications (see also SPARK)
  - And many other interesting applications
    - Medical technology (JEOL)
    - Cable television (Canal Plus)
    - Train signalling systems (TGV, and the new NY subway line)



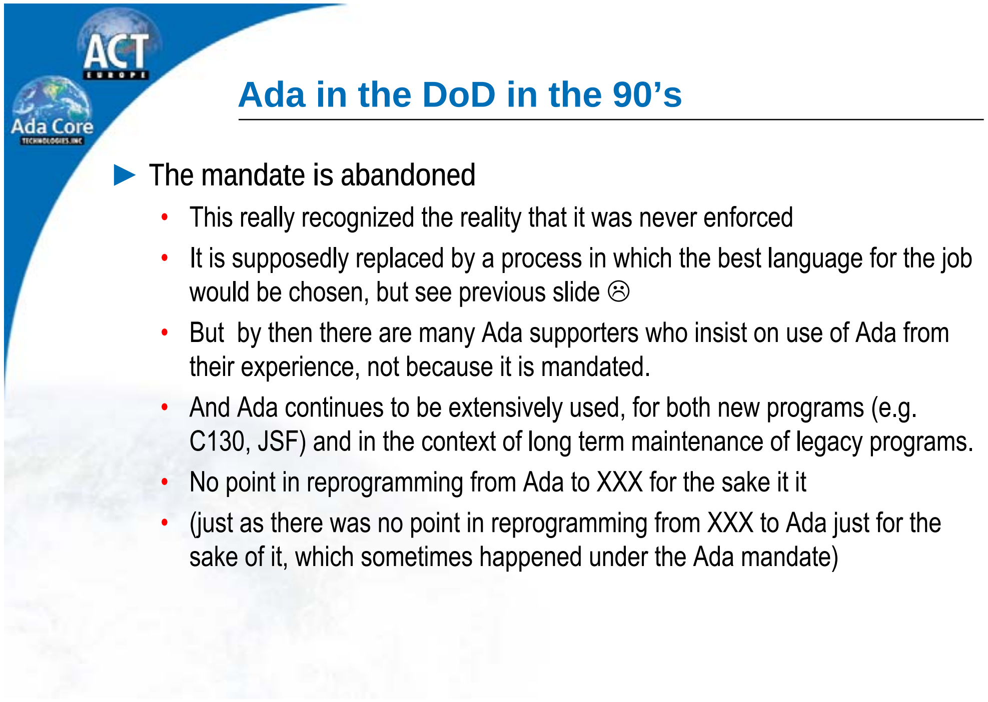
But Why Not More Successful?

- The market in PL is not necessarily oriented to technical superiority
  - COBOL succeeded over PL/1
  - Fortran succeeded over ALGOL-60
  - Many forces dictate choice of languages
- Viable alternatives did appear
  - C++ in particular
- The ideal
  - Each project would carefully evaluate languages and choose the best
- The reality
  - People choose what they know



Ada in the 90’s

- Well funded update to the language
  - Ada 95
  - Same high level of involvement of all segments of community
- Major additions
  - Object Oriented Programming
  - Hierarchical libraries
  - Special needs annexes
    - Distribution, Systems Programming, Information Systems, Real-Time Systems, Safety and Security
- GNAT appears, first Free Software version of Ada
  - Widely available and used in academic communities
  - Also helps to bring down general price levels for Ada products



Ada in the DoD in the 90’s

- The mandate is abandoned
  - This really recognized the reality that it was never enforced
  - It is supposedly replaced by a process in which the best language for the job would be chosen, but see previous slide ☹
  - But by then there are many Ada supporters who insist on use of Ada from their experience, not because it is mandated.
  - And Ada continues to be extensively used, for both new programs (e.g. C130, JSF) and in the context of long term maintenance of legacy programs.
  - No point in reprogramming from Ada to XXX for the sake it it
  - (just as there was no point in reprogramming from XXX to Ada just for the sake of it, which sometimes happened under the Ada mandate)

Ada in the 00's (how is that pronounced?)

- Ada continues to be used in important new programs
- Both military
  - For example, the C130
- Major companies renew commitments to Ada
  - Rockwell chooses Ada 95 as the main development language for AAMP
- And usage continues in commercial arena
  - Continued use on the space station (the Canadian space arm uses GNAT)
  - The Boeing Dream Liner (7E7) (GNAT selected as Ada technology, and all subcontractors automatically get the Ada development system).
  - The new Airbus Jumbo Jet uses Ada 95 (using GNAT)
  - Thalys (European high speed trains) using GNAT
  - Eurospace using GNAT for air traffic control systems in Europe
  - Canal-Plus continues to use VMS and Ada 95 (HP is porting VMS to Itanium, and has commissioned an Ada 95 GNAT compiler)

Ada 2005?

- A new version of the language is being brewed
- Many new features already approved
  - Signatures (inspired by Java)
  - Circular dependencies (inspired by C++)
  - General cleanups
- Many of these new features already in GNAT (use –gnatX)
- New standard expected to be approved in 2005
- An incremental release, not an earthquake
- Strictly upwards compatible

So Will Ada Take Over the World?

- No, it won't
- But it will be around for a long time
- Thought:
  - BMW has only a tiny fraction of the world's car production
  - But is hardly a failure ☺
- Ada will have a significant niche
  - Large critical applications
  - And that's a nice niche to be in
  - Since systems grow and emphasis on safety grows
- There's still hot competition in the Ada world
  - Several vendors competing hard
  - A particularly effective example of Free Software competing