Revisiting the Impact of the Ada Programming Language

Kenneth Magel, North Dakota State University

This installment of From the Archives reflects on a 1978 Computer article and the impact of the US Department of Defense’s effort to standardize software development projects around a single programming language.

FROM THE EDITOR

As part of our 50th anniversary celebration, this special feature revisits influential Computer articles from the past. This month’s author reflects on the US Department of Defense’s effort to reduce development and maintenance costs and improve the quality of software for embedded computer systems by standardizing a programming language. —Ron Vetter, Editor in Chief Emeritus

In 1978, David A. Fisher of the Institute for Defense Analyses wrote an article for Computer that described the four sets of requirements documents that were created successively by academicians and industrial software development professionals under the leadership of a US Department of Defense (DoD) committee. The article (“DoD’s Common Programming Language Effort,” Computer, vol. 11, no. 3, 1978, pp. 24–33) came at a critical time for the DoD, which was seeking to standardize its many embedded software development projects around a single existing or new programming language.

After the article’s publication, a request for proposals was released and four contractors were hired to develop their proposals. In 1979, the winning proposal was selected and named Ada after Ada Lovelace, known for her work on Charles Babbage’s Analytical Engine in the early 1840s and widely considered the first computer programmer. Ada was designed for large, long-lived applications—and embedded systems in particular—where reliability and efficiency are essential. It became an ANSI standard in 1983 and an ISO standard in 1987; in 1995, it became the first ISO standard object-oriented language.
Ada introduced many features—such as packages, scalar ranges, generic templates, exceptions, high-level concurrency, strong typing, and real-time support—that weren’t used in mainstream languages at the time. It’s used around the world in high-integrity, safety-critical domains including commercial and military aircraft avionics, air traffic control, transportation and financial systems, and medical devices. Ada has had four major impacts:

› It influenced most advances in programming language design and implementation from the mid-1970s to the first decade of the 21st century. Developing Ada through four iterations of requirements documents—which the 1978 Computer article details—and four international standards led to extensive public discussions and explorations of the pros and cons of various alternatives for object orientation, programming in the large, real-time programming, and concurrency.

› Its initial success led to several other important DoD standardization efforts in software development methodologies.

› It replaced hundreds of other programming languages in use by the DoD and its counterparts in other nations. By 1996, the DoD was using 37 languages, compared to 450 when the DoD common programming language effort began.

› It popularized the development of tools in an existing language rather than inventing new languages for every major application, as was common before Ada.

Outside of its use in safety-critical applications, Ada has declined in popularity in recent years. The 2016 IEEE Spectrum ranking of programming languages based on relative popularity placed Ada 40th among all languages it highlighted.\(^1\) One major reason for this is that Ada has almost completely ignored the increasing popularity of PC applications over the last 35 years. Ada was developed during a time of major changes in hardware, software, and the use of computers. When Microsoft was founded in 1975, it developed and sold BASIC language interpreters for the new PCs, such as the Altair 8800. The IBM Personal Computer was introduced in 1981, followed by Apple’s Macintosh in 1984 (the Apple Lisa was introduced a year earlier but was not a commercial success), the first mass-produced PC with a GUI. By the end of the decade, PCs and GUIs were widespread and had gained significant inroads into industry, education, and government.

In addition, Ada wasn’t a significant factor in the growth of the Internet and the introduction of the cloud, and is too complex to be a factor in the mobile revolution.

Ada remains an important programming language, as it’s still widely used in military and infrastructure applications, and is an example of how to make a language relatively secure while providing high-level features to utilize the specific attributes of the target hardware and OS. The development process detailed in the 1978 Computer article still describes best practices for using multiple drafts and detailed comments from a wide range of potential users to develop requirements for a complex product such as Ada.\(^1\)

REFERENCES


KENNETH MAGEL is a professor in the Computer Science Department at North Dakota State University. Contact him at kenneth.magel@ndsu.edu.