

ATTACHMENT F

Ada—new language that will impact commercial users

by Kenneth C. Shumate,
CCP, CDP

No, Ada is not pronounced "A," "D," "A."

It is not an acronym. It is properly spelled with only the initial letter capitalized, and is pronounced as a name.

In fact, the newest programming language of the US Department of Defense (DoD) is named in honor of Augusta Ada Byron (1815-1852), Countess of Lovelace and daughter of the poet Lord Byron. She is considered to be the first computer programmer, having worked on the Difference Engine of Charles Babbage in the early 1800s.

Although the language has caused quite a stir in engineering circles and in DoD-oriented organizations, it has not yet been very visible in other data processing environments.

It will be.

The language has been touted as virtually a panacea—"Ada, the Ultimate Language." That's pretty strong stuff. Ada may not be the ultimate language, but it is going to be important in commercial applications as well as DoD embedded computer systems.

Here is information on the language, both its development and its general characteristics.

In the early 1970s, the Federal government recognized that too many different languages were being used for defense applications. A DoD sponsored high order language working group was formed in 1975 with the mission to formulate re-

quirements, evaluate existing languages, and implement a minimal set of languages for DoD use.

A series of requirements documents were developed as shown in Figure 1. These documents were extensively reviewed by government, industrial and academic communities. They represent a series of refinements in the language requirements, culminating in STEELMAN.¹¹ In 1976, the languages shown in Figure 2 were evaluated against the TINMAN requirements. The major conclusions were:

- No language was suitable as it was.
- One language was a desirable and achievable goal.
- Development should be started from a suitable base: Pascal, PL/I or Algol 68.

After a competitive procurement, four contractors—CII Honeywell Bull, Intermetrics, SofTech and SRI International—were selected to design languages in parallel and in competition. The language design documents were color coded Green, Red, Blue and Yellow, respectively, in order to allow unbiased evaluation. All four language design teams chose Pascal as the base language. After three years of design, evaluation and refinement, the revised Green language by Honeywell was accepted as the Ada design Aug. 25, 1980.

The current definition of the Ada language is Reference 3. Reference 2 provides a review of the background, scope and methods of the language development. There is a textbook on a preliminary version of

Ada,¹² and an excellent technical introduction in Reference 1. Many of the engineering and computer industry trade magazines have provided brief introductions. The remainder of this article briefly mentions some of the technical characteristics of the language.

Ada was designed to be a language for the writing of large programs that would be used over a long period of time. It was designed to promote reliability and simplify maintenance, to allow development of programs from independently developed packages, and to be efficient in use of storage and execution time. It is a modern, high-level, strongly typed language, very Pascal-like in many of its constructs.

It has special features for control over visibility of identifiers, for handling concurrent processes and for handling errors in an orderly way. It also allows access, in a controlled way, to the machine-level architecture. It provides for data abstraction and has special constructs, called generic units, to allow building libraries of routines for various data types. Although it is not likely to replace COBOL in typical data processing applications, the language will find use for operating systems, compilers, numerical analysis, simulations, communication systems and a variety of commercial realtime applications. Ada will be of particular value in multiprocessing environments.

Concurrent with the implementation of Ada is development of the Ada Programming Support Environment (APSE), a collection of integrated software tools to assist in

Continued on page 25

Ada language Continued from page 23

Figure 1. Ada requirements generation

- STRAWMAN Apr. 1975
- WOODENMAN Aug. 1975
- TINMAN Jan. 1976
- IRONMAN Jan. 1977
- REVISED IRONMAN Jul. 1977
- STEELMAN Jun. 1978

Figure 2. Ada language evaluation

- Requirements based on TINMAN
- Languages evaluated were:

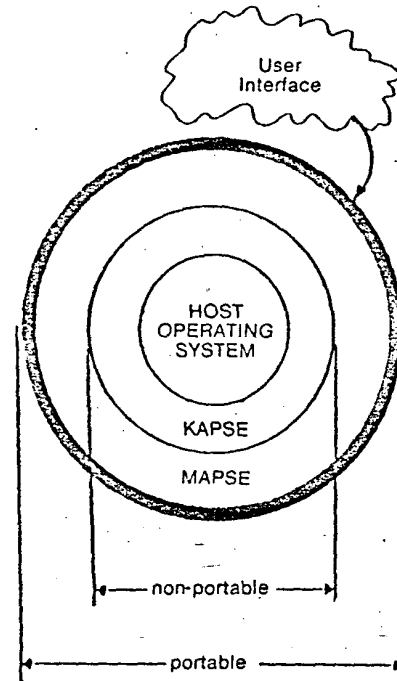
Used for DoD embedded computer systems: Jovial, SPL/I, Tacpol, CMS-2.

Used for process control and similar applications: Coral-66, LIS, Pearl, RTL/s, HAL/S.

Used for research: Euclid, Moral, ECL, Simula-67, PDL/2.

Used for applications outside of DoD: COBOL, FORTRAN, Pascal, Algol, PL/1.

Figure 3. Stoneman architecture



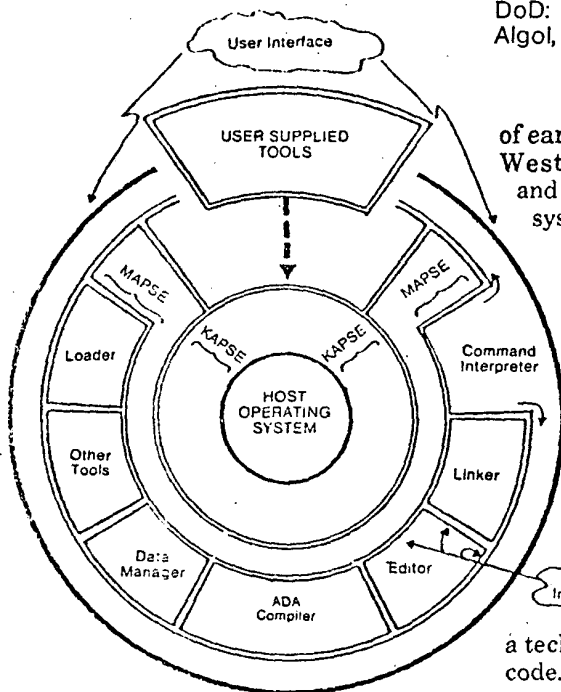
The combination of Ada and the APSE may not be the "ultimate" in languages, but it is likely to be extremely important in the data processing industry over the next 20 years.

About the author

Shumate, CCP, CDP, is technical director of the California div., SofTech. He has been involved with Ada in both government and industry.

Continued on page 42

Figure 4. APSE Architecture



of early availability of Ada on the Western Digital Microengine and the use of Ada as the primary systems programming language for the INTEL iAPX 432.

Ada is an important step forward in the development of programming languages.

The nature of the language has been addressed here only in general terms, but the references provide good technical overviews and introductions, and ought to be available in all technical libraries.

Another DATA MANAGEMENT article will follow, of a technical nature with examples of code.

Ada programming. There was a series of requirements documents, SANDMAN, PEBBLEMAN and STONEMAN, that defined a Kernel APSE (KAPSE) and Minimal APSE (MAPSE) related as shown in Figure 3. The APSE results from the addition of user tools as shown in Figure 4, and has transportability as shown in Figure 5. The APSE provides a common interface between the user and Ada. The MAPSE is intended to be portable among a wide variety of host machines and operating systems.

Although the first DoD Ada compilers will not be available until 1983, there is considerable activity in the commercial arena. Some of the more important are the IBM use of Ada as a program design language, the Telesoftware announcement,

Figure 5. APSE Transportability

