

TWO IMPEDIMENTS TO THE PROPER USE OF ADA

Donald G. Firesmith

Magnavox Electronic Systems Co.

1313 Production Road, Fort Wayne, IN 46808

OBJECTIVES: This brief article has two objectives:

1. To make Ada developers aware of two serious practical impediments to the widespread use of certain popular Ada constructs (e.g., abstract data type packages, abstract state machine packages, and generic packages) that are critical to the development of large, reliable, and maintainable embedded Ada systems.
2. To try to bring pressure on compiler vendors to remove these impediments so that developers can use the language as intended.

THE “GET THEM ALL” PROBLEM: The first impediment is due to the fact that current linkers will link in all subprograms of a withed package, regardless of whether they are needed or not. Thus, if you need only one procedure out of an abstract data type or abstract state machine package that contains ten exported subprograms, you get copies of the other nine as well. This seriously expands object code size without benefit, and unless fixed by the linker vendors, this will for all practical purposes prevent significant reuse and the use of methods such as Object-Oriented Development (OOD), which purposely produce a great many such packages. While this is less of a problem for host development, it is critical for embedded systems in which object code size is highly limited. Developers faced with this problem have sometimes been forced to separate subprograms out of such packages (thus destroying the package concept) and to separate type declarations out of the packages (essentially creating common data pools!). Because this global optimization problem will be difficult to solve for all cases (e.g., when tasks are involved), I am dubious that any quick fixes are on the horizon. However, any vendor interested in the embedded marketplace **MUST** solve this problem, and solve it soon. I would be very interested in talking with any such vendor.

THE “GENERIC COPY” PROBLEM: The use of generics also currently has a major negative impact on object code size. The compilers with which I am familiar implement instantiations by making copies of the code. Surely compiler vendors can reduce some of this duplication. In the mean time, some developers have advocated that generics be instantiated only once using a common type and that the single resulting instantiation be used exclusively in conjunction with explicit type conversions. Surely, this is not what the developers of the language had in mind for generics. I am naturally also interested in talking with all vendors who have solved, or will soon solve, this problem.

CONCLUSION: These impediments must be removed if Ada is to achieve its goals of increased reusability, reliability, and maintainability. In the long run, the compiler vendors must solve these problems if we are to be able to use Ada as intended. In the mean time, we may be forced to write more ADATRAN than we would like.