PLTs in structured programming

Editor:

With regard to the tutorial on structured programming in the March 1981 issue, let us congratulate author R. W. Jensen. The article concisely stated the prevalent attitudes and essential facts concerning structured programming in an enjoyable article.

We do feel that Nassi-Shneiderman charts deserved mention in place of processing logic trees. The NS method forces a structure upon the software design without the shortcomings of flowcharts, while avoiding the cumbersome and lengthy rules of processing logic trees.

Michael A. Neighbors
Associated Technology
Huntsville, Alabama

Malcolm E. Gillis
MEGA Corporation
Toney, Alabama

Author's reply:

Four factors, based on the essential premise that clarity and maintainability are primary goals of structured programming, led to my selection of the processing logic tree over other structured programming notations, including the Nassi-Shneiderman diagram.

First, the non-PLT notations provide only a snapshot of the program design at any instant of time. For example, when the abstract program statement A is refined into its next level of abstraction, the program statement A is deleted from the expanded program and the information is lost. The abstract information (if retained) serves to describe the expanded program block to those individuals (reviewers, managers, maintainers) attempting to understand the program function. The abstract information stored in the tree structure also provides a design history valuable in maintenance.

Second, the work of Peterson, Kasami, Tokura, and later efforts demonstrate the advantage of a "well-formed" program with a single input and possible multiple exits over a "proper" program with only a single input and output allowed. The NS notation as published does not provide for multiple exits.

Third, the stepwise refinement process causes the program detail to expand as the refinement process advances. The standard NS notation prohibits the expansion beyond a few levels of refinement. Variations of the NS notation do allow multiple page diagrams to accommodate the increasing detail.

Fourth, the NS diagram lacks the natural flexibility in handling several logical levels of nesting and special structures such as the case statement containing several cases.

As a closing remark, let me point out the rules for constructing PLTs are only restatements of the laws of stepwise refinement, the primary tool of structured programming.

Randall W. Jensen
Hughes Aircraft Company
El Segundo, California

Computer graphics—
a correction and suggestion

Editor:

In his survey article, Ware Myers refers to the work of MAGI in generating tree images, based on (I presume) a 1974 article.

I would like to point out that we have extensively improved upon this approach, so that the three-hour computer time required for a picture has been reduced to about a half hour. The new method was described in a paper published in 1977. In a companion paper, a full description was given of the scene composition procedure, as well as the methods used to generate forest images.

Herbert A. Steinberg
R & D Director, MAGI
Elmsford, New York


visualizing 3-D data. I think there might be a survey paper in this category that would interest your readership.

Research done at the University of Connecticut from August 1977 through May 1979 on the use of a low-cost computer-controlled model-making machine would give a good starting point. My research shows that the Connecticut machine (which has a patent applied for) is the only machine of its type using direct computer control. Similar machines (referred to in the paper describing the Connecticut machine1) use the intermediate step of producing a punched tape to control the machine.

I hope you find this a useful discussion and keep up the good work. Computer graphics needs the kind of positive exposure writers like Myers provide.

Scott M. Staley
John Deere Component Works
Waterloo, Iowa


Ada—a trademark of DoD

Editor:

The report that Ada, the Department of Defense's new programming language, is being trademarked by DoD brought a sense of déjà vu as I recalled Calvin N. Mooers' effort, circa 1970, to give similar protection to his computer controlling language system TRAC. Because TRAC was trademarked, any mention of it had to be accompanied by a footnote stating that TRAC is "the trademark and service mark of Rockford Research Institute Incorporated in connection with their standard computer controlling languages." The only other trademarked language of that time was JOSS, which had to be footnoted as "the trademark and service mark of The RAND Corporation for its computer program and services using that program." Today language trademarking has proliferated. A glance through the personal computing advertising columns shows a plague of bugs, the R-in-a-circle and the tiny TM attached to all manner of software. (One firm even seems to claim Fortran as a trademark.)

I am sorry to see the DoD put Ada in this company because it has always seemed to me that the ridiculous legalisms associated with language name trademarks and heavy-handed attempts at control impede the free discussion of a language in publications and effectively block the publication of textbooks. If a language is not written about and does not stimulate good textbooks, it will decline and die. I hope that trademarking Ada does not bring about a similar decline in its use and importance.

I am sure that Lt. Col. Druffel's objective, to "ensure that Ada is established as a consistent, unambiguous standard," are admirable, but the evidence of the trademarking of TRAC (the trademark and service mark of Rockford Research Institute Incorporated in connection with their standard computer controlling languages) and JOSS (the trademark and service mark of The RAND Corporation for its computer program and services using that program) may tell us something important. Certainly, *any* impediments to publication harm the spread and use of a language. Perhaps if control is exercised that restrains the computing community from fooling with and attempting to improve a language, the end result is non-use and early death.

Perhaps there is no need to worry. If Ada is good, it probably will be used and will succeed outside the DoD embrace, perhaps under a pseudonym like Ida or Dada.

Eric A. Weiss
Sun Company
Radnor, Pennsylvania

Mr. Weiss' letter to COMPUTER (also sent to and published by COMPUTERWORLD on April 14) was forwarded to Lt. Col. Druffel. His reply appears below.

*Ed.*

Editor:

The term Ada, when used in the context of a computer programming language, is indeed a trademark of the DoD. Applications to register the trademark have been filed in the United States, France, United Kingdom, West Germany, Belgium, Netherlands, Luxembourg, Denmark, Japan, Canada, and Italy. As Mr. Weiss points out, there is adequate precedent for trademarking a programming language.

Our motivation for taking this action is not to discourage use of, or even experimentation with, the Ada language. On the contrary, we have made every attempt to share information about Ada with the international computing community, to encourage implementation in the private sector, and even to encourage research and experimentation with Ada. Our proposed guidelines require only that use of the term be consistent with the definition of the language. The simple requirement to acknowledge the trademark at least once in a publication does not prevent people from writing about Unix* and can hardly be considered a hassle.

We intend to establish Ada as a consistent, unambiguous standard so that one of our primary goals, that of portability, may be realized. In support of that intention, we are committed to every legal mechanism available to discourage dialects. Toward this end, our primary restriction on the use of the trademark is that it may not be used to describe a production compiler that is not validated by the Ada Compiler Validation Office. This restriction does not prohibit an individual from creating a *new* language that is a subset or superset of Ada. It simply prohibits him from using the name Ada for his new creation, which is by definition *not* Ada. We acknowledge that it will take some time to implement a full compiler, so we have agreed that until July 1982, or until the DoD produces a validated Ada compiler, incomplete implementations may be described as Ada compilers if there is a clearly stated intent to complete the implementation without extension.

Any action to restrict the use of Ada would be contrary to our objectives. Our trademark guidelines have specifically excluded non-production compilers used in the university environment to encourage continued research. If it could be convincingly demonstrated that our strategy prevents acceptance of Ada, we would move quickly to alter that strategy. However, I believe that our strategy is a positive motivation for acceptance of the language. The using community recognizes the difficulties caused by lack of standardization in other languages. By aggressively protecting Ada through trademarking, copyrighting, compiler validation, and standardization, I believe we are providing a badly needed service to the computing community as well as to the DoD.

Larry Druffel, Lt. Col., USAF
Ada Joint Program Office
Washington, DC

---

*Unix is a trademark of the Bell Telephone Laboratories.