MASTER

CONF-781052--1

**PORTABILITY AND THE
NATIONAL ENERGY SOFTWARE CENTER**

by
Margaret K. Butler

Prepared for
ACM-SIGNUM Conference
"The Programming Environment for Development of Numerical Software"
Pasadena, CA
October 18-20, 1978

U of C-AUA-USDOE

NOTICE

This report was prepared as an account of work sponsored by the United States Government. Neither the United States nor the United States Department of Energy, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness or usefulness of any information, apparatus, product or process disclosed, or represents that its use would not infringe privately owned rights.

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

ARGONNE NATIONAL LABORATORY, ARGONNE, ILLINOIS
Prepared for the U. S. DEPARTMENT OF ENERGY
under Contract W-31-109-Eng-33

PORTABILITY AND THE
NATIONAL ENERGY SOFTWARE CENTER

Margaret K. Butler, National Energy Software Center

Argonne National Laboratory
Argonne, Illinois, U.S.A.

ABSTRACT

The software portability problem is examined from the viewpoint of experience gained in the operation of a software exchange and information center. First, the factors contributing to the program interchange to date are identified, then major problem areas remaining are noted. The import of the development of programming language and documentation standards is noted, and the program packaging procedures and dissemination practices employed by the Center to facilitate successful software transport are described. Organization, or installation, dependencies of the computing environment, often hidden from the program author, and data interchange complexities are seen as today's primary issues with dedicated processors and network communications offering an alternative solution.

THE NATIONAL ENERGY SOFTWARE CENTER (NESC) is the successor to the Argonne Code Center, originally established in 1960 to serve as a software exchange and information center for U.S. Atomic Energy Commission developed nuclear reactor codes. The Code Center program was broadened to an agency-wide program in 1972, and the scope of the activity expanded further with the organization of the Energy Research and Development Administration (ERDA) and then the Department of Energy (DOE). The Center's goal is to provide the means for sharing of software among agency contractors, and for transferring computing applications and technology developed by the agency to the information-processing community.

To achieve these objectives the NESC:

1. Collects, packages, maintains, and distributes a library of computer programs developed in DOE research and technology programs.
2. Prepares and publishes abstracts describing the NESC collection.
3. Checks library contributions for completeness and executes sample problems to validate their operation in another environment.
4. Consults with users on the availability of software and assists them in implementing and using library software.
5. Compiles and publishes summaries of ongoing software development efforts and other agency-sponsored software not included in the collection because it is felt to be of limited interest.
6. Maintains communications and exchange arrangements with other U.S. and foreign software centers.
7. Coordinates acquisition of non-government software for the Department of Energy.

8. Initiates and participates in the development of practices and standards for effective interchange and utilization of software.

About 850 computer programs covering subject classifications such as mathematical and computer system routines; radiological safety, hazard, and accident analysis; data management; environmental and earth sciences; and cost analysis and power plant economics make up the current collection. Each year over 1000 copies of library software packages or authorizations for their use are disseminated in response to requests from DOE offices and contractors, other government agencies, universities, and commercial and industrial organizations. Over 500 organizations are registered as participants in the program. Of these 333 are registered as contractors, or under exchange arrangements. The remainder pay the cost of the materials and services requested.

It is clear from the enumeration of NESC activities that the major portion of the Center's program is devoted to the task of making it possible for individuals, other than the program author, to use the computer program in their own computing environment, which is different from the environment in which the author developed the software. The Center's dissemination statistics attest to a measure of success in the task. This success is due in part to the DOE sponsors of program development projects who, recognizing the need for software sharing, have encouraged authors to attempt to develop portable software, and to the authors who, self-motivated, entered the initial development stage with the avowed intent of producing a transferrable product. But success must be attributed in large part, too, to the availability of programming language and program documentation standards, and to the program review and evaluation procedures and dissemination practices established by the Center.

STANDARDS

In 1964 when the first FORTRAN language standards were published, program developers welcomed them enthusiastically as the means by which programs, developed for today's environment, could be readily transferred to tomorrow's, or moved to someone else's, quickly and with minimal cost. Installations attempting to achieve this promised portability, however, were frustrated by the variety of implementations produced by the compiler writers, each claiming conformity with the standard, but offering, in addition, special enhancements exploiting their particular hardware.

At the time a standards committee of the American Nuclear Society sent a letter to the Editor of the Communications of the Association for Computing Machinery urging the computing community to exert pressure on the compiler developers to implement the standard, and where deviations existed to flag

accepted non-standard statements and standard statements implemented in a non-standard fashion, describing these variations in the compiler documentation. The computing community, not only failed to endorse this early plea for a standard that could be used to achieve software portability, but the Editor held up publication of the letter for six months because FORTRAN was mentioned explicitly and the ACM might appear to be showing a preference for FORTRAN over other programming languages.

While programming language standards have not proved to be the ready remedy to the portability problem first envisioned, they have provided the necessary first step. Authors pursuing the goal of producing portable software have, by restricting themselves to a "portable" subset of the standard language, common to nearly all compilers, been able to produce an easily transferrable product. Program verification tools, such as PFORT, have proved helpful in this activity.

Over the past decade the American Nuclear Society's ANS-10 standards committee has produced a series of standards to assist authors and developers of scientific and engineering computer programs in preparing software to be used by colleagues at other installations. These include ANSI M413-1974 entitled "Guidelines for the Documentation of Digital Computer Programs" and ANS-STD.3-1971, "Recommended Programming Practices to Facilitate the Interchange of Digital Computer Programs", both of which were adopted by the AEC's Reactor Physics Branch, along with the ANSI FORTRAN standard, for its program development projects. Recently, this committee completed another guidelines document, this one titled "Guidelines for Considering User Needs in Computer Program Development". It is presently under review by the parent committee.

PACKAGING PROCEDURES

The Center's software package is defined as the aggregate of all elements required for use of the software by another organization, or its implementation in a different computing environment. It is intended to include all material, associated with a computer program, necessary for its modification and effective use by individuals other than the author, on a computer system different from the one on which it was developed. The package is made up of two basic components: the computer-media material and the traditional printed material or hard-copy documentation. The material may include all of the following:

1. Source decks: source language program decks or card-image records.
2. Sample problems: test case input and output for use in checking out installation of the software.
3. Operating system control information: operating system control language records required for successful compilation of the software and execution of the sample problems in the author's computing environment. This element includes device assignment and storage allocation information, overlay structure definitions, etc.
4. Run decks: object decks or load modules prepared by the language processors preliminary to execution of the software. This element is redundant when included in conjunction with the equivalent source decks for interchange between users of like systems.

5. Data libraries: external data files required for program operation; e.g., cross section libraries, steam table data, material properties. To provide for exchange across machine-lines a decimal or alphanumeric form is recommended. Routines for transforming the data to the more efficient binary, or machine-dependent representation should be included as auxiliary routines.
6. Auxiliary routines or information: supplementary programs developed for use in conjunction with the packaged software; e.g., to prepare or transform input data, to process or plot program results, edit and maintain associated data libraries.
7. Documentation: traditional reference material associated with the development of the software and its application. If documentation is provided in machine-readable form it is classified as auxiliary information. This item may be a single report or several independent documents.

Not all seven elements are required for every software package; however, items 1,2,3, and 7 are rarely absent from scientific and engineering applications which make up the bulk of our library.

In our review process the package elements are examined for consistency and completeness. Whenever possible, source decks are compiled and test cases and run decks executed to ensure the output generated in another environment agrees with that produced at the developer's installation. This evaluation process provides a good check also, of the adequacy of the documentation elements. If the submitted documentation proves inadequate for our staff to evaluate the software additional information is sought and incorporated in the package documentation. Frequently an NESC Note is written for this purpose. If data libraries are present an effort is made to include these in machine-independent form. This is especially important for the first, or original, version of a program. When conversions to other computer systems, i.e. additional machine versions, are considered, machine dependencies reflecting significant convenience to the users of that system are accepted. An attempt is made to retain in the collection one version of each library program in a form amenable to transfer.

Special software tools have been developed to verify that all routines called are included in the package, to perform rudimentary checks for uninitialized or multiply-defined variables, unreferenced statement numbers, active variables in common and equivalence blocks together with block names and addresses, etc., and to convert between a variety of character codes such as IBM EBCDIC, CDC Display Code, and UNIVAC Fieldata.

DISSEMINATION PRACTICES

The computer-media portion of the program package is generally transmitted on magnetic tape; however, card decks will be supplied for card-image material upon request. The tape recording format to be used in filling a request can be specified by the potential user to suit his computing environment, and, whenever possible, the Center will provide the format, character code, density, and blocking requested.

The Center maintains records of all package transmittals and, should an error be detected or a new edition be received, all recipients of the affected program package are notified.

PROBLEMS

A large number of problems encountered today in software sharing are traceable to in-house modifications of vendor-supplied systems, locally-developed libraries and utility routines, and installation conventions. Recognizing the problem of reproducing the performance of a program at another installation independent of the "local" system, the ANS-10 standards committee introduced the concept of an installation-environment report in its "Code of Good Practices for Documentation of Digital Computer Programs", ANS-STD.2-1967. The idea was that this report would document those timing, plotting, special function, and other local system routines which would have to be transferred with locally-developed software, or replaced with their counterparts at another site before the software could be successfully executed. It was suggested that each program-development installation package their collection of the documented routines as a library package so that users of, for example, the XYZ Laboratory's software would be able to create the necessary XYZ environment either by implementing these environmental routines or with acceptable alternative routines. Several packages in the Center collection are of this nature; the Bettis Environmental Library and the NRTS Environmental Routines are two of them.

Most computer centers, however, have never attempted to define such a package. Program developers are frequently unaware of the routines automatically supplied by the local system and seldom informed when changes or modifications to such routines are effected.

Proprietary software can magnify installation dependency problems, and programs utilizing graphical output are always a challenge to exchange. In most cases the Center, and each receiving organization, is required to insert the equivalent local plotting routines before the test cases can be run. Even when organizations have the same commercial software product, they will probably have different plotting devices, and, if not, you can bet each location chose to implement its own unique enhancements--after all, graphics is an art!

One-of-a-kind compilers and parochial operating systems used at some installations have proved to be a significant deterrent to program interchange. Of growing concern with the increasing use of database systems is data exchange. A DOE Interlaboratory Working Group has committed its members to working with the ANSI Technical Committee X3L5 to develop specifications for an Information Interchange Data Descriptive File.

ARPANET and the Magnetic Fusion Energy Computer Center offer a different solution to the software portability problem. Instead of developing portable software and making it generally available such facilities encourage the development of centers of excellence which provide and maintain the latest and best software for a particular application on a dedicated processor, accessible to the user community. That is the other end of the spectrum.