

N73-19215

COSMIC PROGRAM DOCUMENTATION EXPERIENCE

Martha C. Kalar
University of Georgia

As indicated by the title, this paper deals with the experience of the Computer Software Management and Information Center (COSMIC) in computer program documentation. The first part of this paper will be a brief history of COSMIC as it relates to the handling of program documentation; the second part will summarize the items that seem to be essential for good program documentation.

On July 1, 1966, the University of Georgia was awarded a contract by NASA to receive computer software developed by NASA and its contractors and to supply copies of such material, on request, to all interested domestic parties through COSMIC. Originally COSMIC was to have been a clearinghouse type of operation; i.e., it would send to the requester a copy of exactly what was submitted. No checks were made on either the documentation or the program. This type of operation led to a number of dissatisfied customers.

In order to insure that the user received adequate documentation and a complete, workable computer program at a minimum cost, COSMIC established documentation and program checkout procedures. Time and experience have brought about changes to the original procedure.

COSMIC, today, is composed of 18 employees, 12 of whom are professionals familiar with electronic data processing and hold degrees in a variety of fields, and understand the disciplines to which the programs apply.

The professional staff is divided into two groups, one concerned with the evaluation of the documentation and one concerned with the checkout of the submitted computer program. The evaluation staff checks the documentation for completeness of vital material and assigns a class code to the document. The amount of detail, the complexity of the program, and the uniqueness of the solution all play a part in determining which class code is assigned to these programs. The programmer staff performs a check on each program submitted to the library to determine whether all nonstandard routines are present in the program deck. There are four machine types available to our programmers, the IBM 360, the IBM 7094, the CDC 6400 at the University of Georgia, and the UNIVAC 1108 at the Georgia Institute of Technology. Programs written for any one of these machines are compiled before dissemination; however, programs written for other machines must be assumed to be executable when they are disseminated.

Of some 2800 program packages submitted to COSMIC, 60 percent have been rejected for one reason or another by either the programmer or the evaluation staff. The poor

Preceding page blank

quality of the documentation received is a major factor in the rejection of the program package. Many times illegible documentation has been received, and the program has therefore been rejected. Programs have also been rejected because they are too short or too special purpose to have any value to organizations other than the originator's. Other submitted packages have not contained vital segments of the documentation, making them unusable. For example, COSMIC has received documentation that was a Xerox copy of a listing, with penciled notes on the sides. Documentation of this caliber cannot be disseminated.

COSMIC has encountered a variety of problems in the content of the documentation submitted. Experience has shown that the problem is most often in the user instructions. It is assumed that a purchaser of a COSMIC program is buying the program because it will solve his problem directly or because it can be modified slightly to solve his problem. Therefore, the user knows most of the technology involved or is at least familiar with its purpose. The reason the user needs the program is to obtain the desired results without having to write the program himself. The user, therefore, needs detailed user instructions that are easy to follow. The following is an example of poor user instructions: A Xerox copy of the handwritten instructions, "Use standard IBM OS/360 job control setup," was submitted as documentation. Needless to say, the documentation was rejected. Complete instructions would have contained a listing of a sample deck setup and samples of input and output format. These are needed because machine configurations differ and what is standard to one installation may not be to another. The input and output formats are needed so the user can test his results and knows what to expect of his output appearance.

Because of deadlines and overlapping projects, documentation does not always receive its fair share of the time allotted for these projects. When one works closely with a program for a period of time, certain terminology and concepts become very familiar, and when the documentation is updated, these terms and concepts might be omitted or overlooked. The potential user of the program, however, most likely will not know its routine terminology and familiar concepts; therefore, problems arise. The programmer should be aware of his users and should gear his documentation toward the novice, the user who knows very little, if anything, about the program.

COSMIC's purpose is to disseminate programs that any potential user can employ. Certain areas of documentation are essential and shall be outlined here:

- (1) Program name (official name, acronym, and program title)
- (2) Identification number (NASA, contractor, or other number; COSMIC references programs in our library by the NASA-assigned "flash-sheet" number)
- (3) Installation name (name and location of the center where the program was developed)
- (4) Date (date which program was completed)
- (5) Author(s) and affiliation(s) (The author of the program is usually the person who does the actual programming and design work. If these tasks are separate, both names should be given.)
- (6) Language (the programming language in which the program was written)
- (7) Computer or machine requirements (computer, minimum configuration, level of compiler, and other requirements for the execution of the program)

- (8) Functional abstract (approximately 300 words) including the following:
 - (a) Description of the program (The problem that the program is designed to solve should be presented in such a way that the reader may identify elements that are analogous to his own problem.)
 - (b) Method of solution (When the method is well known or documented in standard publications, it should be identified by reference. Modifications to well-known methods, new methods, or novel combinations of methods should be fully described to indicate their applicability.)
 - (c) Special features of the program (Processing features and options that contribute to the uniqueness of the program should be summarized. Types of input and output should be discussed in terms of their potential value in solutions of problems.)

- (9) User instructions
 - (a) Input preparation formats and options (precise definition of all variables, exact format and arrangement of input parameters, required card or tape format for all input data, and sequence of control statements)
 - (b) Output formats and options (These should clearly explain all output variables; some note regarding accuracy of results also should be included.)
 - (c) Data restrictions (The user should be provided with a full explanation of any data restrictions such as those constituting illegal input, numerical or data-set limitations, and the number of or size of the data sets that can be handled by the program.)
 - (d) Procedural references (manuals and detailed documentation required to use the program)

- (10) Sample input and output models

The documentation that COSMIC receives, in most cases, does not include all these items. Standards at COSMIC have been minimal in the past but are constantly being upgraded. (See appendix.) If the documentation is deemed insufficient, more information is requested from the originating center. If more information is not available, the program must be rejected. On some programs, this is all that can be done. The turnover among programmers is fantastic. A programmer remaining at one job for 2 years many times will have seniority in a department. Therefore, contacting the originator becomes a difficult task. But on the programs being written now, we hope to establish standards to obtain complete documentation initially with as much information as possible in order to anticipate later questions.

APPENDIX—COSMIC DOCUMENTATION AND PROGRAM STANDARDS HANDBOOK

1. INTRODUCTION

COSMIC (COmputer Software Management and Information Center) was established to evaluate computer software developed by governmental agencies and then disseminate the evaluated submittals to other governmental agencies, as well as industrial, educational, and research institutions. To expedite the technical aspects of this process, it is necessary for COSMIC to receive properly prepared documentation and program packages from submitting field centers and contractors. To explicitly state COSMIC's requirements for submittal packages is the primary purpose of this handbook.

COSMIC is cognizant that all documentation packages received will not meet the exact format as outlined in this pamphlet; however, it is imperative that all information requested herein be included with the package regardless of the format chosen.

It is anticipated that this volume will—

- (1) establish a much needed and easily implementable standard for documentation;
- (2) clarify the definition of a complete program deck;
- (3) promote a better understanding among all offices and agencies involved; and thus,
- (4) increase the efficiency and effectiveness of the entire project.

II. DOCUMENTATION CRITERIA

A. General

Documentation which meets the COSMIC standards must include the amount of information necessary to inform a prospective user of the precise problem which the computer program is designed to solve and to enable a qualified programmer to input the required data, successfully run the program, and obtain the desired results. Below is a chart of documentation criteria, each of which will be defined in the following text.

DOCUMENTATION CRITERIA	
SPECIFIC REQUIREMENTS	
1. Description of the Problem	
2. Method of Solution	
3. Program Language	
4. Machine Requirements	
5. User Instructions	
6. Operating Instructions	
OPTIONAL REQUIREMENTS	
1. Program Timing	
2. Accuracy of Results	
3. Sample Input and Output	
4. Flowchart	
5. Listing	

B. Specific Requirements

The following information must be included in the documentation for it to meet the COSMIC standards:

1. **Description of the Problem**—The description must include a **complete** definition of the problem which the program solves. The thoroughness and sophistication of this definition is determined by the sophistication and degree of difficulty of the problem itself. For instance, a simple mathematical routine may be described in one sentence, whereas a description of a program designed to construct electronic printed circuit boards may require a multiple number of pages.

2. **Method of Solution**—This requirement must include the programming techniques or methods used, supporting theory, design, and computational equations with their derivations to substantiate or illustrate the program.

3. **Program Language**—A statement of program language must include all levels of languages found in the submitted deck (e.g., FORTRAN IV, MAP, OBJECT) as well as the compiler necessary to process the languages.

4. **Machine Requirements**—An explanation of machine requirements must encompass not only the computer system for which the program was developed but also all peripheral equipment utilized by the program (e.g., disks, drums, consoles, tape units, display devices, plotters, etc.). Also mandatory is the level of the operating system on which the program executed (e.g., IBM-360/65, Release 14; CDC-6600, Scope 3.1; etc.) as well as the amount of core a program occupies once loaded.

5. User Instructions

a. **Input Instructions**—These instructions must provide the user with the information necessary to prepare his data for input to the program. They should include:

- (1) precise definition of all variables;
- (2) the exact format and arrangement of all input parameters (object time variables); and
- (3) the required card or tape format for all input data to be processed. It must be noted if the input requirement is for a specialized format, e.g., NASA formatted telemetry tapes.

b. **Output Requirements**—The user instructions must also contain a description of the output data formats and types of output devices; e.g., card punch, printer, magnetic tape, etc. In addition, the instructions must include an example to illustrate both the input deck setup and the corresponding output.

c. **Data Restrictions**—The assumption must be made that the user knows nothing of the mechanics of the program; therefore, any data restrictions or illegal input should be specified. For example:

- (1) x cannot equal zero;
- (2) y must be less than 200;
- (3) x cannot equal 5 unless y is less than 4.

d. **Program Structure**—A list of all decks in the program, the main program as well as any subroutines called, must be included. If a routine is to be included in more than one subsection (e.g., chain, overlay, etc.) of a program, please so indicate.

6. **Operating Instructions**—This information must provide the computer operator with step-by-step instructions pertinent to the execution of a program. It must include:

- a. tape assignments or selection (Designate tapes required for input, working, and output for successive runs.);

- b. deck setup;
- c. control and sequencing information; and
- d. special controls and requisite operator actions (e.g., console instructions).

C. Optional Requirements

The following information, although not essential, will facilitate processing and use of a program.

1. Program Timing—Timing information should include the computer time required for a run with a certain number of data points or check points, or the computer time required for an average run.
2. Accuracy of Results—This section should include the number of decimal points or number of significant digits which can be expected in the answer. Where some inputs are based on sampling, both the accuracy of the estimates and the reliability of the output should be supplied.
3. Sample Input and Output—A description of a sample problem, an example of the input data required to run the program, and resulting output from a run of the respective input should be included.
4. Flowchart—This must be a structural flowchart of the sequential logic and decision points included in the program. Machine-produced flowcharts of the exact programming techniques cannot be used to satisfy this requirement as they merely amount to a listing of the programs and do not briefly and concisely reflect the inherent logical flow of decisions.
5. Listing—This must be a post-list of the assembled program submitted to COSMIC to be used as an in-house aid in processing programs.

III. PROGRAM CRITERIA

A. Card Deck and Tape Submittal Formats

Following is a list of requirements compiled by COSMIC in an attempt to standardize program handling processes and to eliminate misidentification of submitted programs:

1. Card Deck Submittals—These must be clearly marked with the respective program identification numbers.
- *2. Tape Submittals—It is requested that 7-track tapes be used. If this is impossible, 9-track will be accepted.
 - a. Tapes must be recorded:
 - (1) at 556 or 800 bpi,
 - (2) in unblocked card image format (84 characters per record for BCD or 168 characters per record for binary),
 - (3) with a complete program package (main deck, subroutines, data, etc.) in the same file,
 - (4) with each complete program package separated by an End-of-File card (blank except for a 7-8 multiple punch in column 1),
 - (5) with multiple 7-8 cards following the final program on tape.
 - b. Programs must be identified by number, title, and file position sequence on tape. This may be accomplished with a cover letter or a label on the tape reel.

**Note added in proof:* These conventions have been revised in line with improved computer technology. The conventions stated here are not presently in use.

B. Definition of a Complete Program

An explanation of COSMIC's definition of a complete program is pertinent at this point. To be considered complete, a program must include:

1. main program;
2. all non-standard (not included with operating system as normally installed by manufacturer) subroutines called within the main program or by other subroutines in the package; and
3. all plotting routines called (If this is impossible for proprietary reasons, submit a dummy subroutine deck with all user called entry points; also, include with the documentation complete input and output variable formats for the routines used.).

C. Mode of Submittal Programs

It is imperative that COSMIC receive source decks rather than object mode decks. It is seldom that a disseminated program can be implemented by a purchaser without modifications being necessary. To facilitate modifications and, thus, wider usability of COSMIC programs, we publish only source programs.

DISCUSSION

MEMBER OF THE AUDIENCE: I wish to raise the question of standards versus guidelines. My understanding is that standards are something required, and guidelines are something to be desired. It seems to me that if documentation standards are insisted upon, many programmers will simply refuse to adhere to them.

KALAR: If most users can use documentation in a certain form, then I think the best thing to do is to try to put it in that form. If the form is pretty well agreed upon, then I think that people ought to try to conform to it. Call it standards or guidelines, I cannot determine between the two. I do not think you can enforce anything.

MEMBER OF THE AUDIENCE: I would like to try to answer that. I do believe that some minimum amount of information should be available to a potential user so that he can make some choice as to whether he wants to make a substantial investment in some of the documentation, which may run into thousands of dollars. I do believe that a standard or a standard requirement or specification may be needed in this area.

MEMBER OF THE AUDIENCE: You have given us a list of things that you desire to see in documentation. Has this been disseminated to your customers?

KALAR: A partial list is in the appendix of my paper. This is a little bit different from the one that COSMIC is now disseminating to its customers.

MEMBER OF THE AUDIENCE: Is it a regular procedure to advise the customers or the people that send you programs of the problems that you see as you go along?

KALAR: Generally, most of these items are covered in the appendix. When programs are submitted, if they are deficient in a certain area, we will tell the senders what areas to send us as documentation. These exact items are not written down yet, but they should be within the next couple of months.

MEMBER OF THE AUDIENCE: To your knowledge, has COSMIC had an opportunity to review the proposed NASA NHB standards on documentation?

KALAR: I do not know.

MEMBER OF THE AUDIENCE: How do you determine the costs for the distribution of programs?

KALAR: By the number of cards in the deck for the programming. The documentation is 10 cents a page. An average cost is about \$275 per program.

MEMBER OF THE AUDIENCE: You said you like to disseminate programs that any individual can employ. Well, I question this. We have taken the other tack in the nuclear field. We have said we want to disseminate programs to installations that have people competent in both computer science and nuclear science because we feel that if you really disseminate programs to anyone, you can spend a fortune trying to train them.

KALAR: I do not think we can be so choosy about our customers. Whoever wants to buy a program can buy one, and if they can understand it, they can use it.

MEMBER OF THE AUDIENCE: It seems to me you have to do a lot more work. More documentation is needed, and these people must be brought in and trained how to use these programs.

KALAR: Most of our programs now being disseminated are accounting-type programs, programs that the small businessman can use without any extensive knowledge.

MEMBER OF THE AUDIENCE: I have a question about your organization. How did a university get into disseminating programs that industry has paid fortunes to get?

MEMBER OF THE AUDIENCE: Could I answer that question? I am the COSMIC specialist at Goddard Space Flight Center with the Technology Utilization Office. COSMIC is mainly a nonprofit institution. NASA has a duty to distribute the technology that NASA develops to people in the public sector, that is, commercial, profit, and nonprofit organizations that may have a need or a desire to use any part of the technology that we develop. Computer programs are considered a part of that technology. COSMIC's function is to distribute those programs to those in the general public who may find them useful, thereby increasing the productivity and welfare of the general public. There is no profit involved to COSMIC. The programs that industry develops under NASA contracts belong to the Government. What the Government does in this instance is make that property available to the commonweal. I hope that answers your question.

MEMBER OF THE AUDIENCE: What is the general turnover in programs and purchases at COSMIC?

KALAR: I think we sell around 60 or 80 packages per month and receive probably an average of 50.

MEMBER OF THE AUDIENCE: One problem with documentation is that we may meet the documentation requirements for a Government contract. Then the contract monitor says to submit it to COSMIC. We submit it to COSMIC and receive a different set of documentation requirements. We go back to the contracting officer and ask for the money to document the program or the system for COSMIC, but they refuse. In other words, who is going to pay for documentation?

MEMBER OF THE AUDIENCE: This is one of the things that falls in my area. I believe that most NASA software documentation requirements now incorporate the COSMIC requirements for program documentation. What often happens is that the contractor does

not regard these as essential, because in the past the documentation specifications really have not been enforced. We now demand that these requirements be met.

MEMBER OF THE AUDIENCE: Do you review the request for proposal (RFP) to see that the requirements . . .

MEMBER OF THE AUDIENCE: I do see some of them, but I believe that most of our RFP's for documentation now include those requirements.