NCE & TECHNOLOGY:

COMPUTER MODEL DOCUMENTATION: A REVIEW AND AN APPROACH

NBS Special Publication 500-39
U.S. DEPARTMENT OF COMMERCE
National Bureau of Standards

NATIONAL BUREAU OF STANDARDS

The National Bureau of Standards[1] was established by an act of Congress March 3, 1901. The Bureau's overall goal is to strengthen and advance the Nation's science and technology and facilitate their effective application for public benefit. To this end, the Bureau conducts research and provides: (1) a basis for the Nation's physical measurement system, (2) scientific and technological services for industry and government, (3) a technical basis for equity in trade, and (4) technical services to promote public safety. The Bureau's technical work is performed by the National Measurement Laboratory, the National Engineering Laboratory, and the Institute for Computer Sciences and Technology.

THE NATIONAL MEASUREMENT LABORATORY provides the national system of physical and chemical and materials measurement; coordinates the system with measurement systems of other nations and furnishes essential services leading to accurate and uniform physical and chemical measurement throughout the Nation's scientific community, industry, and commerce; conducts materials research leading to improved methods of measurement, standards, and data on the properties of materials needed by industry, commerce, educational institutions, and Government; provides advisory and research services to other Government Agencies; develops, produces, and distributes Standard Reference Materials; and provides calibration services. The Laboratory consists of the following centers:

Absolute Physical Quantities[2] — Radiation Research — Thermodynamics and Molecular Science — Analytical Chemistry — Materials Science.

THE NATIONAL ENGINEERING LABORATORY provides technology and technical services to users in the public and private sectors to address national needs and to solve national problems in the public interest; conducts research in engineering and applied science in support of objectives in these efforts; builds and maintains competence in the necessary disciplines required to carry out this research and technical service; develops engineering data and measurement capabilities; provides engineering measurement traceability services; develops test methods and proposes engineering standards and code changes; develops and proposes new engineering practices; and develops and improves mechanisms to transfer results of its research to the utlimate user. The Laboratory consists of the following centers:

Applied Mathematics — Electronics and Electrical Engineering[2] — Mechanical Engineering and Process Technology[2] — Building Technology — Fire Research — Consumer Product Technology — Field Methods.

THE INSTITUTE FOR COMPUTER SCIENCES AND TECHNOLOGY conducts research and provides scientific and technical services to aid Federal Agencies in the selection, acquisition, application, and use of computer technology to improve effectiveness and economy in Government operations in accordance with Public Law 89-306 (40 U.S.C. 759), relevant Executive Orders, and other directives; carries out this mission by managing the Federal Information Processing Standards Program, developing Federal ADP standards guidelines, and managing Federal participation in ADP voluntary standardization activities; provides scientific and technological advisory services and assistance to Federal Agencies; and provides the technical foundation for computer-related policies of the Federal Government. The Institute consists of the following divisions:

Systems and Software — Computer Systems Engineering — Information Technology.

[1]Headquarters and Laboratories at Gaithersburg, Maryland, unless otherwise noted; mailing address Washington,D.C. 20234.
[2]Some divisions within the center are located at Boulder, Colorado, 80303.

The National Bureau of Standards was reorganized, effective April 9, 1978.

COMPUTER SCIENCE & TECHNOLOGY: Computer Model Documentation: A Review and an Approach

Saul I. Gass

Consultant
Potomac, Maryland

Sponsored by the
Institute for Computer Sciences and Technology
National Bureau of Standards
Washington, D.C. 20234

U.S. DEPARTMENT OF COMMERCE, Juanita M. Kreps, Secretary

Jordan J. Baruch, Assistant Secretary for Science and Technology

NATIONAL BUREAU OF STANDARDS, Ernest Ambler, Director

Issued February 1979

Reports on Computer Science and Technology

The National Bureau of Standards has a special responsibility within the Federal Government for computer science and technology activities. The programs of the NBS Institute for Computer Sciences and Technology are designed to provide ADP standards, guidelines, and technical advisory services to improve the effectiveness of computer utilization in the Federal sector, and to perform appropriate research and development efforts as foundation for such activities and programs. This publication series will report these NBS efforts to the Federal computer community as well as to interested specialists in the academic and private sectors. Those wishing to receive notices of publications in this series should complete and return the form at the end of this publication.

National Bureau of Standards Special Publication 500-39
Nat. Bur. Stand. (U.S.) Spec. Publ. 500-39, 89 pages (Feb. 1979)
CODEN: XNBSAV

Library of Congress Cataloging in Publication Data

Gass, Saul I

Computer model documentation.

(Computer science & technology) (NBS special publication ; 500-39)
Supt. of Docs. no.: C13.10:500-39.

I. Electronic data processing documentation. I. Institute of Computer Sciences and Technology. II. Title. III. Series. IV. Series: United States. National Bureau of Standards. Special publication.; 500-39.
QC100.U57 no. 500-39 [QA76.9.D6] 602'.1s [001.4'24] 78-606146

U.S. GOVERNMENT PRINTING OFFICE
WASHINGTON: 1979

For sale by the Superintendent of Documents, U.S. Government Printing Office, Washington, D.C. 20402
Stock No. 003-003-02020-6 Price $2.75
(Add 25 percent additional for other than U.S. mailing).

ABSTRACT

Recent studies and surveys have concluded that, in general, the documents produced to support the understanding and use of computer models are inadequate. This paper describes the issues and concerns of computer model documentation and proposes an approach for the development of adequate documentation. First, a number of documentation studies and reports are reviewed, including software documentation guidelines and model documentation procedures. Then, based on the relationship between the phases of the model life cycle and documentation information needs, a set of documents is proposed and described.

The author takes a highly critical view of the past and present inadequate state of documentation procedures for computer models. The attention of computer model sponsors and developers must be directed to this area. Otherwise, the author feels, there will be an unfortunate decline in the use of decision models as aids in the analysis of important policy issues. The course of action recommended in this report is an extreme position as to the total information and number of documents required to produce adequate documentation. The author calls for the capturing of all information generated during a model's life cycle. Further research is needed to adapt this extreme position to the realities of cost, resources, model complexity, and model use.

Key words: Computer; computer model; documentation; documentation procedures; model; model documentation.

NOTE:

This document is the final report of a study conducted to provide background and planning information for a proposed program to develop standards for the management of government modeling.

As a research result, it is published in its entirety. NBS neither accepts nor rejects the recommended approach to model documentation contained herein.

Paul F. Roth,
NBS Institute for Computer Sciences and Technology

"But 'glory' doesn't mean 'a nice knockdown argument,'" Alice objected.

"When I use a word," Humpty Dumpty said, in rather a scornful tone, "it means just what I choose it to mean--neither more or less."

"The question is," said Alice, "whether you can make words mean so many different things."

"The question is," said Humpty Dumpty, "which is to be master--that's all."

Through the Looking-Glass
Lewis Carroll

"Everybody talks about documentation, but nobody does anything about it."

Anonymous

"You get what you pay for."

Gabriel Biel

CONTENTS

I. Prologue--A Short Play: "Is Documentation Documentation?" or "Quo Vadis Documentus?"

Time: The present.

Scene: The office of a government analyst responsible for assessing the Federal Government's use of policy evaluation models. The analyst is seated behind a desk, a computer CRT is to one side facing the audience. The desk is piled with papers and reports that describe, among others, two of the most important policy models in current use--the Transfer Income Model (TRIM) for welfare program analysis and the Project Independence Evaluation System (PIES) for energy policy analysis (34) and (40). Included amidst the material that describe these two models are a user guide of 700 plus typewritten, double-spaced pages of text, tables, charts and programs for TRIM (33); and a set of 15 volumes that forms the latest PIES documentation (44). As the curtain rises the analyst, who appears confused and bleary-eyed, reaches out and picks up a paper at random and begins to read.

Analyst: "Everyone agrees that documentation is a 'good thing,'..." (The analyst pauses, smiles knowingly, and starts the next sentence.) "Several opportunities are afforded when the concept of documentation is expanded beyond a narrow technical description of a piece of software...." (He breaks off and stares at the two piles of TRIM and PIES documentation. His eyes wander down the page and he continues to read.) "Documentation is not fun. It is hard, nasty and boring business...,"(37). (The analyst stops, puts down the paper and reaches into the desk drawer and takes out a package of matches. He stands, strikes a match and proceeds to ignite the papers on his desk. With a look of relief, but with a sense of failure, he walks out the door, while flames consume the desk. As he exits, a moon-shaped, keep smiling face appears on the CRT followed by an unending stream of text.)

Curtain.

II. The Why of Model Documentation--Reflections After Reading Fifty-plus Documents on Documentation

All general surveys of computer models and all assessments of specific models that have come to our attention conclude that the documents that are supposed to describe and explain these models are either non-existent or are lacking in detail and do not serve the models well (34), (35), (36), (40), (45), (46) and (47). As will be reviewed later, many proposals have been put forth for improving and increasing the information content that describes a computer model. In general, we agree with most of these proposals, but we call attention to our prologue to emphasize the need for documentation procedures that not only produce information, but do so in a detailed, yet comprehensible, clear and timely manner. This might be an impossible task, especially for such complex models as TRIM and PIES. (The prologue was not meant to single out the TRIM and PIES documentation. We used them to illustrate our current abilities and difficulties in transferring information about complex, ongoing modeling efforts.)

Policy makers and other users must understand these models. If not, they will either ignore the power of models and make decisions with degraded information, or become blindly beholden to the computer output and thus, relegate their decision responsibilities to the modeling analyst. It is the professional duty of a modeling analyst to prepare documents that meet, in a timely fashion, the varied needs of the sponsor. It is likewise the duty of the model sponsor to state the needs for documentation and to provide personnel, resources and time to produce it.1/

Most manuals and reports that describe documentation requirements expound variations of the same set of purposes of documentation (6), (8), (9), (25) and (35). Although these reports mainly describe the documentation issues of computer software and not computer models, per se, we feel that they also address model documentation concerns. A composite listing of documentation purposes for models and software includes the following:

- to record technical information that enables system and program changes to be made quickly and effectively.
- to enable programmers and systems analysts other than originators to use and to work on the programs.
- to assist the user in understanding what has been done.
- to increase the model and program-sharing potential.
- to facilitate auditing and verification of the model and the program operations, i.e. model evaluation.
- to provide managers with information to review at significant developmental milestones to determine that requirements have been met and that resources should continue to be expended.
- to reduce the disruptive effects of personnel turnover.
- to facilitate understanding among managers, developers, programmers, operators, and users by providing information about maintenance, training, changes and operation of the software (model).
- to inform other potential users of the functions and capabilities of the software (model), so that they can determine whether it will serve their needs.

1/ Our comments here and throughout are not limited to policy evaluation models, but apply to all complex computer models.

Many claims are made as to the importance of documentation of computer software systems: "To maximize the return on this investment (in computer systems), and to provide for cost-effective operation, revision, and maintenance, sufficient documentation is needed at each stage of the software development life cycle," (8); and "Documentation provides the means for the greatest and most efficient utilization of the system by the user...and the means for careful, well-planned design and integration of the system," (50).

If the purposes and claims for documentation are true, then the modeling and computer programming communities appear to be grossly derelict in their duties in that the "lack" of documentation is one of the main reasons cited for model failures, or for models being little used by their sponsors, or for models not being usable by others (35), (45), and (46). A recent book on large-scale models begins a chapter on documentation by lamenting "Documentation is one of the most neglected aspects of modeling and simulation, partly because it is largely non-creative and therefore uninteresting. Furthermore, because it should be everyone's responsibility, it frequently becomes the responsibility of no one. While documentation should commence at the very beginning of a project, it is often left until the project is otherwise complete. This in turn makes documentation more difficult because it requires searching old records. In addition, most workers find documentation distasteful because it is part of the cleanup operation," (41).

How can we explain the difference between what we expect of documentation and what we actually get? If a model for documentation was constructed using the set of purposes and objectives of documentation as assumptions, it could not be validated using the results of most real-world documentation. We can only conclude that the general run of sponsors and users, model development managers, analysts and programmers do not believe in the high-tone purposes of documentation; or that the normal or usual circumstances of model development precludes the production of documentation that meets the varied needs of a model's audience. Of course, a combination of these two conclusions can hold, along with variations, e.g., a model was originally a research tool and was subsequently elevated to forming the core of a policy evaluation model.

Information that describes the current state of models causes us to be pessimistic about modeling activities with respect to modeling documentation. In the survey (46) it is noted that documentation was considered inadequate to enable other project personnel to set up and run the models (surveyed) in about 75 percent of the cases. To this we may respond with the question: "Was the transferability of the model an objective of the project and if so, was it known to the project personnel and conveyed to them by the allocation of resources and time?" The first part of the question was probably true if for no other reason than that most models for the Federal Government are developed externally by contractors and grantees, and everyone knows that analysts and programmers move about very rapidly from job to job, if not to different organizations. Whether proper resources were committed is the real unknown; and if so, were the resources used properly?

How to improve the documentation activities for computer models and whether such improvement would really overcome any, let alone all of the problems attributed to the lack of documentation, is an open question. Does the answer lie with the lack of motivation in modeling personnel and how they really view the purposes of documentation and/or the emphasis placed on documentation by the sponsors/users? We need to remember that the economic cost of producing an item like model documentation must be compared to its intrinsic and real value. One wonders whether the cost in dollars and modeling labor (applied to unglamorous documentation) is so great that the modeling world is willing to live with (and lament) the inefficiencies caused by not producing worthwhile and useful documentation.

In the following sections, we shall review a wide range of computer model documentation proposals, issues and topics in the hope that the sharing of the collective knowledge of many investigators will offer some directions for research and operational approaches to resolving this open question.

III The What of Model Documentation--Further Reflections

Of the material on documentation that has come to our attention,[1/] most of it describes documentation standards or guidelines designed for automated data systems (ADS). It is our view that documentation requirements for ADS form a subset of the documentation needs for computer models, and that we can build upon the ADS experiences to benefit model documentation. However, a complex policy or decision model has information and documentation needs beyond that of a complex software system. We shall review some of the ADS documentation structures and modify and combine them with model documentation proposals to obtain what we feel are documentation requirements and an approach to documentation for computer models.

1/ Requests for documentation guidelines and standards for computerized models and documentation examples were sent to over 100 individuals and organizations. Their responses are deeply appreciated and gratefully acknowledged. The applicable reports are cited in the References.

First, what is model documentation? From (34) and (40) we have that computer documentation is defined as information recorded during the design, development, and maintenance of computer applications to explain pertinent aspects of a data processing system; including purposes, methods, logic, relationships, capabilities, and limitations. Further, computer model documentation is the principal instrument which allows people in a modeling effort--the user, the model developer, potential users, etc.--to communicate. Complete documentation is important to (1) ensure that the model is thoroughly understood and can be operated and maintained in the present and the future, and (2) facilitate evaluation of the model by a third party (i.e., someone other than the model developer or initial user). The adequacy of model documentation depends on the answer to the question "Is the computer model documentation sufficient to understand, use and maintain the model?"

The above definitions and question automatically pose a number of related questions that have been addressed by model documentation researchers, but have yet to be answered with confidence; possibly they cannot. Some of these questions are: Exactly what type of information is required to form complete and adequate computer model documentation? How and when should it be made available? Who does the writing and how should they be motivated? How are changes made to ensure currency of documentation? How does the extent of documentation vary with the model's use, complexity, cost, etc.? What is the process by which adequate documentation can be obtained? We will, in this report, address these and related issues by summarizing the views of others and by offering our thoughts that have been formed by past experiences and the present study.

A basic but reasonable view of what elements of information should be included in a model's documentation would, we feel, converge to the following (35):

- A precise statement of what the model is supposed to do.
- The mathematical/logical definition, assumptions, and formulation of the problem being modeled.
- A complete set of current input and output and test cases that have been run.
- A complete set of flow charts of the computer program.
- A set of operating instructions for the computer operator.
- An explanation of the various options available in using the model.
- The computer program itself (listing), with comments about various operations in the program.

- The names of the programmers and project managers responsible for the model development and computer program.

The above elements are open to interpretation and a project staff could satisfy their explicit meanings without producing adequate documentation. This would be especially true if a complex policy model were under consideration. What we require is a process for obtaining specific, detailed information, organized to satisfy the needs of each segment of a model's audience. We shall describe such a process subsequently.

IV. Guidelines for Computer-Program Documentation and Their Relationship to Model Documentation

In this section, we review a number of reports that are designed to guide and direct analysts and computer programmers in preparing documentation of computer software. Where appropriate, we emphasize those aspects that relate to the documentation of computer models.

Historically, one of the first formal and official set of Government ADS standards was published in December 1972 by the Department of Defense (DoD).[1/] Although the use of this manual (6) is mandatory, a realistic philosophy as to its applicability is conveyed. It is noted, for example, that documentation standards deal with the communication of information that cannot be rigidly standardized and thus, modifications by a project manager are allowed. The manual stresses (possibly unrealistically) that, although the process of documentation is often identified as a separate phase of project development, documentation activity must be a continuing part of the developmental effort and not relegated and delayed to the last stages of a project.

The DoD ADS documentation manual (6) among others, formed the basis for the important and more generally applicable Federal ADS guideline--the NBS issued Federal Information Processing Standards Publication 38 (FIPS PUB 38) titled "Guidelines for Documentation of Computer Programs and Automated Data Systems," (8). These guidelines are intended to be a basic reference and checklist for planning and evaluation of documentation throughout the Federal Government. In the discussion that follows, we shall describe elements of (6) and (8) that we feel are of value to the aims of computer model documentation.

The key to the DoD and NBS documentation guidelines is the definition of ten major document types and the relating of the production of these documents to the phases and stages of the software life cycle. This life cycle is defined by the following major phases (8):

- Initiation Phase: During this phase the objectives and general definition of the requirements for the software

1/ An earlier set of documentation guidelines was published by NASA (52) are established. Feasibility studies, cost-benefit analyses, and the documentation prepared within this phase are determined by agency procedures and practices.

- Development Phase: During this phase, the requirements for the software are determined and the software is then defined, specified, programmed and tested. The ten major documents are prepared in this phase to provide an adequate record of the technical information developed.

- Operation Phase: During this phase, the software is maintained, evaluated and changed as additional requirements are identified. The documentation is maintained and updated accordingly.

The development phase of the software life cycle is subdivided into four main stages as follows:

-- Definition Stage: When the requirements for the software and documentation are determined.

-- Design Stage: When the design alternatives, specific requirements, and functions to be performed are analyzed and a design is specified.

-- Programming Stage: When the software is coded and debugged.

-- Test Stage: When the software is tested and related documentation is reviewed. The software and documentation are evaluated in terms of readiness for implementation.

The ten document types that are to be produced during the development phase are summarized in Figure 1. The relationship of these documents to the stage in which they may be produced is shown in Figure 2. FIPS PUB 38 notes that the terminology used to describe the stages is arbitrary, but that it provides a convenient framework within which the development of the ten document types may be discussed. It is also stressed that not all document types are required to document software in every case, and that in some cases the various document types may need to be combined.

We shall not discuss the specific contents of the ten documents as that is the function of (6) and (8); detailed content outlines are given in those publications. However, what is of interest are the philosophy and approach to some of the general problems and considerations of documentation discussed in (6) and (8), and related publications such as (25). We review some of these aspects next.

A documentation plan must be developed early in the project. Under

Functional Requirements Document. The purpose of the Functional Requirements Document is to provide a basis for the mutual understanding between users and designers of the initial definition of the software, including the requirements, operating environment, and development plan.

Data Requirements Document. The purpose of the Data Requirements Document is to provide, during the definition stage of software development, a data description and technical information about data collection requirements.

System/Subsystem Specification. The purpose of the System/Subsystem Specification is to specify for analysts and programmers the requirements, operating environment, design characteristics, and program specifications (if desired) for a system or subsystem.

Program Specification. The purpose of the Program Specification is to specify for programmers the requirements, operating environment, and design characteristics of a computer program.

Data Base Specification. The purpose of the Data Base Specification is to specify the identification, logical characteristics, and physical characteristics of a particular data base.

Users Manual. The purpose of the Users Manual is to sufficiently describe the functions performed by the software in non-ADP terminology, such that the user organization can determine its applicability and when and how to use it. It should serve as a reference document for preparation of input data and parameters and for interpretation of results.

Operations Manual. The purpose of the Operations Manual is to provide computer operation personnel with a description of the software and of the operational environment so that the software can be run.

Program Maintenance Manual. The purpose of the Program Maintenance Manual is to provide the maintenance programmer with the information necessary to understand the programs, their operating environment, and their maintenance procedures.

Test Plan. The purpose of the Test Plan is to provide a plan for the testing of software; detailed specifications, descriptions, and procedures for all tests; and test data reduction and evaluation criteria.

Test Analysis Report. The purpose of the Test Analysis Report is to document the test analysis results and findings, present the demonstrated capabilities and deficiencies for review, and provide a basis for preparing a statement of software readiness for implementation.

Software Life Cycle Document Types [8]

Figure 1

<table>
<tr><th>INITIATION PHASE</th><th colspan="4">DEVELOPMENT PHASE</th><th>OPERATION PHASE</th></tr>
<tr><td></td><td>Definition Stage</td><td>Design Stage</td><td>Programming Stage</td><td>Test Stage</td><td></td></tr>
<tr><td></td><td>Functional Requirements Document</td><td>System/ Subsystem Specification</td><td>Users Manual</td><td></td><td></td></tr>
<tr><td></td><td></td><td>Program Specification</td><td>Operations Manual</td><td></td><td></td></tr>
<tr><td></td><td>Data Requirements Document</td><td>Data Base Specification</td><td>Program Maintenance Manual</td><td></td><td></td></tr>
<tr><td></td><td></td><td colspan="2">Test Plan</td><td>Test Analysis Report</td><td></td></tr>
</table>

Documentation Within the Software Life Cycle [8]

Figure 2

guidance from the agency, the project manager needs to determine a plan that details:

- What document types apply and should be prepared.
- The formality, extent and detail of the documentation.
- Procedures and schedule of review, approval and distribution, and the distribution list.
- Responsibilities for documentation maintenance and change control through the development phase.

The formality, extent and level of detail of the documentation is a direct function of the size, complexity and risk of a project. FIPS PUB 38 offers the following two schemes, A and B, for determining how criteria could be established to aid project managers in stating the extent and level of detail of documentation required.

Scheme A

Four levels of documentation are defined:

Level 1 - Minimal level guidelines are applicable to single use programs, one-shot jobs of minimal complexity.

Level 2 - Internal level guidelines apply to special purpose programs which appear to have no sharing potential and are for use only by the requesting user; also to large programs which have a short life expectancy.

Level 3 - Working document level applies to programs which are expected to be used by a number of people in the same installation or which may be transmitted on request to other installations or to contractors or grantees.

Level 4 - Formal publication level applies to programs which are of sufficient general interest and value to be announced outside the originating installation; also included are those programs which are critical to the activities of the installation.

Further definition of these levels in terms of use, project cost and resultant documentation requirements are summarized in Figure 3.

Scheme B

This procedure employs twelve criteria, with weighting factors, and a scale of the total weighted criteria to establish formal documentation requirements. Table 1 shows the weighted criteria, and Table 2 illustrates their application. The way to use these tables is to:

Level	If PROJECT COST:	Or USAGE	Then DOCUMENTATION ELEMENTS	And EXTENT OF EFFORT
1	Less than $1000 Or One Man-month	One Shot (Single Use)	Software Summary plus any incidentally produced documentation.	No special effort, normal good practice.
2	$1000 to $5000	Special or Limited Purpose or Application	Level 1 plus Users Manual and Operations Manual.	Minimal documentation effort, spent on informal documentation. No formal documentation effort.
3	Over $5000	Multipurposed, or Multiuser	Level 2 plus Functional Requirements Document, Program Specification, Program Maintenance Manual, Test Plan, Test Analysis Report, and System/Subsystem Specification.	All basic elements of documentation should be typewritten, but need not be prepared in finished format for publication or require external edit or review.
4	Over $5000	Publicly Announced, or Critical to Operations	Level 3 produced in a form suitable for publication.	At a minimum, all basic elements prepared for formal publication, including external review and edit.

Cost and/or Usage Threshold Criteria--Scheme A [8]

Figure 3

WEIGHTS					
Criteria	1	2	3	4	5
1. Originality required	None—reprogram on different equipment	Minimum—more stringent requirements	Limited—new interfaces	Considerable— apply existing state of art to environment	Extensive—requires advance in state of the art
2. Degree of generality	Highly restricted. Single purpose	Restricted—parameterized for a range of capacities	Limited flexibility. Allows some change in format	Multi-purpose. Flexible format. Range of subjects	Very flexible—able to handle a broad range of subject matter on different equipment
3. Span of operation	Local or utility	Component command	Single command	Multi-command	Defense Department. World wide.
4. Change in scope and objective	None	Infrequent	Occasional	Frequent	Continuous.
5. Equipment complexity	Single machine. Routine processing	Single machine. Routine processing. Extended peripheral system	Multi-computer. Standard peripheral system	Multi-computer. Advanced programming. Complex peripheral system	Master control system. Multi-computer auto input/output and display equipment.
6. Personnel assigned	1–2	3–5	5–10	10–18	18 and over
7. Developmental cost	1–10k	10–50k	50–200k	200–500k	Over 500k
8. Criticality	Data processing	Routine operations	Personnel safety	Unit survival	National defense
9. Average response time to program change	2 or more weeks	1–2 weeks	3–7 days	1–3 days	1–24 hours
10. Average response time to data inputs	2 or more weeks	1–2 weeks	1–7 days	1–24 hours	0–60 minutes
11. Programming languages	High level language	High level and limited assembly language	High level and extensive assembly language	Assembly language	Machine language
12. Concurrent software development	None	Limited	Moderate	Extensive	Exhaustive

Weighting Procedure for Twelve Documentation Criteria--Scheme B [8]

Table 1

TOTAL WEIGHTED CRITERIA	Software Summary	Users Manual	Operations Manual	Program Maintenance Manual	Test Plan	Functional Requirements Document	System/Subsystem Specification	Test Analysis Report	Program Specification	Data Requirements Document	Data Base Specification
0-12*	X										
12-15*	X	X									
12-26	X	X	X	X	X			**		***	***
24-38	X	X	X	X	X	X		**		***	***
36-50	X	X	X	X	X	X	X	X		***	***
48-60	X	X	X	X	X	X	X	X	X	***	***

NOTES: * Additional document types may be required at lower weighted criteria totals to satisfy local requirements.
** The Test Analysis Report logically should be prepared, but may be informal.
*** Preparation of the Data Requirements Document and Data Base Specification is situationally dependent.

Total Weighted Documentation Criteria vs. Required Documentation Types--Scheme B [8]

Table 2

- weight the software by each of the twelve criteria, Table 1.

- Sum the weights assigned to obtain the total weighted criteria.

- For the total weight, find the row in Table 2 that lists the documents to be prepared.

The ADS documentation process described above (and in detail in (6) and (8)) should be viewed as basic and central to the documentation requirements of complex computer models. However, we do not think that it captures the total information needs of a modeling project--especially of a policy or decision model--and it must be expanded. We will do this in Section VI by relating model documentation to the concept of a (to be defined) model life cycle. We next continue our review of other ADS documentation procedures for their insights into the general problem of computer program documentation.

It is interesting to note that even with the DoD (6) and FIPS PUB 38 (8) guidelines, Federal agencies are still developing and using different computer program documentation procedures (10), (11), (27), and (54). This emphasizes that documentation is often viewed as a local concern. However, the flexibility of (6) and (8) allows for local interpretations, while still affording a framework for proper documentation decisions. For example, in (10) and (11) we find a listing of computer program documentation contents that are narrower and more limited than the FIPS PUB 38 requirements. Although we can envision situations that call for local documentation guidelines, it is unclear why (6) and (8) could not be adapted to these local needs, or at least be used to form the core of any in-house documentation procedure. Special local guidelines might be very adequate and appropriate, but if documentation is ever to be more than an afterthought of computer models and systems, then the analysts and programmers must be made aware of and taught to use accepted (if not universal) documentation practices.

Appropriate to local control and development are the guidelines that ensure consistency in computer programs written at a certain installation and/or for a specific purpose, e.g., (11). These type of standards are for subroutine linkage, FORTRAN coding conventions, etc., and are found to be a part of the operating environments of successful computer installations. The programming guidelines (11) address documentation in an interesting manner. First, it realistically notes that there are usually two problems regarding computer program documentation: (1) it does not exist, or (2) it exists but is out-of-date. The recommended solution to both these problems is concurrency, i.e., documentation must be written concurrently with program development in order for it to be accurate and available when checkout is complete.

In (11), two types of documentation are differentiated: (1) inline--documentation which appears in the program code itself and useful to someone attempting to understand and maintain the program, and (2) formal--documentation that includes all reports necessary to define the model adequately. This particular installation uses a program design language (PDL) (53) as a tool in defining and documenting the design, and a specially developed computer assisted documentation (CAD) technique (12). Both such elements are quite appropriate for computer model development and documentation.

Another documentation element of local concern is the general makeup of the documents. For example, the report (7) specifies the physical appearance, form and components of technical computer program and system documentation. It is to be applied to the ten FIPS PUB 38 documents and it details page layouts, flow chart, graphic symbols, etc. Many agencies have similar documents, e.g., (19).

The Army regulation (13) describes in great detail the objectives, policies and procedures and assigns responsibilities for the formulation, design, development, testing, evaluation, installation, operation, maintenance and review of Army Management Information Systems (AMIS). Of interest is the classification of AMIS by multi-users and man-years of effort. A Class A is a multi-user system and/or one that requires more than fifteen man-years; a Class B is a single-user system requiring between three and fifteen man-years of effort; while a Class C is a single-user system which requires less than three man-years of effort. Each such class requires a minimum level of documentation that is contained in only four types of manuals: systems description, functional user, program description and operator's manual. It is not clear why the DoD ADS (6) or the FIPS PUB 38 (8) are not explicitly cited, even though it is stated in (13) that FIPS (in general) must be incorporated in all new systems design or major system change efforts.

The report (18) reviews all Air Force software documentation requirements. It identifies major standard Air Force data items which apply to software, its development, acquisition and use. The term "data item" is used to refer to a formal collection of information (data) acquired during the system acquisition process to support the management of technical objectives of the program. These are termed Data Item Descriptions (DID) and the report lists sixty-five DIDs to software acquisition, development, maintenance and use; with twenty-five designated as major software documents. Each of the latter meets one of the following criteria:

- It is unique to software and this uniqueness is significant in proper use of the document.

- It is not unique to software, but it can provide visibility in critical areas of software development.

Many of these major documents are similar to those of (6) and (8), with some special reports identified, e.g., human operator critical task analysis. A minimum set of software documents is also identified; twelve documents from the major list and five related documents.

The twenty-five major documents are described in (18) in terms of:

- purpose of the document.
- references to more detailed descriptions of content and organization.
- who originates the document.
- categories of usage in the phases of the computer program life cycle.1/
- special conditions necessary for, or requiring the application of the document.
- relationships to other documents.
- adequacy of document delivered.
- potential problems in using the document.

The report (18) also compares its documentation requirements with the DoD ADS documentation standards given in (6). Those of (18), which are more extensive and inclusive, are designed to apply to the acquisition of a large software system composed of software, personnel, equipment and communications subsystems (as in a command and control system), in contrast to an ADS oriented to software only. Software in (18) is defined as computer programs and computer data, with the former also including applications and, by association, computer models.

An early handbook for computer systems documentation is the AEC specifications (19). Here, six major documents are defined and divided into two sets: (1) historical documents--preliminary study report, performance and design specifications, programming specifications; (2) operational documents--system manual, program manual, user's manual. The latter set is the one kept current by updating as changes occur to the system, while the former set is of historical value once each report has served its purpose. Most, if not all of information required by these six reports is also required by the more inclusive ten FIPS PUB 38 reports (8). Of importance to computer model documentation concerns is that (19) calls for a separate preliminary study report (basically a problem definition with a recommended course of action, supported by cost and other data), while FIPS PUB 38 includes similar information into a broader functional requirements document generated during the software definition stage. It is felt that for a computer model such preliminary information should be highlighted to stress user interest and future involvement in the results.

1/ The Air Force computer program life cycle is segmented in to the following phases: analysis, design, code and checkout, test and implementation, installation, and operation and support.

The EPA has published an Automatic Data Processing (ADP) Manual (27) that establishes policy responsibilities and procedures for the management and operations of the EPA ADP program. The manual utilizes an ADP system development cycle consisting of five main steps: system feasibility study, system design specification, system development and implementation, system operation and maintenance, and periodic review and audit. Although not directed to computer models per se, the manual is to be applied when carrying out a program for exploitation of scientific and technical application of computers and mathematical and statistical approaches to the needs of the EPA.

The manual (27) includes a chapter on documentation standards and requirements that delineates the information to be included in the system feasibility study, system design specifications, user manual and system maintenance documentation. Again, FIPS PUB 38 (8) documentation reports are more inclusive. What is of interest is the EPA requirement of a periodic review and audit (step five of the ADP system development cycle above). We will discuss subsequently this concept as applied to model documentation. For now, we note that the EPA manual states that ADP systems should be reviewed and audited on an annual basis to determine:

- if the system is still needed to satisfy valid EPA requirements,

- if the system is performing adequately or needs to be modified,

- if the costs involved with operating the system are justified by the benefits received,

- if the costs/benefits associated with the system justify its continued existence, given the current fund limitations and overall priorities within the user's organization, and

- if adequate user documentation and system maintenance documentation exists to use and maintain the system.

The Transportation Systems Center (TSC) of the Department of Transportation has issued guidelines for the documentation of computer programs (20). As the documentation outline given in the guidelines forms a concise statement of the minimum information required to document a computer model, we present it in Appendix A.

The TSC guidelines and those of the Department of Labor (14) also include a set of criteria for determining the level and extent of the documentation effort. They are similar to the criteria of Figure 3.

Other documentation approaches are given in (16), (21), (22), (26), (28), (31), and (58). Reports (16) and (28) describe the use of notebooks to aid in the documentation of both computer programs and models.

The program notebook (16) is organized to encourage planning and communication, as well as basic documentation. It contains a record of ongoing activity and accomplishments so that a supervisor or subsequent programmer can ascertain the status of the task. In addition, it provides source material for any formal documentation that is required. The program notebook includes an event log (a historical picture of relevant activities) and a compendium of working documentation produced during the phases of software development. The notebook maintains a current record, and material no longer pertinent is removed or indicated as being out of date.

The notebook of (28) is designed to keep a well-ordered and comprehensive record of information, analyses and models employed in the course of a simulation study. It is an attempt to capture the working assumptions, plans and progress of the modeler as an aid in improving the modeling process and model documentation. Both types of notebooks are very appropriate for any computer modeling project.

An example of a well-considered set of documentation standards developed by a non-government group is the report (31), written by the computer services group of SRI. Four kinds of documents are discussed: user's guide, user's reference manual, programmer's maintenance manual and subprogram brief. The report details in outline and narrative forms material to be included in each document. It is structured to serve as a checklist to be used by the documenter to ensure that all relevant subjects have been discussed. However, it is not as extensive or inclusive as FIPS PUB 38 (8), especially in terms of the latter guidelines' ability to capture the information that completely describes a large complex system.

The book (9) includes the material of (6) and (8) with some variation and extensions. One difference is the definition of a project development cycle (as contrasted to a software life cycle) in which the phases are initiation, analysis (definition), design (detailed), development (programming), implementation (testing and conversion) and operation (operation and maintenance). Although here limited to the concern of delivering a new computer software system, the explicit consideration of implementation is important when structuring documentation for the model life cycle.

Of extreme importance to the computer model user and developer communities is the transferability of a computer program to another computer hardware system and/or the ability of a new user to run and interpret the results of the program. This is, of course, what documentation is all about, and the DoD manual (6) and FIPS PUB 38 (8) are designed to facilitate such transfers and use. We have noted that installation should be concerned with stipulating programming procedures to ensure a capability of running and maintaining locally developed programs. A broader issue is how to document, distribute and maintain computer programs that are to be transferred and used by many installations not under the control of a single agency's ADP organization. We discuss next two approaches to this issue.

The URBAN Mass Transportation Administration (UMTA) of the Department of Transportation sponsors software development, including computer models, to be used by the transportation planning community. UMTA has developed a set of software standards that must be adhered to by all developers of programs that are to be part of a program library known as the UMTA transportation planning system (UTPS). The UTPS is used by UMTA and others in planning and evaluating multimodal transportation systems. These standards indicate specific documentation and maintenance requirements, and details such items as the languages of the computer programs and standard test data sets. UMTA defines a project's cycle in terms of four development steps: general functional specifications, detailed technical design specifications, computer code, and test design and results. UMTA requires documentation material for each step, with a fifth step being a review by UMTA of the full documentation and approval for the distribution of the software to the user community. The full documentation also includes a user guide, data set descriptions, and any materials related to briefings or training. The UMTA software standard describes the information to be included in each document, but is not as detailed as FIPS PUB 38. It also contains document preparation standards such as format, spelling, etc. The program writeup standards are designed to produce documents that are an integral part of the program so they can be easily modified and updated. It relies heavily on the use of in-code (embedded) documentation (via comment cards); it also employs an UMTA designed high-level user language (USL) for program description. The USL is used in conjunction with the structured programming (top-down) approach for software development. The UMTA software standards and their implementation include elements for improving not only computer programming procedures, but the specification and documentation of computer models as well.

The second approach that facilitates the transferability and use of computer programs and models is exemplified by the DoD Defense Logistics Studies Information Exchange (DLSIE) and its catalog of logistics models (23) and (24). The DLSIE performs the following functions:

- Acquires, stores, organizes and disseminates information about logistics studies and models (planned, in-process and completed) and other logistics documents, journals and books.

- Maintains a current and historical inventory of all logistics studies which may be of significance to the research and management of logistics.

- Maintains a data base containing relevant information describing logistics modeling efforts.

- Maintains a current and historical inventory of documentation developed pertaining to logistics modeling efforts.

- Develops, publishes and distributes appropriate documents announcing the information accumulated from reporting activities.

- Provides secondary distribution (hard copy) of logistics research and management information to authorized DLSIE users.

The DLSIE maintains a computerized data base that describes each logistics study and model. This data base is used to prepare the catalogue of models (24) by generating a magnetic tape that is sent directly to the Government Printing Office to be read by a linotron (a magnetic tape to film negative to offset printing converter). Individuals can also request a selective dissemination of information and custom bibliographies. We find the DLSIE concept and its implementation an impressive one in terms of a researcher or user being able to find out what has been done (and by whom) in a particular area of interest. Granted it is a rather costly operation, but in many areas of current research and development, e.g., energy and economic modeling, a similar information system would pay for itself in better usage of models, and information exchange and availability.

Additional approaches relative to improving the transferability of models is given in (57) and (58). The report (57) discusses the concepts of modularization as applied to the construction of large and complex models; the development of software interfaces that enable the analyst to link previously developed analysis and data base programs (e.g., statistical and report generators), and the linking of high level languages; and the development of wide-range processors that support the set-up and processing of models that use different methodological approaches (e.g., systems dynamics and econometric models). Modularization is basically an application of the top-down, structured approach to systems analysis and programming, and modular concepts are becoming standard and accepted by many agencies (11) and (54). The use of software interfaces and wide-range processors is not as advanced, but large-scale modeling efforts in energy, welfare and other areas will, we feel, aid in the development of these ideas.

The report (58) is an early standard developed by the American Nuclear Society that recommends programming practices which facilitate the interchange of computer programs prepared for scientific and engineering computations. Its objective is to simplify conversion, modification, and use of computer programs. It includes such recommendations as organizing the program into reasonably sized subprograms, and the minimization of the use of assembly languages. Report (21) is a companion documentation standard.

In this section we have reviewed a number of reports that were developed as guidelines or recommendations to improve the documentation of computer programs. As noted, we feel that some of these ideas and documentation processes are applicable to the documentation needs of computer models. Unlike the literature on the documentation of computer programs, there are no official guidelines that we know of for computer model documentation. Certainly, the programming, user and other documentation guidelines contained in FIPS PUB 38 (8) are applicable to the computer aspects of a modeling activity. However, it is our contention that the information requirements of computer models, especially decision models, are not satisfied by such guidelines. We address these concerns in the next sections where we first review material related to the problem of model documentation, and then discuss an approach for the development of computer model documentation.

V. Current Guidelines for Computer Model Documentation

As evidenced by their publications, the general community of computer model developers and model researchers has not been overly interested in the mundane issues of model documentation. We feel that this disinterest has handicapped severely the utility and acceptance of many valid modeling activities, and, in turn, allowed some invalid models to be used. The situation has improved over the last few years due to studies conducted by and the pressures due to the GAO and other Government agencies (35), (45), (46), (47) and (51), writings by a few researchers (37), (48) and (59), and the difficult documentation requirements of important decision models (32), (33), (34), (40), and (44). Credit should be given to a subgroup of computer model developers--simulation modelers--who appear to have had a continuing interest in related documentation procedures. They have recognized that good documentation aids in establishing the validity of simulation models and in increasing their usage rates. We next review some simulation and other computer model documentation procedures.

The USAF report (17) provides guidelines for preparing computer program or model documentation. It pertains to the specific set of simulation models and computer programs that are developed to support military analyses. These guidelines describe the contents to be included in a user manual, programmer manual, analyst manual and a summary manual. The report suggests that user and programmer manuals are the minimum requirements for documenting a computer model, with the analyst and summary manuals being optional. The purposes of these manuals are as follows.

The user manual must include the information necessary for the user to understand and substantiate the methodology of the model and to prepare accurately the model inputs and interpret the model outputs. The programmer manual must provide the guiding information to allow a programmer to modify promptly and accurately the model or to convert it for another computer system. The analyst manual must be designed specifically to meet the needs of the analyst in (1) the collection and preparation of data for the model, (2) interpretation of the output results which may be expected or desired, and (3) the corrective measures required when such results are not obtained. The summary manual contains a brief overview of the model, explaining its purposes and capabilities (the why, what, and how of the model); it should serve as a handy reference for a quick review of the model.

The report (17) also offers guidelines for organizing each document and for document maintenance. It suggests that new manuals should be published when major changes are made to the model or when a series of minor changes causes change to more than one-fifth of the documentation. The guidelines for preparing the user, programmer and analyst manuals are given in Appendix B.

The guidelines in (17) have been used to prepare the documentation of certain Air Force simulation programs. This was a post-documentation effort in that a modeling contractor was asked to review already existing and running programs and to prepare the necessary manuals. The detailed tables of contents for the user, programmer and analyst manuals for this project are given in Appendix C (39).

In the book (41), House and McLeod comment on the critical issues of documentation, and offer a most comprehensive discussion on how to develop proper documentation of computer models. Although their examples are based on the needs of simulation models, it is felt that the material is applicable to computer models in general.

In (41), the authors differentiate between two types of documentation information: descriptive and technical. For descriptive documentation, the criterion for adequacy is an affirmative answer to the question: "On the basis of this documentation alone, would it be possible for anyone reasonably knowledgeable in the field to determine the suitability and availability of the model for a specific use?" While for technical documentation, the criterion for adequacy is an affirmative answer to the question: "On the basis of this documentation alone, would it be possible for anyone otherwise competent to duplicate and run the model?" The authors note that descriptive documentation of a model is usually available (not necessarily the kind that gives a positive answer to their question, however); but, technical documentation is in short supply. Such technical documentation is a must as without it there can be no evaluation of the model by others, no model transfer, and, in many instances, no model use when key personnel leave the project. They emphasize that both types of documentation should be made a contractual requirement by the funding agency, a position we endorse.

In their chapter on documentation, House and McLeod describe the procedure used to document the complex Strategic Environmental Assessment System (SEAS) of EPA (66), give two examples of documentation formats used to produce both descriptive and technical information (due to McLeod (59) and Meadows et al. (29), and present a documentation checklist to be used as a means of ensuring that all descriptive and technical information is included.

In the article (59),[1] McLeod notes that documentation of the development and running of a model should be complete enough to allow for:

- a peer evaluation of the work,

- the reproducibility of the runs and experiments,

- the ability of others to build on the work reported instead of having to repeat it, or to start it over in their own way.

In reviewing the computer model documentation issues and procedures as described in the Appendices, the reader should note and contrast the differences in the documentation requirements of software systems and that of computer models. For example, the documentation procedure of SEAS includes three categories of documents: system, study, and programming. The system documents include a system definition document and a system implementation plan; the study documents allow for the research nature of many modeling projects, and they, along with the other procedures and standards of SEAS, are designed to strike a balance between the flexibility required by creative model building and the communication of information essential to orderly system development (41) and (66). Also, for computer model documentation, there is a stronger emphasis on user (sponsor) information and scope of user involvement, model assumptions and theoretical basis, validation, and the use and possible experimental nature of the project.

Central to the use of a mathematical/logical model, especially those developed as aids in the making of policy decision, is the determination of the model's validity.[2] Validity, in turn, is key to model evaluation. Evaluation is a process by which interested parties, who were not involved in a model's origins, development and implementation, can assess the model's results in terms of its structure and data inputs to determine, with some level of confidence, whether or not the results can be used in decision making (60). The sine qua non of such third-party evaluations is the availability of a proper set of documentation. Many authors address this concern; see discussions in (34), (37), (38), (40), (41), (42), (60), (61), (63), and (64) for evaluation examples and related material.

1/ See Appendix D for article (59).

2/ Here validation tests the agreement between the behavior of the model and the real world system being modeled; and verification attempts to ensure that a model (i.e., the computer program) behaves as the investigator intended.

We feel that the evaluation of computer models will be a continuing requirement, and any documentation process for computer models must also address the needs of third-party, independent evaluation. The GAO evaluations (34) and (40) were both critical of the status of the documentation of the models in question. The criticisms dealt with the documentation not being up-to-date, weak in program maintenance and test analysis documents, no test plan, and general lack of documentation. Documentation deficiencies are rectified sometimes by having documentation prepared subsequent to the evaluations or to the actual use of the computer model; for example (30), (39) and (44). Such post-documentation tends to be well structured and complete and can aid in the future use and maintenance of computer models. This is especially true for less complex and more directed applications like police patrol car allocation and beat design models (30) and (68) than for more complex models like PIES (44). Reliance on post-documentation to supply the necessary information for model use, transfer, evaluation or maintenance is a poor and a much more costly substitute for a documentation plan initiated at the beginning of a model's development. Concurrency of documentation must be maintained and sustained along with the other major modeling activities.

Extending computer model documentation to include the needs of evaluation requires the development and supplying of information that is beyond that usually documented or even requested. If such information is not generated during the phases of a modeling project, then it is our contention that the modeling process employed was incorrect and the project management was lacking. A basic hypothesis that should apply to the documentation process of a computer model is that upon completion of the model and after its implementation and maintenance phases are entered, documentation should have been produced that furnishes all information that would enable an independent review and assessment to be performed. This set of information includes the software documentation described in (17), (20), (34), (35), (37), (39), (40), (41), (42), (51), (59), (60), and (61). In the next section, we shall address the overall process for computer model documentation that is based, in part on this hypothesis. But before doing that, we conclude this section's review of computer model documentation issues.

In addition to those noted above, other investigators have described what they feel are the necessary information elements that must be contained in computer model documentation. Brewer (37) offers the following as a minimal set of information required to operate a computer model:

- Program listing.
- Variable listings, definitions.
- Flow charts.
- Verbal description of the program.
- Operator's manual.
- Programmer's manual.
- Summary of theoretical bases of the model.

- Data reduction methods and techniques used to specify the model's relationships and to assign input values to parameters.
- Cost data related to production, operation and updating.
- Listing of personnel involved in all phases of the model's existence.
- Existence and history of external professional review, including findings and remedial steps taken.
- Utility analyses types of use, frequency of use, cost, assessment of outcomes.

A similar listing is offered by Shubik and Brewer (45); there the recording of the information associated with those persons involved with the model is stressed. For example, at model initiation--who wanted it built and for what reasons; during model production--who was the master modeler, who was on the model team and what validation procedures did they use; during model operations--what type, if any, external review was made of the model and what was its results; and during use--who used it, when, and with what purpose and outcome. We feel that the capturing of this information is extremely important. A model tends to be an ever-changing entity. Its form and outcomes are reflections of its developers and users. Only by having full knowledge of their involvement can we understand a particular modeling process.

In an attempt to standardize a modeling reporting format, House (42) structured a questionnaire approach that is presented below. It is suggested that each model builder be required to deliver a completed questionnaire as a final product. Given such a set of questionnaires, then a model information exchange system could be initiated similar to the DoD logistics model information system (23).

Model Reporting Format (42)

A. Basic description of model

1. Name or title of the model.
2. Developer(s).
3. Agency or company.
4. Sponsor; purpose or objective of sponsor.
5. When developed?
6. Where developed?
7. How much time and money did it take to develop? Is model proprietary?
8. Developed separately or as part of a larger study?
9. Where used and in use; frequency of use?

B. Subject matter of model

1. Major purpose (objective) of the model.
2. Scope (subject matter) of the model.
3. Was the model based on a particular description or theory? If so, what one?

4. Does the model usage require knowledge of a specific discipline?
5. How is the model different from other similar models?

C. Modeling technique

1. In an analytical sense, what type of model is it?
2. Does the model use any standard packages (e.g. linear programming, statistical, etc.)?
3. Was the model developed from another model? If yes, what one(s) and how?
4. Is its structure clear? Its variables?
5. What are the data requirements of the model?
6. Does the model receive any data from other models? Is it required or optional?
7. What constraints does the model have (e.g. modes of transportation, city size, spatial, etc.)?

D. Computer aspects of the model

1. What computer language(s) is the model written in?
2. What machine(s) is it programmed for?
3. How much time does it take to run?
4. How large is the model (lines of code, core to run, etc.)?
5. How many parameters does the model require?

E. Validation of the model

1. Has the model been validated? How?
2. Is the model documented? How well?
3. Has the model been critiqued or appraised? By whom? At what point?
4. Has there been a sensitivity analysis of any type performed on the model? If so, by whom?
5. Can it be used from the current documentation? Has it been?

F. Model use and transferability

1. If asked, how would you demonstrate the utility of the model? Have you?
2. With whom should one get in touch to discuss use of the model?
3. How much would it cost to transfer the model?
4. Are the model relationships/parameters easy for the user to change?
5. Have there been any papers given or written on the model? Where? When? By Whom?
6. Is the output of the model special purpose or is it designed for a general audience?

The report (43) describes the activities of a workshop on energy policy models. It noted that a call for better documentation was repeated by nearly every speaker.[1/] In fact, the existence, timeliness, completeness, readability, dissemination and purposes of most documentation were challenged or criticized by the workshop participants. The report further notes, however, that the sanctity of belief in good documentation was challenged by counter charges that current documentation is not read; and despite the rhetoric, there is no financial support for documentation preparation because users are not interested in having or reading documentation. These counterclaims do have some validity.

It is incumbent upon all model sponsors, developers and users to confront the documentation requirements at the inception of a model. If one sponsor wishes to specify and support a certain level of documentation, then this should be stated, agreed to and written into the project files. For example, the costs of documentation are increased if an objective of the model is to have it be transferable to other users. If this is not the case, then the sponsor and user should so stipulate and indicate the documentation required for the user's local environment. In this way, the modeling community is forewarned and criticisms of the model for what it was not designed to accomplish would be unwarranted.

In a similar vein, the model developer must specify a documentation plan that fits the resources allocated and the objectives of the model. If the documentation objectives cannot be accomplished, then the developer must face and resolve this problem with the sponsor and user early in the project life cycle.

In the report (49), Chaiken notes that the documentation of a model plays many roles. The existence of a user manual is an absolute prerequisite for dissemination of a model to recipients who do not have technical assistance from the model developers. But, the availability of this manual does not guarantee dissemination. Some documentation can discourage dissemination in that it honestly states the difficulties and cost of transfering and using the model. The developer in so stating does a service to the community of possible users.

Documentation also alerts a potential user to the availability of the model, and for this purpose, a clear and brief executive summary is of more value than the user manual. Chaiken also notes that the availability of an annotated program listing is evidently reassuring to a potential user, even though few users ever read the listings with care. Having the listing suggests that the program is finished and not subject to repeated modifications, even though this may not be true. It also

---

1/ Also see Fromm et al. (46) for results of a survey that discusses model documentation quality, availability and how documentation relates to model use.

indicates that the program is not proprietary, and that the developer has confidence in the model as it is available to be read by others.

As a final item in this section, we comment on recent investigations directed towards the development of software and model documentation aids and languages; see (4), (5), (12), (15), (54), (57), and (65). A few researchers in the areas of simulation, models, large-scale systems models, complex software developments and their documentation have proposed and constructed special procedures for designing and communicating their products. For example, the DELTA (Development of Language Tools for Administration and Research) project (4) has as its objectives the development of a set of concepts and a language for descriptions of systems and communication about systems. The report (4) describes a way of analyzing systems, defined as DELTA-structured systems, and a DELTA-language for communication about such systems. The authors feel that current computer programming languages, simulation languages and natural languages are not appropriate for describing complex systems.

In the report (5), Nance proposes the development of a simulation model specification and documentation language (SMSDL). As proposed, the SMSDL would facilitate model specification and documentation, describe the model at the high and low (top-down) levels, be applicable to diverse problem areas, be compatible with present simulation languages, and aid in validating and verifying a simulation model.

We feel that systems and modeling languages are but a part of the modeling process, and they should not be looked at as being the procedural way to resolve the current difficulties in producing documentation that will satisfy the diverse needs of the modeling community. It is our opinion that these languages will be more aids to the analyst and programmer in doing their work and not be communication devices that improve the user/decision-maker understanding of a model and the modeling process. As modelers, we need to describe the user/decision-maker problem environment and this may be done best by a special language. But, we cannot impose such a language upon the users. They are not modelers and their interests lie elsewhere.

An approach related to the improvement of documentation is the use of computer-aided documentation programs (12), (15), and (65). The report (15) describes a software design and documentation language (SDDL) that is a communications medium that supports the design and documentation of complex software applications. The SDDL provides a design and documentation language, a processor for converting design specifications into a machine reproducible document, and a methodology for using the language and processor. Such procedures enable the system designers and programmers to maintain the currency of their efforts, and to relieve them of some of the tedious aspects of document preparation, e.g. flow-charting, formatting, up-dating. These efforts, plus research and development in special modeling languages, should be encouraged in that they can contribute new tools for understanding and performing the modeling process. However, we must realize that the ultimate needs of computer model documentation are not in new languages, or even a new universal language, and are not in special aids and devices.

The furthering of the use of models and the benefits to be gained by their proper use can be done only by removing the technical barriers that now separate the model developers from the model users. We cannot expect the latter group to be conversant with system languages, computer methodology, etc. We must communicate the bases and results of our modeling efforts to this group--the user, decision maker, sponsor group--via documents that are understandable, inclusive, and timely. And this must be accomplished using a natural language--here English--and a documentation process supported by both model sponsors and developers.

VI. An Approach for the Development of Computer Model Documentation

In this section we describe an approach--a disciplined approach--to resolving the problem of obtaining proper documentation of a computer model. In this context, proper documentation provides specific and detailed information that is organized and presented in a manner that will satisfy the needs of each segment of a model's audience. This audience consists of the model's sponsors and users (possibly non-technically oriented); the model's analysts, programmers, and computer operators; other users; other analysts, programmers and computer operators; and independent model evaluators. Although we are concerned mainly with the documentation requirements for large-scale decision models, our approach is applicable to all computer models.

We base our approach on the following assumptions:

- Computer program and software documentation of a model must follow the guidelines of FIPS PUB 38 (8) and its future amendments.

- Computer model documentation must provide sufficient information that would enable an independent review and evaluation of the model to be performed.

- Computer model documentation must describe all historical, technical, developmental, maintenance and implementation aspects of the model, including assumptions, implications and impact of using the model in a decision situation.

- The organization of a modeling project must include a formal documentation activity with stated objectives and assignment of resources (personnel, funds, time) for their accomplishment.

- As a means of managing the documentation activity of a modeling project, documentation must be produced that corresponds to the phases of the model life cycle, and the production and/or maintenance of the documentation must be concurrent with the time span of each phase.

The rationale behind these assumptions is founded on the following considerations. The FIPS PUB 38 are based on guidelines and practices in Federal agencies and other institutions and were developed after years of study by many Government computer experts. The FIPS guidelines are being adopted by many agencies and we envision their becoming the "standard" guidelines. Also, the National Bureau of Standards is continually monitoring and updating these guidelines. Thus, we see no need to develop other computer program or software documentation procedures. The ten FIPS PUB 38 documents (that were described briefly in Section IV) are adequate for capturing all computer aspects of a modeling project. When developing these and other documents for a modeling project, there will be redundancies. We need to bear in mind that some information must be presented to different audiences and most of the documents must stand alone. In the final analysis, the full set of information must be presented in the right place and at the right time to the corresponding interest group. To err on the side of redundancy to assure completeness is the correct thing to do.

If a modeling project is managed correctly and conducted using proper modeling methodologies, then the project staff would have had to investigate all issues relating to the model's assumptions, analytical basis, development and use. If the results of these investigations are coupled with the results of the computer implementation, and all such information is recorded, then an independent, third party review team would only need to read this information to determine the adequacy of the model. The team would not have to make sensitivity analyses, determine if the data are correct, etc., as the project management would have demonstrated that these matters have been taken care of. The review team would then need to be concerned only with the reproducibility of the results and with spot-checking to guarantee the veracity of the documentation. We believe that every modeling project should be conducted as if an external review of the model and its results will be made, if not in fact, then in concept. If this were the case, then what was just stated above would be the norm for model review teams. However, we recognize that this will not be true for most modeling projects, if for no other reasons than the lack of discipline within most modeling management and the additional costs required. Thus, we only assume that the model documentation will be complete enough for an external team to be able to perform additional tests and analyses to evaluate the computer model.

The final form of a computer model is a function of its origins (who wanted it and why) and its total evolutionary history. The true understanding of a model and its possible utility can only be accomplished by the documentation of this history. Hence, we think that it must be recorded.

As previously described, the overall value of much of the Government's modeling activity is debatable. A good portion of this disutility is due to the lack of proper documentation. Any proposals made here and elsewhere for producing documentation of computer models are worthless unless the modeling community recognizes that documentation is the key to model utility. Project managers must insist upon having documentation objectives and resource support from their sponsors. The delimiting of documentation is the proper thing to do, if for no other reason than to protect the project management from criticism by those who want to use the model beyond the project's requirements.

Finally, documentation must be an integral part of all phases of a modeling project and that documentation, to be of real value, must be an output of the ongoing project. Post documentation of models must be eliminated--it is inefficient, it usually relies on personnel who were not involved in the model's origins and development, and it is not timely. Documentation must be related to the modeling life cycle and all personnel who intersect this life cycle must recognize their duty towards the model's documentation.

Any approach for documenting computer models must not be rigid and must allow for the specific needs of the project. Thus, flexibility and innovation in documentation is encouraged. But, this should not be interpreted as a license to just get by and do the minimum. Also, model documentation should describe not only what the model can do, but also what the model, as designed, cannot do. If a model is experimental and should not be used in an operational or decision setting, this should be stated explicitly. The above comments are just another way of stating that professionalism must be a part of the documentation activities of all computer modeling projects.

As noted in the previous sections, computer programmers, model builders and systems analysts tend to think of their efforts in terms of major phases, stages or steps (8), (19), (51), (60), and (67). In many projects, the activities and resources are divided formally into such segments and progress and expenditures are accounted for by segment. It is recognized, of course, that a modeling project's phases overlap in both time and resources, and a project develops along parallel phases, not serially. In any event, we believe that good model management practice requires the production of documentation to be related to a model's life cycle phases. This concept is just an extension of the FIPS PUB 38 software cycle - document production process to the modeling environment. For the documentation needs of a model, the software life cycle is too limited and aggregated and must be extended and refined, especially for complex decision models.

We next define an appropriate phase segmentation of a computer model life cycle, followed by a discussion of the information to be produced in each phase and the associated documentation. It is not our purpose here to produce a FIPS PUB 38-like set of guidelines for computer models. However, what is offered below can, we feel, be used as a basis for any such development. We emphasize that these model life cycle phases are interdependent, do not necessarily coincide with fixed time periods, and are just convenient groupings of project activities that can be related to the documentation requirements of a modeling project.

COMPUTER MODEL LIFE CYCLE PHASES

- Embryonic: during this phase, the to be sponsor/user contemplates the application of modeling methodology to aid in resolving a problem area, i.e., an idea has been hatched.

- Feasibility: during this phase, the problem is defined and delimited, and specific approaches for solving the problem are conceived and evaluated, i.e., an investigation and decision as to whether the idea can and should be developed further is undertaken.

- Formulation: during this phase, the analytical basis of the selected solution approach is developed, i.e., the idea is represented in terms of a model.

- Data: during this phase, the information requirements to support the model and its development are determined, and activities for the collection and analysis of the data are initiated, i.e., data that describe and support the model are determined to be available and are collected.

- Design: during this phase, the analytical, data and computer requirements are integrated into a set of system specifications for resolving the problem, i.e., the user's problem requirements, as described by the model, are combined with computer and programming approaches to produce a viable technical solution.

- Software Development: during this phase, the design specifications are converted into tested and operating software, i.e., the design is processed through the four stages of the FIPS PUB 38 software development phase--definition, design, programming, test--to produce a verified computer system.

- Validation: during this phase, a validation or acceptance test plan is developed and carried out to validate data extensions (e.g., parameters, forecasts), the model and its subcomponents, and the verified computer system. The plan should include agreed upon test cases or scenarios, sensitivity analyses, tests for robustness, historical validity, etc., i.e., the model, as represented by the computer system, is tested against specified user requirements and/or system objectives to determine user acceptability.

- Training and Education: during this phase, the user groups involved in the future use of the model--decision makers, analysts, computer programmers, computer operators, data collectors, etc.--are trained in appropriate aspects of the computer system, including maintenance of the model and system, i.e., a complete training program must be developed and given (this is most important if the computer system was developed by an external organization).

- Installation: during this phase, the verified and validated computer model is installed, tested and operated on the user's computer, i.e., if the computer system used for development and test is not the user's system, then an installation plan must be developed and carried out to ensure compatability of the computer systems and reproducibility of results.

- Implementation: during this phase, the user organization integrates the computer system into its operating environment and procedures are developed for generating and requesting specific computer analyses, and interpreting and using the results, i.e., the idea has matured into a verified and validated computer model and the model is made part of the organization's (decision) activities.

- Maintenance and Update: during this phase, a process for maintaining the computer model is developed and implemented, including modifications to the model, programming changes, input/output procedures, data and parameter changes, file maintenance, etc., i.e., activities are structured and implemented, and personnel and funds allocated to ensure that the model will continue to represent the user's view of the problem and its environment.

- Evaluation and Review: during this phase, a procedure is established that provides for independent third-party assessments of the model and/or periodic reviews by the user, i.e., depending on the importance of the model in a decision environment, a plan for a detailed independent assessment is developed and implemented, or the user establishes internal review team procedures to ensure that the model is updated properly and is still required by the organization.

- Documentation and Dissemination: during this phase, a documentation plan is developed and implemented for recording of the results of all other phases, i.e., documentation objectives are agreed to, the requirements of specific documents are stated, and the documents are produced. The documentation phase begins during the embryonic phase and continuous throughout the model's life cycle. If appropriate, a plan for disseminating documents and information on the structure, utility and use of the model is also initiated and implemented.

Before describing the documentation requirements for the above model life cycle phases, we next recapitulate our view of what successful model documentation must include. A major reason why models are not utilized properly or utilized at all is due to incomplete and out-of-date documentation. Our approach to correcting this flaw is to be information greedy, i.e., within reason, require all participants to keep informal and current records of their project activities. This information is made available to the documentation staff so they may use it for or combine it with the project's formal documentation. We are describing here a general documentation approach that is directed towards improving the value of complex decision models. The approach can apply also to "simple" models. The principle of information overkill should not be abandoned because the model is simple unless a purposeful decision is made to reduce information.

In a modeling milieu, we must be overly concerned with being able to know and understand the problem situation and its origins, the assumptions of the modeling approach, the decision environment and the user objectives, the validity of the model and data, where the model can be used, etc. We have found that much of this information is never recorded, and sometimes it is never known or stated explicitly. If a model has not been validated or cannot be validated, it should be so stated. If a programmer ran a verification check of a subroutine during the midnight shift, it should be so stated. If the user has ruled out a solution alternative or imposed other restrictions that influence the form of the model, it should be so stated. The acceptability, evaluation and future utility of a model can be determined only by a complete documentation record that relates, not just what has been done, but what has been omitted and why. Thus, the plans and activities required by the above model life cycle phases need to be developed explicity and their results recorded. Much of the information can be contained in informal analyst and programmer notebooks, memoranda and working papers. The documentation phase should include activities for gathering and cataloguing such informal information.

We describe next the formal documents that should be developed for any computer model project. Depending on the scope and ultimate use of the model, some of these documents can be eliminated or combined. In any event, the user/sponsor and the model developer must conclude an agreement as to the documents produced, their contents, uses and audiences. In what follows, we shall sketch out the information to be recorded in each document, recognizing that we are not aiming for completeness in this study. Our purpose has been to review model documentation proposals and to give some direction to future documentation guideline efforts.

Some of the documents are direct products of a particular phase, while others contain information from a number of phases or are the outcome of the total project. The form of the documents can range from a few pages to detailed manuals. We note again that it is incumbent upon all persons involved in the model's development to maintain current records of their activities as they pertain to the model's specifications, assumptions, analytical basis, computational requirements and testing, validation, data sources and collection, implementation and maintenance.

COMPUTER MODEL DOCUMENTS

NEEDS DESCRIPTION: Embryonic Phase

A discussion of the origins of the model idea including who initiated it, why, who are to be the users and what are their needs, extent of problem, general description of problem and decision environment, why modeling was considered, preliminary feasibility considerations, other solution approaches, why a computer model, impact of problem and solution, what is expected from solution, how model and solution are to be used, etc. This is a historical document that can be formed from memoranda, notes, working papers, records of meetings, and possibly user or developer proposals.

FEASIBILITY STUDY: Feasibility Phase

A report that describes the background, purpose, scope, organizations, and participants involved in the study; definition of the problem and issues and objectives, requirements to be met; organizations, functions and systems examined; solution alternatives with costs and benefits; recommended computer model solution and justification, plan of action and schedule of activities; resource requirements (personnel, funds, computer, facilities); etc. This is a historical document that describes the process used to determine that a computer model can and should be developed to resolve the problem. It should describe the role of the model in the user organization, who the users are, and the range of decision situations to be evaluated by the model.

MODEL FORMULATION DESCRIPTION: Formulation Phase

This report describes the complete details of the mathematical/logical model; the theoretical and analytical rationale for its form in terms of the problem definition; assumptions, hypotheses and restrictions; parameter estimation procedures; general data requirements; computational and numerical analysis requirements; computer resources required; approaches and tests for validating the model; sensitivity, robustness and other evaluations required; restrictions on the use and range of the model, etc. This is an operational document maintained throughout the model life cycle. The model structure is usually modified over time and a procedure for updating the description of the model must be initiated.

DATA REQUIREMENTS DESCRIPTION: Data Phase

This report describes the detailed data needs as required by the model; data sources; the process for obtaining the data; experiments, data collections and surveys to be performed; organizational and individual responsibilities for obtaining, updating, and processing the data; numerical and forecasting techniques to be used for parameter estimation; data validation procedures; acceptable data ranges; data input procedures to the computer model, etc. This is an operational document that is maintained throughout the model life cycle.

DESIGN SPECIFICATION: Design Phase

This report is the major document that summarizes the results of the preliminary analyses and details how the model, data and computer aspects of the problem solution are to be integrated into an operational computer system. It is here that the computer requirements, programming, and software specifications are described in more than general detail so that a viable design alternative can be assured and selected. This document includes descriptions of the problem, model, data, and background information; system design alternatives, costs, advantages and disadvantages of each; description of the recommended system design that details assumptions, limitations, restrictions and expected results; software, hardware and interface considerations; overall summary of the major functions, purpose, data requirements, output and users; critical factors affecting system development; system development plan with cost and personnel by task; general plan of action for management and organizational changes and decisions, equipment usage and/or acquisition, personnel training, and user participation by task and level of effort. This document is both an historical and operational one. It records the process by which the computer system was selected, but as the model and related elements are usually modified over time, it will have to be amended to reflect the current system design.

SOFTWARE DESCRIPTION: Software Development Phase

The documents produced here are the ten documents described in the FIPS PUB 38 Guidelines (8). These documents would include some of the information developed in the documents described above.

VALIDATION DESCRIPTION: Validation Phase

This report includes a description of the model validation plan agreed to by the user/sponsor and model developer, and the results of implementing the plan. Validation of the model must include tests of the model's output in terms of comparisons to historical data, acceptability by the user (experiential or intuitive tests), statistical measures, etc. The developers must state and explain the deficiencies and divergences of the model's output, as well as apparent agreements. The validation report should delineate the problem environment in which the model is known to produce results acceptable to the user/sponsor. The validity of the model must be reestablished whenever the model is changed.

TRAINING PLAN: Training and Education Phase

This report details the education and training requirements of the project and describes those materials that need to be produced, including any briefing materials. This plan should describe procedures for turning over the system to the user group (if different than the developer), and tests for ensuring that the new groups understand their aspect of the model, e.g., decision makers should know how to request computations and to be able to interpret the results. The outcomes of the training and education effort should be described.

INSTALLATION PLAN: Installation Phase

This report describes the process for ensuring that the verified and validated computer model is installed correctly on the user's computer system. The test plan is described and the results recorded.

IMPLEMENTATION PLAN: Implementation Phase

This report describes the process by which the computer model is made a part of the user organization, who are responsible for generating and requesting model analyses, how the outputs are to be used in informal and formal reports, security of the system and its inputs and outputs, final authority on output acceptance.

MAINTENANCE PLAN: Maintenance and Update Phase

This report describes the processes for modifying the model and its data, revalidation and who are responsible for the updates. Programming maintenance would be included in the software documents.

EVALUATION PLAN: Evaluation and Review Phase

Depending on the objectives and needs of the user, an evaluation or periodic review process is described. An external third-party evaluation plan cannot be specified by the user or developer, but is, of course, a function of the evaluator team. A process for doing this is described in (60) and (61). A periodic review team would need to develop assessment procedures based on the structure of the model and its use. Thus, a detailed review plan cannot be developed ahead of time. A report should be written at the conclusion of the evaluation or review that describes the status of the computer model and recommendations for its change, future use or discontinuance.

DOCUMENTATION PLAN: Documentation and Dissemination Phase

This report describes the informal and formal documents to be produced, by whom and during what phase(s) of the project, i.e., how the documentation and dissemination phase is to be managed. The report includes a statement of the documentation objectives, dissemination plan and maintenance and updating procedures.

As given above, the relationship between the documents and the model life cycle phases is a one to one correspondence. These phases and documents can be combined depending on the computer model and the needs of the project. Each document will be a result of information that is recorded in project notebooks, studies, programmer logs, memoranda, etc. The documentation plan spells out the responsibilities for the recording of information and the integrating of this information into the project documentation. These documents will describe the historical, developmental and operational aspects of the project and the computer model. However, in terms of the complete needs of users and analysts, the above set of documentation is incomplete. What is required is the coalescing, combining, rewriting, etc. of the information in these documents into very clear and readable new documents termed User's Manual, Analyst's Manual, and Executive Summary. (Note that the documents of the software development phase include programming user, operations and maintenance manuals.) As with the other documents, the production and maintenance of these new documents are described in the documentation plan, along with their contents and assignments for their development. These documents must be available for use during the training and installation phases. A description of typical contents of the user's and analyst's manuals are given in Appendices B and C. Briefly, we note the following with reference to these three documents.

USER'S MANUAL

This report combines information from the other project documents to represent a source document for the user of the computer model. It is to provide the proposed and future users of the model with information necessary to use it effectively. It should allow non-programming personnel to understand the workings of the computer model, how to request runs and interpret the output.

ANALYST'S MANUAL

This report combines information from the other project documents to represent a source document for analysts who have been and will be involved in the development, revisions and maintenance of the computer model. This manual should include only those technical aspects of the model that are essential for practical understanding and application. The detailed technical developments will be contained in the phase-produced documentation, e.g., Model Formulation Description.

EXECUTIVE SUMMARY

This report is an essential one for computer models used in a decision environment. It is directed at executives of the organization who will be required to interpret and use the results of the computer model, and support its continued use and maintenance. Of all the project documents, this one has to be carefully and clearly written in order to convey the full impact of the technical developments. The report should include a description of the problem setting and origins of the project; a general description of the model, including its purpose, objectives, capabilities and limitations; the nature, interpretation, use and restrictions of the results that are produced by the model; costs and benefits to be expected in using the model; the role of the computer model in the organization and decision structure; resources required; data needs; operational and transfer concerns; and basic explanatory material.

The final document required to be produced is the Model Report.

MODEL REPORT

This report is a nontechnical summary of the basic information that describes the computer model. Its purpose is to provide, in a concise fashion, a description of the computer model to other users and analysts so they may determine if the model is of interest to them. It can be included in other documents such as the User's and Analysts's Manuals, or distributed separately. A typical set of contents is that given above in Section V due to House (42).

It should be clear that the production of the full set of documents described above is a major task for any modeling project. Project managers must make a decision early in the model life cycle as to what documentation is required to meet the objectives of the project. Documentation costs can be quite high.1/ But these costs must be compared to the probable additional costs if adequate documentation is not prepared. As noted in (35) and (46), model usage and transferability are direct functions of the quality and quantity of documentation.

It is hoped that the material presented here will aid model users and developers in advancing the cause of model documentation. Our discussion is meant to serve this cause and thus, contribute to improving the development and use of computer models.

1/ Brewer (64) notes that many computer software companies spend as much money on documentation as for the software itself. In (37), Brewer also suggests the possible creation of a new professional class of documentation specialists.

REFERENCES

(1) "Documentation of Computer Library Routines," M. Wright, Systems Optimization Laboratory, Department of Operations Research, Stanford University, Stanford, CA, June 1976

(2) "Analyzing U.S. Energy Options Using the Pilot Energy Model," S. C. Parikh, Systems Optimization Laboratory, Department of Operations Research, Stanford University, Stanford, CA, October 1976.

(3) "The Design and Structure of a FORTRAN Program Library for Optimization," P. E. Gill et al., Systems Optimization Laboratory, Department of Operations Research, Stanford University, Stanford, CA, April 1977.

(4) "System Description and the Delta Language," E. Holbaek-Hanssen et al., Norwegian Computing Center, Oslo 3, Norway, February 1977.

(5) "The Feasibility of and Methodology for Developing Federal Documentation Standards for Simulation Models," R.E.Nance, Department of Computer Science, VPI and State University, Blacksburg, VA, June 1977.

(6) "Automated Data System Documentation Standards Manual," No. 4120.17-M, Department of Defense, Washington, D.C. December 29, 1972.

(7) "Automated Data System Documentation Standards," SECNAVINST 5233.1A, Department of the Navy, Washington, D.C., June 20, 1973.

(8) "Guidelines for Documentation of Computer Programs and Automated Data Systems," FIPS PUB 38, National Bureau of Standards, Washington, D.C., February 15, 1976.

(9) "Documentation Standards for Computer Systems," N.L. Enger, The Technology Press, Inc., Fairfax Station, VA, 1976.

(10) "Computer Program Documentation," ADA OI-3, Department of the Air Force, Headquarters Armament Development Test Center (AFSC), Eglin Air Force Base, FL, September 26, 1977.

(11) "Required Computer Programming/Coding Standards for Two FBM Weapon System Simulations," R.T. Bevan and J.H. Reynolds, Naval Surface Weapons Center, Dahlgren Laboratory, Dahlgren, VA, November 14, 1977.

(12) "Using the Naval Surface Weapons Center (Dahlgren Laboratory) Computer Assisted Documentation (CAD) Technique," R.T. Bevan, Naval Surface Weapons Center, Dahlgren Laboratory, Dahlgren, VA, September 12, 1977.

(13) "Management Information Systems: Policies, Objectives, Procedures, and Responsibilities," AR 18-1, Headquarters, Department of the Army, Washington, D.C., March 22, 1976.

(14) "ADP Documentation Guidelines," Department of Labor Manual Series, DLMS-9, Data Processing Chapter 400, Washington, D.C., November 15, 1976.

(15) "Software Design and Documentation Language," H. Kleine, JPL 77-24, Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA, July 1, 1977.

(16) "Information Processing Division Standards and Procedures on Software Documentation," the Aerospace Corporation, Los Angeles, CA.

(17) "Computer Program Documentation Guidelines," Assistant Chief of Staff, Studies and Analyses, Headquarters, USAF, Washington, D. C., October 1976.

(18) "An Air Force Guide to Software Documentation Requirements," W.L. Schoeffel, Mitre Corporation, Bedford, MA, June 1976.

(19) "Management Information Systems--Systems Documentation Handbook," HQ Manual Appendix 0850-C, Division of Management Information and Telecommunications Systems, U.S. Atomic Energy Commission, Washington, D.C., 1972.

(20) "TSC Computer Program Documentation Guidelines," Transportation Systems Center, Department of Transportation, Cambridge, MA, 1975.

(21) "Guidelines for the Documentation of Digital Computer Programs," N413(ANS-10.3) Draft, American Nuclear Society, Hinsdale, IL, September 1973.

(22) "Software Documentation for Development and Support of Embedded Computer Resources," ARINC Research Corporation, Annapolis, MD.

(23) "Introduction to the Defense Logistics Studies Information Exchange.(DLSIE)," U.S. Army Logistics Management Center, Ft. Lee, VA, January 1976.

(24) "Department of Defense Catalog of Logistics Models--1977," U.S. Army Logistics Management Center, Ft. Lee, VA, January 1977.

(25) "Documentation of Computer Programs and Automated Data Systems," Proceedings of a Symposium, M.A. Krasny, Ed., National Bureau of Standards, Department of Commerce, July 1977.

(26) "Documenting a System of Programs for the Programmer/Analyst," W.E. Judnick, WP-76-1, Vector Research, Inc., Ann Arbor, MI, March 1976.

(27) "Automatic Data Processing Manual," Environmental Protection Agency, Washington, D.C., March 31, 1975.

(28) "Simulation Notebook Procedures," D.L. Meadows, Thayer School of Engineering, Dartmouth College, Hanover, NH.

(29) "Class II Documentation Standards for Simulation Models," D.L. Meadows, et al., Thayer School of Engineering, Dartmouth College, Hanover, NH.

(30) "Patrol Car Allocation Model: Executive Summary (R-1786/1-HUD/DOJ), User's Manual (R-1786/2-HUD/DOJ), Program Description (R-1786/3-HUD/DOJ)," J.M. Chaiken and P. Dormont, the RAND Corporation, Santa Monica, CA, September 1975.

(31) "Computer Services Documentation Standards," K. Hanson, SRI, Menlo Park, CA, October 1977.

(32) "An Analyst's Guide to TRIM - the Transfer Income Model," M.B. Sulvetta, the Urban Institute, Washington, D.C., June 1976.

(33) "TRIM User's Guide," A. Bergsman et al., the Urban Institute, Washington, D.C., April 26, 1974.

(34) "An Evaluation of the Use of the Transfer Income Model--TRIM--to Analyze Welfare Programs," PAD 78-14, U.S. GAO, Washington, D.C., November 25, 1977.

(35) "Improvement Needed in Documenting Computer Systems," B-115369, U.S. GAO, Washington, D.C., October 8, 1974.

(36) "Auditing a Computer Model: A Case Study (CS-4)," U.S. GAO, Washington, D.C., May 1973.

(37) "Documentation: An Overview and Design Strategy," G.D. Brewer, Simulation & Games, Vol. 7, No. 3, September 1976.

(38) "Activities of the Office of Energy Information and Analysis," Professional Audit Review Team, Washington, D.C., December 5, 1977.

(39) "Tables of Contents: User's Manual, Analyst Manual, Programmer Manual," Control Analysis Corporation, Palo Alto, CA 1977.

(40) "Review of the 1974 Project Independence Evaluation System," OPA-76-20, U.S. GAO, Washington, D.C., April 21, 1976.

(41) "Large-Scale Models for Policy Evaluation," P.W. House and J. McLeod, John Wiley & Sons, N.Y., 1977.

(42) "Diogenes Revisited - The Search for a Valid Model," P.W. House, Simulation, October 1974.

(43) "Stanford - EPRI Workshop for Considering A Forum for the Analysis of Energy Options Through the Use of Models,"Stanford Institute for Energy Studies, Stanford University, Stanford, CA, August 1976.

(44) "Project Independence Evaluation System: Volumes 1-15," Federal Energy Administration, Washington, D.C., 1976.

(45) "Models, Simulations, and Games - A Survey," M. Shubik and G.D. Brewer, R-1060-ARPA/RC, Rand Corporation, Santa Monica, CA, May 1972.

(46) "Federally Supported Mathematical Models: Survey and Analysis," G. Fromm, W.L. Hamilton and D.E. Hamilton, Stock No. 038-000-0024-0, U.S. GAO, Washington, D.C., 1975.

(47) "Review of Selected Army Models," Department of the Army, May 1971.

(48) "Professionalism: The Communication Issue," G.D. Brewer, Yale University, April 1977.

(49) "Implementation of Emergency Service Deployment Models in Operating Agencies," J.M. Chaiken, Rand Corporation. Santa Monica, CA, 1977.

(50) "Why Document?" T.D. Puckorius, pp. 5-9 in (25).

(51) "Ways to Improve Management of Federally Funded Computerized Models," LCD-75-111, U.S. GAO, Washington, D.C., August 23, 1976.

(52) "Computer Program Documentation Guidelines," NHB-2411.1, NASA, Washington, D.C., July 1971.

(53) "'PDL' A Tool for Software Design," S.H. Caine and E.K. Gordon, Cain, Farber and Gordon, Pasadena, CA.

(54) "Software Standards (Draft Version)," Report No. UTP.40.73.3.1, Office of Planning Methods and Support, UMTA, DOT, Washington, D.C., November 16, 1977.

(55) "Tactical Digital Systems Documentation Standards," SECNAVINST 3560.1, Department of the Navy, Washington, D.C., August 8, 1974.

(56) "Requirements for Digital Computer Program Documentation," WS-8506 Revision 1, Naval Ordnance Systems Command, Washington, D.C., November 1, 1971.

(57) "Software Requirements for an Improvement of Transfer and Adaptability of Models," S. Dickhoven, Institute for Planning and Decision Systems, Bonn, Germany, April 1977.

(58) "ANS Standard: Recommended Programming Practices to Facilitate the Interchange of Digital Computer Programs," ANS STD.3-1971, American Nuclear Society, Hinsdale, IL, April 1971.

(59) "Simulation: From Art to Science for Society," J. McLeod, Simulation Today, No. 20, pp. 77-80, December 1973.

(60) "Evaluation of Complex Models," S.I. Gass, Computers and Operations Research, Vol. 4, No. 1, March 1977.

(61) "A Procedure for the Evaluation of Complex Models," S.I. Gass, Proceedings First International Conference on Mathematical Modeling, University of Missouri, 1977.

(62) "Models in the Policy Process," M. Greenberger, M.A. Crenson and B.L. Crissey, Russell Sage Foundation, N.Y., 1976.

(63) "Evaluation of Policy Simulation Models," R.E. Pugh, Information Resources Press, Washington, D.C., 1977.

(64) "Politicians, Bureaucrats, and the Consultant," G. Brewer, Basic Books, N.Y., 1973.

(65) "Automated Documentation System User's Manual," H.H. Sayani et al., ISDOS Working Paper No. 123, Dept. of Industrial and Operations Engineering, University of Michigan, Ann Arbor, MI, July 1975.

(66) "Documentation Standards for the Strategic Environmental Assessment System (SEAS)," R.E. Ubico, Control Data Corporation, Rockville, MD, August 24, 1973.

(67) "Data Element Dictionaries for the Information Systems Interface," E. H. Sibley and H. H. Sayani, Proceedings of the First National Symposium on Management of Data Elements in Information Processing, National Bureau of Standards, Gaithersburg, MD, January 24-25, 1974.

(68) "Hypercube Queuing Model: User's Manual," R.C. Larson, R-1688/2-HUD, the RAND Corporation, Santa Monica, CA, 1975.

(69) "Management of the Embedded Computer Systems Software Acquisition Process," J.K. Watson and Y. Omurtag, Headquarters Aerospace Defense Command, Peterson Air Force Base, Colorado, September 1977.

APPENDIX A

ELEMENTS OF COMPUTER MODEL DOCUMENTATION[1]

1/ Adapted from (20).

ELEMENTS OF COMPUTER MODEL DOCUMENTATION

I. Background Information

A. Model name; including any acronyms or short titles.

B. Model sponsors; major agency and direct organizational unit responsible for the model's development, current owners and users of model.

C. Model developers; specific groups, agencies and/or contractors responsible for the model specifications and its development.

D. Model project director and other responsible personnel; including government contract technical monitor, contractor project leader and other government and contractor personnel who had decision-making responsibilities in developing the model and their areas of responsibility.

E. Model abstract; brief description of the purposes of the model, its objectives, general subject matter, methodological approach, planned utilization in a decision-making environment, designated users and decision makers.

F. Model demographics; model start and completion dates (approximately), development location(s), current location(s) of model, model cost, relationship to other models.

II. Computer Model Documentation

This section should be self-contained and of sufficient detail so as to allow the model user or an evaluator to understand the model application and methodology and to be able to reproduce the computing operating environment and the results of the test case included in this section. The documentation should also enable the user or evaluator to modify data inputs and run the program for the specified ranges of parameters and extreme cases. This section contains the following information:

A. Model name,

B. Abstract,

C. Introduction,

The objective or purpose of the program, with any background information, such as feasibility studies or justification writeups. Summaries that describe the total problem environment and general block or flow diagrams should be included here.

D. Problem of Task Description,

A technical description of the problem to be solved or the task to be accomplished and formal requirement specification, when provided, should be included.

E. Method of Solution,

For scientific applications, this includes the mathematical equations, formulas, and technique used and their logical sequencing within the program. A functional flow chart, such as a block diagram or a logic chart, should be included to provide a pictorial display of this logic. For management or business information systems, this includes system specifications, logical sequencing of events within the program, and again the functional flow chart.

F. Program Description,

The minimum contents under program description includes the following:

1. Operating Environment,

   Includes hardware, software, (system or monitor, source language) machine components (memory requirements and tapes), and library routines used.

2. Program Specifications,

   Includes detailed narrative and graphical description of the programming techniques used in writing the program; i.e., calling sequence, overlay structure, test plan, common usage, etc.

3. Subprogram (Other Than Library),

   Any subprogram or program module which is of such a general nature that it has a potential of being used by future programs should be documented in detail as a separate entity. It can then either be referenced or included in the main program documentation. Special emphasis should be given to the model's data file updating and editing programs. Subprograms which have little or no sharing potential should be documented only to the level of detail necessary for program maintenance and modification. The level of detail would vary, depending on the subprogram's size and complexity. Minimum documentation here would be the function of the subprogram and its calling sequence.

4. Source Listing,

   Should be provided or readily available.

5. Detailed Flow Chart,

   Detailed flow charts should be provided.

6. Personnel and Program Requirements,

   State typical personnel time, set up time, and computer time required to make a typical run and to analyze the output; also, types of skills required.

G. Program Use,

A description of the procedure required to set up and run the computer program. The information should clearly and adequately describe how to fill out the data sheets. For a terminal interactive system, complete logon, logoff, and instructions for interpreting the computer requests during operations and final output must be given. A sample, annotated instructional run should be included. For a batch mode system, instructions must be given as to how to submit the system and data deck, with all its program forms, and how to interpret the results of the program. The following items apply to both a batch and terminal system, except item 1a is for batch and 1b for terminal.

1a. Deck Setup,

   An illustration of the deck/tape setup for a computer run, showing placement of control cards, source and/or object decks, data, restart procedures, etc.

1b Terminal Instructions,

   Logon, logoff procedures, complete set of computer queries with ranges of legal responses, data input formats.

2. Input,

   File, record, and data element descriptions and formats, including the origin of each data element.

3. Output,

Files, records, data element descriptions and formats, in addition to information on scratch or intermediate files.

4. Restrictions and/or Limitations,

Hardware and software restrictions, data ranges and capacities, program behavior when restrictions are violated, and recovery procedures. If accuracy characteristics are significant, they should be described in detail.

5. Editing and Diagnostics,

Includes the "cause" and "cure" of program generated diagnostic messages; all checks and balances performed to ensure accurate and complete output.

6. Test Case,

Includes a listing of input and output, and the machine time required to produce this output.

7. Data and Data Files,

Includes description of the data files, programs, and procedures for updating and editing the data.

H. Symbols and Parameters,

Define all meaningful symbols and arrays used in the routine with reference to the mathematical and/or technical notations and terms used in the problem description. Give units where applicable. If possible, this material should be presented in a tabular form. Values of parameters (e.g., a computational zero, step sizes, convergence factors), nominal and initial values should be described, along with their ranges and discussion as to how they affect the computational process.

I. Appendices,

J. References.

APPENDIX B

GUIDELINES FOR PREPARING USER, PROGRAMMER AND ANALYST MANUALS OF COMPUTER MODELS[1]

1/ From (17).

I. GUIDELINES FOR PREPARING THE USERS MANUAL

A. INTRODUCTION

The Users Manual is the source document for the user of the program. It must contain sufficient information for the user to thoroughly understand the inner workings of the model and to accurately use the model. Occasionally the user will be unfamiliar with data processing and computer techniques. Care must be exercised in the preparation of the textual material to insure that a layman "non-programmer" can understand its contents.

The basic text of the Users Manual contains four sections:

DESCRIPTION OF THE MODEL

INPUT DATA DESCRIPTION

OUTPUT DESCRIPTION

INSTRUCTIONS FOR PREPARING COMPUTER RUNS

The contents of each of these sections are described in the paragraphs that follow.

B. SECTION 1 - DESCRIPTION OF THE MODEL

The description of the model will include the following sub-sections:

GENERAL DESCRIPTION
DESCRIPTION OF METHODOLOGY
LIMITATIONS AND ASSUMPTIONS

The specific information that will be included in each sub-section is as follows:

1. GENERAL DESCRIPTION

Provide a complete overview of the model including its purpose, intended use, general magnitude of model applicability, e.g., major limits and assumptions, inputs and outputs, etc. Also include the relationship of this model to any other models, if applicable; e.g., model X prepares the input data for this model. This overview should give a potential user a very comfortable relationship with the model.

2. METHODOLOGY

Include the intimate details of how the model accomplishes its intended purpose. These details should be provided in the sequence in which they are performed in the model. This section will contain the mathematical formulas, derivations, proofs and sufficient detailed description to determine the reasoning behind the approach and logic of the model. The methodology subsection will provide the basis for substantiating the model results. In other words, the description of the logical manipulation of the input data to produce the model results should be included. A gross flow chart of the logic of the program should also be included to assist in understanding the program design methodology.

3. LIMITATIONS AND ASSUMPTIONS

The overall limits of the model concerning the magnitude of the items considered and the level of applicability of the model should be included. Since the assumptions determine the basis for the program design and establish the program limits, a complete explanation of all assumptions will be provided. Assumptions that also have the effect of restricting the program's utility must also be described.

C. SECTION 2 - INPUT DATA DESCRIPTION

This section will include the following subsections:

GENERAL DESCRIPTION OF INPUT DATA
INDIVIDUAL DATA SET DESCRIPTIONS
DATA SET FORMATS

The specific information that will be included in each subsection is as follows:

1. GENERAL DESCRIPTION OF INPUT DATA

This subsection will include a description of the overall data structure, the nature of the data media (e.g., tape cards, etc.) and any general data limitations. It should also include a description of the correlation between data types.

2. INDIVIDUAL DATA SET DESCRIPTIONS

Input data items are normally organized for input in related groups, such as aircraft characteristics, missile characteristics etc., or as the data items that are input on one punch card. These related groups of data establish and define a data set and should be described together. Each data set description should begin on a new page. Logically related data sets should be described in sequence. The information that should be included in each data set follows and is illustrated in Figure 1, Sample Data Set Description.

- DATA SET NAME - Include the name and or acronym.

- DESCRIPTION - Include an overview of the items included in the data set and the purpose and function of the data.

- MAXIMUM NUMBER OF INPUTS - Include the maximum number of inputs of this type that can be prepared.

- CORRESPONDING INPUTS - Specify the names of the related data sets whose data items are dependent upon the data input values for this data set.

- DESCRIPTION OF DATA SET ITEMS - Include a detailed description of each item in the data set with the following information: item name, card columns reserved for each data item, the minimum and maximum range of values or its fixed value, and a definition of the item to include the use of the item in the program, its relationship to other items, its unit of measurement, source (if fixed), and any information that will assist the user in preparing this input. The definition will also include the requirements for right-justified columns (i.e., right-most character must be lined up in the right-most column reserved for the item).

1. GENERAL DESCRIPTION OF INPUT DATA

This subsection will include a description of the overall data structure, the nature of the data media (e.g., tape cards, etc.) and any general data limitations. It should also include a description of the correlation between data types.

2. INDIVIDUAL DATA SET DESCRIPTIONS

Input data items are normally organized for input in related groups, such as aircraft characteristics, missile characteristics etc., or as the data items that are input on one punch card. These related groups of data establish and define a data set and should be described together. Each data set description should begin on a new page. Logically related data sets should be described in sequence. The information that should be included in each data set follows and is illustrated in Figure 1, Sample Data Set Description.

- DATA SET NAME - Include the name and or acronym.

- DESCRIPTION - Include an overview of the items included in the data set and the purpose and function of the data.

- MAXIMUM NUMBER OF INPUTS - Include the maximum number of inputs of this type that can be prepared.

- CORRESPONDING INPUTS - Specify the names of the related data sets whose data items are dependent upon the data input values for this data set.

- DESCRIPTION OF DATA SET ITEMS - Include a detailed description of each item in the data set with the following information: item name, card columns reserved for each data item, the minimum and maximum range of values or its fixed value, and a definition of the item to include the use of the item in the program, its relationship to other items, its unit of measurement, source (if fixed), and any information that will assist the user in preparing this input. The definition will also include the requirements for right-justified columns (i.e., right-most character must be lined up in the right-most column reserved for the item), the use of leading zeroes (i.e., data item will be padded with zeroes to the left-most column reserved for the item after the most significant character of the input value) and the use or specification of required or implied decimal points. The description of each data item should appear in tabular form as illustrated in Figure 1.

INPUT: Vehicle Parameters

Description: This input specifies the weight, empty or loaded, that will be used for vehicles in a basic run or run variation. This input is not optional and must appear in each run. (It doesn't have to be re-input for each run variation since the first setting will be used for each run variation until another input of its type is encountered).

Maximum Cards: One for the basic run and, if desired, one for each run variation.

Corresponding Inputs: Vehicle Characteristics

Definition of Vehicle Parameters:

ITEM	COLUMNS	RANGE/VALUE	DEFINITION
ID	1-14	"PARAMETERS VEH"	Input type Identifier.
EMPTY WEIGHT OPTION	19-24	Blank/Any Char.	If the empty weight of vehicles is desired, then put any nonzero character in any of the columns.
LOADED WEIGHT OPTION	25-30	Blank/Any Char.	If the loaded weight of vehicles is desired, put any nonzero character in any of the columns.

FIGURE 1

SAMPLE DATA SET DESCRIPTION

DATA SET FORMATS

In addition to the description of each data set, a format layout or creation sheet should also be included. This gives the user a visual reference for preparing the input data. Reference Figure 2 for a Sample Data Set Format.

D. SECTION 3 - OUTPUT DESCRIPTION

This section will include the following subsections:

GENERAL DESCRIPTION OF OUTPUTS
INDIVIDUAL OUTPUT DESCRIPTIONS
SAMPLE OUTPUT FORMAT

The specific information that will be included in each subsection is as follows:

1. GENERAL DESCRIPTION OF OUTPUTS

Describe the overall output structure to include types of output, relationship of one output type to another and the quantity of output that can be expected under varying conditions of input. Also include whether or not certain outputs are optional or always produced for each computer run.

2. INDIVIDUAL OUTPUT DESCRIPTIONS

This subsection should include a detailed description of each output report. The description of each output should start at the beginning of a page and the sequence of the output presentations should be provided in the sequence the output reports are printed. A suggested format for each output report description is as follows:

- OUTPUT TITLE - Include the short and long name.

- DESCRIPTION - Include the purpose and interpretation of the output and its relationship to the overall results of the model.

VEHICLE PARAMETERS

ID																		EMPTY WEIGHT OPTION						LOADED WEIGHT OPTION																	
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42
P	A	R	A	M	E	T	E	R	S		V	E	H																												

FIGURE 2

SAMPLE DATA SET FORMAT

- DESCRIPTION OF OUTPUT REPORT ITEMS - Include in tabular form the name of each output item and an explicit description of each item. Also include any information that will assist in validating this result.

- SAMPLE OUTPUT FORMAT - Output formats should be provided. These can be actual samples of the model output or blank formats showing the output headings and X's showing where the data would appear.

E. SECTION 4 - INSTRUCTIONS FOR PREPARING COMPUTER RUNS

This section of the Users Manual includes the following subsections:

ORGANIZATION OF INPUT DATA
INSTRUCTIONS FOR SUBMITTING RUNS

The information to be included in each of these subsections is as follows:

1. ORGANIZATION OF THE INPUT DATA

This subsection must include the organization required to input the data for executing the model as well as any optional or alternate organizations that allow advantage to be taken of inherent execution flexibilities. Both a narrative explanation and a pictorial representation of the logical organization of the input data will be provided. Figure 3 contains a sample illustration of a data deck organization. Input data sets that are always required and those that are optional must also be recapped in this subsection.

2. INSTRUCTIONS FOR SUBMITTING RUNS

Normally, to execute a model, control cards of a varying nature are required in addition to the data inputs. An explanation of how the total run deck is to be set up must be provided including the details of how to prepare the control cards. A complete listing of the control cards in their proper sequence with the card columns annotated for each item on the cards must be provided. If certain control cards are optional, or if the run can be set up in more than one way, a full explanation will be given for each option. A pictorial representation of the

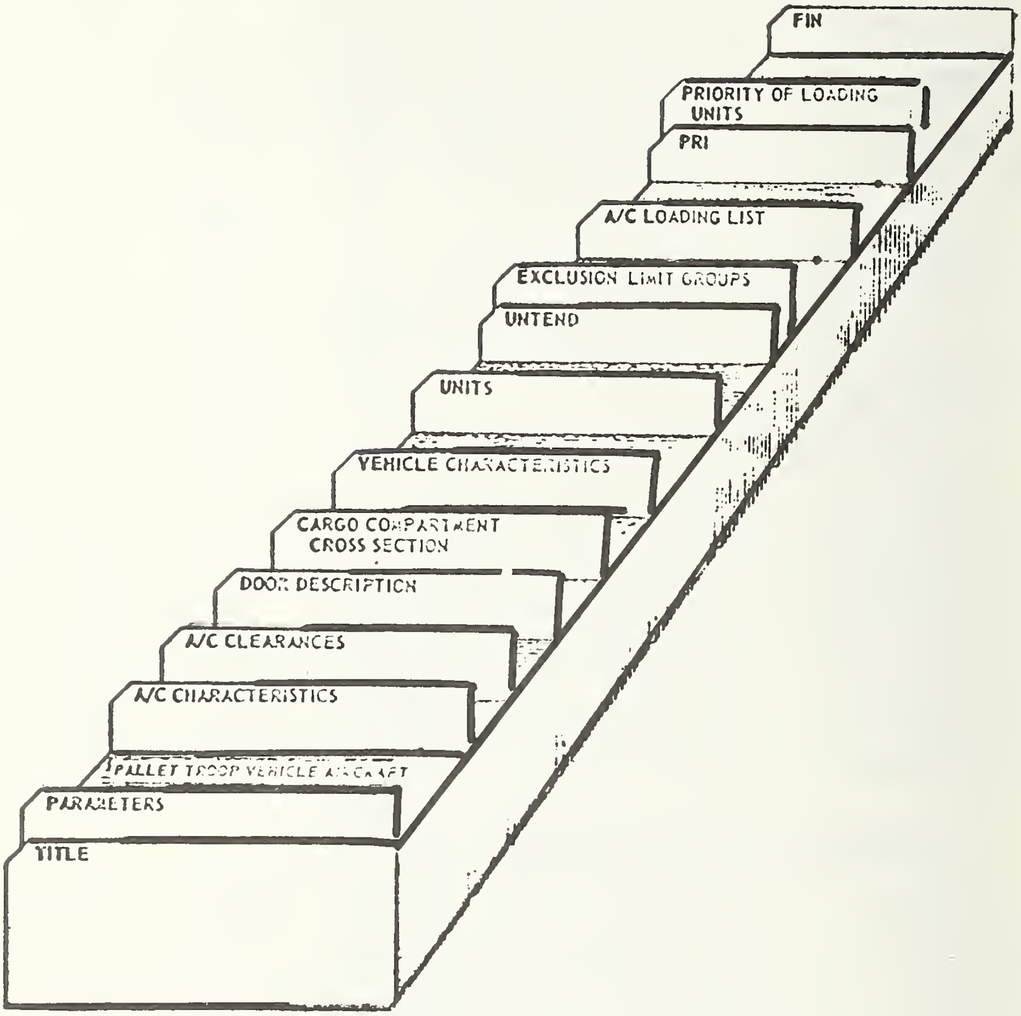

FIGURE 3

SAMPLE DATA DECK ORGANIZATION

integration of the data inputs with the control cards must also be given.

A second general requirement to submit a run for execution is the preparation of a work request form. This form usually requests run specification information such as the estimated execution time, estimated lines of output, input and output tape/disc requirements, core storage requirements, computer language the model is written in and the computer type for which it was written. Therefore, these overall run specifications must also be explained in detail. If the execution time and or number of lines of output vary, then the methods for estimating these variations should be provided.

II. GUIDELINES FOR PREPARING THE PROGRAMMERS MANUAL

A. INTRODUCTION

The primary purpose of this manual is to serve as a guide for modifying the model. The guidance must be provided on two levels, the macro and the micro. The macro level includes an overall discussion of the processing techniques used and their relationships. The micro level contains a detailed description of what happens in each routine. The macro level should give the programmer a grasp of how everything is tied together and highlight those areas of particular vulnerability to change. The micro level provides the factual details about what occurs in each of the parts of the model on an individual basis.

A secondary purpose of this manual is to provide useful information for the programmer who might have to convert the model for a computer other than the one for which the model was originally written. Peculiarities of the model design that are based on the computer hardware and or system software should be discussed with the above purpose in mind.

This manual contains four major sections:

DESCRIPTION OF PROCESSING

DESCRIPTION OF COMMON ARRAYS

DESCRIPTION OF EACH ROUTINE

SOURCE CODE LISTING OF THE MODEL

The specific contents of each of these sections will be described in detail as follows.

B. SECTION 1 - DESCRIPTION OF PROCESSING

The purpose of this section is to provide the macro view of the approach to the model design, i.e., the processing techniques used. The information that is noted here for inclusion can certainly be amplified and should be when more information can be provided to guide a programmer who has never heard of nor seen the model. The aim is to allow the programmer to modify the model properly, so that a change to one routine does not lead to problems in other routines and to provide for a smooth conversion effort. These are difficult tasks but are the true purposes of this section of the Programmers Manual.

The Description of Processing includes the following subsections:

MODEL SPECIFICATIONS
PROCESSING TECHNIQUES
GROSS FLOW CHARTS

The specific information that should be included in each of the subsections is as follows:

1. MODEL SPECIFICATIONS

Include a complete rundown of the computer specifications as follows:

- Computer and operating system for which the model was written.

- Language in which the model was written.

- Number of overlays if any.

- Number of routines - list each routine and provide a brief description of each one of them.

- Number of core locations.

- Execution time.

- Input/output device requirements.

- Use of standard/nonstandard system/library routines.

If there are any peculiarities about any of the above specifications, they should be discussed. For example, if the model was written in the FORTRAN programming language, the nonstandard features of the FORTRAN compiler that are peculiar to the operating system or hardware that was used in the model development should be described. This subsection is really of key importance to the programmer who has to convert the model for another computer. Therefore, the more detail included concerning the model specifications the better.

2. PROCESSING TECHNIQUES

This section describes the program processing design specifications from the vantage point of post-model development. It will include the overall approach used to develop the model, highlighting the design parameters based on the hardware or system software capabilities. For example, since core storage in the G-635 computer system is organized in 36 bit fixed word size and accommodates six characters per word, processing the input data of a model may be organized around this word configuration. It is this type of information that should be included.

If the model is overlayed, the decisions that were made to determine what routines could be overlayed and how they were to be overlayed when the model was designed should be included. Also include the overall control flow of the overlays and their interaction in word or flow chart form or both. If the model is not overlayed, describe the overall processing flow. Other special areas that should be covered are those of initialization and wrap-up. These two areas are generally overlooked, but deserve attention if like tasks accomplished in the normal processing flow are done in a different manner in these two areas. Packed or unpacked features of the model data should be described. If special consideration was given to determining the length of any of the major arrays, these considerations should also be mentioned.

3. GROSS FLOW CHARTS

These charts may be developed as a part of the previous subsection or be located in one central location in the manual. Gross flows should contain operational terminology, not the language of the program. They are more meaningful in this form to a programmer, or to an analyst who might be curious about the actual processing techniques employed by the model. Of course, routine/variable names should be used with a brief description of what they do to tie the charts to the hard cold code. The gross flows provide an overall guide for walking through the code to show the interactive process flow of all routines in the model. They should also depict special purpose areas such as model wrap-up procedures or end-of-file conditions.

C. SECTION 2 - DESCRIPTION OF COMMON ARRAYS

All arrays, variables, tables, data sets, etc., that are shared by more than one routine should be defined. The array name should be provided with a general description of the function of the array. All indices should be defined (both implied and specified), followed by a definition of all elements in the array. A list of the routines that use and/or modify this array should also be provided.

D. SECTION 3 - DESCRIPTION OF EACH ROUTINE

Each routine description should include the purpose and/or function of the routine, i.e., its role in the overall processing logic of the model. A narrative description of the flow is not necessary, and should only be included if it amplifies the code and points out any subtleties that might be lying therein. Flow charts are also optional, but are helpful when the routine is very large and when there are few meaningful comments in each routine.

E. SECTION 4 - SOURCE CODE LISTING OF THE MODEL

If the model or routine is of reasonable size, then the source listing should be included as a section of this manual. If it is large and unwieldy, it should be bound separately and made available upon request.

III. GUIDELINES FOR PREPARING THE ANALYSTS MANUAL

A. INTRODUCTION

The Analysts Manual is the source document for the Analysts using the model. The text shall be factual, concise, clearly worded and illustrated. Sentence form will be simple and direct. Technical knowledge reflected in the manual should be converted into the most easily understood wording possible Discussions of theory shall be omitted except where essential for practical understanding and application. Phraseology requiring a specialized knowledge of computer programming or computers shall be avoided except where no other wording will convey the intended meaning. Where necessary, a "Glossary of Terms" will be added immediately following the Table of Contents. The primary emphasis will be placed upon the specific steps to be followed, the results which may be expected or desired, and the corrective measures required when such results are not obtained.

The basic text of the Analysts Manual contains six sections.

MODEL FUNCTION DESCRIPTION
DATA COLLECTION REQUIREMENTS
INPUT DATA
OUTPUT DATA
COMPUTER RUN ABORT NOTES
TECHNICAL NOTES

The description of each of these sections are described in the paragraphs that follow.

B. SECTION 1 - MODEL FUNCTION DESCRIPTION

The description of the model will contain the following subsections:

A. GENERAL
B.. ASSUMPTIONS AND LIMITATIONS
C. FLEXIBILITY

The specific information that will be included in each subsection is as follows:

1. GENERAL

Provide a summary description of the model including its purpose and intended use. This overview should be sufficiently detailed to give the analyst a very comfortable understanding of the model.

2. ASSUMPTIONS AND LIMITATIONS

A complete identification of all assumptions inherent in the model will be made. Each will be discussed with the view of informing the analysts of the restrictions with which he will be faced in using the model and in interpreting the outputs. A discussion of the overall limits of the model on the magnitude of the items considered and the level of applicability of the model should be included.

3. FLEXIBILITY

Provide a description of the capability for adapting the program to changing requirements, such as anticipated operational changes, interaction with new or improved programs, and planned periodic changes. Components and procedures designed to be subject to change will be identified. Limitations on the flexibility of the model will also be included.

C. SECTION 2 - DATA COLLECTION REQUIREMENTS

This section will include the following subsections:

SCOPE
INPUT SOURCES OF DATA
SUPPORT PROGRAMS

The specific information that will be included in each subsection is as follows:

1. SCOPE

This subsection will describe the types of information required by the program in order to establish the data values of each data element. It shall discuss, as a minimum, those types of information needed to describe the data element in accordance with the information required for the data element library. It shall also specify information to be collected by the user, logically grouped and presented in a manner which will enable the user to make an effective response to the program's requirements for data elements.

2. INPUT SOURCES OF DATA

This subsection will name recommended sources from which the data elements should be gathered. The source of origin of the data will be, for example, a document, an organizational unit, etc. Recommendations as to whom should be responsible for providing specific data inputs will, also, be stated here. This will include recommendations regarding the establishment of a user input reporting organization, if required. Those data inputs dependent on interfacing systems, unrelated agencies, or specific documents should be the source defined. Specific instructions for data collection procedures shall be given.

3. SUPPORT PROGRAMS

All of the support programs available for handling the entry of data into the data base will be discussed briefly. Descriptions shall include program name, functions, and major program operating considerations.

D. SECTION 3 - INPUT DATA

This section will include the following subsections:

DATA ELEMENT PREPARATION
DATA BASE IMPACTS
ACCURACY AND VALIDITY

The specific information that will be included in each of the subsections is as follows:

1. DATA ELEMENT PREPARATION

This subsection will provide a detailed description of all computer program inputs. There shall be a description of each type of input applicable to the program. Each input shall be categorized by the types applicable to the program (such as parameter, data type A, B, etc.) and described in detail to include the following, as applicable:

a. Title and tag.

b. Format and acceptable range of values.

c. Number of items.

d. Description of each item to include number and type of characters (numeric, alpha, decimal, signed, unsigned), range of values, accuracy requirements.

e. Means of entry and initiation procedures; e.g., typewriter, card, tape, internal.

f. Flexibility, such as capability of omitting and adding items.

In addition, the following supplementary information shall be given for each data element where applicable:

a. Critical Value. Many elements that have a range of values will have one value that is particularly significant to the analyst. This may be a breakpoint, a minimum stock level, a critical wind velocity, etc. When applicable, the critical value and its significance to the analyst should be included.

b. Scales of Measurement. For numeric scales, if the successive steps are not equal to "one" on the units of measurement, then the increment shall be specified. For example, a data element representing pressure in pounds per square inch (the unit of measurement) may be incremented by pound, half-pound, or ten-pound intervals. For numeric scales, the "scale zero" should be specified if it is not implicit in the units of measurement: e.g., pressure in pounds per square inch may be measured relative to absolute zero pressure or to atmospheric pressure. For numeric scales where averaging functions other than the arithmetic mean are appropriate, these other functions (e.g., geometric mean, root mean square, harmonic mean) should be indicated. For non-numeric scales, any relationships indicated by the legal values should be stated if not otherwise specified. For example, a code indicating lubricant type of values A, B, C, etc., should show whether the value is arbitrary (A = paraffin oils, B = graphite, etc.) or whether it indicates the ordering of the lubricants by some parameter (A = viscosity less than SAE 10, B = viscosity SAE 10 through SAE 40, etc.).

c. Conversion Factors. Measured quantities that must go through analog digital conversion processes shall have the conversion factors specified.

d. Cross-References. Many input values are so closely related that a change in one value should cause a change in another. Describe the relationship of each input with other inputs, i.e., if the input value of A is changed, must the values of X, Y, or Z also be changed. Build a table of cross-references for each input value: e.g.

	A	B	C	.	.	.	.	.	.	.	.	.	Z
A	X	-	X	.	.	.	.	.	.	.	.	.	X
B	-	X	-	.	,	.	.	.	.	.	.	.	X
C	X	-	X	.	.	.	.	.	.	.	.	.	
.	.	.	.	.	.	.	.	.	.	.	.	.	
.	.	.	.	.	.	.	.	.	.	.	.	.	
Z	X	.	.	.	.	.	.	.	.	.	.	.	

where A, B, C . . . Z are data item names and an X in any row alerts the analyst to check the value of the data item indicated for the column. For example, A might be an aircraft name and C might be its weight and Z its fuel co-efficients.

2. DATA BASE IMPACTS

This subsection will describe the impacts associated with collection and maintenance of the data base on equipment, software, organizational, operational, and developmental environments. Impacts on the system resulting from deficiencies in the data base shall also be given.

3. ACCURACY AND VALIDITY

This subsection will provide a description of accuracy requirements imposed on the system (i.e., the program and computer system). The following accuracy requirements must be considered:

a. Accuracy requirements of mathematical calculations.

b. Logical and legal accuracy of alphanumeric data.

c. Accuracy of transmitted data.

E. SECTION 4 - OUTPUT DATA

This section will include the following subsections:

OUTPUT DESCRIPTION
OUTPUT FORM/DEVICE
ACCURACY OF OUTPUT

The specific information that will be included in each subsection follows:

1. OUTPUT DESCRIPTION

This subsection will provide a detailed description of all program outputs. A description of each type of output applicable to the program will be included. Each output item will be categorized by type (such as data type A, B, etc.) and described in detail to include the following as applicable:

a. Title and tag.

b. Format to include headings, line spacing, arrangement, totals, etc.

F. SECTION 5 - COMPUTER RUN ABORT NOTES

This section will include the following subsections:

ABORT CODES AND MEANING
WARNING NOTICES AND MEANING

The specific information that will be included in each subsection follows:

1. ABORT CODES AND MEANING

This subsection will identify all O/S System abort codes as well as codes built into the program. The meaning of the code will be clearly defined. Action to be taken by the analyst (including consultation with Programmer or Operator) will be noted.

2. WARNING NOTICES AND MEANING

This subsection will identify all O/S warning notices as well as warning notices put out by the program. The meaning of each notice and possible effects on the operation of the program and on outputs from the program will be clearly identified. A notation will be made as to whether or not the warning notice can be safely ignored by the analyst.

G. SECTION 6 - TECHNICAL NOTES

This section will describe all algorithms in the model necessary for the analyst's understanding of how the computer program uses the input data to calculate information reflected in the output. For example, the method of calculating the fuel consumption of the aircraft will be fully described. Other examples include: description of algorithm used for vectoring an interceptor against an intruder, description of algorithm used for calculating impact points of weapons, etc. Input values used in calculations will be specifically identified by name. Headings will be used for each algorithm such as "Fuel Consumption Calculations", "Weapon Impact Point Calculations", etc. As many headings as necessary will be used and will be of the following form:

1. FUEL CONSUMPTION CALCULATION
2. WEAPON IMPACT POINT CALCULATIONS
etc.

Depending upon the length of this section, it may be structured as a separate volume, or it (or its subsections) may be structured as an appendix(ces) to this volume.

c. Number of items.

d. Description of each item to include number and type of characters (alpha, numeric, symbol, etc.), range of possible values.

e. Data selection criteria will be presented to establish the basis for selecting information for display.

f. Description of plots or graphic displays will include the coordinates used, symbols used, type of graphic technique (i.e., points or continuous), number of curves per sheet, etc.

g. Means of display; e.g., CRT, printer, typewriter, projector, plotter.

h. Length of output, including special handling requirements due to variations in length (such as pages of printout, feet of paper tape, etc.).

2. OUTPUT FORM/DEVICE

Output data elements may be presented to the user symbolically, graphically, or may be used as input to some other automated system. If the user is to receive a sensible presentation, the description should specify whether the user will receive the data element as part of a hard copy printout, a symbol in a CRT display, a line on a drawing, etc. Any limitation on the presentation due to the form of display will be discussed.

3. ACCURACY OF OUTPUT

This subsection will discuss the accuracy of the output values in relation to the method of their derivation. A review of the algorithms used for their derivation may be necessary at this point. Any limitations on the use of the data by the analyst should be discussed. Any peculiarities of the system that may result in erroneous output (such as, overflows causing loss of significant digits in some tables) should be noted. Corrective actions to be taken by the analyst in the cases noted should be described (i.e., should he contact the programmer to modify the program, or can he modify the program or modify his input data to take care of the problem). Any corrective action described must be clearly stated so that there is little chance for error on the part of the analyst.

APPENDIX C

DETAILED TABLES OF CONTENTS FOR USER, PROGRAMMER AND ANALYST MANUALS OF COMPUTER MODELS[1/]

1/ From (39).

USER'S MANUAL

TABLE OF CONTENTS

PROGRAMMER MANUAL

TABLE OF CONTENTS

ANALYST MANUAL

TABLE OF CONTENTS

ANALYST MANUAL - TABLE OF CONTENTS (continued...)

ANALYST MANUAL - TABLE OF CONTENTS (continued...)

APPENDIX D

SIMULATION: FROM ART TO SCIENCE FOR SOCIETY[1/]

1/ From (59).

SIMULATION TODAY

The Society for Computer Simulation (*Simulation Councils, Inc.*) P.O. Box 2228, La Jolla, Ca. 92037

SIMULATION: FROM ART TO SCIENCE FOR SOCIETY

by
John McLeod

ABSTRACT

Simulation can be a more effective tool for the study of problems of our times if the practitioners will adopt procedures which will make simulation more a science than an art. A big step in that direction would be the development and use of a standard format that would assure adequate and uniform documentation. Such a format is proposed, and criticism is solicited.

INTRODUCTION

The fact that simulation today is still more an art than a science is unfortunate for many who could use simulation to advantage. Although the "man in the street" doesn't know it matters (he is in no mood to trust either simulation or science), it would be well if we practitioners of the art start applying more scientific principles in our work.

Simulation, even in its current state of development, is important. And it is becoming more so. There is no more powerful tool to help us understand the current and future problems of our society. However, so long as practitioners of the art continue to do their own thing in their own way, and report their work in such a way that it is difficult or impossible for others to check or repeat their work—or don't report it at all—simulation will remain an art. That may be all right for the artist, but it does nothing to inspire our decision-makers to have confidence in us. Before we can sell—and not oversell—simulation to those who can use it to help alleviate the problems of our society, we must take steps to discourage simulationists from riding off in all directions. This will require better communication among all concerned. By better communication, I don't mean just more comprehensive communication, but more comprehensible communications. Complete and uniform documentation will be a prime requisite.

The documentation of the development of a model and of simulation runs should be complete enough to allow

The reporter's peers to evaluate his work
Others to repeat the experiment
Others to build on the work reported instead of having to repeat it—or start over in their own way.

This is not a new idea. This writer and others have made efforts in that direction in the past.[1] Some have listened, but most simulationists, independent thinkers that they are, have continued to build individualized models to suit their own purposes (and too often, consciously or otherwise, their own egos). This tendency is detrimental to our long-range interests. If simulation is to become a respected and accepted tool for the study of important problems, it must be made more of an organized discipline. We know how to wield one of the most powerful tools that technology has placed at the disposal of society. I think we should take our responsibility for its further development and proper use very seriously.

I was obviously groping for a more professional way of reporting our work when I wrote that article in 1970[1] urging uniform documentation. We've come a long way since then, but I'm still groping, still looking for a better way in a field that is developing fast—but not with as much discipline as it should have. Proper documentation of a simulation project will establish a benchmark, a point of departure from which one can determine how far he has progressed, and to which he can return if he gets lost.

I would now like to update my thinking of more than three years ago and propose a more complete format for documentation.

OUTLINE

The documentation format should cover the following items describing the overall project, each model developed under the project, and each simulation for which each model is used.

1. *Project information*
   - 1.1 Project title
   - 1.2 Responsible organization
   - 1.3 Contact
   - 1.4 Project objective
   - 1.5 Project duration
   - 1.6 Funding
     - 1.6.1 Source
     - 1.6.2 Amount
     - 1.6.3 Period

2. *Model development information*
   - 2.1 Name of model
   - 2.2 Name of modeler(s)
   - 2.3 Purpose for which model was developed
     - 2.3.1 Specific
     - 2.3.2 General
   - 2.4 Discipline(s) involved
     - 2.4.1 Primary
     - 2.4.2 Supporting
   - 2.5 Data requirements
   - 2.6 Method of development
   - 2.7 Assumptions
   - 2.8 Cost of development
   - 2.9 Availability
     - 2.9.1 To developer
     - 2.9.2 To others
   - 2.10 Compatibility
     - 2.10.1 Computer system used in development
     - 2.10.2 Other systems used
     - 2.10.3 Language(s)
   - 2.11 Extent of use
     - 2.11.1 By developer
     - 2.11.2 By others

3. *Description of model*
   3.1 Classification of model
      3.1.1 Focus
      3.1.2 Scope
      3.1.3 Sophistication
   3.2 Block diagram of system modeled
   3.3 Program flow or wiring diagram of model
   3.4 Notation
   3.5 Validation
   3.6 Reference information
   3.7 Distinctive features
   3.8 Antecedents of model
   3.9 Current related models

4. *Simulation(s)*
   4.1 Title
   4.2 Purpose
   4.3 Assumptions
   4.4 Experimental design
   4.5 Data requirements
   4.6 Data used
   4.7 Run time
   4.8 Cost per run
   4.9 Results
   4.10 Justifications of assumptions
   4.11 Analysis

5. *Discussion*
   5.1 Comments
   5.2 Conclusions

6. *Literature*
   6.1 Project reports
   6.2 References
   6.3 Bibliography

Obviously, it would be foolish to expect rigorous adherence to any such format. Some of the information requested may be considered "sensitive," while in other cases it might not be of sufficient pertinence to warrant inclusion. And in some instances the information may simply not be available. But if simulationists will use the proposed format—or something like it which others might propose—as a checklist to aid in the preparation of more complete and more uniform documentation, it will certainly go a long way toward making simulation, if not a science, at least a more respectable technology.

SUGGESTIONS FOR USE OF OUTLINE

1. *Project information*

   1.1 *Project title*
   This should be the title of the overall project of which the modeling or simulation may be just a part.

   1.2 *Responsible organization*
   This is the name and address of the organization responsible for the overall project. If the project is supported by an external source (e.g., by a grant), this would be the organization responsible for the money and equipment furnished.

   1.3 *Contact*
   This is the person or persons to contact for further information. If the address is other than that given in 1.2, the full mailing address should be given. The organizational mail-stop or code and telephone number and extension should also be given.

   1.4 *Project objective*
   This is the objective of the overall project, which may be of greater scope than that of the modeling or simulation to be covered later.

   1.5 *Project duration*
   Give date that project was established and expected completion date.

   1.6 *Funding*

      1.6.1 *Source*
      Give name of funding organization.

      1.6.2 *Amount*
      Give total dollars. If equipment is contributed, list major items or estimate value.

      1.6.3 *Period*
      Give dates covered by support listed in 1.6.2

2. *Model development information*

   2.1 *Name of model*
   This might be the computer-callable name of the program. If an acronym, spell it out (e.g., WLDREC: WorLD RECycling model).

   2.2 *Name of modeler(s)*

   2.3 *Purpose for which model was developed*
   "The same" here will indicate the same as 1.4.

      2.3.1 *Specific*
      Give reason(s) for developing model (e.g., to test hypothesis that . . .).

      2.3.2 *General*
      Give other uses to which the model has been or might be put (e.g., to study other problems related to . . .).

   2.4 *Disciplines involved*
   These need not be fields of endeavor recognized as distinct disciplines (e.g., economics), but may be more descriptive of the work (e.g., land use).

      2.4.1 *Primary*
      Give the discipline(s) that the model was primarily developed to serve (e.g., international relations).

      2.4.2 *Supporting*
      List other disciplines required in the development of the model, preferably in descending order of importance.

   2.5 *Data required*
   Give kind of data (e.g., population) and source (e.g., census).

   2.6 *Method of development*
   Give method of development (e.g., theoretical, empirical, other).

   2.7 *Assumptions*
   List all assumptions concerning both data and causality that led to the model's being developed in the way it was.

   2.8 *Cost of development*
   Give actual or estimated total cost of the model and what the cost includes.

2.9 *Availability*

2.9.1 *To developer*
Is the model operative? What will it take to make it operative?

2.9.2 *To others*
Is the model proprietary or classified? Can it be obtained by others? How? In what form (e.g., computer listing, deck, paper or magnetic tape, other)? What are the charges?

2.10 *Compatibility*

2.10.1 *Development of computer system*
On what equipment was the model developed?

2.10.2 *Other systems*
On what other computer systems has it been or might it be run with negligible change?

2.10.3 *Language(s)*
In what language(s) was the program written? Is it available in others?

2.11 *Extent of use*

2.11.1 *By developer*
What actual use has been made of the model by the developer? What use is planned?

2.11.2 *By others*
Has the model been used by others? By whom? To what extent?

3. *Model description*

3.1 *Model classification*
What kind of model is it? How is it run—batch or interactively? Locally or remotely? Is the computer time-shared?

3.1.1 *Focus*
Give the primary fields of interest that the model serves (e.g., political science, resource usage, etc.).

3.1.2 *Scope*
Give entity modeled (e.g., an industrial plant, a river basin, the U.S. Senate, etc.).

3.1.3 *Sophistication*
Where does the model fit in the "Fuzz to Fact"[2] spectrum (e.g., preliminary studies, evaluating alternatives, predicting the future)?

3.2 *Block diagram of system modeled*
This should have a block for each component of the real-world system modeled, and show lines between the blocks indicating the causal relationships of the components as well as exogenous inputs and outputs.

3.3 *Program or wiring diagrams*
A program flow diagram should be shown in the case of digital models, a wiring diagram in the case of analog, and both in the case of a hybrid model.

3.4 *Notation*
A complete description of the notation used in 3.2 and 3.3, as well as any narrative description, should be included here. The notations and definitions must be carefully checked for consistency throughout the documentation.

3.5 *Validation*
Describe how the model was validated.

3.6 *Reference information*
This should be a computer listing of the program and the output of a standard check run for a digital model, a plot of a standard check run for an analog, or both in the case of a hybrid model. It should give all "numbers" used to set up the run, and be annotated in such a way that either the developer at a future date, or another user of the model, can make sure that if the model is to be rerun or used by someone else, he will be working with the model that he thinks he is working with.

3.7 *Distinctive features*
How does the model differ from related models? How is it better? What are its limitations? What are the possible pitfalls that might be encountered in its use?

3.8 *Model antecedents*
Have similar models been built before? If so, give proper credit. Is the current model a follow-on or a distinct "mutant"?

3.9 *Current relations*
Do other models exist that have the same or a closely related purpose? How does this model relate to them? Are they another attempt to solve the same problem, or can the results be expected to be complementary, i.e., to present two aspects of a larger problem? Are there possibilities of online interconnection and interaction between the models?

4. *Simulation(s)*

A simulation will be taken to mean an *experiment* performed on a model instead of the real-world simuland. Multiple runs using the same experimental design may be considered one simulation. However, if the design of the experiment or the procedure is changed, it should be considered another experiment and items 4.1 through 4.11 should be covered again.

4.1 *Title*
This should be descriptive of the simulation experiment, and may be made up of the model name plus a subtitle (e.g., "WLDREC: Effect of cost of recycling").

4.2 *Purpose*
This is the purpose of the individual experiment.

4.3 *Assumptions*
List assumptions made in the design of the experiment.

4.4 *Experimental design*
This should give the procedure to be followed in the experiment, step by step.

4.5 *Data requirements*
Give the data requirements for the individual simulation run(s) that differ from those for the model's reference run (item 3.6).

4.6 *Data used*
This might best be a computer listing of those lines of data that differ from the reference run.

4.7 *Run time*
This should be total as well as mainframe time.

4.8 *Cost per run*
This should be given for both mainframe and peripherals.

4.9 *Results*
This can be "raw data" (e.g., a computer printout) and plots, graphs, etc., prepared by hand or machine.

4.10 *Justification of assumptions*
This is most important, and should be done before any analysis of the results is attempted. Assumptions that influenced the development of the model as well as those related to the specific simulation experiment should be considered.

4.11 *Analysis*
Describe conclusions drawn from the results, and give reasoning where the conclusions are not obvious.

5. *Discussion*

5.1 *Comments*
Add any comments here that might further illuminate aspects of the project not covered elsewhere or, if preferred, give a brief narrative description for the benefit of the casual reader.

5.2 *Conclusions*
Relate development of the model and the simulation experiments to the overall project objective given in 1.4.

6. *Literature*

6.1 *Project reports*
List reports, presentations, and articles generated by the project.

6.2 *References*
List publications actually referred to in the documentation.

6.3 *Bibliography*
List publications which influenced the work documented or which are closely related to it.

CONCLUSION

I realize that just scanning the foregoing list of suggested items might stop some potential users before they start. If so, they should fill out the easy parts, then look at the blanks. For each part they have skipped, they should ask themselves, *Is this information necessary for a person unfamiliar with the details of my work to understand it? Is it information that would be necessary to reconstruct the model and rerun the simulation?* If the answers are *No*, the information may be left out with impunity. If, however, the answer to either question is *Yes*, then every effort should be made to give the necessary information. On the other hand, if more information than is called for by this format would be necessary to repeat the experiment, then that information should be added.

I will appreciate suggestions for additions, or justifiable deletions, or just plain comments.

REFERENCES

1 McLEOD J
Toward Uniform Documentation—PHYSBE and CSMP
SIMULATION May 1970 pp 215-220

2 McLEOD J
Simulation Today—from Fuzz to Fact
SIMULATION March 1973 center section

NBS-114A (REV. 9-78)

U.S. DEPT. OF COMM. BIBLIOGRAPHIC DATA SHEET	1. PUBLICATION OR REPORT NO. NBS SP 500-39	2. Gov't. Accession No.	3. Recipient's Accession No.

4. TITLE AND SUBTITLE	5. Publication Date
COMPUTER SCIENCE & TECHNOLOGY: COMPUTER MODEL DOCUMENTATION: A REVIEW AND AN APPROACH	February 1979
	6. Performing Organization Code
7. AUTHOR(S) Saul I. Gass, Consultant	8. Performing Organ. Report No.
9. PERFORMING ORGANIZATION NAME AND ADDRESS	10. Project/Task/Work Unit No.
Potomac, Maryland	11. Contract/Grant No. PO 713006
12. SPONSORING ORGANIZATION NAME AND COMPLETE ADDRESS *(Street, City, State, ZIP)* Institute for Computer Sciences and Technology, National Bureau of Standards, Washington, D. C. 20234	13. Type of Report & Period Covered: Final
	14. Sponsoring Agency Code

15. SUPPLEMENTARY NOTES

Library of Congress Catalog Card Number: 78-606146

☐ Document describes a computer program; SF-185, FIPS Software Summary, is attached.

16. ABSTRACT *(A 200-word or less factual summary of most significant information. If document includes a significant bibliography or literature survey, mention it here.)*

Recent studies and surveys have concluded that, in general, the documents produced to support the understanding and use of computer models are inadequate. This paper describes the issues and concerns of computer model documentation and proposes an approach for the development of adequate documentation. First, a number of documentation studies and reports are reviewed, including software documentation guidelines and model documentation procedures. Then, based on the relationship between the phases of the model life cycle and documentation information needs, a set of documents is proposed and described.

The author takes a highly critical view of the past and present inadequate state of documentation procedures for computer models. The attention of computer model sponsors and developers must be directed to this area. Otherwise, the author feels, there will be an unfortunate decline in the use of decision models as aids in the analysis of important policy issues. The course of action recommended in this report is an extreme position as to the total information and number of documents required to produce adequate documentation. The author calls for the capturing of all information generated during a model's life cycle. Further research is needed to adapt this extreme position to the realities of cost, resources, model complexity, and model use.

17. KEY WORDS *(six to twelve entries; alphabetical order; capitalize only the first letter of the first key word unless a proper name; separated by semicolons)*

Computer; computer model; documentation; documentation procedures; model; model documentation.

18. AVAILABILITY ☒ Unlimited	19. SECURITY CLASS (THIS REPORT)	21. NO. OF PRINTED PAGES
☐ For Official Distribution. Do Not Release to NTIS	UNCLASSIFIED	89
☒ Order From Sup. of Doc., U.S. Government Printing Office, Washington, DC 20402, SD Stock No. SN003-003-	20. SECURITY CLASS (THIS PAGE)	22. Price $2.75
☐ Order From National Technical Information Service (NTIS), Springfield, VA. 22161	UNCLASSIFIED	

USCOMM-DC

☆ U.S. GOVERNMENT PRINTING OFFICE: 1979 O—281-067 (383)

ANNOUNCEMENT OF NEW PUBLICATIONS ON COMPUTER SCIENCE & TECHNOLOGY

Superintendent of Documents,
Government Printing Office,
Washington, D. C. 20402

Dear Sir:

Please add my name to the announcement list of new publications to be issued in the series: National Bureau of Standards Special Publication 500-.

Name ____

Company ____

Address ____

City ______________________ State ______________ Zip Code __________

(Notification key N-503)

NBS TECHNICAL PUBLICATIONS

PERIODICALS

JOURNAL OF RESEARCH—The Journal of Research of the National Bureau of Standards reports NBS research and development in those disciplines of the physical and engineering sciences in which the Bureau is active. These include physics, chemistry, engineering, mathematics, and computer sciences. Papers cover a broad range of subjects, with major emphasis on measurement methodology, and the basic technology underlying standardization. Also included from time to time are survey articles on topics closely related to the Bureau's technical and scientific programs. As a special service to subscribers each issue contains complete citations to all recent NBS publications in NBS and non-NBS media. Issued six times a year. Annual subscription: domestic $17.00; foreign $21.25. Single copy, $3.00 domestic; $3.75 foreign.

Note: The Journal was formerly published in two sections: Section A "Physics and Chemistry" and Section B "Mathematical Sciences."

DIMENSIONS/NBS

This monthly magazine is published to inform scientists, engineers, businessmen, industry, teachers, students, and consumers of the latest advances in science and technology, with primary emphasis on the work at NBS. The magazine highlights and reviews such issues as energy research, fire protection, building technology, metric conversion, pollution abatement, health and safety, and consumer product performance. In addition, it reports the results of Bureau programs in measurement standards and techniques, properties of matter and materials, engineering standards and services, instrumentation, and automatic data processing.

Annual subscription: Domestic, $11.00; Foreign $13.75

NONPERIODICALS

Monographs—Major contributions to the technical literature on various subjects related to the Bureau's scientific and technical activities.

Handbooks—Recommended codes of engineering and industrial practice (including safety codes) developed in cooperation with interested industries, professional organizations, and regulatory bodies.

Special Publications—Include proceedings of conferences sponsored by NBS, NBS annual reports, and other special publications appropriate to this grouping such as wall charts, pocket cards, and bibliographies.

Applied Mathematics Series—Mathematical tables, manuals, and studies of special interest to physicists, engineers, chemists, biologists, mathematicians, computer programmers, and others engaged in scientific and technical work.

National Standard Reference Data Series—Provides quantitative data on the physical and chemical properties of materials, compiled from the world's literature and critically evaluated. Developed under a world-wide program coordinated by NBS. Program under authority of National Standard Data Act (Public Law 90-396).

NOTE: At present the principal publication outlet for these data is the Journal of Physical and Chemical Reference Data (JPCRD) published quarterly for NBS by the American Chemical Society (ACS) and the American Institute of Physics (AIP). Subscriptions, reprints, and supplements available from ACS, 1155 Sixteenth St. N.W., Wash., D.C. 20056.

Building Science Series—Disseminates technical information developed at the Bureau on building materials, components, systems, and whole structures. The series presents research results, test methods, and performance criteria related to the structural and environmental functions and the durability and safety characteristics of building elements and systems.

Technical Notes—Studies or reports which are complete in themselves but restrictive in their treatment of a subject. Analogous to monographs but not so comprehensive in scope or definitive in treatment of the subject area. Often serve as a vehicle for final reports of work performed at NBS under the sponsorship of other government agencies.

Voluntary Product Standards—Developed under procedures published by the Department of Commerce in Part 10, Title 15, of the Code of Federal Regulations. The purpose of the standards is to establish nationally recognized requirements for products, and to provide all concerned interests with a basis for common understanding of the characteristics of the products. NBS administers this program as a supplement to the activities of the private sector standardizing organizations.

Consumer Information Series—Practical information, based on NBS research and experience, covering areas of interest to the consumer. Easily understandable language and illustrations provide useful background knowledge for shopping in today's technological marketplace.

Order **above** *NBS publications from: Superintendent of Documents, Government Printing Office, Washington, D.C. 20402.*

Order **following** *NBS publications—NBSIR's and FIPS from the National Technical Information Services, Springfield, Va. 22161.*

Federal Information Processing Standards Publications (FIPS PUB)—Publications in this series collectively constitute the Federal Information Processing Standards Register. Register serves as the official source of information in the Federal Government regarding standards issued by NBS pursuant to the Federal Property and Administrative Services Act of 1949 as amended, Public Law 89-306 (79 Stat. 1127), and as implemented by Executive Order 11717 (38 FR 12315, dated May 11, 1973) and Part 6 of Title 15 CFR (Code of Federal Regulations).

NBS Interagency Reports (NBSIR)—A special series of interim or final reports on work performed by NBS for outside sponsors (both government and non-government). In general, initial distribution is handled by the sponsor; public distribution is by the National Technical Information Services (Springfield, Va. 22161) in paper copy or microfiche form.

BIBLIOGRAPHIC SUBSCRIPTION SERVICES

The following current-awareness and literature-survey bibliographies are issued periodically by the Bureau:

Cryogenic Data Center Current Awareness Service. A literature survey issued biweekly. Annual subscription: Domestic, $25.00; Foreign, $30.00.

Liquified Natural Gas. A literature survey issued quarterly. Annual subscription: $20.00.

Superconducting Devices and Materials. A literature survey issued quarterly. Annual subscription: $30.00. Send subscription orders and remittances for the preceding bibliographic services to National Bureau of Standards, Cryogenic Data Center (275.02) Boulder, Colorado 80302.

U.S. DEPARTMENT OF COMMERCE
National Bureau of Standards
Washington, D.C. 20234

OFFICIAL BUSINESS

Penalty for Private Use, $300

POSTAGE AND FEES PAID
U.S. DEPARTMENT OF COMMERCE
COM-215

U.S.MAIL

SPECIAL FOURTH-CLASS RATE
BOOK