COMPUTER MODEL DOCUMENTATION

Saul I. Gass

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 some of the issues and concerns of computer model documentation and proposes an approach for the development of adequate documentation. A set of computer model documents is proposed and described that is based on the relationship between the phases of the model life cycle and documentation information needs.

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. (This paper is based on the more detailed report [11].)

THE WHY OF MODEL 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 nonexistent or are lacking in detail and do not serve the models well [2], [3], [4], [5], [6], [8] and [18]. 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 emphasize the need for documentation procedures that not only produce information, but do so in a detailed, yet comprehensible, clear and timely manner.

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.

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," [1]; 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," [17].

If the purposes and claims for documentation are true, then the modeling and computer programming communities appear to be grossly derelect 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 [3], [8] and [18]. 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," [14].

Information that describes the current state of models causes us to be pessimistic about modeling activities with respect to modeling documentation. In the survey [8] 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.

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.

THE WHAT OF MODEL DOCUMENTATION

Of the material on documentation that has come to our attention, 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.

First, what is model documentation? From [2] and [5] 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) insure 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?

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 [3]:

- 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 was 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 next describe such a process.

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 nontechnically 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 [1] 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.

Computer programmers, model builders and systems analysts tend to think of their efforts in terms of major phases, stages or steps [1], [7], [9] and [19]. 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. 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 explicitly 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 paper. Our purpose has been 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 are the users 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 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 a 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 Developmnet Phase

The documents produced here are the ten documents described in the FIPS PUB 38 Guidelines [1]. 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), the 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 process 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 [9] and [10]. 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 must be available for use during the training and installation phases. 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 Analyst's Manuals, or distributed separately. A typical set of contents is that given in House [13].

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 documenation is required to meet the objectives of the project. Documentation costs can be quite high. But these costs must be compared to the probable additional costs if adequate documentation is not prepared. As noted in [3] and [8], 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.

REFERENCES

1. Anon.; "Guidelines for Documentation of Computer Programs and Automated Data Systems," FIPS PUB 38, National Bureau of Standards, Washington, DC, February 15, 1976.
2. Anon.; "An Evaluation of the Use of the Transfer Income Model--TRIM--to Analyze Welfare Programs," PAD 78-14, U.S. GAO, Washington, DC, November 25, 1977.
3. Anon.; "Improvement Needed in Documenting Computer Systems," B-115369, U.S. GAO, Washington, DC, October 8, 1974.
4. Anon.; "Auditing a Computer Model: A Case Study (CS-4)," U.S. GAO, Washington, DC, May, 1973.
5. Anon.; "Review of the 1974 Project Independence Evaluation System," OPA-76-20, U.S. GAO, Washington, DC, April 21, 1976.
6. Anon.; "Review of Selected Army Models," Department of the Army, May 1971.
7. Anon.; "Ways to Improve Management of Federally Funded Computerized Models," LCD-75-111, U.S. GAO, Washington, DC, August 23, 1976.
8. Fromm, G., W.L. Hamilton and D.E. Hamilton; "Federally Supported Mathematical Models: Survey and Analysis," Stock No. 038-000-0024-0, U.S. GAO, Washington, DC, 1975.
9. Gass, S.I.; "Evaluation of Complex Models," Computers and Operations Research, Vol. 4, No. 1, March 1977.
10. Gass, S.I.; "A Procedure for the Evaluation of Complex Models," Proceedings First International Conference on Mathematical Modeling, University of Missouri, 1977.
11. Gass, S.I.; "Computer Model Documentation: A Review and An Approach," P.O. 713006, National Bureau of Standards, Washington, DC, April 1978.
12. Greenberg, M., M.A. Crenson and B.L. Crissey; Models in the Policy Process, Russell Sage Foundation, NY, 1976.
13. House, P.W.; "Diogenes Revisited - The Search for a Valid Model," Simulation, October 1974.
14. House, P.W. and J.McLeod; Large Scale Models for Policy Evaluation, John Wiley & Sons, NY, 1977.
15. Krasny, M.A. (Ed.); "Documentation of Computer Programs and Automated Data Systems," Proceedings of a Symposium, National Bureau of Standards, Department of Commerce, July 1977.
16. McLeod, J.; "Simulation: From Art to Science for Society," Simulation Today, No. 20, pp. 77-80, December 1973.
17. Puckorius, T.D.; "Why Document?" pp. 5-9 in [15].
18. Shubik, M. and G.D. Brewer; "Models, Simulations, and Games - A Survey," R-1060-ARPA/RC, Rand Corporation, Santa Monica, CA, May 1972.
19. Sibley, E.H. and H.H. Sayani; "Data Element Dictionaries for the Information Systems Interface," Proceedings of the First National Symposium on Management of Data Elements in Information Processing, National Bureau of Standards, Gaithersburg, MD, January 24-25, 1974.