MODEL DOCUMENTATION
Hinderance or Help

John M. Corliss, Jr.
Office of Policy, Planning, and Analysis
U.S. Department of Energy
Washington, D.C. 20585

ABSTRACT

Just as the cobbler's children are the last to have their shoes repaired, simulators--individuals who spend careers structuring the world into systematic models--have not developed tools and techniques to systematize and structure their own procedures. As a result, quick and easy communication among all interested parties during model development is often made extremely difficult. Inputs which might be helpful to the modeller are consequently lost. It is also typically impossible for a client to maintain control over the implementation of his or her model design especially when the client and model builder are physically separated. Finally, when a version of the model is completed, there can be a considerable delay while an entirely new model description, in laymans terms, is prepared. Unfortunately, as a result of time constraints such a description is sometimes never completed.

Based on work done at Purdue University and the frustration of designing and overseeing the implementation of a system dynamics simulation model at the Department of Energy, this paper describes a structured development and documentation approach to modelling. A systematic approach of this type forces the analyst to think out the implications of a given representation of the world before sitting down at a terminal. It provides a living (continually updated), standardized, written document which not only helps improve the quality of the work but allows for efficient communication between the client and the implementor and eliminates the need for most post model development documentation efforts.

Structured analysis and documentation require a significant pre-coding investment in time and manpower and add to the tasks the modeler must perform. Structured techniques may, however, allow the project manager or client to maintain control over the project. Systematic methods of operation may also streamline the process thus decreasing the overall cost of the effort. In fact, my experience at the Department of Energy indicates that modelers may discover that a structured approach and the resulting documentation is more of a help than a hinderance.

In the fall of 1980 the Secretary of Energy needed to be informed of the possible oil market impacts of the Iran-Iraq war. To meet this need, a simulation model which traced the flow of oil from the ground through the crude markets, the refineries, the product markets and to the final consumer was designed [1]. This first version of the Short Term Energy Markets (STEM) model was then computerized by a consulting firm using the dynamo simulation language and a report on the model was written [2]. Modifications were then made and documentation of the improved model is now in preparation. This process of incremental improvements will probably continue.

Although STEM is not a particularly large model, management of its development has proved difficult. Over the past two years, a supervisor, a designer/project manager,

a series of 4 programer/modelers, a user and a number of experts have been involved in the model development. Thus, in order to simulate short term oil markets, very specific information has been transferred among at least a dozen individuals. This complex transfer of information has frequently caused the model's design to diverge from what was originally envisioned and has resulted in significant delays.

Overseeing STEM's development has taught me a great deal. Although it has reinforced some basic concepts, it has also made me reevaluate and refine the methods developed for my Industrial Engineering Master's Thesis [3] and apply those methods to a project of a different type than originally intended. It is my hope that the refinement and application of these techniques will be useful to others in their modeling efforts.

BASIC CONCEPTS

Although stressed in a large number of colleges and universities, many modelers perceive documentation as something between a necessary evil and an afterthought. This is understandable given the way many models are documented. The documentation tends to be aimed at those unfamiliar with model and is typically of little real use to the modeler. Ideally however, documentation should be useful to all who come in contact with the model throughout the model's lifetime. In order to develop techniques to provide such documentation, one must first examine how models are developed and used and then determine how documentation might support this process.

Model Life Cycle

Computer models have much in common with the humans who develop them. They evolve much like people. Just as many of the characteristics of our ancestors have been passed on to the current generation, many of the characteristics of past models can be found in their current day counterparts. The life of a model is also similar to the human life cycle: The model is born, matures, has a period of productivity and is finally replaced by a member of the next generation of models. The length of a model's productive period depends on the care it is given. Insufficient investments in a model's health can lead to early retirement. Poorly documented models are often, poorly maintained models. Such models lose their usefulness because new personnel find it difficult to learn how to use them and are unable to keep them up-to-date. Although each stage of the modeling process can be as random as life seems to be, efforts have been made to structure such activities. Models do have at least one advantage over humans, they develop

in an iterative manner (Figure 1). When a mistake is discovered, the modeler simply returns to the appropriate previous step and corrects the error.

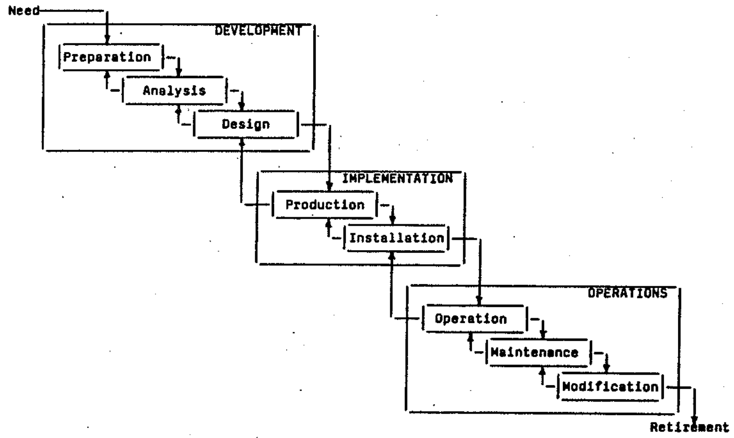

Figure 1. Model Life Cycle

Models begin as the gleam in someone's eye. A need is perceived and, if appropriate, a model conceived to meet the need. For example, given the need to quickly access the possible impacts of an oil supply disruption, the development of a short term oil market model was suggested. Although the first thing some modelers would like to do is sit at a terminal and start entering code, this could complicate the development of the model and lead to otherwise avoidable implementation problems. The development period consists of three stages: preparation, analysis and design. During the preparation stage, the effort is planned, the shortcomings of the present methods are reviewed, resources and other constraints are considered and the scope of the effort determined. It is important to note that the scope of the analysis must be larger than the processes of direct concern if the linkages to the rest of the world are to be clearly understood. In the analysis stage the modeler tries to learn as much as possible about the world within the previously defined boundaries. This is accomplished through research and interaction with experts. In this stage the modeler also specifies the inputs, the output variables of interest and the historical behavior to be simulated. Finally the model is designed on paper and discussed with the consulted experts to verify their thoughts are correctly interpretted.

Implementation now begins--the modeler can get to the terminal. The design is computerized and the model is put into service. The transition from diagrams on pieces of paper to a computer simulation model, is not always a smooth one. It is in fact often a painful process with many regressions back to the design activity. This is especially true if little effort was

spent on the development stage. Although models are almost always being improved, the earlier major problems are found the less embarassing they are and the less expensive they are to correct.

Modelers sometimes view their job completed once they have produced the required model. A model cannot be considered operational, however, until it is installed and the users are trained. The time it takes for a developed model to become a regularly used one can be short or long depending on the effort the implementor makes to transfer knowledge to the users. I have seen examples of models built for D.O.E. which have been "left on the shelf" for years. This happened because the implementors never trained users who did not have time to read the voluminous documentation supplied with the model.

Once the model is placed in the hands of a trained user, it is operational. It can now be used for the purpose it was intended. While the model is being used, new data may become available, new behavior may become evident, errors in the code or model structure may be discovered or new issues for analysis may surface. Such events may require model maintenance (a narrowly focused rapid repeat of the development and implementation phases) or could result in model modification (a more thorough repetition of the first two phases).

As the model and the people using it change, the operation of the model may become increasingly hard to understand. Eventually the model may become irrelevant given the issues of the day or the implementation of further incremental changes may become too difficult to be cost effective. At this point someone might suggest that a new version of the model or perhaps an entirely new model be built to serve the current needs of the organization. The life cycle of a new model will thus begin. This new model soon will become operational and the old model can be retired.

As mentioned previously, in each of the stages described above a rather structured pattern may be (and perhaps should be) followed. The input to each stage is information. Through a series of feedback loops this information is transformed into different information which is an input to the next stage in the process (see figure 2). First, the input information is analyzed and the specifications for this stage of the process are developed. Next, alternative means of meeting the specifications are synthesized, tested and then evaluated against the specifications. Finally a decision is made as to whether the solution is suitable or whether additional work is required. In the development of large models, those approving the specifications and making the final decisions maybe and often are different from those completing the other steps. Thus, good communication during each phase of the modeling process is essential to the development and implementation of a useful product.

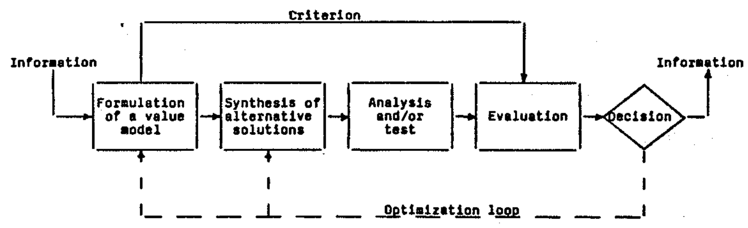

Figure 2. A Model of the System Design Process (the Horizontal Structure of Systems Engineering) Source: Kline and Lifson

Model Documentation

Communication between the designers, implementers, users and experts on the topic being modeled is critical to the smooth and cost effective evolution of a model. Both for clarity and to provide a record of progress, much of this communication is in written form. Traditionally, model documentation has taken two forms: an anotated listing of the model and a description of the model in one or more volumes. Often the documentation is written after the production stage and is viewed only as an installation tool. Such a narrow perspective limits the usefulness of the documentation and actually adds to the implementors work load (each time the model undergoes a major modification the written documentation may have to be rewritten). My experience indicates that, documentation can be developed which builds upon and contributes to the modeling process and which supplies most of the necessary written material.

Both the process of building and documenting a large model and the management of that process are difficult. Information is gathered from and then, after being recorded by the modeler, must be validated by busy experts. There are a seemingly infinite number of interconnections in a large system. As a result, significant feedbacks may be hidden among the multitudes. Once the system is understood, the model must be designed, the design coded, and the code tested. This entire process must be documented in a way which satisfies the needs of individuals ranging from those interested in using the results to those hired to operate or modify the model. Over time, the model and its documentation will be periodically revised. Thus, if it is to help make model development easier, the documentation method should:

- o allow specific pieces of the design to be easily reviewed without including extraneous material;

- o highlight primary feedbacks and clearly present secondary and tertiary loops;
- o contribute to a structured modularized modeling approach which makes debugging and modifying the model easier and less expensive;
- o allow the reader to quickly find and interpret the desired information at the level of detail suitable to his or her needs;
- o be easily and inexpensively updated; and
- o aid the project manager in maintaining control over both the model design and the modeling process.

A STRUCTURED APPROACH

Industrial engineers, computer scientists and management information systems development specialists have developed techniques which serve as the building blocks of this modeling and documentation methodology. Such an approach is based on structured analysis and information-transformation theories. Structured analysis provides the overall framework. Information-transformation provides both the perspective and the theoretical basis for decision making.

Structured analysis is a disciplined approach to understanding a system. Such methods incorporate diagrams supplemented by texts, an assumption dictionary and a glossary to facilitate communication between involved parties. Written language tends not to be very precise or concise. Structured graphics are both and when used properly expose detail in a gradual, controlled and comprehensive manner [5]. Such a method attempts to minimize the chances of making critical errors during model development [6]. The desired results are an effective model (or data management system, organization, etc.) and its support documents, all of which can be easily modified to meet changing needs.

Most processes can be viewed as the transformation of a set of information and resources into a new set of information and resources (e.g. the system-design process discussed earlier). Since most activities can be viewed in this manner, tracing the flow of information and objects through the system provides us with a fairly complete description of how the system operates. Another advantage of such an approach is that looking at a system from inside its very fabric keeps us from getting a one-sided view of what is happening. This "big picture" perspective helps ensure that most of the important feedbacks are included.

Whether building a new model or revamping an old one, the information transformation approach provides guidelines for code modularization. A benefit of nested modularization is that it high-lights the significant feedbacks and linkages at each level of abstraction. Modularization is accomplished by a grouping procedure aimed at simplifying the presentation of information and ensuring that all inputs are supplied and that there are sufficient inputs to a process to produce the output. There are two simple rules for grouping: all the activities in the group must be part of a single process, and the flow of information across the group boundaries should, to the greatest degree possible, be minimized.

WORKING DOCUMENTATION

The previous sections described the modeling process which the working documentation will support, the desirable characteristics it should have and the theories upon which it is based. The basic components of the documentation, the way in which these components fit together and the procedure through which the documentation is used and developed can now be discussed.

Diagraming components

As implied earlier, the working documentation is centered around diagrams which depict the flow of resources and/or information. A diagram in its simplest form symbolically shows a set of resources/information entering a process which transforms them into another set of resources/information. A complex system is a network of these simple transformations.

Arrows represent flows [6] of information or resources. Physical flows are shown as arrows with solid stems. Information flows are shown as arrows with dashed line stems. To simplify the diagram, flows of the same type, with the same source and destination, and a related purpose are bundled (e.g. gasoline, diesel fuel and jet fuel become transportation fuels). A definitive descriptive noun-phrase is used to name each flow. If a suitable name can't be found then the current conceptualization of the system needs to be reevaluated.

Circles represent processes [6]. Processes occur over time and may be concrete (transforming crude oil into petroleum products) or conceptual (transforming a producer's desire to sell crude and a refiner's desire to buy crude into a sales agreement). A strong action verb and a singular object is used to name each process. Again if a suitable name can't be found the process is probably improperly defined.

A third symbol will sometimes be required. A small circle with a line through it (O) represents a storage facility [6]. On occasion a flow will come from or go into such a facility. To differentiate between storage facilities, each is given a name. If the same facility used in more than one place, however, the same name is used.

A simple diagram of domestic oil refining, as conceptualized in the STEM model, is presented in Figure 3. It illustrates how each of the various symbols are used.

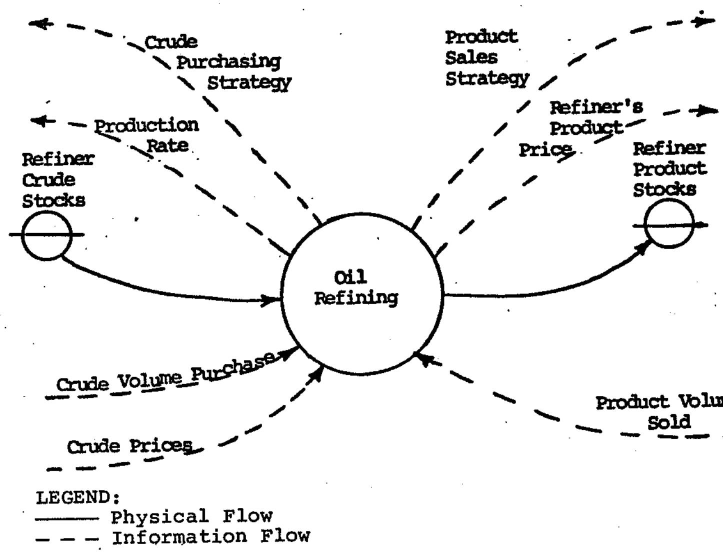

Figure 3. Domestic Oil Refining Process

Diagram Organization

Using these methods to diagram the operations of any complex system would take up a very large sheet of paper. Such an approach has two major drawbacks which decrease the value of the diagram as an aid to communication. First, anything on a piece of paper larger than 8 1/2" by 11" is hard to handle. In fact, without a master in the art of paper folding, such a large diagram could not be conveniently transported from one place to another. The second drawback involves complexity. Anything with over seven activities is generally too difficult to follow. The solution lies in a set of progressive expansions.

Using such a strategy, one begins with a very general description of the organization's operation using between three and seven circles on an 8 1/2" by 11" diagram. This diagram is labeled DO. Each activity is named and numbered and then expanded, exposing its details on a new diagram with three to seven named and numbered circles. These diagrams are labeled D1, D2, D3, etc. This process of enlarging the subactivities on a new diagram can continue until all tasks are shown in satisfactory detail. At its most detailed level, dynamo flow diagram symbols should be used. The blow-ups of the more detailed diagrams should be labeled D1.1, D1.1.1, D1.1.2, etc. Figure 4 shows how all these diagrams fit together.

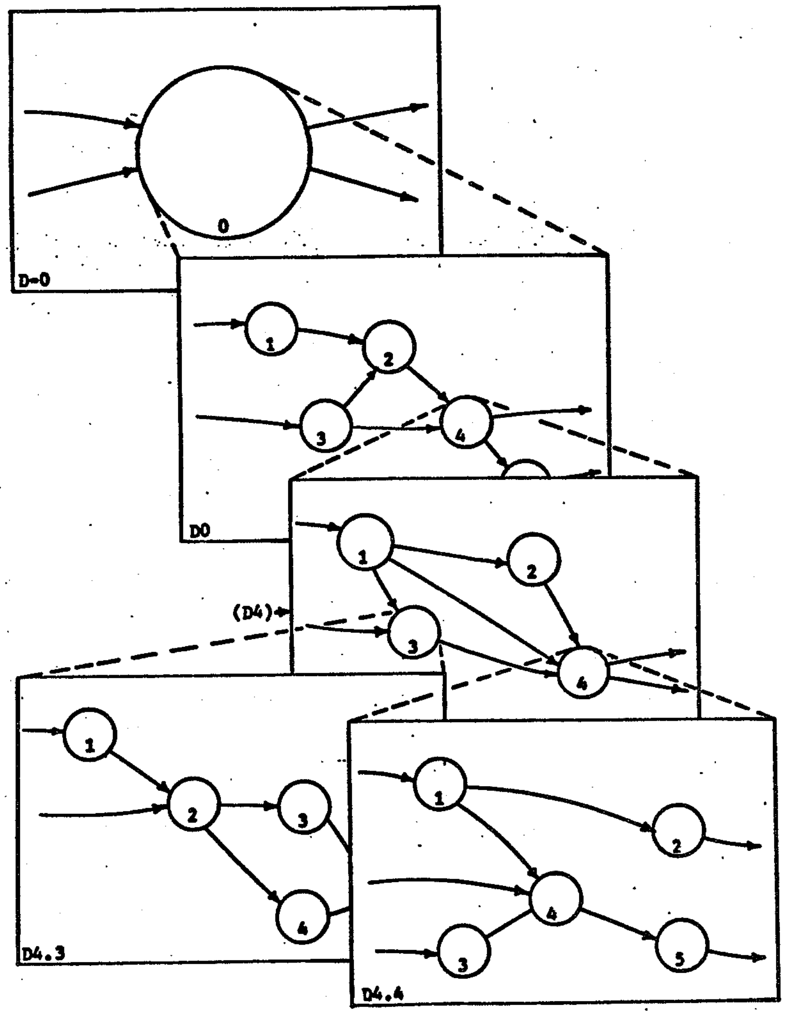

Figure 4. Diagram Relationships.

Diagram D1 is called the parent diagram to Diagram D1.1, D1.2, ... , D1.N. A diagram is completely bounded by its parent. In other words, all flows leaving and entering the diagram must also be depicted on the parent. This is not to say bundled flows cannot be decomposed on the more detailed drawing, only that everything must be accounted for. The way bundled flows are presented is illustrated in Figure 5. On

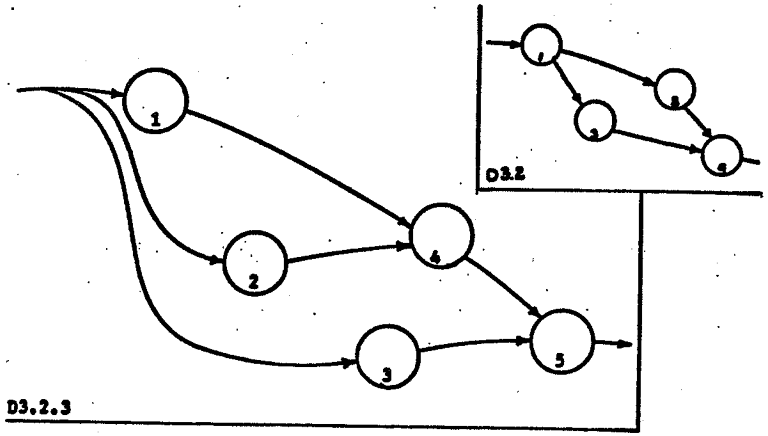

Figure 5. Decomposition of Bundled Flows.

diagrams D3.2, process circle 3 has only one input. On Diagram D3.2.3 where process circle 3 is shown in detail, it can be seen that this one input really has three parts.

Because the parent diagram for DO (labeled D-O) consists of only one circle representing the entire system and shows the models interactions with the outside world, it is referred to as the context diagram. Should it be determined that diagrams showing the environment in more detail would better establish the context in which the system operates, then diagrams D-1, D-2, etc., can be developed.

Diagraming Guidelines

Although it is not the purpose of this paper to describe in great detail how to create a complete set of diagrams, a few guidelines may be in order.

When creating a new diagram:

1. bound the diagram with the list of inputs to and the outputs from the process being decomposed;

2. sketch out the relationships between the inputs and outputs seeking natural patterns of interconnection;

3. label the processes and flows;

4. critically review the drawing: are the names suitable? are the inputs to a process sufficient to produce the outputs? is the diagram easy to read? etc.; and

5. repeat the process until the diagram is acceptable.

When selecting a process to decompose, there are several possible decision rules:

- o Proceed in numerical order;

- o Select the process whose decomposition would expose the most information about the other processes; or

- o Select the most difficult process to decompose, the one about which are the you are the least clear [9].

There are also a few general diagramming guidelines which might be helpful. If the diagram is based on a conversation with an expert, it should be submitted for his or her review as soon as its finished. A change to one diagram may effect others. Changes to these diagrams should be made quickly, otherwise they might never be made. Details should not be exposed too quickly, if necessary add an extra level of

diagrams. Be precise but try not to use jargon. Limit your diagram to seven or fewer processes. While working on a new diagram try to keep the parent diagram in view, stick within the bounded context and be sure the present diagram and parent diagram match when finished [7].

Implementation

Although the methodology is built around diagrams it does not end with diagrams. The diagrams must be cataloged throughout the process, supporting material must be prepared and a final report produced.

When the project begins the very first thing to do is acquire two or more loose leaf note books. Both the "Report Book" (Volume 1) and the "Workbook" (Volume 2 or more if needed) should have dividers for the introduction, the model description, the assumptions dictionary and the glossary. Additional dividers for each level of diagrams should be added to the report book as they are developed. Additional dividers for each diagram, in order by level and by number (D0; D1, . . . , DN; D1.1, . . . , D1.N; etc.) should be added to the work book(s) as needed. Standardized forms with a place to note the diagram name and number, the author and the date it was prepared should be photocopied. During the modeling process each diagram should be drawn on a standard form and put in the workbook with the most recent version on top. In this way the development process can be traced if necessary. Only diagrams which are currently approved and up to date go in the report book.

Once a "final" version of all the diagrams is in the report book the supporting documents should be prepared. The most important of these is the diagram detail sheet which will be placed opposite each diagram form (see Figure 6). This standard form has three portions. Half the page is reserved for a concise walk through the system. Don't just state what is on the diagram. Explain the real world process being modeled. Many times the development of these descriptions exposes model errors which require diagram revisions. The upper corner is for a miniature of the parent diagram. This illustrates to the reader the context in which the current diagram fits. The remainder of the page is for supplemental material. Decision rules, statistics regarding the processes, model code or other material which will enhance the readers understanding of the system and the model.

The introduction should state the purpose of the model and briefly describe how the diagrams are read and the report is laid out. Since the diagrams are presented in order, one level at a time (the context diagram, the first level diagrams, the second level diagrams, . . . the most detailed level diagrams) it should be noted that the reader is welcome to read

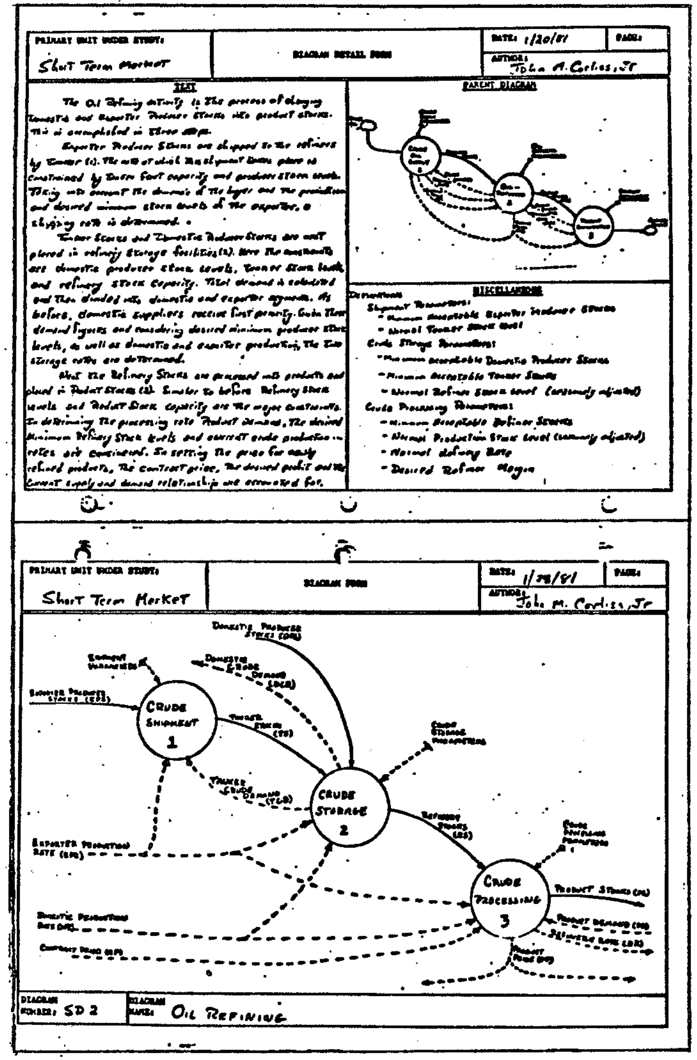

Figure 6. Page Pair Format

to what ever level of detail statisfies his or her needs. The assumptions dictionary lists, in alphabetical order, all of the constants and table functions in the model. Each variable name should be accompanied by its description, its value, the source or derivation and the uncertainty. Such a list is very useful for sensitivity analysis or when answering inquires about specific assumptions. If necessary a glossary should also be prepared.

The final document should be kept in loose-leaf form. This increases the ease with which revisions can be made and will encourage the modeler to keep the documentation up to date and to use it in further model development. The use of standard forms helps to ensure completeness and uniformity [8], [9].

Conclusion

The process of building a model can be made easier, can require less time and can be more easily managed if effort is invested in proper documentation throughout the modeling process. The working documentation described possesses many desirable qualities. It allows for the quick communication of modeling efforts to the project manager, (submitting a marked up diagram is one of the quickest ways to propose a model

change). It facilitates communication between all those involved in the effort. It helps to structure the efforts of the modelers. It is easily revised. It is easily converted into final report form. When completed, it concisely presents the material while exposing details in a gradual controlled and comprehensive manner. In short, if used, this method may convince modelers that documentation can be much more than a necessary evil.

BIBLIOGRAPHY

[1] Corliss, John M., Jr. and Stanley-Miller, John, Understanding Oil Market Behavior During Disruptions: The Development of the Short-Term Energy Markets Model DOE/PPA Working Paper, Washington D.C. (1982).

[2] Sloan, Mike and Salama, Samir, International Short-Term Oil Markets Model: Preliminary Description, Energy and Environmental Analysis Inc. for DOE/PPA, Washington, D.C. (1982).

[3] Corliss, John M. Jr., Government Organization Improvement: Definition Phase Procedure Manual, Industrial Engineering Thesis, Purdue University, West Lafayette Indiana (1980).

[4] Kline, Melvin B. and Lifson, Melvin W., "System Engineering" in English, J. Morley (editor), Cost-Effectiveness the Economic Evaluation of Engineered Systems, John J. Wiley and Sons, New York, New York (1968).

[5] "IDEF - Introduction to IDEF (version 0)," Softech, Inc., Waltham, Massachusetts (1978).

[6] Demarco, Tom, Structured Analysis and System Specification, Yourdon, Inc., New York, New York (1978).

[7] "IDEF - Authors Guide to Creating Activity Diagrams (version 0)," Softech, Inc., Waltham, Massachusetts (1978).

[8] "IDEF - Forms and Procedures Guide (version 0)," Softech, Inc., Waltham, Massachusetts (1978).

[9] System Development Methodology - Forms Guidelines, Katch and Associates and Atlantic Software Inc., Philadelphia, Penn. (1975).