Simulation Modeling by Stepwise Refinement

J. N. Beauchamp
R.C. Field

E-Systems, Inc., ECI Division
St. Petersburg, Florida

Abstract

Application of the structured programming concept of stepwise refinement to modeling is described and developed. Modeling by simulation with GPSS (General Purpose Simulation System) is emphasized, even though the full modeling activity undertaken included analytical and brassboard modeling as well. Stepwise refinement was applied rigorously to the simulation model, and also used for the analytical and brassboard models.

The simulation model is described in detail. The network, node and implementation analysis that provided the numerical basis for operation of the model are also described. How the concept of stepwise refinement was applied to the development of the simulation model in GPSS is thoroughly described. Results of the simulation models show how stepwise refinement can maximize useful output with respect to cost and effort invested.

1. INTRODUCTION

In designing a new and complex system, most designers would acknowledge the benefit of a tool for testing design decisions before rendering judgments into actual hardware and software. Such a tool exists, but is rarely used. The tool is computer modeling; the reasons that it is rarely used include misconceptions about its expense and a lack of engineers schooled in its application. Indeed, it is possible to spend unconscionable time and money building and running a model. It is further possible for the end product to so little resemble the system being modeled that it is of no practical value.

Cost is a factor with fundamental roots and serious implications. The application of the technique of top-down structured development to modeling directly affects cost and accuracy. Top-down structured development, applied as described herein, reduces costs. It keeps the level of model detail consistent with answers sought; it enhances accuracy by providing specific points of interface between the modeling activity and objective reality. Coupled with certain tests of convergence, this concept has both practical value and theoretical elegance.

The subject is stepwise refinement, and we will be using examples taken from one particular application, the design of a tactical store-and-forward message switching network. We were fortunate in being able to start with a top-level military mission to be supported by this system, and to derive the network, nodes, and hardware and software designs therefrom.

The network is characterized by having up to 25 nodes, approximately 500 communications channels, and the-order-of-200 users. The channels operate at 16 kb/s, employ FEC (Forward Error Correction) as required, utilize ADCCP (Advanced Data Communications Control Procedure) protocol, and have bit error rates in the range of 10^{-3} to 10^{-6} (including the effects of FEC).

Each node serves up to twelve subscribers. The subscribers generate traffic of two precedence classes, and perishable or nonperishable within each class. Some subscribers primarily transmit; others primarily receive; traffic occurs at a variety of average rates. The size, weight, power and environmental requirements imposed upon the nodes dictated a state-of-the-art approach, precluding the use of existing designs. So the design was challenging, and the stepwise technique contributed greatly to success.

2. KEY CONCEPTS

In the process of performing the modeling, several concepts emerged that have broad application. These are contained in the following general categories: program development; partitioning; and use.

Program development includes the software system that is the model. Good software design practices can be used to great advantage. These practices include a rigorous application of top-down design, using stepwise refinement to move between levels, and minimum commitment at each level. Use of these principles is slightly different in modeling,

CH1437-3/79/0071-0081$00.75 © 1979 IEEE
1979 Winter Simulation Conference

since a model may never be completed, and approximations are widely used. The purpose is the same: define function and interface in increasing detail; minimize the development effort.

Top-down design starts with the highest level mission or system requirements available, and defines major system (or network) elements from these requirements. At each level of the design, the requirements are further refined and partitioned, until a level is reached at which design-to (Type B1) specifications can be prepared.

Our structured modeling approach requires that all functional requirements and interfaces of top-level system elements be defined before the model is carried to the next level or detail. This ensures that all major system requirements are incorporated into the model, and that these requirements are tested before the details are investigated. This philosophical tenet goes hand-in-hand with minimum commitment. At each level of definition of a system, the minimum number of constraints are placed upon the model. This approach has two important results:

a. It forces the design of the lower-level subsystem and components to the lower levels of the models.

b. It avoids defining subsystems and components not required for an adequate description of the system.

Partitioning provides a method of subdividing the model of complex systems into manageable units. This implies the passing of information between the units through feedforward and feedback. Our approach has developed partitions based upon a top-down system analysis which deals with abstractions at high levels (summaries) through physical devices at the lowest levels. To manage the feedforward and feedback process, we use the concept of adjacency. This states that feedforward and feedback occurs only between adjacent hierarchical model levels.

Use focuses attention upon the goals of modeling. These goals must be identified so that the effort can be expended in achieving these goals with minimal digressions. Goals also define the endpoint of the models. This forces the models to be applicable to the design process, and leads directly to the definition of convergence.

The important principle employed in the use of these models is the application of appropriate tests of convergence. As the stepwise refinement process is repeatedly applied, the results tend to change by increasingly smaller amounts. The process can be carried to any level of detail required. However, complete closure is likely never possible. The process must be terminated when the results of interest converge to within the error of estimate as determined by the quality of the input.

Two sets of parameters are taken as absolutes: technological constraints, and the mission requirements. This permits us to bridge the enormous conceptual range between the ends of such a spectrum. We predict, for example, the effect of change in the speed of an integrated circuit on the probability of success of an armored attack on Western Europe. While this example may be a bit extreme, let us use it to explain how the feedforward and feedback processes are used to couple the models to bridge this conceptual gap.

Mission requirements establish message delivery speed and distributions. This information creates the network model and its connectivity. Results of the network model identify a tentative connectivity, an optimum number of nodes, and node performance. Node performance and architectural analysis is used to develop the node model. This results in a tentative subsystem definition and subsystem performance. Equipment models are built based upon the subsystem definitions, implementation analysis, and technological constraints of devices arrived at through implementation analysis. Feedforward proceeds similarly. A change in a technological constraint modifies subsystem performance in a node subsystem. This change in subsystem performance results in a node performance change within the network model that may result in changes in the network performance.

3. MODEL DEVELOPMENT

By using the stepwise refinement approach, a complex network can be easily modeled. An abstraction of the communications network modeled is shown in Figure 1. The objectives for this simulation study were to learn as much as possible, and to converge upon a solution for the network being analyzed. The range of information that was required was great. At one end of the spectrum, major decisions concerning network control and network operating procedures had to be made. At the other, detailed hardware and software designs had to be developed to implement the communications system.

The model design approach consisted of making early decisions on major segments with minimum detail. As information was developed and analyses completed; increasingly greater depth of understanding and detail were made available to be included in the model. The end point consisted of identifying convergence within the models. Given this approach and the definition of convergence, and bringing out convergence so that it can be seen is comparatively easy.

For the complexity of the network shown in Figure 1, a partition into three submodels was chosen. These are shown in Figure 2. Other partitions are certainly possible. It turned out, however, that the choice made was extremely fortuitous for our task definitions and systems organization.

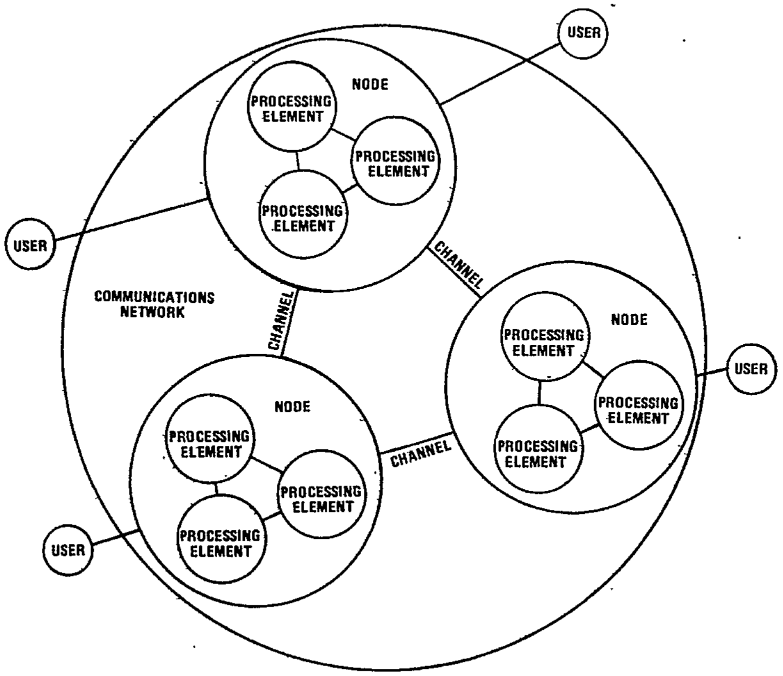

FIGURE 1.

COMMUNICATIONS NETWORK MODEL

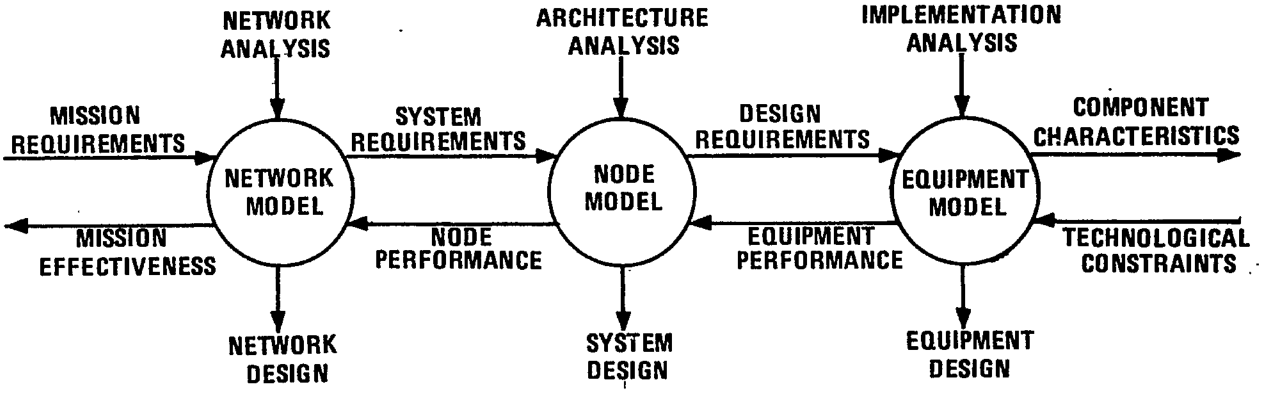

FIGURE 2.

MISSION MODEL

Simulation Modeling by Stepwise Refinement (continued)

Definite partitioning criteria were established early in the modeling effort. Some of these were: the level of effort required to implement a submodel; estimation of computational costs to run the model; the detail and level of information required to implement the model; the ease of information entry into the model; and data extraction from the model.

Feedforward and feedback of information is easily accommodated by this methodology. It is extremely important to note that feedforward and feedback of information occurs only between adjacent models. The information passed between models is extremely condensed or abstracted.

Modeling does not replace conventional analysis, and this partition provides a simple vehicle for entering analysis information into the modeling process. In fact, raw information from conventional analysis is used to create the model. The models serve to validate the conventional analysis used. In all simulation modeling situations, validation of the model itself is an extremely important aspect of the modeling process. Results that cannot be justified based upon conventional analysis certainly cause the simulation model to be suspect. Conversely, simulation models can be used to confirm the results of the conventional analysis.

Further, modeling amplifies the conventional analysis by providing information that is difficult, or in some cases almost impossible, to obtain in any other way. Some areas that have been modeled do not have a closed form solution, or have closed form solutions that are extremely difficult to solve. These cases are typically easily modeled to obtain the desired results.

The partitioning used provides familiar outputs at the levels required for the design process. The task definitions are at the levels of network, node, and implementation bubbles. Partitions of the model into the network, node, and equipment provide direct and relevent outputs.

Network Model

The network model, Figure 3, is designed to answer network-level questions. Network-level questions deal with the interactions of major system elements performing the assigned mission. These elements are nodes, channels and traffic. The structure and major elements of the model itself are provided by network analysis. In this case, the traffic model is shown impacting the channel model and not the node performance summary. This is because the nodes generate no traffic themselves. The mission requirements yield such things as aggregate traffic, its distribution throughout the network if not uniform, and the channel characteristics that must be accommodated.

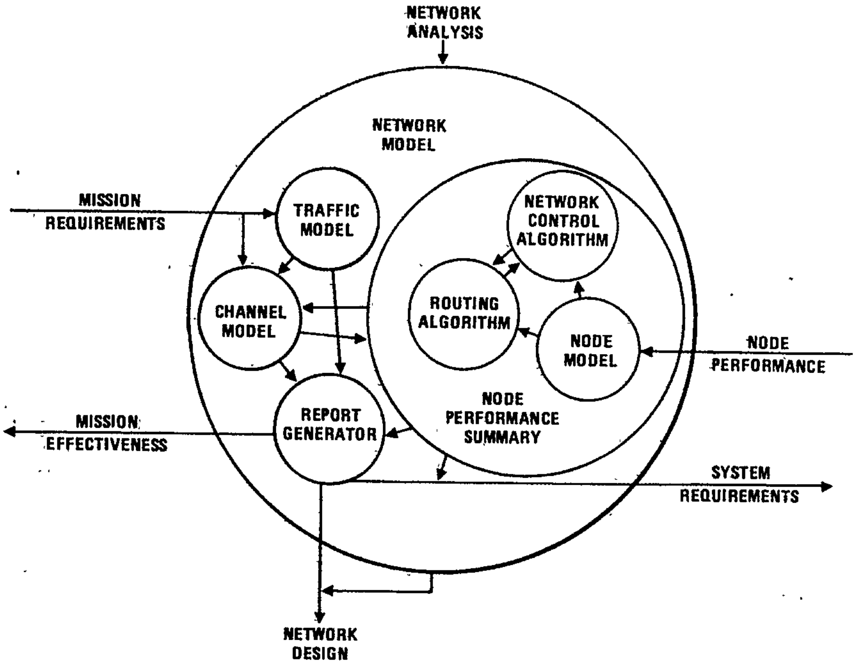

FIGURE 3.

NETWORK MODEL

The report generator provides measures of mission effectiveness, such as the speed of service through the network, probabilities of losing messages, and the ability of the network to respond to threats (such as jamming and other hostile acts), as well as operate within a realistic communications environment. Network design outputs of the model reveal the numbers of nodes and channels required, their utilization under the simulated operating conditions, and the validity of the network analysis results that went into the model.

The elements of the network model are network abstractions. Only gross aspects of system elements are required to analyze the network. As a consequence, the network model is constructed using major performance measurements, and gross limits of system element performance. Some factors of the network in the model will always remain at a nonphysical level. These factors can be lumped under one category called "Network Policy." In this example, network policy includes network operating procedures, flow control, traffic routing and channel-use discipline.

The network model answers network-level questions by supplying statistical data on each network abstraction. These abstractions have been built as behavior summaries of all system and subsystem entities, down to the lowest level of the equipment model. When the network model acts on these parameters, they have been condensed twice by the more detailed models (node model and equipment model). All of the relevant information that is known about the system to the lowest level is contained within these summaries, but the level of detail is carefully limited to that having a major effect on network performance and mission effectiveness.

To illustrate these levels of abstraction, the example of block size in a message-switching system will be used. The selection of block size effects design and performance of the system, from the highest level of abstraction to the most detailed level of the design tradeoffs. This design parameter interacts with the relatively uncontrollable environmental factor, the bit error rate. This interaction effects almost every aspect of system behavior. Figure 4 gives an example of this network-level question being answered by the network model. At this level of abstraction, transmission efficiency is the independent variable, the figure-of-merit for network design. This simple parametric graph encompasses all of the network policy decisions concerning ACK/NAK procedures, error detection and correction, and transmission procedures, as well as all relevant aggregates of node and equipment performance.

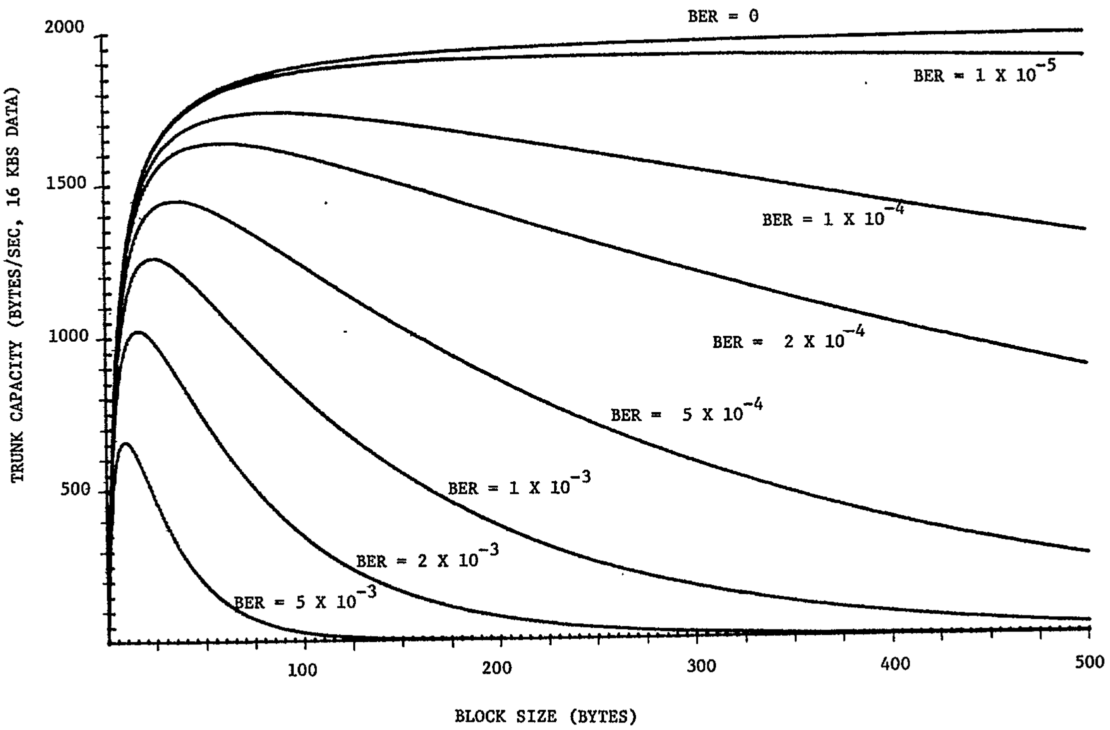

FIGURE 4.

TRANSMISSION EFFICIENCY

Node Model

The node model, Figure 5, is constructed to answer system-level questions. System-level questions deal with the interactions of the major subsystems to meet the system requirements within the constraints imposed by equipment performance. Elements of the node architecture are: partitioning of the processing and associated subsystems; use of node resources; and the processing flow determined by the architecture analysis. This architecture model interacts with the network performance summary and the equipment performance summary models to emulate the behavior of the node. The system requirements provide the characteristics of the traffic that must be handled by the node, and the characteristics of the particular channels that are connected to the node.

The equipment performance summary provides relevant abstractions of equipment behavior from the equipment model, such as procedure execution times and storage speed and capacity. The report generator delivers to the network model measures of node performance such as throughput, delay characteristics, and lost messages; and to the equipment model design requirements such as processor speeds, utilization of resources, and processing bottlenecks to be relieved.

The system design outputs of the model reveal the adequacy of the architecture simulated to perform acceptably under the specified operating conditions. If designed properly, the report generator will supply data for improving the architecture by either improved performance or reduced slack.

The elements of the node model are physically realizable entities. These can be generally characterized as major subsystem elements that are used to construct a node. Modeling at the node level then becomes an effort to analyze subsystem performance. System analysis has defined major subsystems within the node and their interconnection through the architectural analysis. The model, then, contains performance modules that represent these subsystems in some moderate degree of detail along with associated connectivity.

The node model answers system-level questions by supplying statistical data on the physically realizable entities within the node. These entities have been built as summaries of the behavior of the elements of the equipment model from the lowest physical level of interest. When the node model acts on these parameters, they have already been condensed by the more detailed equipment model. All of the information that is known about the equipment to the lowest level is contained within these summaries, but the level of detail is care-

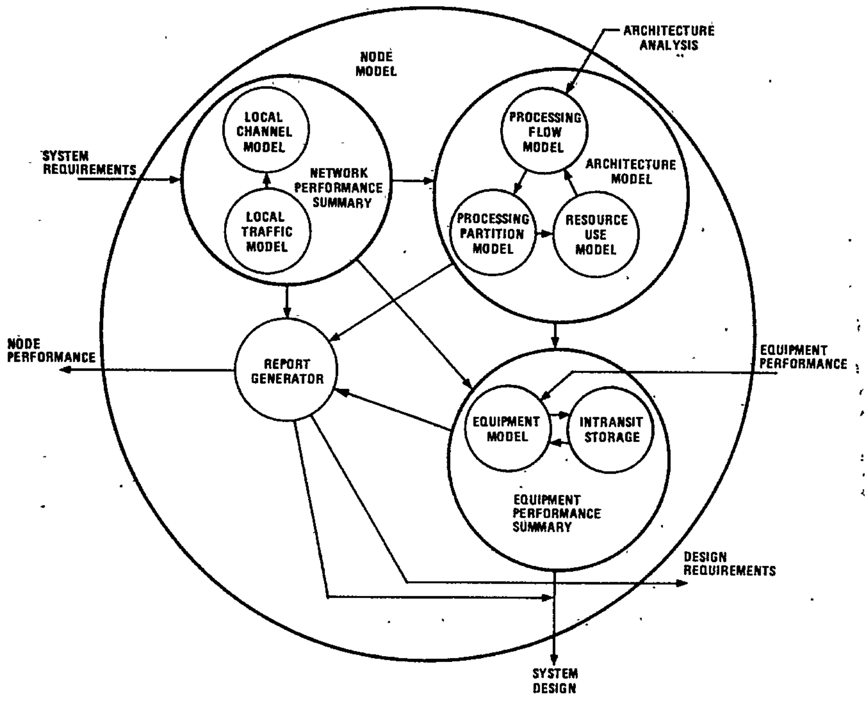

FIGURE 5.

NODE MODEL

fully limited to include only that having a noticeable effect on node performance.

Figures 6 and 7 continue the example of the effects of block size at the node level. The independent variable chosen at this level was speed-of-service through the node. Again, the speed-of-service numbers shown on these graphs subsum all of the detailed analysis and modeling done at the equipment level, such as procedure execution rates and resource allocation methods. In addition to being useful in its own right, this information is one node performance abstraction that is used in the network model to determine the delay through each node.

Equipment Model

The equipment model, Figure 8, is constructed to answer implementation questions. Implementation questions deal with the interactions of system components to meet the design requirements imposed by the node model within the constraints of the technology chosen for system implementation. The elements of the implementation model are memory, processors and resource contention mechanisms, both hardware and software. The technological constraints determine such things as memory access time and instruction execution rate. The node performance summary translates the stimuli impinging on the node into demands for particular hardware and software resources. The report generator provides the node model aggregate hardware and software performance, assesses the adequacy of the implementation to perform in the specified node environment, and produces component characteristics such as bus speed required, degree of hardware assist required in processing interrupts, and speed and quantity of interprocess communications required.

The elements of the equipment model can be refined to the lowest physical and performance levels of interest. In terms of the feedforward and feedback required within the modeling structure, equipment models refine subsystem modules. Detail can be added to the equipment model to both validate the implementation design, and to stabilize the subsystem model performance used in the node model. The general procedure is to take a subsystem model used within the node simulation, and to expand this model to do detailed equipment modeling.

The equipment model answers implementation questions by supplying statistical data to the lowest physical level of interest. The level of the simulation can vary from case to case, and within a given case; it is determined by the level of detail desired in the results, and the level of detail in the technological constraints that must be met. The model depends upon implementation analysis and technological constraints; the implementation

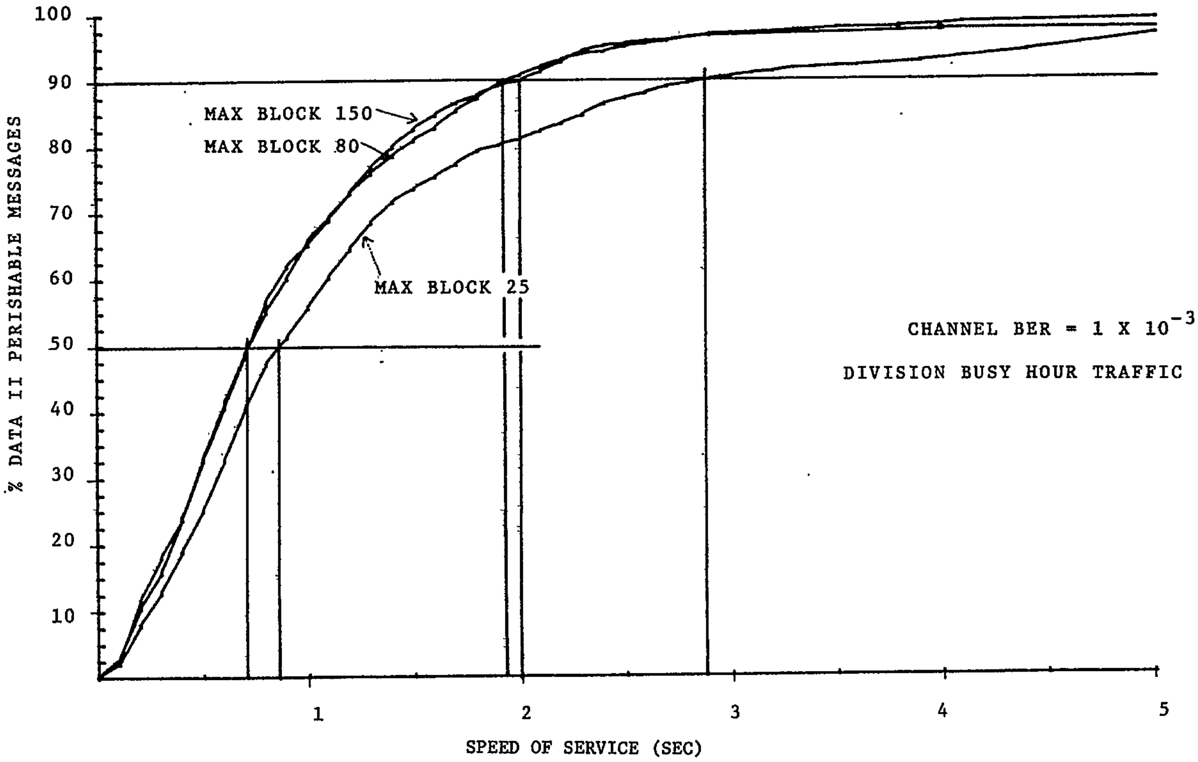

FIGURE 6.

SPEED-OF-SERVICE

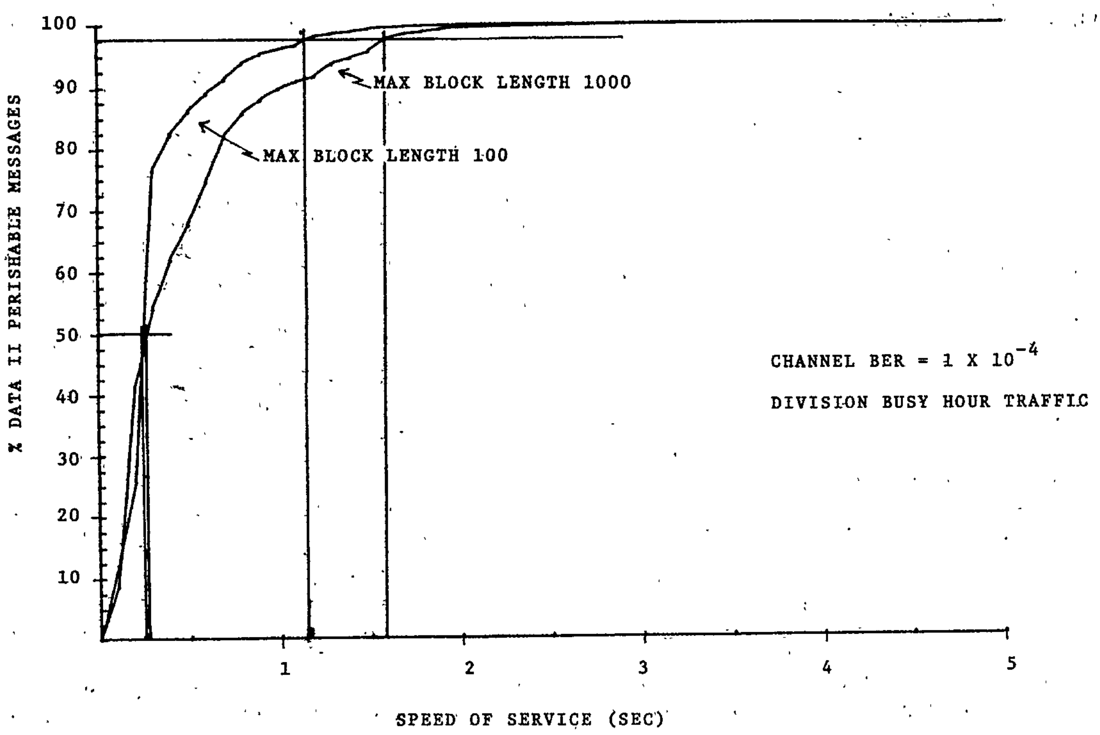

FIGURE 7.

SPEED-OF-SERVICE

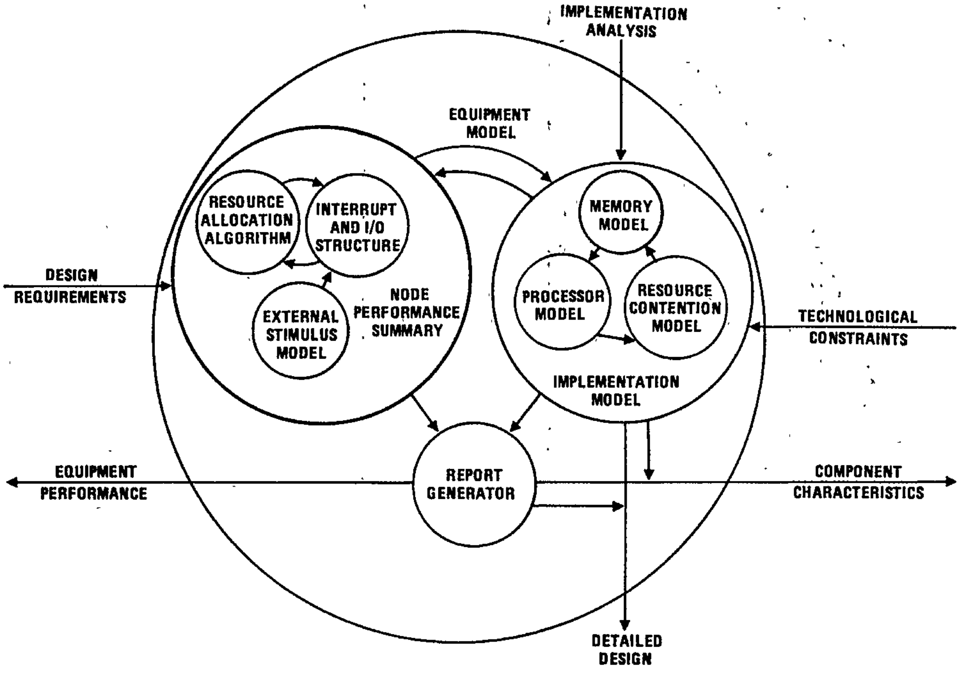

FIGURE 8.

EQUIPMENT MODEL

analysis is analogous to the architecture analysis and network analysis used in the higher-level models, but technological constraints have no such analogue. These constraints are laws-of-nature and other fundamental limitations of the technology chosen for equipment implementation. Care must be exercised in selecting and modeling these because rules-of-thumb and desiderata often present themselves disguised as laws-of-nature. An error in judgment or analysis at this point can have a dramatic effect on the outcome of all of the modeling activities. A brief sensitivity analysis can usually determine the potential effect of these assumptions.

Figures 9, 10 and 11 further continue the example of the effects of block size at the equipment level. Figures 9 and 10 show the effect of block size on the intransit storage required. Figure 11 shows its effect on processor utilization. The processor in this case was an aggregate of three TMS 9900s contending for a common program store. This level of detail gives specific design guidance, and provides the equipment performance summary information needed by the node model.

To this point, we have been describing the individual models and have indicated that they interact. The system of models is synergistic with respect to the individual models in that it has properties that go beyond the sum of the properties of the individual models.

Returning to the armored attack on Western Europe, the speed of the integrated circuit in question is a technological constraint on the equipment model and can be varied as a parameter. If and only if its speed is critical to the performance of the equipment, a change will be observed in the equipment performance report that is generated by the equipment model, and is input to the node model as a constraint or performance parameter. The node model, in turn, uses this information in generating the node performance summary that is a part of the network model. The network model produces quantitative measures of mission effectiveness - one such measure is probability of success of friendly and enemy tactics.

As stated previously, this is a deliberately extreme example, but it illustrates several important features of the system of models constructed according to these principles. Chains of events like that described in this example have been observed in actual models. Such a chain of event will only occur if the speed of that transistor is actually critical to mission success. This is a rare occurance; the ability of the system of models to find such critical variables is one of their most valuable features.

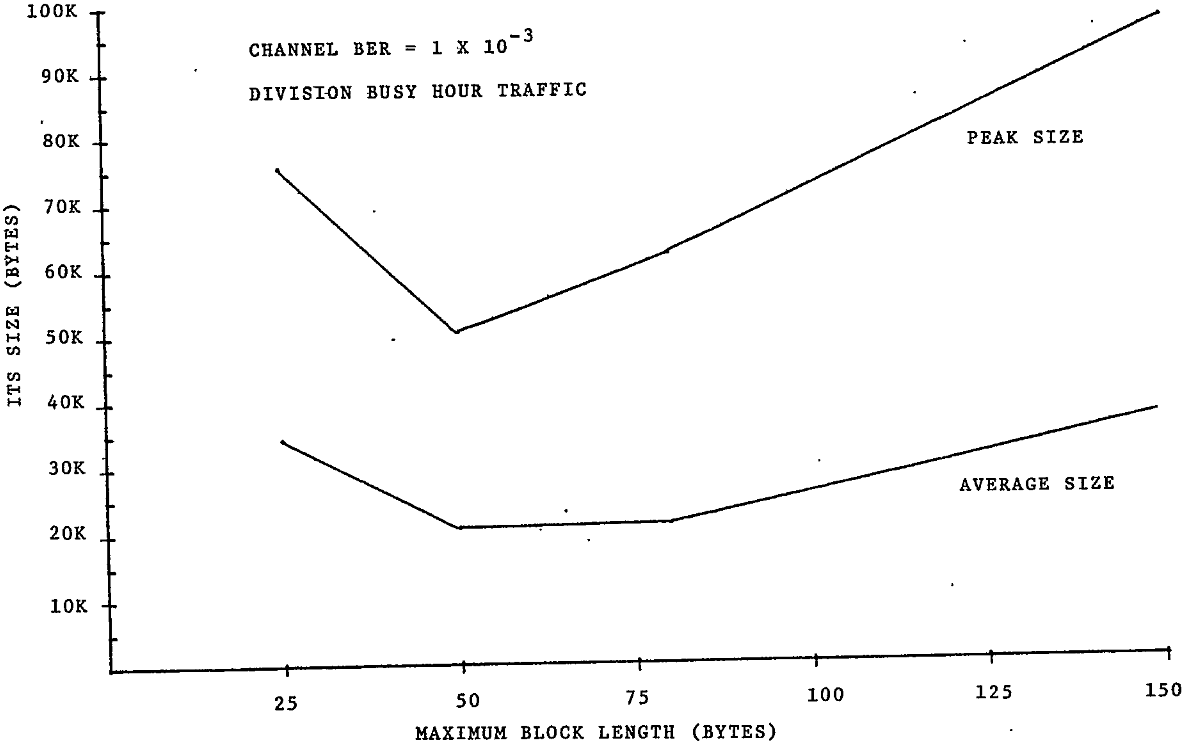

FIGURE 9.

ITS SIZE VERSUS MAXIMUM BLOCK LENGTH

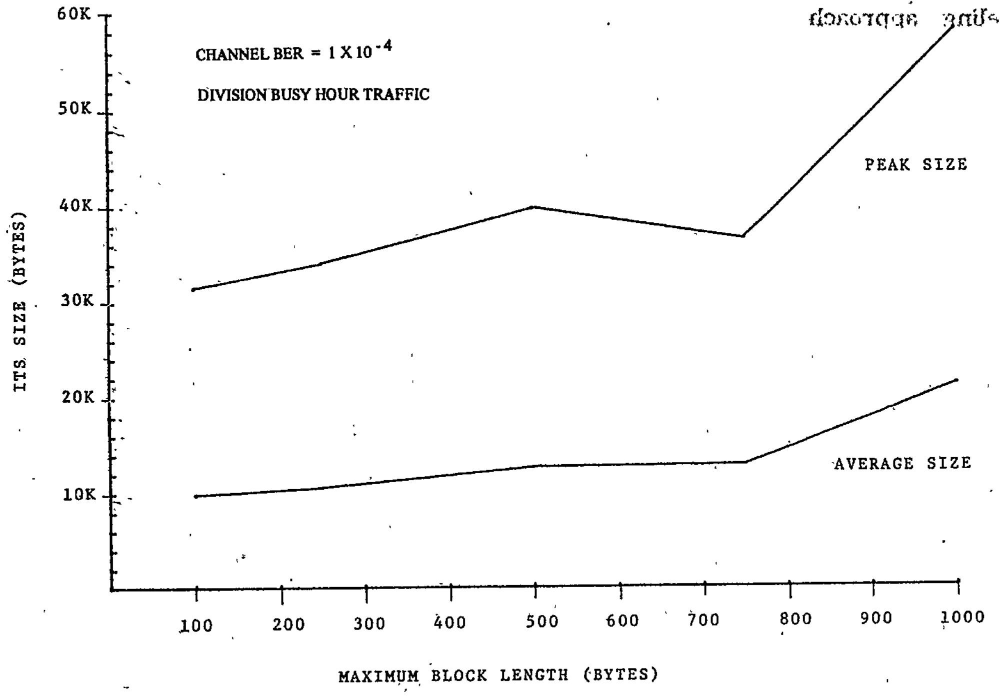

FIGURE 10.

ITS SIZE VERSUS MAXIMUM BLOCK LENGTH

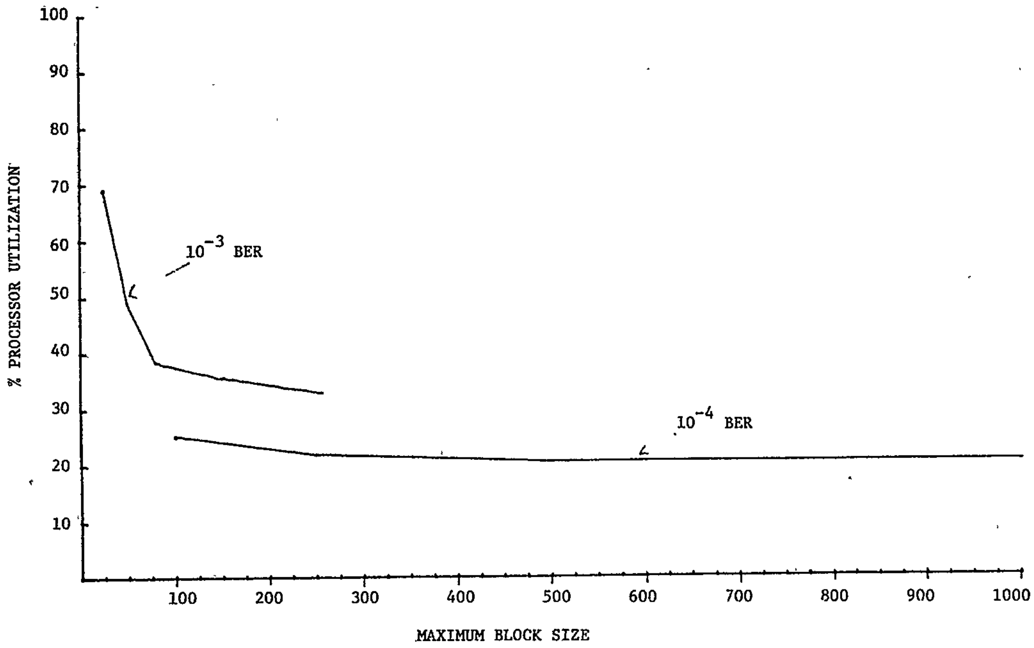

FIGURE 11.

PROCESSOR UTILIZATION

4. SUMMARY AND CONCLUSIONS

The single, most important ramification of our structured modeling approach is that simulation becomes an effective design tool from the beginning of the project. The top levels of the network and node models can usually be produced after a brief study of the mission and system requirements. Equipment models are developed in parallel with the design requirements. Refinement of the models proceeds in parallel with the continuing analysis tasks. This has three implementations: first, the modeler is a member of the design team; second, simulation is a continuing activity throughout the design process; third, simulation results are available throughout the project.

Our experience with this technique provides the following information: manpower and cost for the modeling are small; the models are approximately equal in size in terms of source lines of code; this process minimizes the amount of extraneous information for each level.

System design of the network discussed occurred over an 18-month period. During this time, a total of about eight man-months was expended in the development of all three models. For the amount of insight and information gained, this was an extremely modest investment of labor. The cost of each simulation run varied depending upon the number of cases. Typical parameteric runs, involving six to eight cases, cost $100 to $200 from commercial time sharing services.

Each of the models contained an average of 700 source lines of code, including comments to internally document the model at the conclusion of this project. Original top-level models started as 100 to 200 source lines, depending upon the level of detail when the modeling started. This implies that the simulation effort can be easily managed, and total generated code can be minimized.

Each of the analysis tasks requires specific information from the models. Given our partitioning of network, node, and equipment and associated models, simulation results are created to meet the needs of the particular analysis task.

Our experience with this design tool has been totally positive. It is unlikely that we will ever embark again on a major system development without it. It has been to us the functional equivalent of having the end product to test and optimize all through the design process. While it is not possible to state the time and money saved, we are confident that it was many times that expended on the modeling activity.

BIBLIOGRAPHY

1. McGowan, C.L., J.R. Kelly, Top-Down Structured Programming Techniques, Mason/Charter, 1975.
2. Pritsker, A.A.B., C.D. Pedgen, Introduction to Simulation and SLAM, John Wiley, 1979.
3. Reitman, J., Computer Simulation Applications, John Wiley, 1971.