Proceedings of the 1983
Winter Simulation Conference
S. Roberts, J. Banks, B. Schmeiser (eds.)

INTERACTIVE GRAPHICS AND MENUS FOR COMPUTER NETWORK MODELS

Donald F. DuBois
General Electric Company
Space Systems Division
1277 Orleans Drive
Sunnyvale, Ca. 94089
408-734-4980

The Hierarchical Modeling System (HMS) is a single software package that provides a discrete event simulation and analytic modeling capability for computer networks. The analyst may reconfigure the network model depending on the number of nodes and their interconnections by making a few changes to some simple data files-no recoding of the model is necessary. This paper describes an interactive graphics front end for HMS which allows the analyst the capability for reconfiguring the model at a graphics terminal. This provides a faster, less error prone input method. In addition, changes in the network topology are more readily understood when presented graphically. Menus are used with the graphics input in order to make changes to nodal specific models.

1. INTRODUCTION - HMS OVERVIEW

The Hierarchical Modeling System is a single software package that provides a simulation and analytic modeling capability for computer network. There are two major sub-systems within HMS, (DuBois 1982) the Interactive Modeling System and the Distributed System Simulator (Figure 1-1). IMS is used to produce high level models of computer networks. Since it relys on analytic modeling techniques it has the advantage of producing model results interactively. In addition, there is an optimizing capability which allows the user to relax the values of some model parameters in order to estimate device characteristics necessary to achieve certain performance goals such as device utilization factors and message delay times. DSS (DuBois 1981) is a modeling tool especially designed to simulate computer

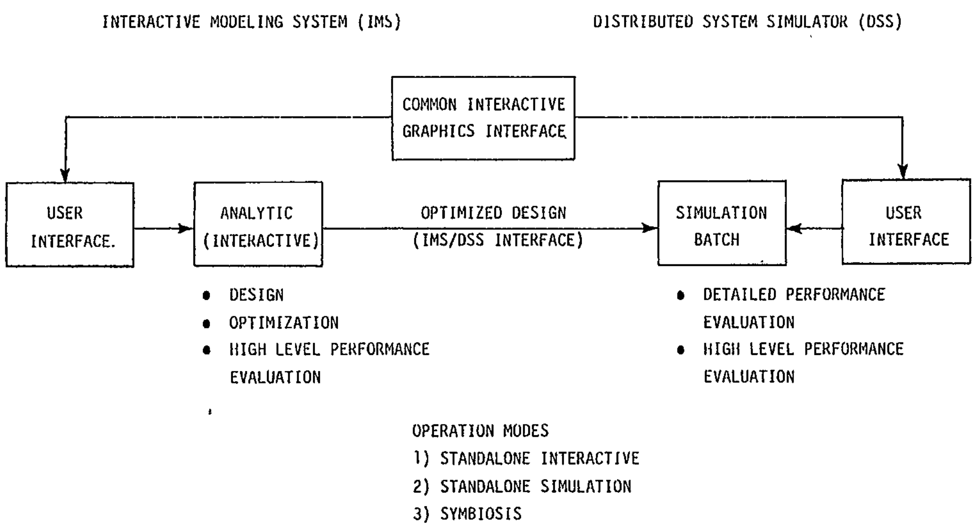

Figure 1-1 Hierarchical Modeling System Overview

networks. It has high level constructs that facilitate the development of simulators for a wide variety of networks at any level of detail. Trace facilities and a broad range of output reports aid in the debugging and validation phases for simulators.

The user interfaces for IMS and DSS have been made as compatible as possible. This minimizes the time needed to become familiar with the system. Files are created which describe the system to be modeled along with its operating characteristics and workload. In the case of IMS these input files are translated into analytic model formulas which describe on a gross level the behavior of the system. On the other hand DSS interprets the input files as components of a discrete event simulation model which is then run in batch mode.

As stated before IMS can produce estimates for device characteristics that would be required for the modeled systems to meet specified performance criteria. The DSS simulator has access to these device estimates through the IMS/DSS interface connecting the two sub-systems. In this way a simulation model can provide a detailed analysis of a system which uses as part of its input the optimized parameters from the IMS model. This interface provides an easy way of passing device characteristics from IMS to DSS. The IMS/DSS Interface provides a bridge between the analytic and simulation sub-systems which makes possible the routine application of the hierarchical modeling technique a practical reality. Besides giving the network designer an easy way of going from IMS to DSS this interface provides an almost automatic method for verifying the analytic model through simulation.

We can also go in the opposite direction, from a model in DSS to one in IMS. A detailed model of certain parts of a network can produce results which are then summarized and used for input parameters to the IMS model. For example, a simulator could be designed which models only the contention for resources in a host site which is intended to be at a later stage a site within a computer network. Part of the output from this simulator might be the number of resource requests over a period of time that can not be satisfied at the host site. These unsatisfied requests could be considered as the job input rate to the communication sub-network in the high level IMS model. In this way a detailed simulation model has helped to parameterize the IMS model. Since there are no standard means of summarizing detailed output from a simulator so that it may be factored into a high level model, HMS does not provide an automatic interface for this purpose.

The rational for the design of HMS is that analytic and simulation models of computer networks both have distinct advantages and costs. As such a system which gives the analyst easy access to both techniques can provide a powerful tool to aid in the network design process. Analytic models usually require less time to run and they can be used more easily on an interactive basis. When performance goals are quantified such as minimum average response time or maximum throughput analytic models can be frequently optimized to meet those specifications (Kleinrock 1976). Simulation models on the other hand require fewer simplifying assumptions but their developmental and computational costs can be significantly greater and they are difficult to optimize. Analytic and simulation models can be used in a complementary way, however, during the design stages for computer networks. In the beginning the number of options available to the designer is usually quite large. This suggests that an efficient method of narrowing the number of options while keeping in mind the performance goals is a desirable approach. Analytic models are therefore ideal at this stage. When fewer options remain a detailed study of the remaining candidates might be appropriate in which case simulation models would be the preferred approach. The hierarchical technique for modeling complex systems is not new (Bhandarkar 1976, Browne 1975, Sekino 1971, Brown 1977, Courtois 1977, Chiu 1978). HMS allows the analyst, however, to routinely apply hierarchical decomposition methods using analytic and simulation techniques.

2. HMS INPUT FILES

As stated above the user interfaces for IMS and DSS have been made as compatable as possible to minimize the time required to go from a model in one system to the other. The three main input files to HMS are the topology file (TP.FILE), Model Library file and M.FILE.

HMS views a network as a set of nodes and the TP.FILE describes how those nodes are interconnected. A network node is a set of resources which may be considered as a separate entity which communicates with the rest of the network through some kind of communication interface. Switching computers and host sites are probably the most widely used examples of what is meant by a node in a distributed network (Kleinrock 1976). However, the essence of a network is that fully or partly autonomous resources may communicates with each other. For instance, in a satellite/terrestial system one node may be identified with the satellite as a transmission medium while the earth stations may be individual nodes (Jacobs 1973). In a local area network each of the workstations may be thought of as a separate node communicating with one another, and perhaps a central site, over a shared bus. There is some latitude in what is defined as a node depending on the detail level of the model. A simulator which models gateways between interconnected local networks (C.A. Sunshine 1977) may identify an entire local network with a node because what is of interest is the required bandwidth between networks, not the behavior of a particular network.

As a computer network evolves in the design phase or as a functioning system nodes in a network may be added or deleted. Even if the number and types of nodes remain the same the topology of the network can be altered. This suggests that modeling nodes as separate entities and then connecting the different models together would be a natural means of simulating this type of system. In addition, it often occurs in computer networks that many nodes, such as the nodes in the communication network, have the same architecture. HMS provides a means to exploit this redundant feature of computer networks so that the analyst will not be required to write a separate and distinct program for each of the nodes. We would also like to study different nodes at the same or different levels of detail within the same model. We can go one step further and require that our computer network model incorporate different levels of detail within the same node when different areas are modeled. For example, a network designer might be interested in a detailed analysis of flow control with less emphasis on routing algorithms or concurrency control.

To provide this flexibility in modeling computer networks HMS incorporates as its first main input file the topology file (TP.FILE). The topology file contains node link statements which describe nodal interconnections, propagation delays - if any - and the data rates of the internodal devices. The second main input file is the Model Library file. Each model is a separate self contained program that models a specific node in a network. For similar nodes in a network a particular model may be duplicated any number of times. A model contains three main sections: The System Description section which describes the hardware components and the paths between these components at a node; a Workload Description section which describes the resource utilization at nodes and finally, the Resource Manager section which describes the operating system components. Each model is coded in a language specially designed for simulation of computer systems called the Extendable Computer System Simulator (ECSS) (Kosy 1975, FEDSIM). The third input file is the M.FILE. The M.FILE is a data file which maps a model to a specific node: Node N will be modeled by model type M. An example of these files is given in the following section.

3. HMS AND RECONFIGURABLE NETWORK MODELS

In building network wide models HMS allows for an incremental approach in the debugging and verification stages of network models. By making simple changes to the TP.FILE and M.FILE the network configuration can be changed from a two node network (Complexity Level A) consisting of two host sites, to a three node network (Complexity Level B) consisting of two host sites and a switching node and finally to an arbitrary configuration (Complexity Level C) consisting of nine nodes (See Figure 3-1). By concentrating on the simpler complexity levels (A & B) the work required in the debug and verification stages are considerably reduced. When complexity Level C is reached the analyst is usually assured that the network model is performing as planned.

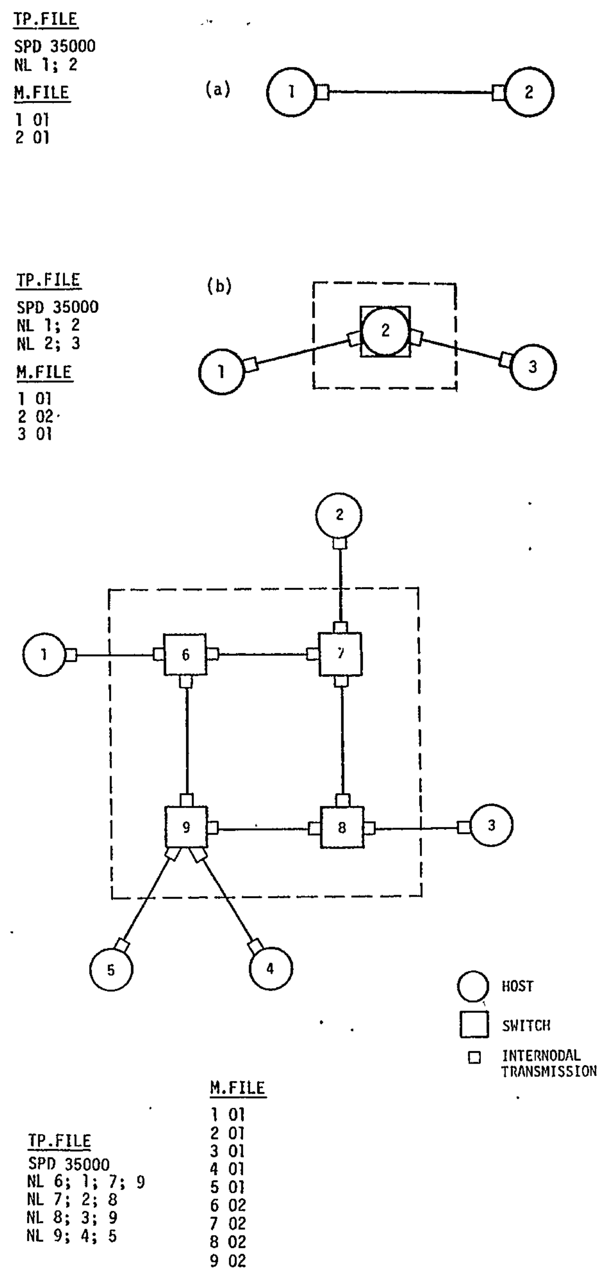

Figure 3-1 Complexity Levels
(a) Complexity Level A - Host to Host Configuration
(b) Complexity Level B - Host/Switch/ Host Configuration
(c) Complexity Level C - Arbitrary Configuration

Besides the fact that network simulators are difficult to verify they also tend to be very time consuming and costly to run. By having a direct correspondence between models and nodes in the network we can mix and match models depending on the type of experiments that are performed. For instance, suppose that one set of experiments has to do solely with the performance evaluation of the routing algorithms in the switching nodes of a particular network. A host site node for this set of experiments is of little interest except as a source and sink for messages. In order to reduce the complexity of the network wide model and therefore the run time costs we could use a simplified host site architecture with a detailed switching node model. Similarly, we could mix and match other models from the DSS Model Library depending on the type of experiments being performed while keeping the complexity of the network wide model to a minimum.

In the more traditional approach a single model is designed and built with the object of answering all of the questions which motivated the study at the outset. However, in the incremental approach we can be more selective. The model can be reconfigured depending on the requirements of a particular subset of the proposed experiments. This makes the experimentation stage less time consuming in terms of run time costs and the model output is reduced and directly applicable to the questions the experiments were designed to answer.

4. GRAPHICS AND MENU INPUT

Due to the fact that HMS has been designed so that the network topology data resides in one file (TP.FILE) it is a reasonably straight forward task to implement a graphics package for inputing that data. Figure 4-1 is a view of a multi-nodal network that can be created at a terminal. It has all of the information of the TP.FILE and M.FILE. By moving a hair line cursor on the screen and hitting a function key a circle will appear that represents a particular node of the network. At this point a menu appears for that node which requires several pieces of information including network nodal number, model number, the number and data rates of internodal paths (Figure 4-2). The information entered at this point in the menu is then displayed on the screen graphically. In this way a network topology is built incrementally which can be used either by the IMS or DSS systems.

NODE NUMBER ______________
MODEL TYPE______________
NODAL CONNECTIONS
NODE NUMBER______________
NUMBER OF PATHS__________
PATH DATA RATE___________

Figure 4-2 Menu Description of Network Nodes

Once a model type has been designated for a particular node it is then possible to parameterize the system description for that node by means of menus. A model type, in this instance, may be viewed as a template for a given node; the outline is there and default values for hardware characteristics may be used or they may be modified depending on the particular application. For example, the System Description section for Model Type 2 may consist of a central processor (CPU), Main Memory (MEMORY), channels (CHANNELS), terminals (TERMINALS), and disk drives (DISKS). These system components appear in a menu as depicted in Figure 4-3 along with their default characteristics. By entering numbers into the spaces provided the default characteristics of Model Type 2 are changed to the new inputs. Hence, no programming for this part of the model is required by the user with, naturally, a having in time and possible errors.

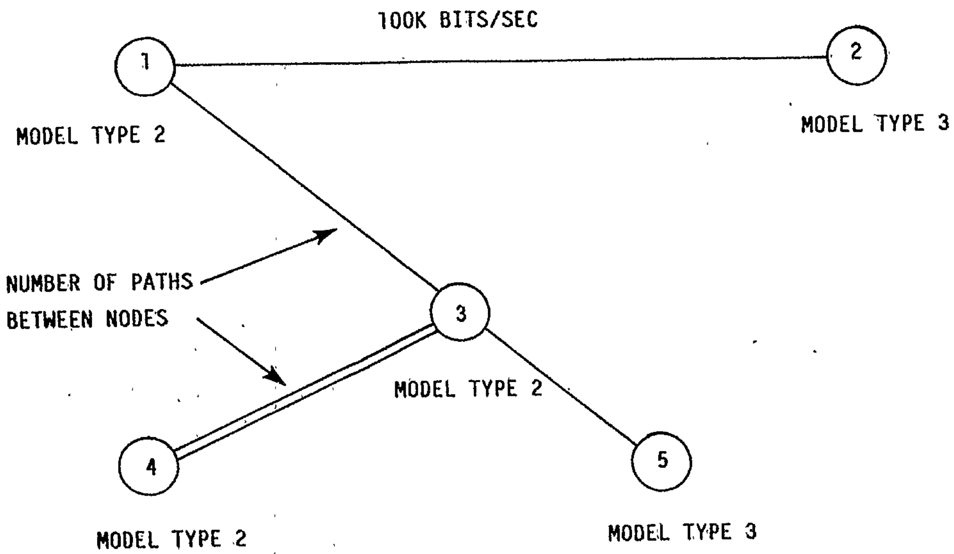

Figure 4-1 High Level View of Network

```
CPU
   NUMBER 1
   PROCESSING SPEED 1000000 INSTRUCTIONS/SEC
MEMORY
   NUMBER 1
   CAPACITY 2000000 BYTES
CHANNELS
   NUMBER 2
   DATA RATE 110000 BYTES/SEC
TERMINALS
   NUMBER  20
   DATA RATE 50000 BYTES/SEC
DISKS
   NUMBER 5
   CAPACITY 5000000 BYTES
   DATA RATE 95000 BYTES/SEC
T. PATH
   TERMINALS ARE CONNECTED TO CHANNELS
C. PATH
   CHANNELS ARE CONNECTED TO DISKS
```

Figure 4-3 System Description Menu Options for Model Type 2

5. SUMMARY

The Interactive Graphics front end for HMS greatly improves the ability of the analyst to reconfigure network models and experiment with different network topologies while minimizing the number of input errors. In addition, menus provide the capability of tailoring a model for a particular node while avoiding unnecessary coding. Future work will concentrate on developing these techniques for other phases of building and using computer network models including real time graphical monitors and back end data analysis.

6. REFERENCES

Bhandarkar, D.P. (1976), "A hierarchy of analytical models for complex computer systems." The European Comp. Conf. Computer. Syst. Evaluation Sept.

Brown, R.M., et. al., (1977), "Memory management and response time" CACM, Vol. 20, No. 3

Browne, J.C., et. al., (1975), "Hierarchical techniques for the development of realistic models of complex computer systems. "Proc. IEEE, Vol. 63, pp. 966-975

Chiu, W.W., Chow W.M., (1978), "A performance model of MVS." IBM Jour. Resc. and Dev. Vol. 17 No. 4

Courtois, P.J., (1977), Decomposability - Queueing and Computer System Applications. Academic Press, N.Y.

DuBois, D., (1981), "Distributed System Simulator (DSS) for Computer Networks," Proc. Twelth Annual Conf. on Modeling and Simulation, IEEE, the Society for Computer Simulation, University of Pittsburgh, Pittsburgh, PA.

DuBois, D., (1982), "A Hierarchical Modeling System for Computer Networks", Proc. Computer Network Symposium, ACM SIGCOM, SIGMETRICS and SIGOPS, University of Maryland.

FEDSIM (Federal Computer Performance Evaluation and Simulation Center), Washington, D.C. 20330 ECSS II Extended Statistics and Trace Features

Jacobs, I.M., et. al., (1973), General Purpose Packet Satellite Networks, IEEE Proc., Vol. 66, No. 11, pp. 1448-1496.

Kleinrock, L., (1976), Queuing Systems, Volume II: Computer Applications. New York: Willy Interscience

Kosy, Donald W., (1975) The ECSS II Language for Simulating Computer Systems Rand, Santa Monica

Sekino, A. (1971), "Performance evaluation of multiprogrammed time-shared computer systems." Tech. Rep. 103, MIT, Cambridge, Mass.

Sunshine, C.A., (1977), Interconnection of Computer Networks, Computer Networks, 1, 3, pp. 175-195.