<u>Algorithm</u>

```
BEGIN
REPEAT
     READ IN-FUNOP;
     GET-CORRES-SDESC;
     REPEAT
          GET-NEXT-SCREC;
          INCLUDE-ARC-COST;
          IF BASIC-NODE
               BEGIN
               REPEAT
                    GET-NEXT-PSPEC;
                    ADD-TO-MIN-TOTAL;
               UNTIL END-OF-DATA-PSPEC;

               REPEAT
                    GET-NEXT-DEPSP;
                    UPDATE-CORRES-PWT-TOT;
               UNTIL END-OF-DATA-DEPSP;
               END
          ELSE IF FUNCTION-NODE
               BEGIN
               GET-CORRES-FUNOP;
               ADD-TO-MIN-TOT;
               REPEAT
                    GET-NEXT-PVAL;
                    GET-CORRES-PWT;
                    ADD-TO-MIN-TOT;
               UNTIL END-OF-DATA-PVAL;

               REPEAT
                    GET-NEXT-DEPSP;
                    GET-CORRES-PWT;
```

```
                IF MATCH
                     BEGIN
                     CALCULATE-PARM-WTS;
                     UPDATE-CORRES-PWT-TOT;
                     END
                ELSE UPDATE-CORRES-PWT-TOT;
           UNTIL END-OF-DATA-DEPSP;

           IF UNMATCHED-PWTS
                UPDATE-CORRES-PWT-TOT;
           END;
UNTIL END-OF-DATA-SCREC;
UPDATE-MIN-RESOURCE-REQ;
REPEAT
     NEXT-CALCULATED-PWT-TOT;
     IF MATCHES-EXISTING-PWT
     UPDATE-PWT
     ELSE INSERT-PWT;
     IF UNMATCHED-EXISTING-PWT
     DELETE-PWT;
UNTIL END-OF-CALCULATED-PWT;
UNTIL END-OF-FILE-IN-FUNOP;
END;
```

GENREP Design Specifications
Basic Performance Reports

Purpose:

To demonstrate the automatic model parameter derivation, resource demand analysis, and workload characterization capabilities.

General Description:

Analyze the performance data in the data base and produce the following reports:

1. Derive queueing network model parameters
2. Derive estimated (average) elapsed time of the scenario
3. Print estimated (average) elapsed time for each IPIP request within the scenario
4. Print distribution of IPIP requests.

Input:

Scenarios to be analyzed

SDESC data base, schemas:

1. SDESC
2. SCREC
3. PSPEC
4. PVAL

FUN data base

CPU and I/O wait times (from CADS)

Output

Performance reports listed above.

Algorithm:

```
REPEAT
    READ IN-SCENARIO;
    GET-CORRES-SCENARIO-DESCR;
    REPEAT
        GETNEXT-SCREC;
        INCLUDE-ARC-COST;
        IF TYPE = FUN
            BEGIN
            GET-CORRES-MIN-RESOURCE-REQ;
            UPDATE-TOTALS;;
            REPEAT
                GET-NEXT-PVAL;
                GET-CORRES-PARMWT;
                COMPUTE-RESOURCE-REQUIREMENTS;
                UPDATE-ELAPSED-TIME;
            UNTIL END-OF-DATA-PVAL;
            END
        ELSE BEGIN
            IF TYPE = BASIC
                BEGIN
                GET-PERF-SPECS;
                ADD-RESOURCE-REQUIREMENTS;
                END
            END;
    UNTIL END-OF-DATA-SCREC;

    COMPUTE-PRINT-MODEL-PARMS;
    COMPUTE-PRINT-RESPONSE-TIMES;
    COMPUTE-PRINT-IPIP-REQUEST-DATA;
UNTIL END-OF-DATA-IN-SCENARIO;
END;
```

The ADEPT demonstration consists of the performance evaluation of the "Query" IPIP user scenario. The fundamental IPIP operations are first defined in the SDESC data base. FUNA is used to compute the resource requirements for each of them and update the FUN data base. Figure 5.2 shows the FUNA results of the evaluation of the IPIP component RETRIEVE. The minimum resource requirements for each component are shown. The variable requirements are represented by parameters associated with components with dependent specifications. Information is provided on the status of the updated FUN data base.

Next, the software scenarios are defined and loaded into the SDESC data base. GENREP is run to produce the queueing network model parameters as shown in Figure 5.3. CADS is then run with these parameters giving the results shown in Figure 5.4. The CPU and DISK wait time from the CADS results are used to run GENREP again to produce the reports shown in Figures 5.5 and 5.6.

Figure 5.5 shows the number of times each component is executed, and the CPU time and number of I/O's required each time. The estimated elapsed time is computed by multiplying the CPU time and number of I/O's by the appropriate wait times from the CADS results.

Figure 5.6 shows the estimated elapsed times in 100 ms. intervals and their frequency of occurrence. The frequencies are also reported for several types of components. The FIND and RETRIEVE are the components used most often. OTHER represents the remainder of the IPIP components and USER DEF represents statistics for components that are defined by the application system designer.

The elementary model results in Figure 5.4 indicate that the response times for the scenarios are unsatisfactory and that the CPU is the bottleneck resource. The performance report in Figure 5.5 shows the average elapsed time and CPU time requirements of each of the software components. Figure 5.6 is the frequency of the elapsed times of the software components. It indicates, in this example, that the FIND and OTHER (actually the run-time binding) operations have the longest elapsed times. Improvements to these modules will improve the overall response time. Since the RETRIEVE operation is called most frequently, small improvements in the design (CPU requirements) will have a large impact on the overall performance.

A comprehensive example of the IPIP evaluation is in the Appendix. The investigation of design and configuration alternatives is demonstrated there.

The ADEPT system is functional and is suitable for interactive use for design evaluations. Many enhancements are still needed for it to meet the functional specifications of the comprehensive system.

```
EVALUATION OF FUNDAMENTAL OPERATION * RETRIEVE    * 15 JUL 80

                                        CPU (SECS)  #I/O

COMPONENT NAME SENDMSG
          MINIMUM REQUIREMENT              .0140      0

COMPONENT NAME DBCS
          MINIMUM REQUIREMENT              .0050      0

COMPONENT NAME REC.XLATE
          MINIMUM REQUIREMENT              .0020      0

COMPONENT NAME DMS-RETRV
          MINIMUM REQUIREMENT              .0170      0
          DEPENDS ON  OVHD                 .0180      3     ** FOR EACH
          DEPENDS ON  READS                .0060      1     ** FOR EACH

TOTAL:    LINKAGE                          .0030

          MINIMUM REQUIREMENT              .0410      0

          PARAMETER   OVHD                 .0180      3     ** FOR EACH

          PARAMETER   READS                .0060      1     ** FOR EACH

********************

ADDED RETRIEVE
 MINIMUM REQUIREMENTS:

              QUAL= EA     CPU=     .0410  I/O=    0.0000  OS=    0.0000

  PARAMETERS:

     ADDED PARM OVHD

                   DESCR= ORT,LDT    CPU=    .0180 I/O=   3.0000 OS=   0.0000

     ADDED PARM READS

                   DESCR= DATA       CPU=    .0060 I/O=   1.0000 OS=   0.0000

********************
```

FIGURE 5.2. FUNA RESULTS FOR COMPONENT RETRIEVE

```
PERFORMANCE REPORTS FOR SCENARIO * QUERY      * AS OF  15 JUL 80

10 USERS

==================

CALCULATED MODEL PARAMETERS:
CPU RATE =   28.1690
BR PROB:
  BP TERM: .0238
  BP CPU: .9762

SPECIFIED MODEL PARAMETERS:
TERM RATE =  .0333
NUMBER OF USERS = 10

==================
```

FIGURE 5.3. QUEUEING NETWORK MODEL PARAMETERS FOR QUERY SCENARIO

Response Time	4.03 seconds
CPU Wait Time	.0566 seconds
Service Time	.0355 seconds
Queue Time	.0211 seconds
I/O Wait Time	.0393 seconds
CPU Utilization	44%
I/O Utilization	38%

FIGURE 5.4. QUERY MODEL RESULTS

PERFORMANCE REPORTS FOR SCENARIO * QUERY * AS OF 15 JUL 80

10 USERS

STATISTICS BY COMPONENT

	DESCRIPTION	#REQUESTS	ELAPSED SECS.	CPU SECS.	#I/O
COMPONENT QLP	PARSER SPECIFIED	1.	.0319	.0200	0.
COMPONENT SENDMSG	TO IPIP MINIMUM	1.	.0223	.0140	0.
COMPONENT DBCS-BIND	RUN TIME SPECIFIED	1.	.5749	.1390	9.
COMPONENT FIND	NODAL DEFL OVHD	1.	.3822	.0920	6.
COMPONENT RETRIEVE	NODAL DEFL OVHD	1.	.2118	.0590	3.
	READS	24.	.1142	.0470	1.
SUB-TOTAL		25.	2.9519	1.1870	27.
COMPONENT PRES-SERV	TO USER SPECIFIED	1.	.0159	.0100	0.
**********		-------	--------	-------	----
ALL COMPS		30.	3.9791	1.4620	42.
LINKAGE			.0462	.0290	
TOTAL		30.	4.0253	1.4910	42.

FIGURE 5.5. QUERY COMPONENT STATISTICS

PERFORMANCE REPORTS FOR SCENARIO * QUERY * AS OF 15 JUL 80

10 USERS

FREQUENCY OF ESTIMATED ELAPSED TIMES

INTERVAL (SECS)	TOTAL	#FINDS	#RETRIEVES	#OTHER	#USER DEF
0.000- .099	3.	0.	0.	1.	2.
.100- .199	24.	0.	24.	0.	0.
.200- .299	1.	0.	1.	0.	0.
.300- .399	1.	1.	0.	0.	0.
.400- .499	0.	0.	0.	0.	0.
.500- .599	1.	0.	0.	0.	1.
.600- .699	0.	0.	0.	0.	0.
.700- .799	0.	0.	0.	0.	0.
.800- .899	0.	0.	0.	0.	0.
.900- .999	0.	0.	0.	0.	0.
1.000-1.099	0.	0.	0.	0.	0.
1.100-1.199	0.	0.	0.	0.	0.
1.200-1.299	0.	0.	0.	0.	0.
1.300-1.399	0.	0.	0.	0.	0.
1.400-1.499	0.	0.	0.	0.	0.
1.500-*****	0.	0.	0.	0.	0.
TOTALS	30.	1.	25.	1.	3.

FIGURE 5.6. QUERY ELAPSED TIME FREQUENCIES

CHAPTER 6
CONCLUSIONS

6.0 Summary

This document defines a methodology for the prediction and evaluation of the performance of software from design specifications. It satisfies the requirements identified in the first chapter. A basic methodology is presented that is sufficient for designs without complex interrelationships of components and when the performance does not depend on the environment in which it will execute. Additional techniques are given for analyzing the effect of environmental factors: data dependency, competitive effects, and memory contention. The modeling of design complexities (internal concurrency, synchronization, mutual exclusion and blocking) are explained. Specifications are included for a comprehensive performance prediction tool. A prototype demonstrating its feasibility is presented.

The methodology has been successfully used for the evaluation of several software systems. The Appendix illustrates its use through several design iterations. Minimal performance specifications are required for the analysis. The effort required to obtain the model solution is also minimal. The results have, in the IPIP application, impacted early design decisions, thus precluding performance crises and costly revisions to the software after implementation.

The methodology embodies a number of new proposals that are essential for software performance prediction:

1. The specification of performance determining factors
2. The graphical representation of the significant structural elements of software systems including:
    1. hierarchical structuring and recursion
    2. typical control representations
    3. concurrency and synchronization
    4. blocking for mutual exclusion
3. Enhancements to static graph analysis techniques for data dependent execution characteristics, hierarchical structuring, and the spectrum of nodes and arcs.
4. The uniting of graph analysis techniques and queueing network models: the algorithms for the computation of elementary model parameters as well as swapping, memory, synchronization, and blocking model parameters.
5. Queueing network model formulations for analyzing synchronization and blocking.

The proposals are consolidated to produce the desired performance prediction and evaluation capabilities.

6.1 Limitations

The methodology has several limitations. The first is the reliance on the performance specifications; the predictions are only as good as the data upon which they are based. Several compensating features are used:

1. The use of upper and lower bounds to evaluate the potential range of results

2. The iterative evaluation approach with actual performance measurements replacing the specifications during implementation
3. The identification of critical components whose actual performance measurements are most important with respect to producing accurate predictions
4. The feedback on actual resource requirements versus the specifications to improve the accuracy of future specifications.

Queueing network models are used as the basis for the analyses, however, only mean values for performance metrics are obtained from them. Approximation techniques are used to analyze design complexities. The models are easily analyzed, but the results are not exact. This approach is preferable, in most cases, to more complex analyses since the early design specifications are imprecise and the design is likely to be modified before the final implementation. When the performance of new software is crucial, more detailed analysis is desirable as soon as accurate data is available. A simulation model could then be automatically generated using measurement data and information from the specifications and execution graphs.

Unusual software design and resource usage combinations may not yield to the iterative model solution approaches. Nevertheless, model solutions for most, actual software designs will likely be within acceptable bounds. Anomalies have not yet been detected, therefore, further investigations have not been necessary.

Other design issues such as the performance of distributed software have not been addressed. Such additional features can be handled within the framework of the methodology by extensions to the queueing network models.

6.2 Further Research

Several areas for further research are suggested. The performance prediction tool should be integrated with a functional specification tool and perhaps a verification tool. A common data base should be used for all the design-aid tools to reduce redundancy, ensure consistency, and ease the burden on the designer.

Several of the functions of the performance prediction tool need further work. It should include a mechanism for mapping various formats for design specifications into the desired performance specifications and for providing suitable default values. Some guidelines are needed for system designers for determining reasonable initial performance specifications. The graphical input and output features need to be developed. Good automatic heuristics for the iterative queueing network model solutions are needed.

Additional features would enhance the usefulness of the tool. Data and file structures should be more explicit in the representation and analysis techniques. This would facilitate more automatic evaluation of design alternatives.

Work is needed to automatically detect and correct system bottlenecks. Proposed resolutions to the bottlenecks may be configuration changes, software design modifications or both. Some typical design alternatives should be studied and the tool modified to automatically evaluate them. This would minimize the effort required of the designer.

The capabilites for supporting and tracking the versions of the software should be automated and integrated with the automatic evaluation of design alternatives. A project tracking system should be incorporated to monitor progress and detect serious discrepancies between specifications and actual measurements. Problems should be detected as early as possible during development.

Additional queueing network modeling techniques are needed to evaluate distributed processing execution environments. The evaluation of the execution of software systems in a paged memory environment is also a problem. The prediction of paging activity is complicated by its dependence on the load on the host system. The information in the execution graphs should be sufficient to provide queueing network model parameters for such an analysis.

The applicability of the methodology to software systems with critical real-time constraints should be investigated. The approximation techniques are adequate for early design evaluations; however, they may need to be supplemented by other evaluation techniques at some point in the design process. The critical point and the supplementary techniques need to be formulated.

Finally, additional software systems need to be evaluated throughout the design and implementation phases. This approach was used throughout the development of this methodology to ensure that the major problems were resolved. Additional problems will be discovered as computer hardware and software technology continues to advance and new software systems are developed to utilize the technology.

APPENDIX

The scenario used to demonstrate the application of the methodology through several design iterations is the "Indentured Parts List" (IPL). It is a query from a terminal asking for information about all parts that are contained in a particular section of an aircraft. Each part contains 20 smaller parts. The smaller parts contain 10 other parts which contain 5 parts.

The highest level processing steps are depicted in the execution graph in Figure A.1. Figure A.2 shows the graph of the next level of detail for the "Get contains-parts relations." Figures A.3 through A.8 show the processing details of some of the data management system (IPIP) fundamental operations called by this application program.

The performance specifications are collected and the information is loaded into the ADEPT data base. Next, FUNA is used to calculate the resource requirements for each of the fundamental operations. Figures A.9 through A.13 show the results of this step. For example, the "Send message" (SENDMSG) operation always requires a minimum of 14 ms. of CPU time (Figure A.9) while "Resource manager" (RMGR) requires a minimum of 5 ms. plus 6 ms. and one I/O each time data must be retrieved from the disk (Figure A.10).

The host system configuration that will be used to run the scenario is illustrated by the queueing network model in Figure A.14. It consists of up to 10 terminals, a CPU with a processing rate of one million instructions per second (1 MIP), four disk drives, and two controllers shared by the disks. Model parameters for disk and controller service rates are provided with the environmental specifications. The terminal think time and the number of users are specified with the scenario description.

GENREP is run to calculate the remaining model parameters (the CPU service rate and the branching probabilities) from the graphs. These model parameters are shown in Figure A.15. The performance reports in Figures A.16 and A.17 are also produced. They show the estimated elapsed processing time, CPU, and I/O requirements of each component (running without competing work) and a frequency distribution of elapsed time requirements.

The queueing network model is then run to obtain the average response time for one to five IPL users with no other competitive effects. The results in Figure A.18 indicate that the estimated average response time with one user is 49.5 seconds and 195.1 seconds with 5 users. The response time is unacceptable for an interactive application. Investigation of the total time spent at each of the devices in the system indicates that the bottleneck device is the CPU.

GENREP is run again to obtain information about the elapsed processing time of each of the components when 5 terminals are active. As shown in Figures A.19 and A.20, the majority of the elapsed time is spent in the FIND and RETRIEVE of the contains-parts relations.

Analysis of the graph in Figure A.2 shows that 3 of the FIND operations occur in loops. Processing begins with a FIND to obtain all contains-parts at level 1. The first of these is retrieved. Another FIND is done to obtain a list of all its contains-parts at level 2; and so on. After all of the contains-parts have been retrieved at level 4, processing returns to level 3 to RETRIEVE the next part. A FIND was previously executed to locate all of these parts; however, since IPIP only retains the most recent find list, another FIND is required. The result is many "redundant" FIND's. If IPIP is modified to retain multiple find lists, the "redundant"

FIND´s can be eliminated.

The revised graph for the "Get contains-parts relations" using the multiple find list feature is in Figure A.21. The ADEPT data base is updated to reflect the revised processing requirements for FIND´s and GENREP is run to obtain the revised queueing network model parameters shown in Figure A.22. The model is then run to obtain the response times and queueing delays in Figure A.23. The performance reports for the revised scenario with 5 active users are in Figures A.24 and A.25.

The response time was reduced dramatically (from 195 seconds to 86 seconds for 5 users), but it is still unsatisfactory. The next study determines the hardware configuration required to support the scenario. A faster CPU is needed since it is the current bottleneck. The queueing network model is run, with an increased CPU rate, until suitable response times are obtained. As the CPU speed increases, however, the I/O request rate also increases and it is necessary to add additional I/O devices to handle the load.

The resulting proposed hardware configuration is shown in Figure A.26: a CPU that processes 5 million instructions per second (5 MIP), 8 disk drives, and 4 channel controllers are required. The model results for the upgraded configuration are in Figure A.27 and the performance reports are in Figures A.28 and A.29.

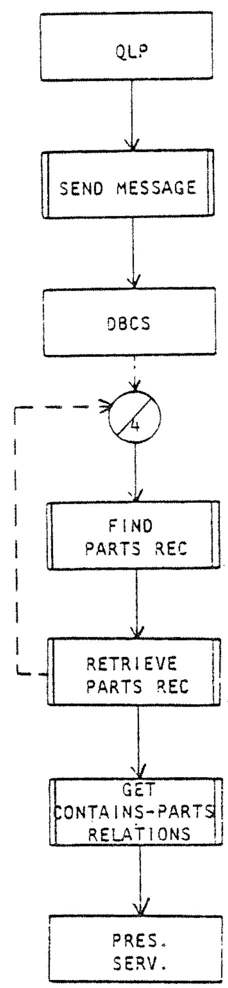

FIGURE A.1. INDENTURED PARTS LIST (IPL) SCENARIO

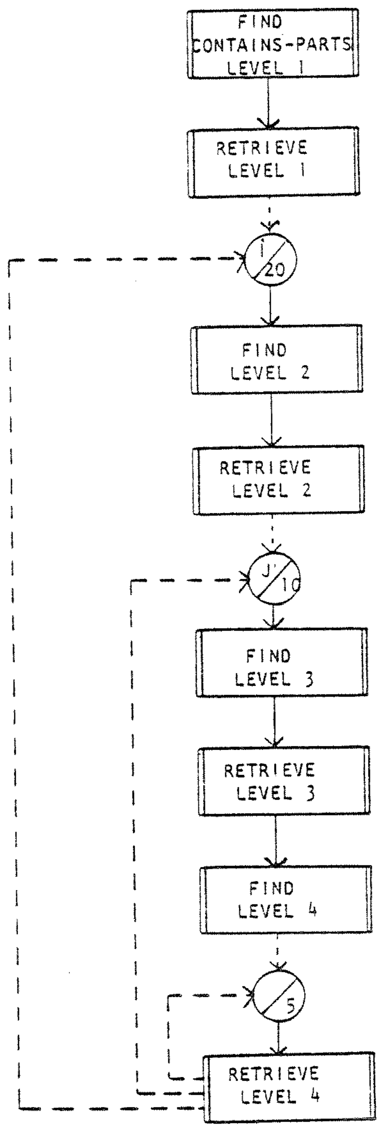

FIGURE A.2. GET CONTAINS-PARTS RELATIONS

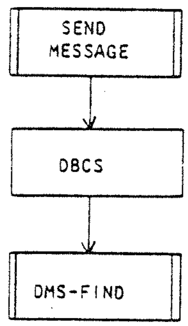

FIGURE A.3. FIND

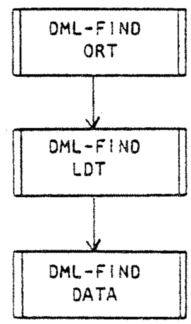

FIGURE A.4. DMS-FIND

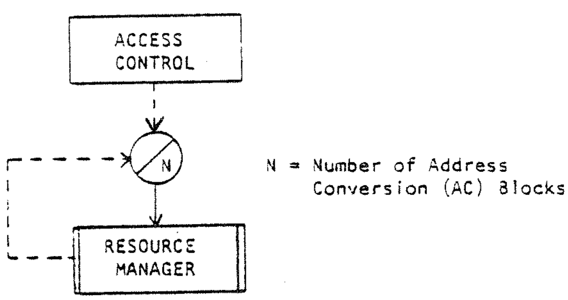

FIGURE A.5. DML-FIND

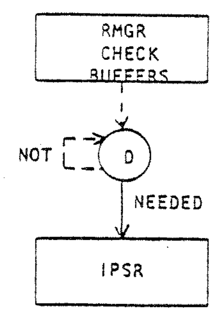

FIGURE A.6. RESOURCE MANAGER

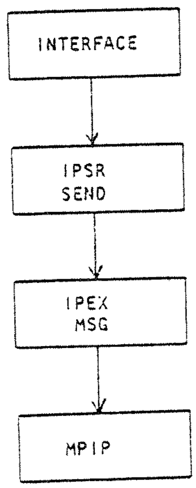

FIGURE A.7. SEND MESSAGE

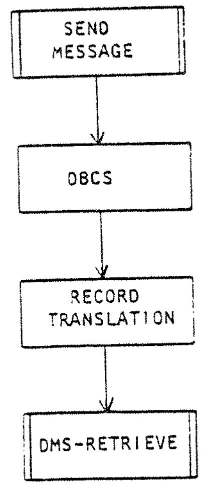

FIGURE A.8. RETRIEVE

EVALUATION OF FUNDAMENTAL OPERATION * SENDMSG * 15 JUL 80

		CPU (SECS)	#I/O
COMPONENT	NAME INTERFACE MINIMUM REQUIREMENT	.0020	0
COMPONENT	NAME IPSR SEND MINIMUM REQUIREMENT	.0020	0
COMPONENT	NAME IPEX MSG MINIMUM REQUIREMENT	.0020	0
COMPONENT	NAME MPIP MINIMUM REQUIREMENT	.0020	0
TOTAL:	LINKAGE	.0060	
	MINIMUM REQUIREMENT	.0140	0

****

ADDED SENDMSG
MINIMUM REQUIREMENTS:

QUAL= EA CPU= .0140 I/O= 0.0000 OS= 0.0000

PARAMETERS:

****

FIGURE A.9. SEND MESSAGE RESOURCE REQUIREMENTS

```
EVALUATION OF FUNDAMENTAL OPERATION * RMGR        * 15 JUL 80

                                  CPU (SECS)  #I/O

COMPONENT NAME RMGR
          MINIMUM REQUIREMENT           .0050     0

COMPONENT NAME IPSR
          DEPENDS ON  READS             .0060     1     ** FOR EACH

TOTAL:    LINKAGE                      0.0000

          MINIMUM REQUIREMENT           .0050     0

          PARAMETER   READS             .0060     1     ** FOR EACH

********************

ADDED RMGR
 MINIMUM REQUIREMENTS:

               QUAL= EA    CPU=     .0050  I/O=    0.0000  OS=    0.0000

  PARAMETERS:

     ADDED PARM READS

                    DESCR= READS NEED CPU=    .0060 I/O=   1.0000 OS=   0.0000

 ********************
```

FIGURE A.10. RESOURCE MANAGER RESOURCE REQUIREMENTS

```
EVALUATION OF FUNDAMENTAL OPERATION * DMS-FIND   * 15 JUL 80

                              CPU (SECS)  #I/O

COMPONENT NAME DML-FIND
          MINIMUM REQUIREMENT        .0110      0
          DEPENDS ON  OVHD           .0120      2     ** FOR EACH

COMPONENT NAME DML-FIND
          MINIMUM REQUIREMENT        .0110      0
          DEPENDS ON  OVHD           .0120      2     ** FOR EACH

COMPONENT NAME DML-FIND
          MINIMUM REQUIREMENT        .0110      0
          DEPENDS ON  USR-AC-BLK     .0060      1     ** FOR EACH
          DEPENDS ON  OVHD           .0120      2     ** FOR EACH

TOTAL:    LINKAGE                    .0020

          MINIMUM REQUIREMENT        .0350      0

          PARAMETER   OVHD           .0360      6     ** FOR EACH

          PARAMETER   USR-AC-BLK     .0060      1     ** FOR EACH

********************

ADDED DMS-FIND
 MINIMUM REQUIREMENTS:

              QUAL= EA    CPU=     .0350  I/O=    0.0000  OS=     0.0000

  PARAMETERS:

     ADDED PARM OVHD

                   DESCR= ORT         CPU=     .0360 I/O=    6.0000 OS=    0.0000

     ADDED PARM USR-AC-BLK

                   DESCR= USR ADDR    CPU=     .0060 I/O=    1.0000 OS=    0.0000

 ********************
```

FIGURE A.11. DMS-FIND RESOURCE REQUIREMENTS

```
EVALUATION OF FUNDAMENTAL OPERATION * FIND        * 15 JUL 80

                                CPU (SECS)  #I/O

COMPONENT NAME SENDMSG
          MINIMUM REQUIREMENT          .0140     0

COMPONENT NAME DBCS
          MINIMUM REQUIREMENT          .0050     0

COMPONENT NAME DMS-FIND
          MINIMUM REQUIREMENT          .0350     0
          DEPENDS ON  OVHD             .0360     6     ** FOR EACH
          DEPENDS ON  USR-AC-BLK       .0060     1     ** FOR EACH

TOTAL:    LINKAGE                      .0020

          MINIMUM REQUIREMENT          .0560     0

          PARAMETER   OVHD             .0360     6     ** FOR EACH

          PARAMETER   USR-AC-BLK       .0060     1     ** FOR EACH

********************

ADDED FIND
 MINIMUM REQUIREMENTS:

               QUAL= EA    CPU=     .0560  I/O=    0.0000  OS=    0.0000

  PARAMETERS:

     ADDED PARM OVHD

                    DESCR= ORT        CPU=    .0360 I/O=    6.0000 OS=    0.0000

     ADDED PARM USR-AC-BLK

                    DESCR= USR ADDR   CPU=    .0060 I/O=    1.0000 OS=    0.0000

 ********************
```

FIGURE A.12. FIND RESOURCE REQUIREMENTS

```
EVALUATION OF FUNDAMENTAL OPERATION * RETRIEVE   * 15 JUL 80

                                      CPU (SECS)  #I/O

COMPONENT NAME SENDMSG
          MINIMUM REQUIREMENT            .0140      0

COMPONENT NAME DBCS
          MINIMUM REQUIREMENT            .0050      0

COMPONENT NAME REC.XLATE
          MINIMUM REQUIREMENT            .0020      0

COMPONENT NAME DMS-RETRV
          MINIMUM REQUIREMENT            .0170      0
          DEPENDS ON   OVHD              .0180      3     ** FOR EACH
          DEPENDS ON   READS             .0060      1     ** FOR EACH

TOTAL:    LINKAGE                        .0030

          MINIMUM REQUIREMENT            .0410      0

          PARAMETER    OVHD              .0180      3     ** FOR EACH

          PARAMETER    READS             .0060      1     ** FOR EACH

********************

ADDED RETRIEVE
 MINIMUM REQUIREMENTS:

             QUAL= EA     CPU=      .0410  I/O=     0.0000  OS=     0.0000

  PARAMETERS:

     ADDED PARM OVHD

                  DESCR= ORT,LDT     CPU=     .0180 I/O=    3.0000 OS=    0.0000


     ADDED PARM READS

                  DESCR= DATA        CPU=     .0060 I/O=    1.0000 OS=    0.0000

********************
```

FIGURE A.13. RETRIEVE RESOURCE REQUIREMENTS

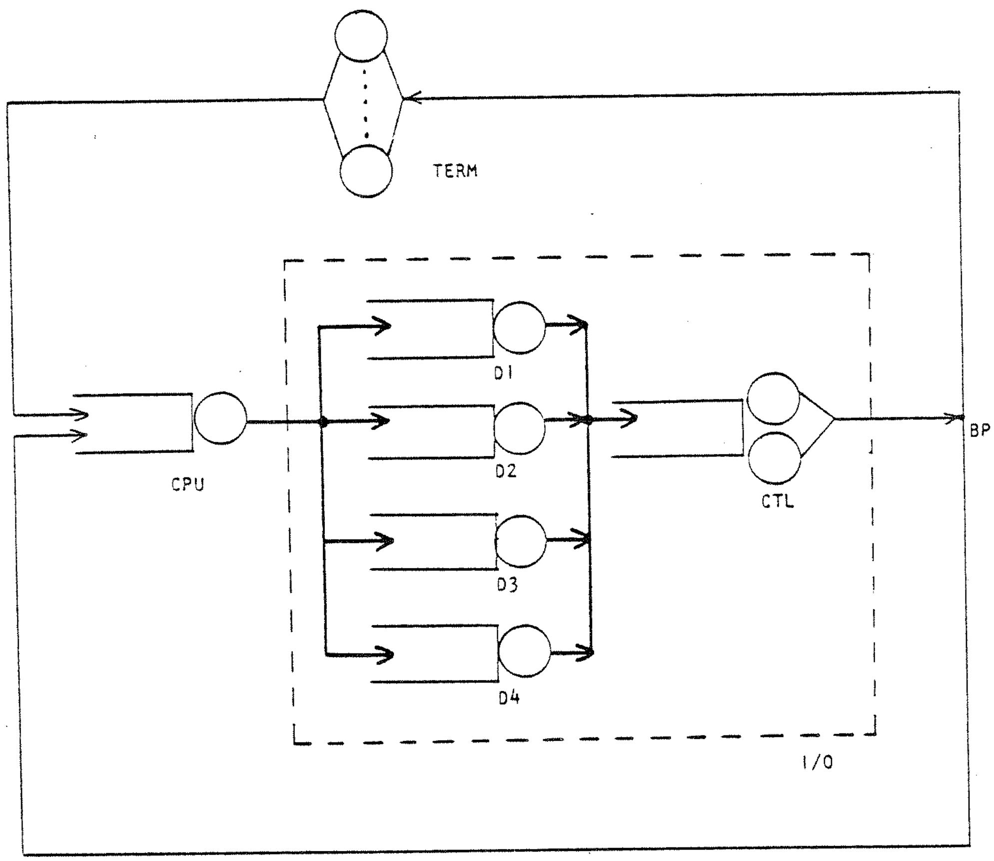

FIGURE A.14. IPL HOST SYSTEM CONFIGURATION

```
PERFORMANCE REPORTS FOR SCENARIO * IPL          * AS OF  15 JUL 80

5 USERS:

===================

CALCULATED MODEL PARAMETERS:
CPU RATE =      .9435
BR PROB:
  BP TERM: .0238
  BP CPU: .9762

SPECIFIED MODEL PARAMETERS:
TERM RATE =  .0333
NUMBER OF USERS =  5

===================
```

FIGURE A.15. CALCULATED QUEUEING NETWORK MODEL PARAMETERS

PERFORMANCE REPORTS FOR SCENARIO * IPL * AS OF 15 JUL 80

RUNNING WITHOUT COMPETING WORK:

STATISTICS BY COMPONENT

	DESCRIPTION	#REQUESTS	ELAPSED SECS.	CPU SECS.	#I/O
COMPONENT QLP	PARSER SPECIFIED	1.	.0200	.0200	0.
COMPONENT SENDMSG	TO IPIP MINIMUM	1.	.0140	.0140	0.
COMPONENT DBCS	RUN TIME SPECIFIED	1.	.9462	.2730	18.
COMPONENT FIND	PARTS REC				
	OVHD	1.	.3164	.0920	6.
	USR-AC-BLK	3.	.0994	.0620	1.
SUB-TOTAL		4.	.6146	.2780	9.
COMPONENT RETRIEVE	PARTS REC				
	OVHD	1.	.1712	.0590	3.
	READS	3.	.0844	.0470	1.
SUB-TOTAL		4.	.4244	.2000	6.
COMPONENT FIND	CONTAINS				
	OVHD	1.	.3164	.0920	6.
	USR-AC-BLK	0.	.0994	.0620	1.
	MINIMUM	440.	.0560	.0560	0.
SUB-TOTAL		441.	24.9565	24.7320	6.
COMPONENT RETRIEVE	CONTAINS				
	OVHD	1.	.1712	.0590	3.
	READS	0.	.0844	.0470	1.
	MINIMUM	440.	.0410	.0410	0.
SUB-TOTAL		441.	18.2113	18.0990	3.
COMPONENT PRES-SERV	TO USER SPECIFIED	1.	.0100	.0100	0.
**********		-------	--------	-------	----
ALL COMPS		894.	45.1970	43.6260	12.
LINKAGE			.8930	.8930	
TOTAL		894.	46.0900	44.5190	42.

FIGURE A.16. IPL COMPONENT STATISTICS

PERFORMANCE REPORTS FOR SCENARIO * IPL * AS OF 15 JUL 80

RUNNING WITHOUT COMPETING WORK:

FREQUENCY OF ESTIMATED ELAPSED TIMES

INTERVAL (SECS)	TOTAL	#FINDS	#RETRIEVES	#OTHER	#USER DEF
0.000- .099	889.	443.	443.	1.	2.
.100- .199	2.	0.	2.	0.	0.
.200- .299	0.	0.	0.	0.	0.
.300- .399	2.	2.	0.	0.	0.
.400- .499	0.	0.	0.	0.	0.
.500- .599	0.	0.	0.	0.	0.
.600- .699	0.	0.	0.	0.	0.
.700- .799	0.	0.	0.	0.	0.
.800- .899	0.	0.	0.	0.	0.
.900- .999	1.	0.	0.	0.	1.
1.000-1.099	0.	0.	0.	0.	0.
1.100-1.199	0.	0.	0.	0.	0.
1.200-1.299	0.	0.	0.	0.	0.
1.300-1.399	0.	0.	0.	0.	0.
1.400-1.499	0.	0.	0.	0.	0.
1.500-*****	0.	0.	0.	0.	0.
TOTALS	894.	445.	445.	1.	3.

FIGURE A.17. IPL ELAPSED TIME FREQUENCIES

	Number of Users				
	1	2	3	4	5
Response Time (secs)	46.1	72.2	107.4	149.0	192.7
Wait Time (secs)					
Per visit:					
CPU	1.06	1.68	2.52	3.51	4.55
I/O	.0374	.0374	.0375	.0375	.0375
Total:					
CPU	44.5	70.6	105.8	147.4	191.1
I/O	1.6	1.6	1.6	1.6	1.6

FIGURE A.18. MODEL RESULTS FOR IPL SCENARIO

PERFORMANCE REPORTS FOR SCENARIO * IPL * AS OF 15 JUL 80

5 USERS:

FREQUENCY OF ESTIMATED ELAPSED TIMES

INTERVAL (SECS)	TOTAL	#FINDS	#RETRIEVES	#OTHER	#USER DEF
0.000- .099	3.	0.	0.	1.	2.
.100- .199	440.	0.	440.	0.	0.
.200- .299	443.	440.	3.	0.	0.
.300- .399	5.	3.	2.	0.	0.
.400- .499	0.	0.	0.	0.	0.
.500- .599	0.	0.	0.	0.	0.
.600- .699	2.	2.	0.	0.	0.
.700- .799	0.	0.	0.	0.	0.
.800- .899	0.	0.	0.	0.	0.
.900- .999	0.	0.	0.	0.	0.
1.000-1.099	0.	0.	0.	0.	0.
1.100-1.199	0.	0.	0.	0.	0.
1.200-1.299	0.	0.	0.	0.	0.
1.300-1.399	0.	0.	0.	0.	0.
1.400-1.499	0.	0.	0.	0.	1.
1.500-*****	1.	0.	0.	0.	0.
TOTALS	894.	445.	445.	1.	3.

FIGURE A.19. IPL ELAPSED TIME FREQUENCIES WITH 5 USERS

PERFORMANCE REPORTS FOR SCENARIO * IPL * AS OF 15 JUL 80

5 USERS:

STATISTICS BY COMPONENT

	DESCRIPTION	#REQUESTS	ELAPSED SECS.	CPU SECS.	#I/O
COMPONENT QLP	PARSER				
	SPECIFIED	1.	.0859	.0200	0.
COMPONENT SENDMSG	TO IPIP				
	MINIMUM	1.	.0601	.0140	0.
COMPONENT DBCS	RUN TIME				
	SPECIFIED	1.	1.8472	.2730	18.
COMPONENT FIND	PARTS REC				
	OVHD	1.	.6200	.0920	6.
	USR-AC-BLK	3.	.3037	.0620	1.
SUB-TOTAL		4.	1.5312	.2780	9.
COMPONENT RETRIEVE	PARTS REC				
	OVHD	1.	.3658	.0590	3.
	READS	3.	.2393	.0470	1.
SUB-TOTAL		4.	1.0838	.2000	6.
COMPONENT FIND	CONTAINS				
	OVHD	1.	.6200	.0920	6.
	USR-AC-BLK	0.	.3037	.0620	1.
	MINIMUM	440.	.2405	.0560	0.
SUB-TOTAL		441.	106.4194	24.7320	6.
COMPONENT RETRIEVE	CONTAINS				
	OVHD	1.	.3658	.0590	3.
	READS	0.	.2393	.0470	1.
	MINIMUM	440.	.1760	.0410	0.
SUB-TOTAL		441.	77.8262	18.0990	3.
COMPONENT PRES-SERV	TO USER				
	SPECIFIED	1.	.0429	.0100	0.
**********		-------	--------	-------	----
ALL COMPS		894.	188.8966	43.6260	42.
LINKAGE			3.8344	.8930	
TOTAL		894.	192.7310	44.5190	42.

FIGURE A.20. IPL COMPONENT STATISTICS WITH 5 USERS

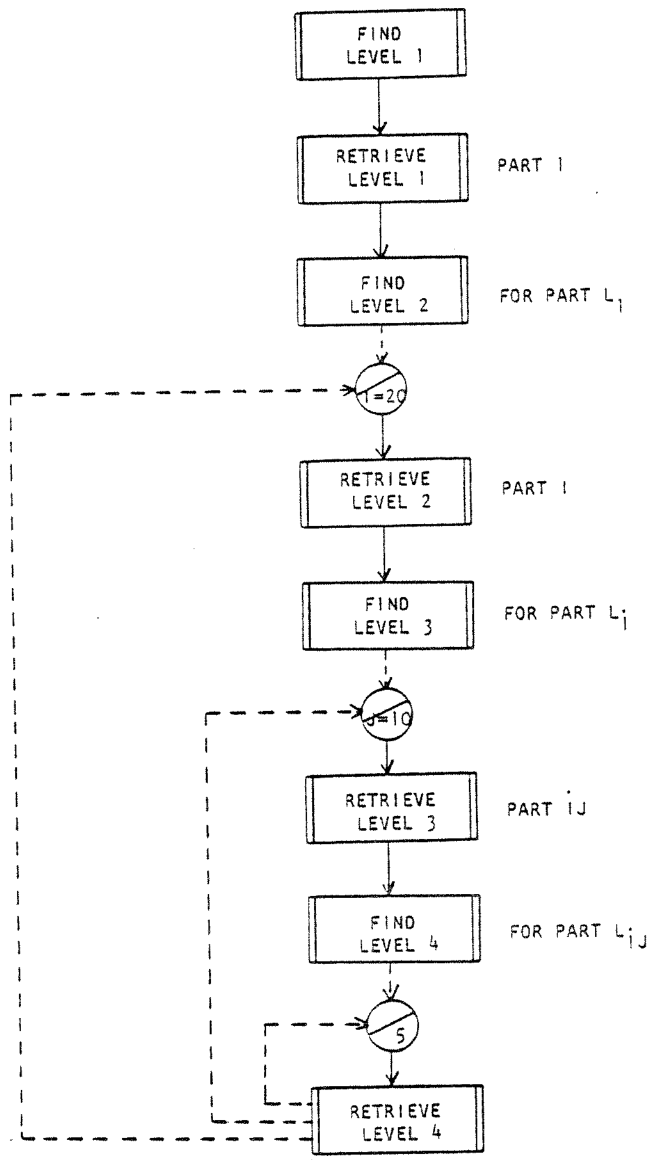

FIGURE A.21. GET CONTAINS-PARTS RELATIONS: MULTIPLE FIND LISTS

```
PERFORMANCE REPORTS FOR SCENARIO * IPL        * AS OF  15 JUL 80

MULTIPLE FIND FILES.  5 USERS:

==================

CALCULATED MODEL PARAMETERS:
CPU RATE =     2.5400
BR PROB:
  BP TERM: .0172
  BP CPU: .9828

SPECIFIED MODEL PARAMETERS:
TERM RATE =  .0333
NUMBER OF USERS =  5

==================
```

FIGURE A.22. CALCULATED QUEUEING NETWORK MODEL PARAMETERS: MULTIPLE FIND LISTS

	Number of Users				
	1	2	3	4	5
Response Time (secs)	25.1	34.6	48.0	65.4	85.6
Wait Time (secs)					
Per visit:					
CPU	.3937	.5573	.7886	1.087	1.437
I/O	.0374	.0375	.0376	.0377	.0377
Total:					
CPU	22.9	32.4	45.8	63.2	83.4
I/O	2.2	2.2	2.2	2.2	2.2

FIGURE A.23. IPL MODEL RESULTS: MULTIPLE FIND LISTS

PERFORMANCE REPORTS FOR SCENARIO * IPL * AS OF 15 JUL 80

MULTIPLE FIND FILES. 5 USERS:

FREQUENCY OF ESTIMATED ELAPSED TIMES

INTERVAL (SECS)	TOTAL	#FINDS	#RETRIEVES	#OTHER	#USER DEF
0.000- .099	3.	0.	0.	1.	2.
.100- .199	211.	0.	211.	0.	0.
.200- .299	235.	223.	12.	0.	0.
.300- .399	2.	0.	2.	0.	0.
.400- .499	0.	0.	0.	0.	0.
.500- .599	2.	2.	0.	0.	0.
.600- .699	0.	0.	0.	0.	0.
.700- .799	0.	0.	0.	0.	0.
.800- .899	0.	0.	0.	0.	0.
.900- .999	0.	0.	0.	0.	0.
1.000-1.099	0.	0.	0.	0.	0.
1.100-1.199	0.	0.	0.	0.	0.
1.200-1.299	0.	0.	0.	0.	0.
1.300-1.399	0.	0.	0.	0.	0.
1.400-1.499	0.	0.	0.	0.	0.
1.500-*****	1.	0.	0.	0.	1.
TOTALS	454.	225.	225.	1.	3.

FIGURE A.24. IPL ELAPSED TIME FREQUENCIES: MULTIPLE FIND LISTS

PERFORMANCE REPORTS FOR SCENARIO * IPL * AS OF 15 JUL 80

MULTIPLE FIND FILES. 5 USERS:

STATISTICS BY COMPONENT

	DESCRIPTION	#REQUESTS	ELAPSED SECS.	CPU SECS.	#I/O
COMPONENT QLP	PARSER				
	SPECIFIED	1.	.0728	.0200	0.
COMPONENT SENDMSG	TO IPIP				
	MINIMUM	1.	.0510	.0140	0.
COMPONENT DBCS	RUN TIME				
	SPECIFIED	1.	1.6723	.2730	18.
COMPONENT FIND	PARTS REC				
	OVHD	1.	.5611	.0920	6.
	USR-AC-BLK	3.	.2634	.0620	1.
SUB-TOTAL		4.	1.3512	.2780	9.
COMPONENT RETRIEVE	PARTS REC				
	OVHD	1.	.3279	.0590	3.
	READS	3.	.2088	.0470	1.
SUB-TOTAL		4.	.9542	.2000	6.
COMPONENT FIND	CONTAINS				
	OVHD	1.	.5611	.0920	6.
	USR-AC-BLK	7.	.2634	.0620	1.
	MINIMUM	213.	.2038	.0560	0.
SUB-TOTAL		221.	45.8227	12.4540	13.
COMPONENT RETRIEVE	CONTAINS				
	OVHD	1.	.3279	.0590	3.
	READS	9.	.2088	.0470	1.
	MINIMUM	211.	.1492	.0410	0.
SUB-TOTAL		221.	33.6965	9.1330	12.
COMPONENT PRES-SERV	TO USER				
	SPECIFIED	1.	.0364	.0100	0.
**********		-------	--------	-------	----
ALL COMPS		454.	83.6571	22.3820	58.
LINKAGE			1.6489	.4530	
TOTAL		454.	85.3060	22.8350	58.

FIGURE A.25. IPL COMPONENT STATISTICS: MULTIPLE FIND LISTS

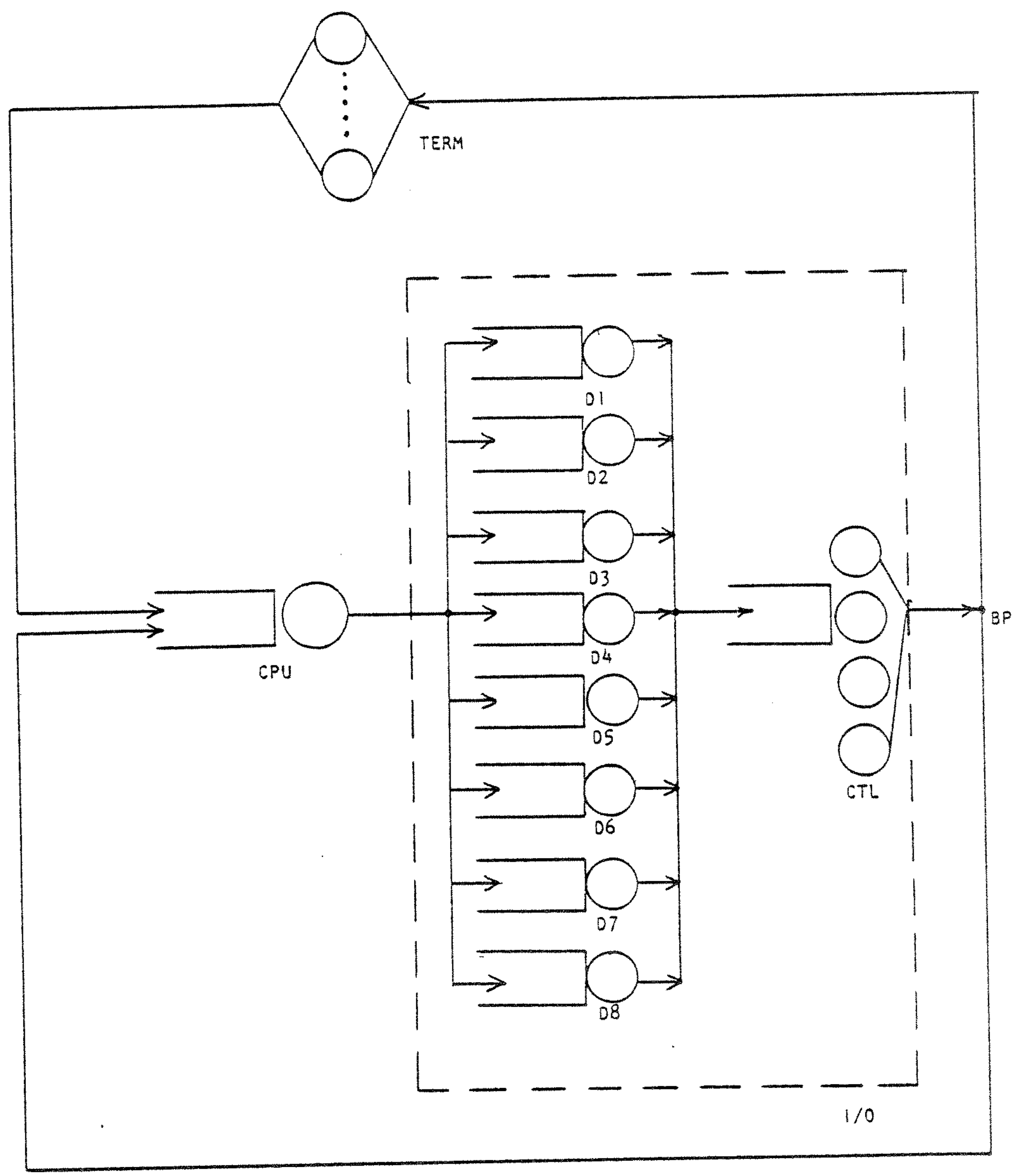

FIGURE A.26. HOST SYSTEM CONFIGURATION REQUIRED FOR IPL

	Number of Users				
	1	2	3	4	5
Response Time (secs)	6.8	7.3	8.0	8.9	9.9
Wait Time (secs)					
Per visit:					
CPU	.0787	.0885	.1004	.1150	.1329
I/O	.0374	.0375	.0376	.0377	.0378
Total:					
CPU	4.6	5.1	5.8	6.7	7.7
I/O	2.2	2.2	2.2	2.2	2.2

FIGURE A.27. REVISED IPL MODEL RESULTS: UPGRADED CONFIGURATION

PERFORMANCE REPORTS FOR SCENARIO * IPL * AS OF 15 JUL 80

UPGRADED HARDWARE CONFIGURATION. 5 USERS:

FREQUENCY OF ESTIMATED ELAPSED TIMES

INTERVAL (SECS)	TOTAL	#FINDS	#RETRIEVES	#OTHER	#USER DEF
0.000- .099	449.	223.	223.	1.	2.
.100- .199	2.	0.	2.	0.	0.
.200- .299	2.	2.	0.	0.	0.
.300- .399	0.	0.	0.	0.	0.
.400- .499	0.	0.	0.	0.	0.
.500- .599	0.	0.	0.	0.	0.
.600- .699	0.	0.	0.	0.	0.
.700- .799	1.	0.	0.	0.	1.
.800- .899	0.	0.	0.	0.	0.
.900- .999	0.	0.	0.	0.	0.
1.000-1.099	0.	0.	0.	0.	0.
1.100-1.199	0.	0.	0.	0.	0.
1.200-1.299	0.	0.	0.	0.	0.
1.300-1.399	0.	0.	0.	0.	0.
1.400-1.499	0.	0.	0.	0.	0.
1.500-*****	0.	0.	0.	0.	0.
TOTALS	454.	225.	225.	1.	3.

FIGURE A.28. ELAPSED TIME FREQUENCIES: UPGRADED CONFIGURATION

PERFORMANCE REPORTS FOR SCENARIO * IPL * AS OF 15 JUL 80

UPGRADED HARDWARE CONFIGURATION, 5 USERS:

STATISTICS BY COMPONENT

	DESCRIPTION	#REQUESTS	ELAPSED SECS.	CPU SECS.	#I/O
COMPONENT QLP	PARSER SPECIFIED	1.	.0068	.0200	0.
COMPONENT SENDMSG	TO IPIP MINIMUM	1.	.0047	.0140	0.
COMPONENT DBCS	RUN TIME SPECIFIED	1.	.7726	.2730	18.
COMPONENT FIND	PARTS REC				
	OVHD	1.	.2579	.0920	6.
	USR-AC-BLK	3.	.0587	.0620	1.
SUB-TOTAL		4.	.4341	.2780	9.
COMPONENT RETRIEVE	PARTS REC				
	OVHD	1.	.1333	.0590	3.
	READS	3.	.0537	.0470	1.
SUB-TOTAL		4.	.2943	.2000	6.
COMPONENT FIND	CONTAINS				
	OVHD	1.	.2579	.0920	6.
	USR-AC-BLK	7.	.0587	.0620	1.
	MINIMUM	213.	.0189	.0560	0.
SUB-TOTAL		221.	4.6959	12.4540	13.
COMPONENT RETRIEVE	CONTAINS				
	OVHD	1.	.1333	.0590	3.
	READS	9.	.0537	.0470	1.
	MINIMUM	211.	.0138	.0410	0.
SUB-TOTAL		221.	3.5369	9.1330	12.
COMPONENT PRES-SERV	TO USER SPECIFIED	1.	.0034	.0100	0.
**********		-------	--------	-------	----
ALL COMPS		454.	9.7486	22.3820	58.
LINKAGE			.1529	.4530	
TOTAL		454.	9.9015	22.8350	58.

FIGURE A.29. COMPONENT STATISTICS: UPGRADED CONFIGURATION

BIBLIOGRAPHY

[ALL79] F.W. Allen, "A Predictive Performance Evaluation Technique for Information Systems," Proc. Fourth International Symposium on Modeling and Performance Evaluation of Computer Systems, Vienna, February 1979.

[BAE72] J.L. Baer, R. Caughey, "Segmentation and Optimization of Programs from Cyclic Structure Analysis," Proc. AFIPS (SJCC) 40, 1972.

[BAK78] J.W. Baker, D. Chester, R.T. Yeh, " Software Development by Stepwise Refinement," Report SDBEG-2, University of Texas at Austin, January 1978.

[BAS75] F. Basket, K.M. Chandy, R.R. Muntz, F. Palacios-Gomez, "Open, Closed and Mixed Networks of Queues with Different Classes of Customers," J. ACM 22,2, April 1975.

[BER58] C. Berge, Theorie des Graphes et ses Applications, Paris: Dunod, 1958.

[BGS79] "POD, A Software Engineering Tool for Life Cycle Management of System Performance - Applications," BGS Systems, Inc., Waltham, MA, April 1979.

[BGS79] "POD, Performance Oriented Design," BGS Systems, Inc., Waltham, MA, December 1979.

[BOE80] "Development of Integrated Programs for Aerospace-vehicle Design," Report D6-IPAD-70036-D, Vol.9, Boeing Commercial Airplane Company, Seattle, WA, March 1980.

[BOO79] T.L. Booth, "Use of Computation Structure Models to Measure Computation Performance," Proc. Conference on Simulation, Measurement and Modeling of Computer Systems, Boulder, August

1979.

[BOX79] O.J. Boxma, A.G. Konheim, "Approximate Analysis of Exponential Queueing Systems with Blocking," IBM Research Report RC7741, Yorktown Heights, June 1979.

[BRO75] R.M. Brown, J.C. Browne, K.M. Chandy, T.W. Keller, D.F. Towsley, C.W. Dissly, "Hierarchical Techniques for the Development of Realistic Models of Complex Computer Systems," Proc. IEEE 63, 1975.

[BRO77] R.M. Brown, J.C. Browne, K.M. Chandy, "Memory Management and Response Time," CACM 20,3, March 1977.

[BUZ76] J.P. Buzen, "Fundamental Operational Laws of Computer System Performance," Acta Informatica 7,2, 1976.

[CHA77] K.M. Chandy, J.H. Howard, D.F. Towsley, "Product Form and Local Balance in Queueing Networks," J. ACM 24,2, April 1977.

[DEN78] P.J. Denning, J.P. Buzen, "The Operational Analysis of Queueing Network Models," Computing Surveys 10,3, September 1978.

[DRA67] A.W. Drake, Fundamentals of Applied Probability Theory, McGraw-Hill, 1967.

[FOR78] H.C. Forsdick, R.E. Schantz, R.H. Thomas, "Operating Systems for Computer Networks," IEEE Computer 11,1, January 1978.

[GRA73] R.M. Graham, G.J. Clancy, D.B. DeVaney, "A Software Design and Evaluation System," CACM 16,2, February 1973.

[GRA78] G.S. Graham (editor), Special Issue: Queueing Network Models of Computer System Performance, ACM Computing Surveys 10,3, September 1978.

[HAR79] M. Harada, T.L. Kunii, "A Design Process Formalization," Department of Information Science Report, University of Tokyo, May 1979.

[HEB78] P.G. Hebalkar, S.N. Zilles, "TELL: A System for Graphically Representing Software Designs," IBM Research Report RI2351, San Jose, September 1978.

[IRA75] User's Manual for the CADS System, Information Research Associates, Austin, 1975.

[IRA79] J.C. Browne, T.W. Keller, C.U. Smith, "Performance Analysis of the Integrated Programs for Aerospace-vehicle Design," Report 1022, Information Research Associates, Austin, TX, October 1979.

[KEL74] J.C. Kelly, "The Theory of Repetition Networks with Application to Computer Programs," Ph.D. Dissertation, Purdue University, December 1974.

[KER71] B.W. Kernighan, "Optimal Sequential Partitioning of Graphs," J.ACM 18, 1971.

[KLE75] L. Kleinrock, Queueing Systems, Volume I; Theory, Wiley Interscience, 1975.

[KLE76] L. Kleinrock, Queueing Systems, Volume II; Computer Applications, Wiley Interscience, 1976.

[KOD78] U.R. Kodres, "Discrete Systems and Flowcharts," IEEE Software Engineering SE-4,6, November 1978.

[KON76] A.G. Konheim, M. Reiser, "A Queueing Model with Finite Waiting Room and Blocking," J. ACM 1976.

[LAM77] S.S. Lam, "Queueing Networks with Population Size Constraints," IBM J. Research Development 21, 1977.

[LAZ78] E.D. Lazowska, K.C. Sevcik, "Approximating Response Time Distributions in Queueing Networks," Proc. IRIA Operating Systems Conference, Paris, October 1978.

[LIT61] J.D.C. Little, "A Proof of the Queueing Formula $L=\lambda w$," Operations Research 9, 1961.

[LOW70] T.C. Lowe, "Automatic Segmentation of Cycle Program Structure Based on Connectivity and Program Timing," CACM 13, 1970.

[MAR67] J. Martin, Design of Real-Time Computer Systems, Prentice Hall, Englewood Cliffs, N.J., 1967.

[MIL77] R.E. Millstein, "The National Software Works: A Distributed Processing System," Massachusetts Computer Associates, Wakefield, MA, August 1977.

[MRI73] System 2000 Reference Manual, MRI Systems Corporation, Austin, TX, 1973.

[NG 78] N. Ng, "A Graphical Editor for Programming Using Structured Charts," IBM Research Report RJ2344, San Jose, September 1978.

[RID78a] W.E. Riddle, J.H. Sayler, A.R. Segal, A.M. Stavely, J.C. Wileden, "A Description Scheme to Aid the Design of Collections of Concurrent Processes," Proc. National Computer Conference, Anaheim, June 1978.

[RID78b] W.E. Riddle, J.C. Wileden, J.H. Sayler, A.R. Segal, A.M. Stavely, "Behavior Modeling during Software Design," IEEE TSE 4, 1978.

[SAN77] J.W. Sanguinetti, "Performance Prediction in an Operating System Design Methodology," Ph.D. Dissertation, RSSM/32, University of Michigan, May 1977.

[SAN78] J.W. Sanguinetti, "A Formal Technique for Analyzing the Performance of Complex Systems," Proc. Computer Performance Evaluation Users Group 14, Boston, October 1978.

[SAN79] J.W. Sanguinetti, "A Technique for Integrating Simulation and System Design," Proc. Conference on Simulation Measurement and Modeling of Computer Systems, Boulder, August 1979.

[SCH78] R.E. Schantz, "A Performance Investigation of the National Software Works System," Bolt, Bernak, Newman Report 3847, July 1978.

[SCH77] R.E. Schantz, R.E. Millstein, "The Foreman: Providing the Program Execution Environment for the National Software Works," BBN Report 3442, January 1977.

[SHA79] M. Shaw, "A Formal System for Specifying and Verifying Program Performance," Report CMU-CS-79-129, Carnegie-Mellon University, June 1979.

[SHO75] H.A. Sholl, T.L. Booth, "Software Performance Modeling Using Computation Structures," IEEE Software Engineering 1,4 December 1975.

[SMI79a] C.U. Smith, J.C. Browne, "Performance Specifications and Analysis of Software Designs," Proc. Conference on Simulation Measurement and Modeling of Computer Systems, Boulder, August 1979.

[SMI79b] C.U. Smith, J.C. Browne, "Modeling Software Systems for Performance Predictions," Proc. Computer Measurement Group X, Dallas, December 1979.

[SMI80] C.U. Smith, J.C. Browne, "Aspects of Software Design Analysis: Concurrency and Blocking," Proc. Performance 80, Toronto, May 1980.

[SPR80] J. Spragins (editor), Special Issue: Analytical Queueing Models, IEEE Computer 13,4, April 1980.

[STA78] A.M. Stavely, "Design Feedback and its Use in Software Design Aid Systems," Proc. Quality and Assurance Workshop, San Diego, November 1978.

[TOW75] D.F. Towsley, "Local Balance Models of Computer Systems," Ph.D. Dissertation, Dept. of Computer Sciences, University of Texas at Austin, December 1975.

[TOW78] D.F. Towsley, K.M. Chandy, J.C. Browne, "Models for Parallel Processing Within Programs: Application to CPU:I/O and I/O:I/O Overlap," CACM 21,10, October 1978.

[VAN71] E.W. Van Hoep, "Automatic Segmentation Based on Baerlean Connectivity," Proc. AFIPS (SJCC) 38, 1971.

[ZAH79a] J. Zahorjan, "An Exact Solution for the General Class of Closed Separable Queueing Networks," Proc. Conference on Simulation Measurement and Modeling of Computer Systems, Boulder, August 1979.

[ZAH79b] J. Zahorjan, "Computational Algorithms for Queueing Networks with Product Form Solutions," Report CSRG-100, University of Toronto, July 1979.