The Performance Evaluation and Prediction SYstem for Queueing NetworkS PEPSY-QNS

Matthias Kirschnick

Department of Computer Science IV

University Erlangen-Nürnberg

Martensstraße 1; D-91058 Erlangen

E-mail: kirschnick@informatik.uni-erlangen.de

Abstract. Performance evaluation often involves the use of queueing network models. PEPSY-QNS has been developed to create these models and analyze them in a comfortable and easy-to-use environment. Its main application is the performance analysis of complexes like transport systems and operating systems but it is also used for educational purposes and the validation of analysis methods.

The article demonstrates the work with PEPSY-QNS and shows what types of queueing networks can be examined, what analysis methods are available and what performance measures are be calculated.

1. Overview

PEPSY-QNS [3] has been developed at the University of Erlangen-Nürnberg over the last six years. Today there are more than 50 different analysis algorithms embedded in an comfortable user environment.

At the beginning of PEPSY-QNS' development, the main goals were to implement the known methods of queueing network analysis and to create and validate new ones. In the last two years the intention has changed from method study towards the use of queueing theory in different areas of engineering.

PEPSY-QNS consists of the basic system - so called PEPSY - and the graphical front end XPEPSY [8] that has been developed since 1990. PEPSY consists of three parts that were designed to fit together homogeneously: the interactive model input, the guided analysis selection, and the analysis program. These parts can be used on almost all UNIX machines, whereas XPEPSY needs a more powerful equipment due to the resource consuming X11-Windows-system.

Shortly after information about PEPSY was made public and the system was presented on the 6. GI/ITG-Fachtagung MMB in Munich in 1991, it was installed at other sites. Some of the interested people did not come from Germany, so the PEPSY was internationalized and now provides German and English messages and texts, now.

Nowadays PEPSY and XPEPSY are installed on more than 20 sites all over the world and further installations will follow.

PEPSY-QNS is used for educational purposes, for the comparison to other tools, for real performance evaluation of operating, transport and communication systems and for the development of new analysis methods.

2. PEPSY

The first step when using PEPSY is the creation of a model description. PEPSY has a mask oriented interactive user interface.

Enter the network specifications, like the number of job classes, the number of nodes, the node types and the arrival and service process descriptions. After the model-description-file is created, you can ask the system what methods are applicable to your model, select one ore more out of them and initiate the analysis. A short time later - in most cases a few seconds - the results are presented in a table.

There are some more utilities available in PEPSY. For example, you can get information about the analysis methods including a short description, a bibliography, implementation details and references to similar or related algorithms.

2.1. Model types

Network types

PEPSY-QNS can handle queueing networks with one or more open or closed job-classes and mixed networks with open and closed job-classes. The routing information can be given either by visit ratios or transition probabilities, but cannot be mixed for different job-classes. The routing information is given for each job-class separately and no change of a jobs class is possible. It is possible on principle to convert every model with job class switching to a network without, to get around this limitation of PEPSY.

Node types

Of course the BCMP-node types (M/M/m-FCFS, M/G/1-Processor-Sharing, M/G/.-Infinite Server, M/G/1-LCFS) are available for the service stations. In addition to this, the more general node-type ./G/m-FCFS and service stations with preemptive or nonpreemptive priority policies can be chosen. Two types of multiple server nodes with the different service stations M/M/m-FCFS-ASYM and M/G/m-FCFS-ASYM are provided as well.
The service process is defined by its first and second moments - the service rate and the squared coefficient of variation.

2.2. Methods

The implemented algorithms are classified and described shortly in the following paragraphs. In addition to these methods, some more techniques are available that do not fit exactly to one of these groups. One example is a method called *bounds* that performs a calculation of upper bounds for the throughput and lower bounds of the average response time.
Most of the implemented methods are described in [2].

Techniques that calculate the performance values using state probabilities

Although analysis methods that are based on the computation of state probabilities (Markov Analysis) need an enormous amount of computing resources, they have a great importance because exact performance results can be calculated. Due to the limited power of the computers, the queueing networks that have to be examined cannot exceed a certain size. PEPSY-QNS disposes the possibility of numerical analysis of queueing network models.

Methods that calculate the performance values using a normalization constant

In this section the well known convolution algorithm [4] and the REcursion by Chain ALgorithm (RECAL) [6] have to be mentioned. The are used to analyze closed product-form networks.

Strategies that base on the Mean Value Analysis

One of the most known analysis methods for closed networks is the Mean Value Analysis (MVA) as described by Reiser in [16]. This algorithm is available in PEPSY-QNS, of course. Some closely related methods, e.g. the algorithm by Bard [1] and Schweitzer [17] or the SCAT (Self Correction Approximation Technique) [13] are included, some improvements and extensions of them, to handle special node-types with priorities, for example.

Analysis by decomposition

The algorithm described by Raymond Marie [11][12] and some more methods that can compute approximate performance values for non BCMP-networks are implemented in PEPSY-QNS. The method from Kühn [10], Whitt [18], Pujolle [14], Gelenbe [7], Chylla [5] are available individually or integrated in an universal decomposition method that uses some heuristics to calculate the performance values for the service stations after the network analysis is done.
The summation method, that has been developed at the University of Erlangen-Nürnberg, is available, too.

Product form approximation techniques

Another major group of queueing network analysis methods are the product form approximation techniques. The diffusion approximation method described by Reiser [15] and Kobayashi [9] and the EPF-method by Shum [17][18]are available.

Simulation

There is a discrete event simulation module integrated that can be used like the other analytical methods, the only difference being time consumption. The simulation will be started in the background so the user can use other analysis methods to examine the model.
The advantage of the integration of the simulation method is the usage of the same model description and the presentation of the performance values provided for the other methods. This leads to an easy comparison or validation of the analytically calculated values and those from the simulation.

2.3. Results

The analysis of a network creates a results file for each applicated method. It is very comfortable for the comparison or validation of the results to have almost the same output format for every method.
The performance values calculated by every algorithm are listed below. In addition to this, some methods append further information, e.g. the simulation method *sim2* can give some measures about the width of confidence intervals for the simulated results. Some iterative working methods add information about the accuracy achieved or the number of iterations made.

Calculated performance measures

The results of all analysis methods contain at least:

- Performance values that are calculated for each node:
  - throughput,
  - visit ratio,
  - average service time,
  - utilization,
  - average response time,
  - average number of jobs,
  - average waiting times, and

- average queueing length.

They are listed:

- for each job class separately, and
- summarily over all job classes.

• Performance values for the entire network calculated for each job class separately:

- throughput,
- average response time,
- average number of jobs,

2.4. Example

The main steps in working with the PEPSY are described in this paragraph. Starting with the model, first create the model file. After this is done select the analysis method and, last but not least, start the performance evaluation.

The queueing network model

Let's take this often used central server model shown in figure 1.

CPU: M/G/1-PS, $\mu = 10s^{-1}$,

$p_{DISK} = 0.7, p_{TERM} = 0.3$

DISK: M/M/1-FCFS, $\mu = 3s^{-1}$

TERM: M/G/∞-IS, $\mu = 0.2s^{-1}$

Number of jobs: 10

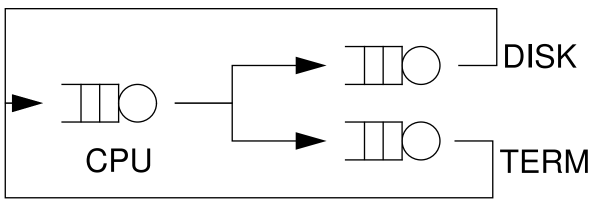

Figure 1: Queueing model of a central server computer system.

Creating the model description

The model input program is started by entering the command "`eingabe`".
The user will be asked to enter the kind and number of job-classes (closed, 1), the number

of nodes (3) and then the type of each node (M/G/1-PS, M/M/1-FCFS, M/G/.-IS) and its service-rate (10.0, 3.0, 0.2).

After that, the routing information has to be entered. In PEPSY-QNS there is one special thing to consider when talking about routing: open job-classes contain a (combined) source/sink-node as usual, and although a special node like this is not absolutely necessary in closed job-classes, PEPSY-QNS uses the “outside”-node as reference node for relative performance values. For example when transition probabilities are given, the visit ratios of the nodes in the network will be calculated by setting the “outside”-node’s visit ratio to 1. Finally the number of jobs in the network (10) is specified the model’s name is entered in this case *first*, and the model description file *e_first* will be generated (figure 2).

```
#
# filename  e_first
#

NUMBER NODES: 3
NUMBER CLASSES: 1

NODE SPECIFICATION

  node   |       name          |     type
---------+---------------------+---------------------
     1   |       CPU           |     M/G/1-PS
     2   |       DISK          |     M/M/1-FCFS
     3   |       TERM          |     M/G/0-IS

CLASS SPECIFICATION

  class    | arrival rate   number of jobs
----------+----------------------------------
     1    |     -             10

CLASS SPECIFIC PARAMETERS

CLASS  1

     node          |  service_rate    squared_coeff._of_variation
-------------------+----------------------------------
     CPU           |  10              1
     DISK          |  3               1
     TERM          |  0.2             1


SWITCHING PROBABILITIES

from/to     | outside    CPU        DISK       TERM
-----------+-------------------------------------------
outside     | 0.000000  1.000000  0.000000  0.000000
CPU         | 0.000000  0.000000  0.700000  0.300000
DISK        | 1.000000  0.000000  0.000000  0.000000
TERM        | 1.000000  0.000000  0.000000  0.000000
```

Figure 2: Model description file *e_first* generated by *eingabe*

Get the applicable methods

The next step is the selection of the analysis method. PEPSY-QNS supports the user by a built-in database with the algorithms' limitations and gives the list of applicable analysis methods on the query "`auswahl first`" (figure 3).

```
first:
    usable                  |  need further specification
---------------------------------------------------------
    ammva                   |
    bol_aky                 |
    bounds                  |
    cmva                    |
                            |  hm
    marie                   |
    mmva                    |
    monosum                 |
    multisum                |
    num_app                 |
    num_single              |
    pm_2                    |
    priomva2c               |
    priomva2m               |
    recal                   |
                            |  sim2
```

Figure 3: The list of applicable methods presented by *auswahl first*

This list contains all methods that can be used to analyze the network *first*.

In addition to the standard network description that has been entered in the beginning, some methods need further specification of parameters. The method *sim2* (the simulation method) shown in the right column needs some information about its maximum duration or the desired accuracy.

If the user isn't familiar with the methods alias names or the shortcuts, the PEPSY-QNS assists with its on-line database.

Performing the analysis

Now select the method's name, e.g. *marie*, and start the performance value calculation by entering the command "`analyse marie first`".
One second later the results are presented as follows: (figure 4)

```
PERFORMANCE_INDICES FOR NET:  first

description of the network is in file 'e_first'

the closed net was analysed with method 'marie'   .


jobclass 1

marie       |  lambda      e    1/mue     rho     mvz     maa     mwz    mwsl
-----------+------------------------------------------------------------------
CPU         |   3.922   1.000   0.100   0.392   0.156   0.613   0.056   0.221
DISK        |   2.746   0.700   0.333   0.915   1.276   3.503   0.943   2.588
TERM        |   1.177   0.300   5.000   0.000   5.000   5.883   0.000   0.000




characteristic indices:

marie       |  lambda     mvz     maa
-----------+--------------------------
            |   3.922   2.549  10.000


legend

e   : average number of visits            mue   : service rate
rho : utilisation                         lambda: mean throughput
mvz : average response time
maa : average number of jobs
mwz : average waiting time
mwsl: average queue-length
```

Figure 4: The performance values of the network *e_first* calculated by the method

More analyses can be done with different methods or with a modified queueing network and comparisons of the calculated results can be made easily. Case studies are easy to realize by editing the model description file using any ASCII-editor; the user does not have to enter the whole model again.

3. XPEPSY

As opposed to the basic PEPSY where several programs have to be used, the graphical front-end XPEPSY hides the different parts of PEPSY-QNS.

In spite of this, it is fully compatible to PEPSY and uses the same model files, analysis methods and result files. This allows the user to utilize both so differently appearing systems at the same time without converting files or having duplicate information.
XPEPSY has some big advantages over the standard PEPSY. First, it is very easy to enter a model. The interactive model input provided by PEPSY is quite simple and after one or two tries the user should be able to enter any queueing network he wants, but the user interface of XPEPSY is more intuitive. You place the service centers, connect them as given by the routing. It looks more like constructing a queueing network on a piece of paper. It is very easy to change or extend the existing model for doing case studies.
When the design of the model is finished, XPEPSY offers a list of the applicable methods. The user chooses simply by clicking the methods name. The integrated On-line-Help-System does not only support the usage of the program itself, but it gives information on basic topics of queueing theory.
XPEPSY was designed for easy use of PEPSY, especially for engineers or students that know about queueing network theory but don't need to know about the usage of the computer's operating system.

3.1. How to work with XPEPSY

To demonstrate the work with XPEPSY, the same queueing model is used as in the paragraph about PEPSY above.

After the program start, XPEPSY presents its main window which looks like shown below in figure 5.

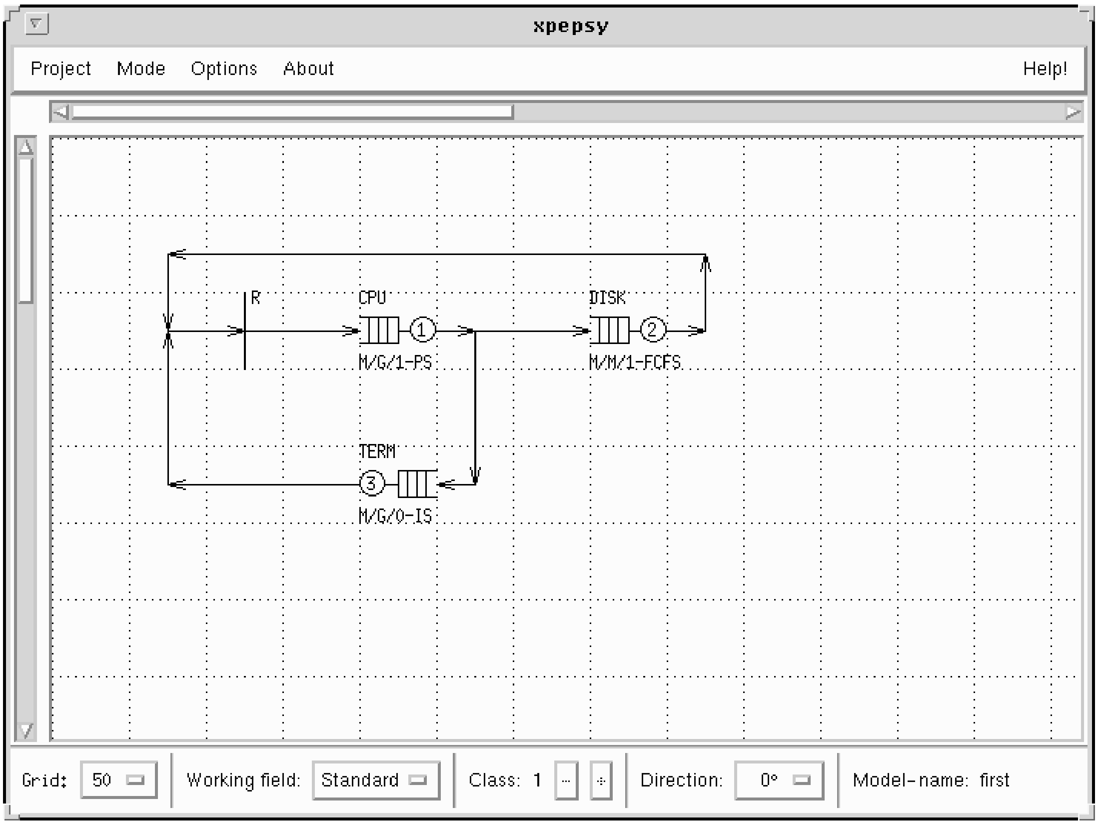

Figure 5: The main XPEPSY window

You can see the working area with the model that was created with operations such as create and place the service center or establish the connections between them. Nodes or connections can be created, moved or deleted by simple mouse-clicks. The necessary parameters are specified in dialogs. See e. g. the node dialog (figure 6) where the parameters for the service centers are entered.

When the queueing network is fully specified, change from the edit-mode to the analysis-mode and a new set of menus will appear.

Now the PEPSY-QNS assists in doing the method selection. There are four possibilities in doing this:

- select from a limited set of methods, i.e. the ones that are especially fast or precise,
- select from all feasible methods,
- show only methods that already have been used and valid results can be found without calling the algorithm, or
- show all methods that can not be used due to algorithm inherent restrictions.

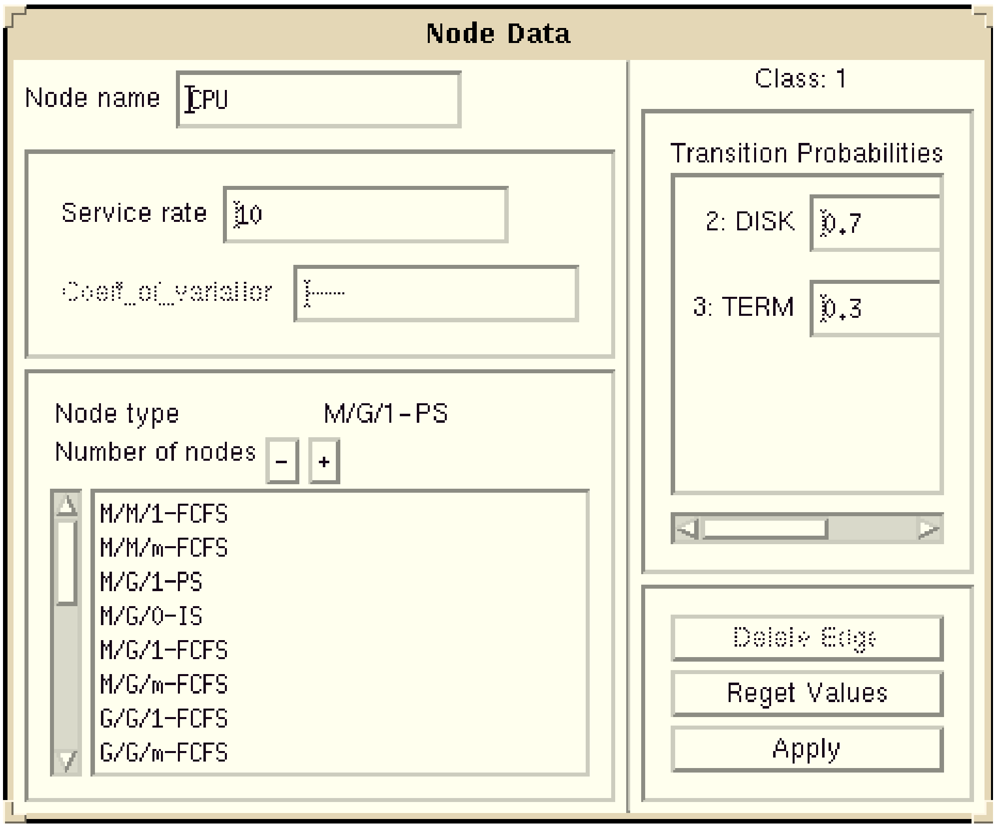

Figure 6: The node-data dialog appears by clicking on the service station on the drawing-area.

Now the analysis can be performed by simply activating the "Analysis"-button from the analysis menu (figure 7) and the results will appear in a new window in the same format as shown in figure 4.

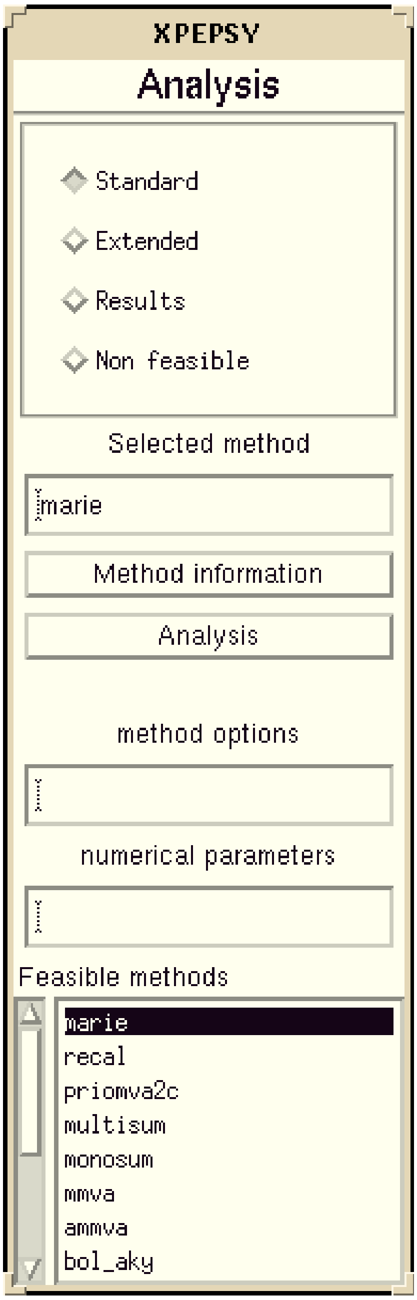

Figure 7: The analysis-menu

Some additional programs from PEPSY are integrated into XPEPSY as well. Using the "Method Information"-button, the PEPSY-QNS internal database can be searched for information about the selected method.

4. Further information

PEPSY contains an optimization component, too. Some examinations for throughput maximization or cost minimization can be done.

Some filter programs have been developed to convert the analysis result files to give a graphical representation using tools like the GNU-plot system or xgraph. A tool specialized on PEPSY result files named XGAD is available as well.

PEPSY-QNS is written in ‘C’ and ported to many different UNIX platforms, e.g. HP-UX, SunOS, Ultrix, AIX, Sorix, NetBSD and some more. It is available on PC compatible computers as well as on workstations or high performance servers. The standard PEPSY works on ASCII-terminals and does not need special equipment or other software. The XPEPSY uses X11-Windows and the Motif widget-set. It can be installed easily on most systems, too.

Further information can be obtained by using the World-Wide-Web (WWW). Look at this URL:http://www4.informatik.uni-erlangen.de/tree/Departments/IMMD-IV/Projects/PEPSY/en/pepsy.html or contact Dr. Gunter Bolch[1] (bolch@informatik.uni-erlangen.de).

5. References

1. Bard, Y.: *Some Extensions to Multiclass Queueing Network Analysis*, 4th Int. Symp. on Modelling and Performance Evaluation of Computer Systems, Vol. 1 Vienna, Feb. 1979.

2. Bolch, G.: *Leistungsbewertung von Rechensystemen mittels analytischer Warteschlangenmodelle*; B.G.Teubner-Verlag, Stuttgart 1989; German.

3. Bolch, G., Kirschnick, M.: *PEPSY-QNS - Performance Evaluation and Prediction SYstem for Queueing NetworkS*; Universität Erlangen-Nürnberg, Tech. Report TR-I4-21-92, Oct. 1992; German.

1. Dr. Gunter Bolch; Universität Erlangen-Nürnberg, IMMD 4, Martensstraße1, D-91058 Erlangen

4. Buzen, J. P.: *Queueing Network Models of Multiprogramming*, Ph.D.Thesis, Div. of Engineering and Applied Physics, Haward University, 1971.

5. Chylla, P.: *Zur Modellierung und approximativen Leistungsanalyse von Vielteilnehmer-Rechensystemen*, Dissertation an der Fakultät für Mathematik und Informatik der TU München. 1986.

6. Conway, A. E., Georganas N. D.: *RECAL-A New Efficient Algorithm for the Exact Analysis of Multiple-Chain Closed Queueing Networks*, Journal of the ACM, Vol. 33, No. 4, pp. 768-791, Oct. 1986.

7. Gelenbe, E.: *Introducing to Queueing Networks*, J. Willey, 1987.

8. Kirschnick, M.: *XPEPSY-Handbuch*; Universität Erlangen-Nürnberg, Tech. Report TR-I4-18-93, Sep. 1993; German.

9. Kobayashi, H.: “Application of the Diffusion Approximation to Queueing Networks, Part I: Equilibrium Queue Distributions”, Journal of the ACM, Vol. 21, No. 2, pp. 316-328, April 1974.

10. Kühn, P. J.: *Approximate Analysis of General Queueing Networks by Dekomposition*, IEEE Transactions on Communications, Vol. Com-27, No. 1, pp. 113-126, Jan. 1979.

11. Marie, R.A.: *An Approximate Analytic Method for General Queueing Networks*, IEEE, TSE, 1979.

12. Marie, R.A.: *Calculating Equilibrium Probabilities for* $\lambda(n)$ */C/1/N Queues*, ACM Sigmetrics Performance Evaluation Review, Vol. 9, No. 2, Summer 1980, pp. 117-125.

13. Neuse, D., Chandy, K.M.: *SCAT: A heuristic Algorithm for Queueing Network Models of Computing Systems*, ACM Sigmetrics Performance Evaluation Review, Vol. 10, No. 3, pp. 59-79, Fall 1981.

14. Pujolle, G., Wu, A.: *A Solution for Multiserver and Multiclass Open Queueing Networks*, INI-OR, Vol. 24, No. 3, 1986, pp. 221-230.

15. Reiser, M., Kobayashi, H.: *Accuracy of the Diffusion Approximation for some Queueing Systems*, IBM Journal of Research and Development, Vol. 18, No. 2, pp. 110-124, March 1974.

16. Reiser, M., Lavenberg, S.S.: *Mean Value Analysis of Closed Multichain Queueing Networks*; Journal of the ACM, Vol. 27, No. 2, Apr. 1980.

17. Shum, A. W.: *Queueing Models for Computer Systems with General Service Time Distributions*; Ph.D. Thesis, Div. of Engineering and Applied Physics, Havard Univ., Dec. 1976.

18. Shum, A. W., Buzen, J. P.: *The EPF Technique: A Method for Obtaining Approximate Solutions to Closed Queueing Networks with General Service Times*; Proc. 3rd Symp. on Measuring, Modelling and Evaluating Computer Systems, North-Holland, pp.201-220, 1977.

19. Schweitzer, P.: *Approximate Analysis of Multiclass Closed Networks of Queues*, Proc. Int. Conf. on Stochastic Control and Optimization, Amsterdamm, 1979.

20. Whitt, W.: *The Queueing Network Analyzer*, The Bell System Technical Journal, Vol. 62, No. 9, Nov. 1983, pp. 2779-2815.