

# Software Tools for Network Modelling

**Attila Kuki<sup>a</sup>, János Sztrik<sup>b</sup>, Günter Bolch<sup>c</sup>**

<sup>a</sup>Department of Information Technology University of Debrecen  
e-mail: kuki@inf.unideb.hu

<sup>b</sup>Department of Information Technology, University of Debrecen  
e-mail: jsztrik@inf.unideb.hu

<sup>c</sup>Department of Operating Systems, University of Erlangen, Germany  
e-mail: bolch@informatik.uni-erlangen.de

## Abstract

A small and very powerful tool, called PEPSY-QNS (Performance Evaluation and Prediction SYstem for Queueing NetworkS), was developed in the 90's at University of Erlangen to obtain the performance measures of queueing systems. It has a very simple and user-friendly interface. So far there has been more than 50 analysing methods built in the system.

PEPSY-QNS consists of the basic system - so called PEPSY - and the graphical XPEPSY. PEPSY consists of three parts that were designed to fit together. This paper demonstrates using of PEPSY on queueing networks and handling different system files eg. input and output data.

An example has been given for calculating the main system characteristics such as throughput, utilization, average response time, average waiting times etc.

WinPEPSY is the new version of PEPSY-QNS. It was developed to run under MS Windows operating systems (PEPSY-QNS runs under Unix operating system). You can use WinPEPSY as a usual Windows application. The model specification is very simple. Graphical tools and dialog boxes can be used as well. The results can be obtained in chart and table form and in addition scenarios can be used, where a value of a parameter can be run between given limits and the results will be computed for all value.

**Key Words and Phrases:** Queueing theory, queueing networks, tools, performance evaluation

# 1. Introduction of queueing theory problems

If we consider a complex computer system we can observe that the programs are not simply run but compete each other for the limited resources, eg. CPU, RAM, peripheries etc. The high level of complexity of these systems causes that to achieve an optimal performance is a very difficult task.

A possible solution is using hardware and software monitors in existing systems, but a significant disadvantage of this method is that building a monitor system is an expensive and complex process and it could not be used at the designing and engineering phases. The other method is the modelling. Making some abstractions the considered system may be so simple that it can be handled and investigated easily.

Large number of problems in computer systems and manufacturing systems can efficiently be modelled by the help of queueing systems or queueing networks (see in. [1], [2], [3]).

The considered queueing networks consist of one or more nodes with a given structure. The jobs to be served move between these nodes.

The nodes consist of a waiting queue and one or more servers. If an arriving job does not find a free server it joins the queue. The members of this queue will be served subject to a pre-defined serving discipline. When a job is served it moves to another node with a given transition probability or leaves the node.

The queueing systems can be described by the help of Kendall's notation:

$A/B/m$ -serving principle,

where  $A$  is the distribution of arrival times of jobs,  $B$  is the distribution of serving times and  $m$  is the number of servers.

If the considered distribution is exponential it is denoted by  $M$  (Markovian), in case of general distribution it is denoted by  $G$ .

Serving principles can be the followings:

- FCFS - First Come First Served,
- LCFS - Last Come First Served,
- PS - Processor Sharing,
- IS - Infinite Server,
- FCFS PRE, (FCFS NONPRE) jobs with higher priority can (can not) interrupt jobs with lower priority,
- FCFS ASYM servers works with different serving times.

The  $M/M/m$  FCFS,  $M/G/1$  PS,  $M/G/\text{inf}$  IS and  $M/G/1$  LCFS PRE nodes are called as product form nodes. For queueing networks which consist of only these types of nodes an exact solution can be given.

The network is called open network if jobs arrive from outside (with a given rate) and after visiting the appropriate nodes leave it. In case of closed networks given number of jobs circulate within the network

We can compute the characteristics of a queueing network, eg.

- overall throughput,
- utilization,
- average queue length,
- average waiting times, etc.

by the help of exact and approximate algorithms.

## 2. The solutions: PEPSY-QNS and WinPEPSY

The PEPSY-QNS (Performance Evaluation and Prediction SYstem for Queueing NetworkS) system was developed at University of Erlangen in the 90's. There are more than 50 analysing methods built in the system.

A PEPSY-QNS consists of two systems. The first one is the PEPSY basic system, which can be used under UNIX OS and it has three integrated modules:

- interactive model input,
- guided choice of analysing methods,
- analysing modul.

The second one is the so called XPEPSY which requires graphical interface. The bases of XPEPSY are the same modules as in program PEPSY.

## 3. The programs of PEPSY-QNS

The main programs of PEPSY-QNS are the followings:

Program '*eingabe*' The structure and characteristics of the network be can defined by the help of '*eingabe*'. You can specify

- Type of network (open, closed or mixed),
- Number of nodes,
- Type of nodes (with Kendall's notations),

- |                        |                      |
|------------------------|----------------------|
| (1) M/M/1-FCFS         | (2) M/M/m-FCFS       |
| (3) M/G/1-PS           | (4) M/G/0-IS         |
| (5) M/G/1-FCFS         | (6) M/G/m-FCFS       |
| (7) G/G/1-FCFS         | (8) G/G/m-FCFS       |
| (9) M/G/1-LCFS-PRE     | (10) M/M/1-FCFS-PRE  |
| (11) M/M/1-FCFS-NONPRE | (12) M/G/m-PS        |
| (13) G/G/m-PS          | (14) M/G/1-FCFS-PRE  |
| (15) M/G/1-FCFS-NONPRE | (16) M/M/m-FCFS-PRE  |
| (17) M/M/m-FCFS-NONPRE | (18) M/G/m-FCFS-PRE  |
| (19) M/G/m-FCFS-NONPRE | (20) M/M/m-FCFS-ASYM |
| (21) M/G/m-FCFS-ASYM   |                      |

- Number of job classes,
- Arrival rate,
- Serving rates,
- Quadratic coefficients,
- Transition probabilities,
- Visiting ratios.

Program *'zusatz'* Some analysing methods need further input information which can be given by using of this program.

Program *'auswahl'* After defining the network, *'auswahl'* lists the applicable analysing methods. The list consists of two parts. There are methods which can be used without specifying any further information and there are ones which require the *'zusatz'* program for some additional data.

Program *'analyse'* This program is the heart of PEPsy-QNS. It solves the problem and generates an output which consist of the required system characteristics.

## 4. Working with WinPEPSY

So far this easy and powerful tool works on Unix based platforms. There arises a natural demand for a version that can be used on Windows PCs as well. This version of WinPEPSY can be installed under operating systems Windows 95/98, Windows Me, Windows NT/2000 and Windows XP.

Though this German language software is localised only to English, the usage is very simple for users who are familiar with PEPsy-QNS and the basic technics of Windows operating system.

In the following section we demonstrate how the known PEPsy tasks look like in WinPEPSY.

*To define a model*

Models can be defined by dialogue boxes or by the help of graphical tools.

The first way is the dialogue boxes. The network parameters can be provided by using the radio buttons and text fields of the appearing dialogue boxes:

- Open network
- Closed network
- Mixed network
- Number of classes

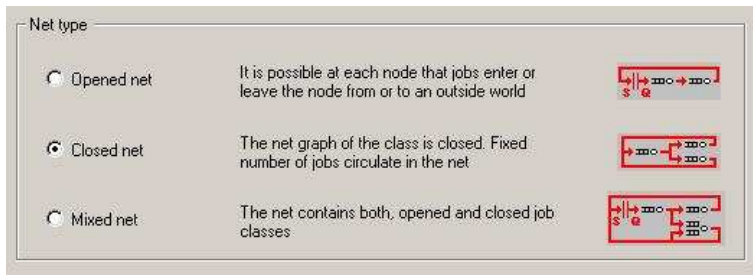


Figure 1: Type of network

- Name of nodes
- Type of nodes (number of servers, serving principles)
- Number of tasks
- Arriving and serving rates
- Coefficient of variation
- Transition probabilities
- Visiting ratio

*To analyze a model*

The appropriate analyzing methods are listed in the bottom right pane of the program window. You can choose one from there.

The values of certain parameters (arriving and serving rates, number of tasks, transition probabilities) can be run between specified values. That way a series of solutions can be obtained which improves the sensitivity and the completeness of the model analysis.

The system characteristics can be obtained in the form of graphics and tables as well.

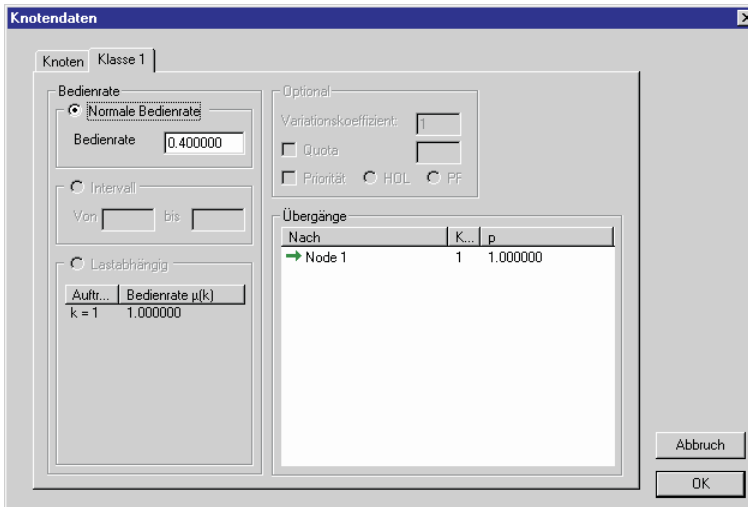


Figure 2: Job classes

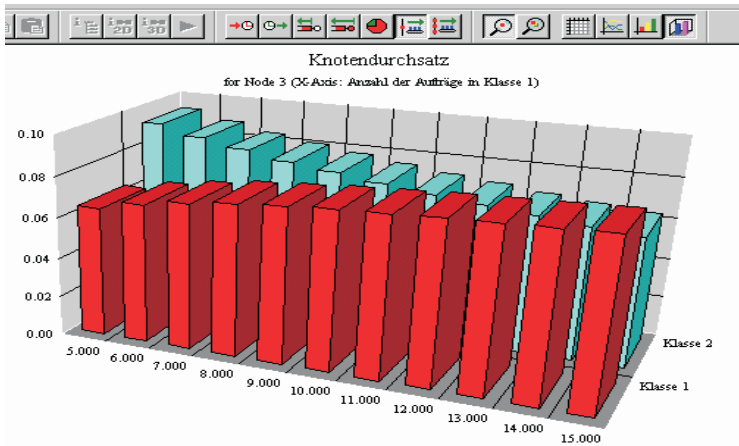


Figure 3: Result of a scenario

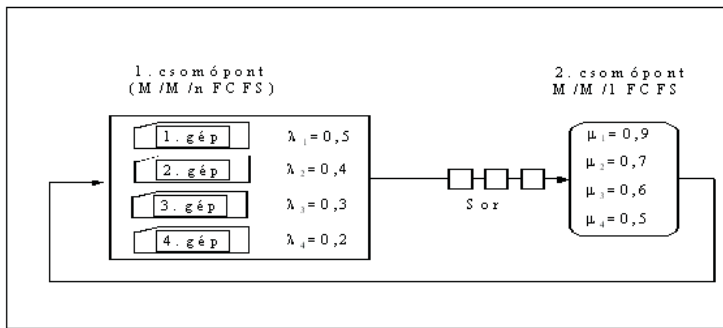


Figure 4: Heterogeneous finite source model

## 5. Analyzing finite source queueing networks

For example, consider how the WinPEPSY can be used to solve the following network: There are four job sources (eg. four computers) with different intensity values and there is one serving station with four servers with different serving rates.

To solve this model you have to specify two nodes with four classes (one job for each class). The type of servers and the arriving and serving rates can be seen on Figure 4. The description of the model in WinPEPSY is the following:

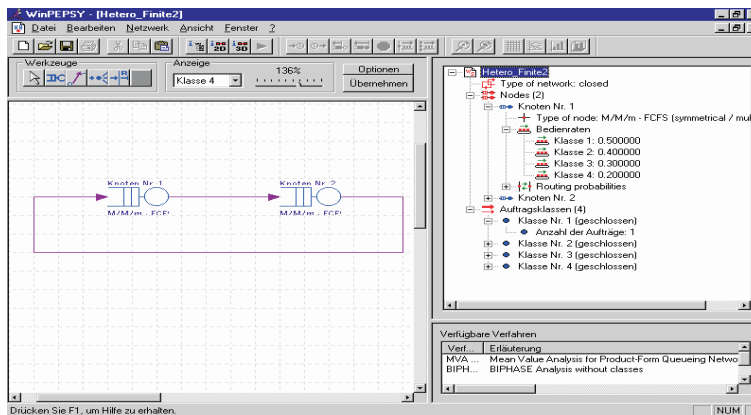


Figure 5: Model description in WinPEPSY

The numerical values of the system characteristics are:

- Utilization: 0,906
- Average number of jobs: 2,195

- Average response times: T1=2,658, T2=3,405, T3=4, 045, T4=4,999

## References

- [1] Bolch G., Greiner S., de Meer H., Trivedi K.S. *Queueing Networks and Markov Chains*, John Wiley & Sons Inc. New York, 1998.
- [2] Haverkort B. *Performance of Computer Communication Systems*, John Wiley & Sons Inc. New York, 1998.
- [3] Nelson R. *Probability, Stochastic Processes and Queueing Theory*, Springer-Verlag, New York, 1995.

## Postal addresses

### **Attila Kuki**

*Dpt. of Information Systems  
and Networks  
University of Debrecen  
Egyetem tér 1., 4010  
Hungary*

### **János Sztrik**

*Dpt. of Information Systems  
and Networks  
University of Debrecen,  
Egyetem tér 1., 4010  
Hungary*

### **Günter Bolch**

*Universität Erlangen-Nürnberg, Informatik 4  
Martensstr. 1, 91058 Erlangen  
Germany*