QUEUEING NETWORKS AS MODELS OF HUMAN PERFORMANCE AND HUMAN-COMPUTER INTERACTION

Yili Liu
Department of Industrial and Operations Engineering
The University of Michigan
Ann Arbor, MI 48109-2117

Technical Report 93-32

November 1993

Preprint
To appear in Proceedings of the '94 Symposium on Human Interaction with Complex Systems

Queueing Networks as Models of Human Performance and Human-Computer Interaction

Yili Liu
Dept. of Industrial and Operations Engineering
University of Michigan
1205 Beal Ave. Ann Arbor, MI 48109-2117, USA

ABSTRACT

This article describes several queueing network models of human performance and human-computer interaction that we have developed recently and illustrates the great value of queueing network methods in establishing models of human performance and human-computer interaction at all analysis levels and in establishing an integrated, computational framework for unifying some currently isolated models. The article starts with a theoretical discussion of the most "micro" level of performance and presents a queueing network model for reaction time as a model of elementary mental processes. As a continuous-flow network model, it includes discrete serial-stage and continuous-flow overlapping-stage models as well as discrete network models as special cases. The second section moves to the more "macro", behavioral level and describes a 3-node queueing network model of multitask performance that includes single-channel, queueing theoretic models and parallel-processing, multiple resources models as special cases. The third section reaches the level of applications and discusses queueing network models of human-computer interaction and human-computer networks and their potential applications in standalone and networked environments. In order to illustrate the modeling capabilities of queueing networks, the first two sections discuss sojourn times and waiting times in an open queueing network, and the last section selects queue length distributions in a closed network as the tool of modeling.

The computational models of human performance and human-computer interaction that we have developed recently and are described in this article are based on our view that human information processing system and human-computer systems are, in many respects, analogous to a queueing network, in which information processing tasks may assume a wide range of complex structural and temporal arrangements. The structural arrangements include both serial selection and parallel execution, and the temporal arrangements include both immediate activities and delayed processing. Queueing network methods employed widely in industrial engineering and systems analysis can serve as a valuable tool in modeling human performance and human-computer interaction.

The idea of a queueing network arises naturally when one thinks of a network of service stations (also called service centers, or simply, nodes), each of which provides a service of some kind to the demanders for service (called customers), either immediately or after a delay. Each center has a waiting space for customers to wait if they cannot immediately receive their requested service, and thus multiple queues may exist simultaneously in the system. The service centers are connected by arcs over which customers flow from node to node in the network. Telephone communications systems, computer networks and road traffic networks are examples of queueing networks [1] [2].

It is not difficult to see the close resemblance between a queueing network and the current views of a human cognitive system. Before embarking on a detailed technical elaboration, the following is a brief illustration of the potential relevance of queueing network concepts to human performance analysis.

In a queueing network representation of a cognitive system, the customers are stimulus components or information processing tasks, which may enter the cognitive system at some node, traverse from node to node in the system, and depart from some node, not all tasks necessarily entering and leaving at the same nodes (e.g., the use of different sensory and motor modalities), or taking the same path once having entered the system (e.g., the use of separate memory and decision routines). Tasks may return to nodes previously visited (e.g., performance feedback or decision loops), skip some nodes entirely (e.g., skill acquisition and automaticity), and even remain in the system for a long time (e.g., memory rehearsal). Some customers may failed to enter a busy system (e.g., perceptual tunneling), leave a busy system before they have been fully serviced (e.g., speed-stress induced errors), jockey for position by switching from one queue to another, or preempt earlier customers if the queue discipline allows this to happen (task scheduling). Multiple queues may improve their joint performance by adopting some

coordinated service schemes (e.g., task integration), or lose effective communication in the face of overwhelming information (e.g., confusion, cross-talk, and outcome conflict) [3] [4].

In order to give mathematical substance to the models presented in the following sections, we introduce the following notations, which are now rather standard in the queueing network literature. Two sets of notations are needed, the first for describing a stochastic queueing process at a service station, and the second for stochastic processes in a queueing network.

A queueing process at a service station in a network is described by a series of symbols and slashes such as A/B/C/D/E, where A indicates the arrival pattern of customers as described by the probability distribution for interarrival-time or arrival rate, B the probability distribution for service time, C the number of parallel service channels at the station, D the restriction on waiting room capacity in front of the station, and E the queue discipline (the manner by which the customers are selected from the queue for service). For the most part, this article will focus on the class of queueing process that has received most research attention and enjoyed a most fruitful history of producing usable analytical results. This queueing process is denoted as M/M/c/∞/FCFS (or M/M/c for short), representing a queueing process with exponential interarrival times (also called Poisson arrival), exponential service times, c identical servers at a station, no restriction on the maximum number of customers allowed in the queue, and first-come, first-served queue discipline. The importance and justifications of employing this type of queueing process in performance modeling are discussed in all standard textbooks on queueing theory [2].

We use the following symbols to represent a queueing network:

1) K: number of nodes,

2) i: identity of nodes,

3) γi: mean arrival rate to node i from outside the network (also called external arrival rate),

4) pij: the probability that a customer visits node j immediately after departing from node i (also called routing probability or switching probability), i=1,..., K, j=0,.., K, with pi0 representing the probability that a customer leaves the network immediately after visiting node i,

5) λi: the total mean arrival rate into node i (from outside and from other nodes), (according to "traffic equation", $\lambda i = \gamma i + \sum pji\lambda j$, summed over j=1 to K)

6) μi: mean service rate for each channel of node i.

For the most part of the paper, we will mainly be concerned with queueing networks with the following characteristics:

1. Arrivals from the "outside" to node i follow a Poisson process with mean rate γi,

2. Service times for each channel at node i are independent and exponentially distributed with parameter μi,

3. The routing probabilities (pij's) are independent of the state of the system, which is a vector representing the number of customers at each station.

Networks that have these properties are called separable networks or product-form networks. They are also called Jackson networks, named after the author who showed that this class of networks have the following amazing property: the network *acts as if* each node can be viewed as an independent M/M/c queue, with parameters λi and μi. The joint probability distribution for the number of customers at each node can be written as a product of marginal M/M/c's [5]. This amazing property makes it possible to derive many important results for the Jackson network that are often not available or analytically intractable for other types of networks. Jackson networks have subsequently received the most research attention and enjoyed a great success in model development. The models have also been successfully applied in diverse areas, because separable networks can be evaluated quite efficiently. Furthermore, many authors have demonstrated that many of the results for Jackson networks provide close approximations to non-Jacksonian networks [6]. In computer system analysis, the pragmatic, "operational" framework for queueing network analysis, pioneered by Buzen and Denning, relies heavily on the assumption of separable queueing networks. It has been pointed out that, in practical applications, inaccuracies resulting from violations of Jackson's assumption typically are not worse than those arising from other error sources (e.g., inadequate measurement data) [7] [8].

With these notations and introductions in hand, we are ready to present a number of queueing network models for human performance and human-computer interaction. The presentation proceeds as follows. We will start with a theoretical treatment of the most "micro" or "molecular" level of performance and present the models for elementary mental processes, and then we move to the more "macro", "aggregated" behavioral level and describe a 3-node model of multitask performance. The last section reaches the level of applications and discusses queueing network models of human-computer interaction and human-computer networks and their potential applications in both standalone and

networked environments. In order to illustrate the modeling capabilities of queueing networks, the first two sections focus the discussion on sojourn times and waiting times in an open queueing network, and the last section selects queue length distributions in a closed network as the tool of modeling.

QUEUEING NETWORKS AS MODELS OF ELEMENTARY PSYCHOLOGICAL PROCESSES

Why is there a delay between stimulus presentation and response initiation? This has been one of the most enduring and fundamental questions that psychologists have been fascinated with. Although it appears that everyone can offer an answer to this seemingly simple question, the exact nature of and the causes for this delay remain a mysterious domain. The current belief of cognitive psychologists is that this delay is a reflection of the dynamic activities of an underlying mental architecture that transforms stimulus into response. And most importantly, since the cognitive system is not amenable to open inspection, the characteristics of this delay--also called reaction time (RT)--may offer important clues to the possible configurations of the mental architecture.

Theoretical models that use reaction time as the primary performance measure to infer the general structure of mental systems are also called models for RT. Of great interest to the present discussion are two dimensions along which RT models can be classified. One of the two is a dynamics dimension distinguishing discrete-processing from continuous-flow models, and the other an architectural dimension distinguishing serial-stage models from network models. Discrete processing models assume that a mental process will not transmit its processing output to other processes until it is completed and it transmits its output in an indivisible unit. Thus a process can not begin until all of its preceding processes are completed. Continuous-flow models assume that each process transmits its available partial outputs to other processes continuously rather than waiting for the full completion of processing, and thus a process can begin processing even though its preceding processes are still active. Serial-stage models assume a serial arrangement, whereas network models assume a network configuration of the mental processes. The two dimensions jointly define four classes of models as shown in Figure 1.

In the top-left quadrant that defines serial discrete processing models, we find the most traditional interpretation of information processing. Donders' subtractive method (1868) and Sternberg's additive factors method (1969) both assume non-overlapping durations of serially arranged processes [9] [10]. This class of models are also referred to as serial discrete-stage models. Donders assumed that processes can be added or deleted from the processing chain while leaving intact the rest of the chain (called the assumption of pure insertion). Sternberg assumed that the duration of a process can be changed by experimental manipulations while this change will not produce indirect effects on the rest of the chain (called the assumption of selective influence). Based on the assumption of selective influence, Sternberg proposed an additive factors methodology for RT analysis, according to which experimental factors that influence a common process will interact with each other, whereas those influencing separate processes will be additive. The serial discrete-stage model and the additive factors methodology have been the fundamental basis of a large body of experimental literature. McGill and Gibbon (1965) noted that reaction time in a serial discrete-stage model can be described by the general-gamma distribution, if the durations of each stage is exponentially distributed with different duration means [11].

	Series	Network
Discrete	1. Subtractive 2. Additive Factors 3. General-Gamma	1. PERT (Critical-Path)
Continuous-Flow	1. Cascade 2. Queue-Series	1.Queueing Network

Fig. 1. Reaction Time Models

Some theorists have studied sequentially arranged processes that permit temporal overlapping of process activities. Prominent among the RT models include McClelland's (1979) Cascade model [12], and more recently, Miller's (1993) queue-series model [13]. Both models try to mimic the behavior of serial discrete-stage models, and examine the conditions under which the two classes of models converge or diverge in their predictions. This class of models belong to the bottom-left quadrant of Figure 1.

Townsend (1972) challenged the notion of

serial arrangement of mental processes and examined possibility of parallel activities [14] Schweikert (1978) later developed a class of so-called PERT networks [15], which assume that mental processes can be arranged as a network, with serial and parallel structures as special cases. Although the PERT models allow processes that are not on the same path of a network to be active at the same time, they assume that processes on the same path operate in strict sequence--a process can not start until all the preceding processes on the same path are completed. Furthermore, the PERT method for RT analysis follows the postulate of selective influence and assumes that each experimental manipulation prolongs the duration of one process, but does not change the duration of any other process. This class of models can be classified as discrete-network models and is shown in the top-right quadrant of Figure 1.

In the following, we present a queuing network model for elementary mental processes. The model, in its most general form, is a continuous-flow-network model. As will be shown below, the model takes the existing models in the other three quadrants of Figure 1 as special cases. Furthermore, this model allows consideration of a broader range of possible mental structures that can be subjected to empirical testing. The purpose is to expand the set of modeling tools available to psychologists and contribute to the psychological endeavor of discovering new mental architectures as possible models of cognition.

General description of the model.

General assumptions. The model assumes that a reaction time task is carried out by a network of processing centers, each of which provides a distinct type of information processing service to the customers (stimulus or task components). Analogous to other continuous-flow models, the model assumes that each center can begin processing as soon as it receives some customer (1 or more stimulus component) from outside the system or from another center. A center immediately transmits any available output (a satisfied customer) to other centers or to the outside of the network without waiting for the full completion of its processing of all its customers. Similar to the cascade model and the queue-series model [12][13], we assume that there is a separate response unit at the end of the processing network, which is activated when it has accumulated N signal components.

Stimulus components as customers. I will adopt the term "stimulus components" used in Miller (1993) to refer to these customers or demanders [13]. The model assumes that a stimulus is composed of a number, C, of distinct classes of components, with Ni components of class i, i=1,..., C. In the simplest case, there is only one class of stimulus components that is responsible to RT (this is the case considered by Miller). We may call them signal components. In a more general case, there may be two classes of stimulus components-- signal components and noise components. It will be shown below that this distinction is critical for queueing network analysis of some RT behavior such as speed-accuracy tradeoff (SAT) and violations of the selective influence assumption. It is easy to image situations in which a finer distinction between the classes of stimulus components is necessary, but this paper will not extend the discussion further to include those cases. As nicely summarized in Miller (1993), "the stimulus components may be regarded as elementary stimulus features, complex semantic codes, objects, or the associated neural activations" ([13], p.703), and as in Miller, this article will not attempt to develop the empirical means of identifying stimulus components.

Component arrivals and services. As pointed out by Pachella, the definition of stimulus onset is not always psychologically obvious [10]. Consistent with the common assumptions adopted in queueing literature, the model assumes that the arrival sequence of stimulus components can be described as a Poisson process. The model assumes that at the node i, customers have an exponentially distributed service time requirement with a mean of $1/\mu i$. μi is often referred to as the service rate of node i. As discussed by numerous authors, this assumption is not as strong as it appears to be [2], and is also consistent with one of the most common assumptions of other RT models [11] [14] [16].

Reaction time as network sojourn time. From the perspective of RT analysis, the most interesting performance measure of a queueing network is customer sojourn time--the time a customer spends in the network (or part of it). Several decades of queueing network research has shown that determining the sojourn time distribution of a customer in queueing networks is a very complicated problem and among the hardest in queueing network theory. For non product-form networks, almost no explicit results exist. Even for product-form networks, complicated correlations among waiting times at various nodes exist and thus very little can be said about the sojourn times of a customer at successive nodes. An important exception to this statement is provided by the sojourn time distribution of a customer along a path in a product-form network when the path is "overtake-free", which means that customers can not overtake or bypass one another. When this overtake-free condition is satisfied, the sojourn times of a customer at various nodes along

the path are independent [1] [6] [17]. However, it should be emphasized here that this independence does not mean that the sojourn times of successive customers are independent. As pointed out by Disney and Konig (1985), the complete characterization of the joint sojourn times of a sequence of customers is still an unsolved problem ([1], p.377).

In this article, we only consider the types of networks in which signal components cannot overtake each other, although noise components may overtake signal components (discussed below). For the models discussed in this article, we assume that each center has a single processing channel and a FCFS discipline. The assumption of single channel processing nodes is similar to that of Miller's queue-series model and is common in psychological theory.

For this type of networks, the Nth signal component to depart from the network is also the Nth signal component to arrive from the outside. Thus, RT is the sum of the Nth customer's network sojourn time (T) and the time interval between the first and the Nth arrival (Ta). For Poisson arrivals, Ta follows the Erlang distribution and is independent of T, which means that Ta is neither influenced by nor offering any insight into the structure of the network. Thus we only need to analyze the network sojourn time of the Nth signal component (T). For Poisson arrivals, signal components arrive at the network independently with each other, and thus the Nth signal component (called a tagged component) could be any of the signal components with equal probability, and is stochastically indistinguishable from any other signal components.

Let us use road traffic as an analogy. In order to study how changes in the traffic environment (e.g., road structure) influences the traveller behaviror--in this case the time to reach a destination, we can either study the travel time of a large number of customers at once, or study a randomly selected "tagged" traveller a large number of times (who is either the same traveler or preferably randomly selected each time). We adopt the "tagged" customer approach, because research results are only available for this case. Actually, adopting this approach allows us to model queuing networks that are not single-channel FCFS-based. But we will not extend the present discussion into those cases.

With these important results in mind, we can proceed to analyze some interesting cases and examine the previous models. We will show that the discrete or continuous-flow serial models are special cases of a special type of queueing network called tandem queues, corresponding to the situation in which each stimulus activates one or a large number (possibly infinite) of stimulus components respectively. PERT networks are shown to be a special case of another type of queueing network called acyclic fork-join networks when each stimulus activates exactly one stimulus component.

Tandem Queues

Network sojourn time in a tandem queue. A special type of queueing network (also the simplest type) is a tandem queue, also called a series queue, in which the service stations form a series system with flows always in a single direction from the first node to the last node. Customers may enter from the outside only at node 1 and depart only from the last node. If each stimulus activates only 1 component, then the successive nodes that component has to visit will operate in strict sequence. In this case, we have a serial discrete-stage processing system. If it activates multiple components, successive nodes along the customer's route will operate with temporal overlap.

More formally, we have a open network where

$\gamma i = \lambda$ (i=1)
$= 0$ (elsewhere)

and

$pij = 1$ $(j=i+1;\ 1 \le i \le k-1)$
$= 1$ (i=k, j=0)
$= 0$ (elsewhere)

It has been shown that the network sojourn time, T, for a M/M/1 tandem queue has as distribution the convolution [6]

$$\Pr\{T<t\} = (1- e^{-(\mu_1-\lambda)t}) * \ldots * (1- e^{-(\mu_K-\lambda)t}),\ t>=0$$

which has been shown to be the general gamma distribution:

$$Fk(t) = 1 - \Sigma C_{ik}\, e^{-(\mu i-\lambda)t} \quad (1)$$

where

$$Cik = \prod(\mu j-\lambda)/(\mu j-\lambda - (\mu i-\lambda)) \quad (2)$$

Now let's compare this result with those of the serial discrete-stage and other continuous-flow models.

Serial discrete-stage model. McGill and Gibbon (1965) have shown that the general gamma is the RT distribution of a serial discrete stage model in which the duration of each stage is exponentially distributed [11]. More specifically, McGill and Gibbon showed that in a serial discrete-stage model with k stages, the RT distribution has the following form:

$$Fk(t) = 1 - \Sigma Cik\, e^{-(\mu i)t} \quad (3)$$

where

$1/\mu i$ is the mean duration of the exponentially distributed passage time through stage i.

Comparing (1) and (3), it is clear that the general-gamma distribution of RT is not limited to serial discrete-stage models. Actually, the serial discrete stage model of McGill and Gibbon can be treated as a special case of the tandem queuing model by replacing (μi-λ) with μi ((μi-λ) is often called the "effective service rate" of a node). The major conceptual difference is that the serial discrete stage model has the largest possible grain size of transmission (a stimulus is indivisible). Only one stimulus component is allowed in the tandem network, and no other components are allowed to enter the network until the current one has completed processing (a situation in which λ=0 and thus μi-λ = μi). In the tandem queueing model, a stimulus is regarded as consisting of a number of components, and they pass through the network like a traffic flow ($\lambda > 0$). Furthermore, the tandem queueing model allows the existence of noise components in the network. Their main effect on RT is a reduction of effective service rate and the corresponding increase in RT. In following analysis of the cascade model, we will show that the introduction of noise components facilitates modeling the well-known phenomenon of speed-accuracy tradeoff.

McClelland's Cascade model. McClelland proposed a cascade model as a continuous-flow serial-processing model [12]. The model assumes that the human information processing system functions like a series of parallel linear integrators. These linear integrators take a weighted sum of a subset of the outputs of the integrators at the preceding level and produces continuous output that is always available for processing at the next level. The central assumption of the cascade model is that the rate of activation of a linear integrator depends on the difference between the level its output are driving it to and the level of activation the unit has already reached. A cascade equation--the heart of the cascade model--was derived based on this assumption, which gives an expression for the activation of linear integrator j at processing level n to a stimulus S presented at time t = 0. The equation has the following form:

$$a_{nj/S}(t) = a_{nj/S}(1 - \sum Ki\, e^{-(ki)t}) \qquad (4)$$

where $a_{nj/S}$ is the asymptotic activation of the linear integrator that would result if the stimulus were left on indefinitely, and the kis are the rate constants of the different processes in the system. McClelland examined the effects of manipulating rate constants and asymptotic levels on RT and derived a set of prediction of RT behavior. Similar to the serial discrete-stage models, the cascade model shows that experimental factors affecting the rate of the same process will interact, whereas those affecting the rates of different processes are additive. However, the predictions become more complicated when at least one of the experimental factors affect the asymptotic level of activation. A particularly interesting result is that the cascade model is able to fit the shape of the well-known time-accuracy curve closely.

The readers may have already noticed the general-gamma function in equation 4. In fact, a more general form of the tandem queueing model makes the same set of predictions as the cascade model. Instead of allowing the Nth signal component to activate the response unit unconditionally when it leaves the last node (always possessing enough activation strength), the more general form of the tandem queueing model assumes that the response-activation strength of the Nth signal component can be manipulated by experimental factors. Analogous to the cascade model, we assume that in yes/no experiments, the response-activation strength of the Nth signal component is ay/y, and the response-activation strength of other stimulus components is ay/n. As in the cascade model, we assume that actual response execution is a discrete event that adds the duration of a single discrete stage (e.g., 0.1 sec.) to the time between the stimulus presentation and the registration of the overt response. Then for the tandem queueing model, the observed value of d' at time t is given by

$$d'(t) = (ay/y - ay/n)\, \Gamma_n[t - .1]/\{1+\sigma^2(\Gamma_n[t - .1])^2\}^{1/2} \qquad (5)$$

This equation is identical to equation (13) of [12], which has been shown to fit the time-accuracy curve closely.

The derivation of equation should be the same as that for equation (13) of [12]. The major difference between the two continuous-flow models is in the interpretation of the general-gamma function. In the cascade model, the general gamma function, $\Gamma_n(t)$, is an activation function and represents the relative activation of a unit at Level n at time t. In the tandem queueing model, the same function represents the probability that a being-observed stimulus component has passed through the last node of the network adn thus reached the response unit at time t.

Miller's queue series model. Miller's model considered a special type of tandem queue [13]. In Miller's model, stimulus components arrive at the queue series at the same time (called bulk arrival in queueing literature). Components are not served in the same order as they arrive at the various servers (not FCFS). Therefore, the nth customer to depart from the end of the queue series is not likely to be the nth to arrive at the front of series. Miller used PERT representation and numerical simulation to examine the time for N customers to traverse through the system and concluded that, within the class of queue

series models he considered, experimental factors affecting different processing stages always have additive effects on reaction time with sequential stages but rarely do so with overlapping stages, and thus, observations of factor additivity support discrete-stage models. As Miller stated: "From the ubiquity of additive factor effects in RT experiments, it appears that nondiscrete queue-series models must be regarded as fairly implausible general descriptions of human information processing". ([13], p.712).

As mentioned above, virtually no analytical result is available in the queueing literature about sojourn times in the type of tandem queue considered in Miller's model. It appears that Miller's conclusion should only be restricted to the type of nondiscrete model he considered, since the nondiscrete M/M/1 tandem queue we considered and discussed above mimics McGill and Gibbon's discrete serial model precisely. There are at least two related issues that are worthy of further exploration. First, in Miller's simulation, the time for the ith component to pass through stage j (tij's) were either constants or independently randomly selected from each of three distributional families: normal, uniform, or exponential-plus-constant. However, as mentioned above, the sojourn times of a customer at successive non-FCFS stages are dependent random variables, rather than constants or independent random variables. Second, the queue-series model follows the postulate of selective influence--an essential assumption for discrete models. "In the queue-series model, a factor affecting stage i changes only the values of tij" ([13] p.711.). There has been a substantial amount of debate about how reasonable this assumption is for discrete models. For a continuous-flow model, this assumption should be even more debatable.

Miller has pointed out that although the queue-series model is able to approximate the shape of the activation functions of the cascade model, the two models produce different effects on RT. The explanation was that the cascade model allows experimental factors to have downstream effects, whereas the queueing series model does not consider such propagation. This could also explain why the predictions of the cascade model and the tandem queueing models converge, while both diverge from the queue-series model. Both models allow the effects of experimental manipulations to propagate through the system, and neither model assumes selective influence. Without a further clarification of these issues, the following statement should be regarded as tentative and debatable: "In view of the prevalence of additivity, then, the most plausible conclusion is that nondiscrete queue-series models are generally inappropriate" ([13], p.713).

Acyclic Fork-Join networks

This is a special type of queueing network, in which a "fork" node simultaneously creates several new customers, which are sent to separate queues, and the corresponding join occurs at a "join" node when the services of all these new customers are completed. Apparently, just as the serial discrete-stage model can be treated as a special case of the tandem queueing model, a PERT network can be treated as a special case of an acyclic fork-join network, in which each stimulus activates only one stimulus component (N=1). No other stimulus components are allowed to enter the network until the offsprings of the newly admitted stimulus component have completed processing.

Stochastic PERT networks and fork-join queueing networks are both extremely difficult to analyze. Schweikert considered deterministic PERT networks [14]. Fisher and Goldstein considered stochastic PERT networks using a method they proposed called order-of-processing (OP) diagram [16]. Fork-join networks is a new research area in the queueing network research community [18]. Future breakthroughs in the research on stochastic PERT networks or fork-join queueing networks will hopefully improve the applicability of these methods for RT analysis.

Simon-Foley Queueing Network

We have mentioned that if a Jackson network does not allow customers overtake each other (e.g., if single-channel Jackson network has atmost one path from node i to node j for every i, j) then sojourn times at nodes are mutually independent and exponentially distributed random variables and passage time along a path can be described as a general gamma distribution, which has been shown to play a central role in the McGill and Gibbon's model, the cascade model, and the tandem queue model. We now consider what happens if a network allows customers to overtake or bypass each other.

A classic example of a non-overtake-free network is the so-called Simon-Foley network (Figure 2). Simon and Foley (1979) considered a three-node Jackson network with single servers at each node [17]. Customers only enter the system at node one and exit the system at node three. After visiting node 1 a customer goes directly to node 3 with probability (1-p), or goes to node 2 with probability p. If the customer goes to node 2, he goes directly to node 3 after receiving service at node 2. Simon and Foley showed that the sojourn time in the first and the third queue (T1 and T3) are not

independent for those customers who go through the second queue. T1 and T2 are independent, T2 and T3 are independent, T1 and T3 are independent for those customers who go directly from node 1 to node 3. Foley and Kiessler later showed that T3 is stochastically increasing in T1 for a customer that goes through node 2, i.e., $P\{T3 > t|T1\}$ is increasing in T1. A result that is particularly useful for mean sojourn time analysis is derived by Walrand and Varaiya [19], who showed that

$$E\{T3|T1=t'\} > E\{T3|T1=t\}, \quad t'>t>0 \qquad (6)$$

where $E\{T\}$ represents the mean of T.

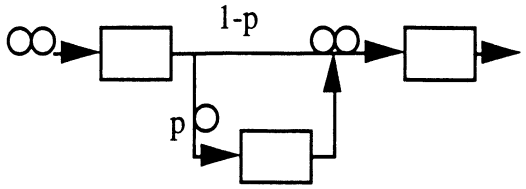

Fig.2. A Simon-Foley Network (The network does not follow Sternberg's assumption of selective influence)

These results are important in that they may suggest a new class of mental architectures that can be subjected to empirical tests. The test for the existence of a Simon-Foley arrangement of the psychological processes makes no more assumptions than those for testing the validity of the Schweikert's PERT methodology for RT analysis. It assumes that we are able to prolong the duration of a process of interest, and we are able to record time at several points in the network [15]. According to Equation 6, if in a task situation in which prolonging a process produces a corresponding increase in the duration of another process, then there is a great possibility that the task situation involves a Simon-Foley network of mental processes, particularly if such a network also " 'makes sense' in terms of other knowledge" [9]. For example, it is possible that in certain task situations node 1 has the function of distinguishing signal from noise components. After passing through node 1, signal components must go through node 2 for a high level cognitive analysis, while noise components go directly from node 1 to node 3. Experimental manipulations that change T1 or p will produce corresponding changes in T3 that are positively correlated with changes in T1.

We are in the process of reviewing published data in the literature to search for possible evidence of this mental network. A series of experiments are also being prepared to test this method. Results along this line of investigation should have significant theoretical implications. Although previous studies have discussed the possible existence of indirect influence of experimental factors [9] [10], Simon-Foley network offers a possible method of testing and quantifying one possible type of indirect influence. Furthermore, since Simon-Foley network is among the simplest continuous-flow network that is neither strictly serial nor parallel, data collected will provide critical insights into the architecture and function of human cognitive system.

In this section, queueing network methods are applied to the analysis of reaction time and elementary mental processes. Customers in a network are indistinguishable components of the same stimulus or of the same task. The models and the methods can be extended to situations in which customers in a network are components of separate stimuli or parts of separate tasks. The next section partly illustrate this point through the models of multitask performance.

QUEUEING NETWORKS AS MODELS OF MULTITASK PERFORMANCE

This section applies the queueing network methods to the analysis of human multitask performance. We are concerned with psychological behavior at a more macro and aggregated level than in the previous section. The modeling work has a strong motive for application, and the approach is engineering and approximative. A 3-node queueing network model is described in this section that integrates the concerns of single channel, queueing theoretic models of selective attention and parallel processing, multiple resources model of divided attention. The two schools of models have fundamental differences in their views of the nature of multitask performance and in their research and modeling methodology. The single channel, serial processing models treat multitask performance as an issue of task selection and scheduling. The multiple resources, parallel processing models, in contrast, treat multitask performance as an issue of parallel allocation and division of processing resources among simultaneous tasks.

From the perspective of computational modeling, the single channel assumptions have thus far enjoyed a greater success, as indicated by the existence of a set of well-established models such as the queueing theoretic models reviewed in the following section. These models provide formal mechanisms for representing and codifying the single channel assumptions of task selection in computational terms. The multiple resources models, in contrast, have only recently started to see some of their concerns being gradually accommodated in several simulation models

of human performance, and there is still a lack of a set of computational methods to transform the assumptions of simultaneous execution and resource allocation into engineering terms. Furthermore, there does not exist a set of integrated engineering-based methods to model the concerns of both schools of models and to bridge the gap between the two. As indicated in a recent report of the Committee on Human Factors of the National Research Council report, "there is no unique method to model the two most important features of macromodel: task selection and simultaneous execution ([20], p.40)." After a brief review of the queueing theoretic and the multiple resources models, we will illustrate that queueing networks may provide a useful method for modeling the two features.

Queueing theoretic models of selective attention

These models postulates that the human functions like a time-shared computer with a single central processing unit (CPU), which quickly switches and allocates its processing capacity among a variety of tasks in a sequential and all-or-none fashion. The models view human multitask performance as a single server queueing problem or multitask sequencing problem in which multiple tasks or diverse sources of information are queued for service from the human information processing system [21].

A number of queueing theoretic models have been developed, focusing on human visual sampling and monitoring behavior. Senders (1966) developed a instrument monitoring model, which integrated the single channel concept and the sampling theorem of Shannon's information theory in making its predictions about the observer's fractional dwell time on each monitored instrument [22]. Carbonell (1966) proposed a single server priority queueing model of multi-instrument visual sampling and used simulation to solve the model [23]. Senders and Posner (1976) further developed the queuing theoretic approach to instrument monitoring and provided analytical solutions to a model that they developed for display sampling [24]. Schmidt (1978) applied queueing theoretic method to the analysis of air traffic control task [25].

An extensive effort in applying the queuing theoretic methods to the modeling of human machine systems can be found in a series of studies conducted by Rouse and his colleagues. Rouse (1977) described human-computer interaction as a queueing system with the human and the computer as two servers [26]. He formulated a queueing theoretic model of dynamic allocation of responsibility between the human and the computer in multitask situations, and illustrated the potential utility of this model with simulation experiments. Chu and Rouse (1979) later investigated the predictive power of the model with a behavioral experiment that simulated a multitask flight management situation [27]. A similar task scenario was also used in an earlier study by Walden and Rouse (1978) that investigated the suitability of a single server queueing model of pilot decision making [28]. Greenstein and Rouse (1982) integrated a pattern recognition technique called discriminant analysis with queueing theory methods in their 2-stage model of human decision making in multi-process monitoring situations [29].

Although the single channel based models have demonstrated tremendous success in modeling visual sampling and task scheduling, "We must recognize that people in fact can do more than one thing at a time and normally do (Adams, Tenney and Pew, 1991; [30], p.5)". In multitask situations, single channel assumptions and the analogy of a single-CPU time-shared computer or a single server queueing system may only capture part of the nature of human performance and may not be adequate to portray the complex cognitive mechanisms for concurrent processing. It may be necessary to address the parallel processing aspect of performance, to consider the intensity as well as the time characteristics of task demand, and to analyze the structural similarity of concurrent tasks. These issues have been the focus of investigation of models of divided attention.

Multiple Resources Models of Divided Attention

In contrast to the single channel assumption that attention capacity can only be switched in a sequential and all-or-none fashion among competing tasks, divided attention theorists have generally suggested that the limited human information processing capacity can be simultaneously allocated to multiple tasks in a graded fashion, and that information for simultaneous tasks can be processed in parallel as long as the total processing load does not exceed a person's processing capacity. Since the 1970s, the concept of capacity has evolved and become more commonly known as a "resource". This concept has also evolved from referring to a single undifferentiated pool to models that human information processing should include multiple pools of resources (Wickens, 1984; [31]). The models suggest that task interference should only be manifest to the extent that they compete for the same pool of resource. Of the various definitions of processing resources that have been proposed so far, the one that has received the most consensus in the literature is the definition based on a distinction between spatial and verbal processing codes. The dichotomy of spatial and verbal processing codes distinguishes operations of perception, working memory and response that have a linguistic and symbolic base from those that have a spatial analog base. Ample evidence has demonstrated the role of processing codes in accounting for variances in task interference. Recently, several simulation models of

human performance, e.g. the MICROSAINT model [32] and the WINDEX model [33], have started to accommodate some of the assumptions of the multiple resources models.

Since the multiple resources models were originally proposed to address the characteristics of parallel allocation of scarce yet divisible processing resources to concurrent activities, the models do not provide a formal mechanism to model the serial processing bottlenecks and the selective and scheduling aspects of task performance. Thus, a typical research strategy of investigators in this research paradigm has been to treat these bottlenecks of serial processing as extraneous factors or to keep the influence of these factors as small or constant as possible. However, as reviewed in the previous section, processing bottlenecks do exist and play an important role in many task situations. The next section will argue that the existence of processing bottlenecks might be a major cause of interference between tasks that do not use the same processing code.

A 3-Node Queueing Network Model of Multitask Performance

A variety of experimental studies have demonstrated that task interference is expected to be greater when concurrent tasks use the same processing code than when they require separate codes. However, the data do not indicate that two tasks demanding separate codes will always be perfectly time-shared [31]. The data seem to suggest a pattern of task interference as follows [3]. First, task interference will not be observed when the total demand of the concurrent tasks is low, regardless of the processing codes involved. Second, when the total task demand is sufficiently high, both within-code and between-codes interferences could be observed, but within-code interference is more likely to occur. Third, increases in task difficulty would produce a faster increase in within-code than between-codes interference.

It appears that this pattern of task interference is also consistent with intuition and experiences. For example, in a perfect driving environment an easy secondary task can be performed concurrently with the primary driving task without causing any performance decrement, no matter whether the easy secondary task is spatial (e.g., imagining a simple road map or tuning a radio) or verbal (e.g., recollecting a previous conversation or talking to a passenger). But under the same driving condition, a difficult secondary spatial task (such as mentally "walking" around a complex spatial layout or using a complicated navigational device) would be more likely to disrupt driving than would an equally difficult verbal task (such as reciting a difficult poem or engaging in a challenging conversation). As the driving environment becomes more hostile, a driver would very likely experience great difficulty in performing even a simple secondary spatial task but could still perform a secondary verbal tasks such as a light conversation. But as the difficulty of driving further increases--such as driving on a winding and narrow road in a stormy weather or in a heavy-traffic area--the driver would have to concentrate on driving, which could be easily disturbed by a slight disruption of any kind.

Since the queueing theoretic models have not attempted to address the parallel and structural aspects of task demand, it is not clear how the models would account for differential effects of spatial and verbal tasks. A natural approach to this problem from the queueing theoretic perspective might be to develop a set of complicated scheduling algorithms that allow the single server to vary task priorities and service rates according to the processing codes of the to-be-processed tasks. But it is not clear whether this is a feasible approach, and the problem remains until the algorithms are developed. A concept that might be invoked by the multiple resource theorists in explaining the existence of between-codes interference is the concept of concurrence cost. But the concept appears limited in this context, since it fails to explain why between-codes interference could be absent in some situations, but present in other situations, and perhaps more importantly, why between-codes interference could show a pattern of gradual increase as seen in a performance/difficulty tradeoff, and why the concurrence costs exist.

While it remains to be seen how the single channel and the multiple resources models would address these issues, the following three-node queueing network model offers a plausible account of the research findings. This queuing network model has a parallel processing component and a serial processing component. The parallel processing component refers to the parallel operation of a "spatial server" (S) and a "verbal server" (V), analogous to the spatial and verbal processing mechanisms or resources advocated in the multiple resources theory. The serial processing component refers to the serial operation from either S or V to the third server, which can be tentatively referred to as the central server (C).

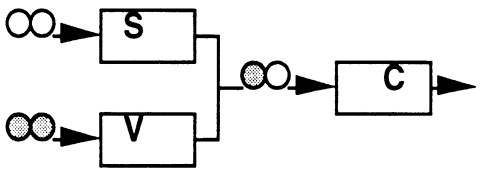

Figure 3: A 3-Node Queueing Network Model of Concurrent Spatial and Verbal Tasks

The model assumes that spatial and verbal tasks take separate routes of the network: a spatial task must be serviced by the server S, and a verbal task must

be serviced by the server V, before they receive the required service of the server C prior to their completion. The capacity of the servers in meeting the service needs of the arriving tasks can be represented as the service rate of the servers (μ). (the number of customers that can be serviced per unit of time). The model also assumes that a task is composed of a number of task components, and the arrival rate (λ) (the number of arriving customers per unit of time) of the task components is a rough measure of the difficulty or the service demand of a task. As in other queueing theoretic models of human performance, this model assumes that there is a performance cost associated with delaying service to a task (called the cost of waiting).

Using the symbols introduced at the beginning of the article, the essential constituents of the 3-node queueing network can be represented as follows:

1) K=3,

2) i=1, 2, 3, representing the spatial, verbal, and the central server, respectively,

3) $\gamma i = \lambda i$, for i = 1, 2
$= 0$, for i = 3

4) $\lambda i = \lambda i$, for i = 1,2
$= \lambda 1 + \lambda 2$, for i=3

5) $p_{12} = p_{21} = p_{31} = p_{32} = 0$,
$p_{ii} = 0$, for $\forall i$,
$p_{13} = p_{31} = 1$

6) $p_{10} = p_{20} = 0$, $p_{30}=1$

A number of performance measures can be computed using queueing network methods. Of most interest to the present analysis is the customer waiting time in front of each server. We continue to assume that the network is a separable queueing network, and for this type of network, we have,

$$Wqi = 1/(\mu i - \lambda i) - 1/\mu i,$$

where Wqi is the mean waiting time of customers in front of server i,

μi is the mean service rate of server i,

λi is the total mean arrival rate at server i.

For separable queueing networks, the arrival rate at the central server is the sum of the arrival rates at the spatial and the verbal servers. By making assumptions about the capacities of each server and the cost of waiting in front of each server, the queueing network model allows the modeling of a variety of patterns of task interference, considering both the difficulty and the processing codes of concurrent tasks. We assume that the μi's are constants, which is an assumption consistent with one of the most common assumptions in multitask research. The relationship between waiting time and arrival rate is diagrammed in Figures 4 and 5. It can be seen that waiting time increases monotonically as the arrival rate increases, indicating that performance decrements will increase as the difficulty of concurrent tasks increases. Waiting time approaches infinity as the arrival rate approaches the service rate, indicating a situation in which the tasks are too difficult to be performed simultaneously (beyond the processing limit). Therefore, of most interest to the model is the change in performance when λ is smaller than μ at each server.

As seen in Figure 3, customer waiting could occur in front of S or V or both servers (similar to the predictions of the multiple resources models), or in front of C (similar to the predictions of the single channel or queueing theoretic models), or a combination of both cases. A way to model the pattern of task interference discussed earlier in this section is to assume that the serial processing bottleneck (server C) has a capacity that is smaller than the sum of the capacities of the spatial and the verbal server, but greater than the capacity of either of the two servers. That is, $\mu C < \mu S + \mu V$, $\mu C > \mu S$, and $\mu C > \mu V$. For example, if we assume that $\mu C=14$, $\mu S=10$, $\mu V=10$, where μC, μS, μV refer to the capacity of the three server nodes respectively, then the relationships between waiting time and arrival rate at the three servers are shown in Figure 4 and Figure 5. If we further assume that the cost of waiting in front of the servers are identical, then the relationship between performance decrements and task difficulty should show a similar pattern as Figure 2 and Figure 3. Different patterns of relationship can be obtained if we make different assumptions about the cost of waiting.

Four important cases of customer waiting (task interference) emerge in the 3-node network:

Case 1: This is the case when the customer arrival rate at each of the centers is significantly smaller than the service rate (i.e., $\lambda i < \mu i$) (a situation in which easy tasks are time-shared with each other). Minimum waiting is expected, and thus performance decrements would be minimal in performing the simultaneous tasks, regardless of whether the same or separate processing codes are involved.

Case 2: When the customer arrival rate approaches the service rate of either the spatial (S) or the verbal server (V) but not that of the central server, significant waiting is expected in front of S or V, respectively, but not in front of C (e.g., when $\lambda s=8$, $\lambda v=1$, $\lambda c=9$). This is the case when within-code interference is the only source of task interference. This within-code interference will increase quickly as the arrival rate to the congested server S or V continues to increase.

Case 3: When the customer arrival rate approaches the service rate of the central server but not that of the spatial or the verbal server, customers are expected to wait in front of C, but not in front of S or V (e.g., when $\lambda s=6$, $\lambda v=6$, $\lambda c=12$). This is the case when between-codes interference is the only source of task interference. Progressive increase in λs or λv will produce progressive increase in λc, and produce a

corresponding progressive increase in the between-codes interference. However, as long as λs and λv are both still much smaller than μs and μv, within-code interference would be minimal.

Case 4: This is the most general case, in which performance decrements are caused by a combination of within- and between-codes interferences (a combination of Case 2 and Case 3). Due to the heavy demands of the tasks, queues could be observed in front of all three servers.

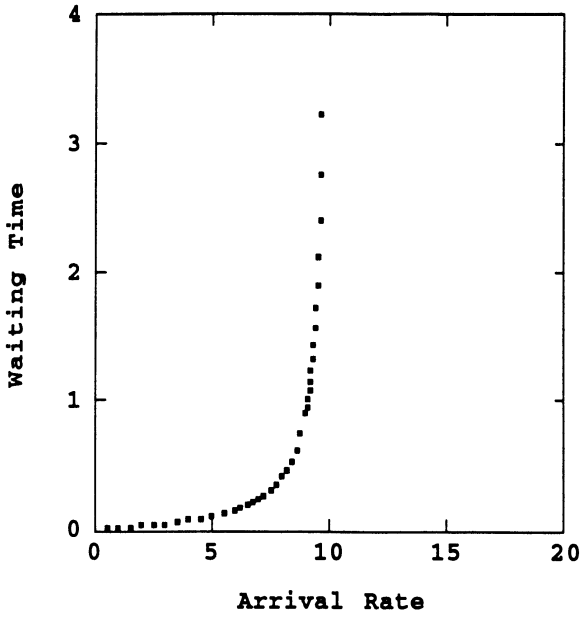

Figure 4: Relationship between waiting time and arrival rate at the spatial or verbal server

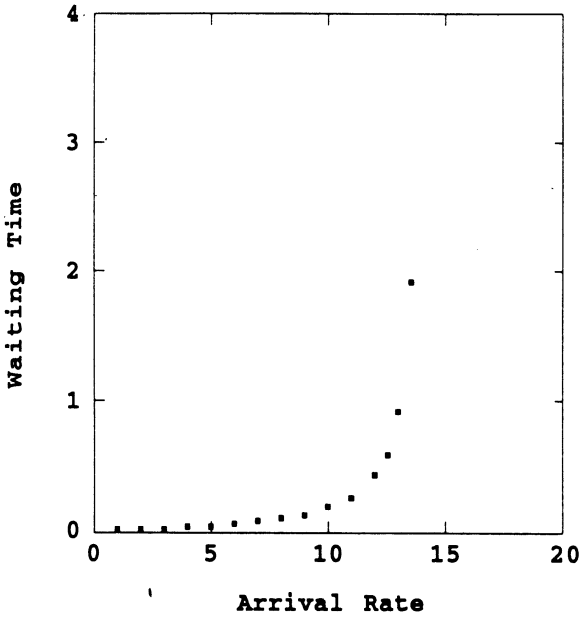

Figure 5: Relationship between waiting time and arrival rate at the central server

QUEUEING NETWORKS AS MODELS OF HUMAN-COMPUTER INTERACTION AND HUMAN-COMPUTER NETWORKS

In essence, models of multitask performance reviewed in the previous section are also models of human-computer interaction, although the role of the computer is not explicitly addressed. In this section, we consider the joint function of the human and the computer agents in a human-computer system. The two types of agents could very well be in charge of different functions and have different performance features, and we use queueing networks with different types of servers as a general modeling framework, which apparently treats the queueing theoretic models of a single server or identical servers as special cases.

Along this line of thinking and modeling philosophy, we have developed and are currently validating several queueing network models of human-computer interaction, both in standalone and in networked environments. Since the previous two sections of the paper have demonstrated the value of sojourn times and waiting times as performance measures, this section shifts the attention to another important stochastic process and performance measure--queue length distributions (or the states of a queueing network system). Also, since networks in the previous two sections are both open networks, we turn to closed networks in this section. However, it should be obvious that sojourn time, waiting time and open networks can be similarly applied in modeling human-computer interaction as well.

In a closed network, the same customers circulate eternally through the network. A closed network can also be interpreted as an open network with the total number of customers held fixed. In this system, a new customer arrives when and only when a customer leaves the system. The sliding window protocol of message communication and the paging policy in computer memory management are examples of this type of system.

One of the models we have developed is in the context of a failure management system (e.g., an aircraft or a process plant). In the simplest, standalone situation, a human and a computer work together to detect and correct system failures. One or more of the system components could fail (e.g., one or more engines or one or more boilers). The system is designed in such as way that when a system component fails, the computer will work on it first (e.g., to perform detection, warning and preliminary analysis functions) and then the human controller will work on it to perform higher level tasks. After the human controller have successfully completed his/her

task, the failed component will return to normal operation. This human-computer-machine system can be modeled as the simple queueing network model shown in Figure 6. For the 3-node (machine, computer, human) closed-series network, the state of the queueing system at any time instant t is a vector p(n1, n2, n3), representing the number of customers at node ni (i=1, 2, 3) at time t. For the system described above, p(n1, n2, n3) means that there are n1 machine components in normal operation, n2 being serviced by the computer, and n3 by the human. The total number of customers (denoted by S) should be a known quantity (e.g., S=k could mean that there is a total of k engines).

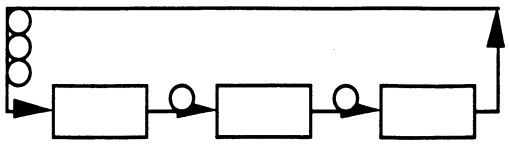

Fig.6. A closed queueing network model of human-computer interaction in a failure management system described in the text

p(n1, n2, n3) can be computed easily with the following set of equations, derived from the results of Jackson (1963) [34]:

$p(n1, n2, n3|S=k) = \omega^*(n1, n2, n3)/T^*(S=k)$;

$\omega^*(n1, n2, n3) = \prod^3_{i=1}\prod^{ki}_{j=1}(1/\mu ij)$;

$T^*(S=k) = \sum\omega^*(n1, n2, n3)$, summed over (n1, n2, n3) with S=k;

where μij is the mean service rate of node i when there are j customers at node i. Apparently, the "service rate" of the machine (node 1) is the rate at which it causes machine components to fail. The values for uij are usually obtainable from measurements, specifications or historical data.

The above set of equations allow us to predict a number of interesting performance features of the system. For example, it is easy to compute the proportion of time during which the human operator will have at least one machine component to repair ($\sum p(n1, n2, n3)$, summed over (n1, n2, n3) with n3>0 and S=k), or the proportion of time during which the machine will have at least two components working normally (e.g., at least two engines are running) ($\sum p(n1, n2, n3)$, summed over (n1, n2, n3) with n1>1 and S=k).

We have extended the work to modeling more complicated systems involving more than one humans and more than one computers--a human-computer network. A specific example is a failure management system in which there are two type of machine component failures, each type is handled by a computer and then by a human operator. A possible scenario is that two computers and two human operators work cooperatively in a manner illustrated in Figure 7, where the "copilot" completes his/her task alone with a probability of p, but need to forward the problem to the "pilot" with a probability of (1-p), before the component is returned for normal operation.

In order to compute the queue length distributions, we need the routing probability of the customers--pij, the probability that a machine component will immediately visit node j after departing from node i, which is specified by the task structure. In Figure 7, we have,

p12 = q (the probability that a failure is of type 1),

p13 = 1-q (the probability that a failure is of type 2),

p54 = p (the probability that human operator 2 needs help from human operator 1),

p56 =1 - p (the probability that human operator 2 can complete his/her job alone)

p24 = p35 = p46 = p60 = p01 = 1

pij = 0, for all other i and j's.

The expected value of the number of appearances of node i on a routing is computed with the following recursive equation,

$e_i = p0i + \sum^i_{m=1}(e_m\, pmi)$

With a total of k machine components in the queueing system, p(n1, n2, n3, n4, n5) can be computed easily with the following set of equations, derived from the results of Jackson (1963):

$p(n1, n2, n3, n4, n5|S=k) = \omega^*(n1, n2, n3)/T^*(S=k)$;

$\omega^*(n1, n2, n3, n4, n5) = \prod^5_{i=1}\prod^{ki}_{j=1}(e_i/\mu_{ij})$;

$T^*(S=k) = \sum\omega^*(n1, n2, n3, n4, n5)$, summed over (n1, n2, n3, n4, n5) with S=k;

A number of question can be answered with the computed queue length distributional values. The type of questions include the relative workload of operator 1 versus operator 2, the proportion of time during which the machine has at least c components operating normally, and the effects of changing network configuration or service rates.

Although the models are presented in the context of a network of human and computer agents interacting with each other toward a common goal, an area known as computer-supported cooperative work (CSCW). The same methodology can be applied to the broader area of human-computer networks, which also includes situations in which competitive or confrontive agents may compete with each other for

limited network resources and cause delays in servicing other agents' processing needs.

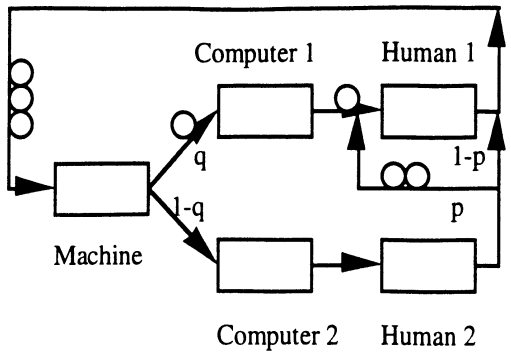

Fig. 7. A queueing network model of a human-computer network in the failure management system described in the text

Although a multitude of human-computer networking tools and CSCW applications have been developed, there is a substantial lack of predictive models and theories. As Schneiderman (1992) pointed out, this is a "vast uncharted territory: theories are sparse, measurement is informal, data analysis is overwhelming, and predictive models are nonexistent" ([35], p.391). The model presented in this section illustrates that queueing network methods could serve as a useful tool for establishing performance theories and predictive models of human-computer networks and for establishing theory-guided, systematic ways of performance measurement and analysis, particularly the issues of concern involve timing, scheduling and resource allocation.

The models presented in Figures 6 and 7 are currently being evaluated with lab experiments using a simulated failure management system and human subjects. We are also in the process of preparing experiments to validate a model of human-computer network with competing agents.

We hope that this article has illustrated the potential power of queueing network methods in establishing new models of human cognition, human performance and human-computer interaction on various analysis levels, and in establishing an integrated, computational framework for unifying some currently isolated models.

REFERENCES

[1] R. Disney and D. Konig, "Queueing networks: A survey of their random processes," SIAM Review, vol-27, 335-403, 1985.

[2] L. Kleinrock, "Queueing Systems," New York: Wiley, 1975.

[3] Y. Liu, "Visual scanning, memory scanning, and computational human performance modeling," Proc. of the Human Factors Society 37th Annual Meeting, 1993.

[4] Y. Liu, "A queueing network model of human multi-task performance," Tech. Rep. Univ. of Michigan, Dept. of IOE. 1993.

[5] J. Jackson, "Networks of waiting lines," Oper. Res., vol-5, pp.518-521, 1957.

[6] O. Boxma and H. Daduna, "Sojourn times in queueing networks," In H. Takagi (Ed.), Stochastic Analysis of Computer and Communications Systems, p.401-450, 1990, North Holland.

[7] J. Buzen, "Fundamental operational laws of computer system performance," Acta Informatica, vol-7, pp.167-182, 1976.

[8] P. Denning and J. Buzen, "The operational analysis of queueing network models," Computing Surveys, vol-10, pp.225-261, 1978.

[9] S. Sternberg, "The discovery of processing stages: Extensions of Donders's method," Acta Psychologica, 30, p.276-235, 1969.

[10] R. Pachella, "The interpretation of reaction time in information processing research," In B. Kantowitz (Ed.), Human information processing: Tutorials in Performance and Cognition, p.41-82, 1974, Hillsdale, N.J.: Erlbaum.

[11] W. McGill and J. Gibbon, "The general-gamma distribution and reaction times," J. of Math. Psych., 2, 1-18, 1965.

[12] J. McClelland, "On the time relations of mental processes: An examination of systems of processes in cascade," Psychological Review, 86, 287-330.

[13] J. Miller, "A queue-series model for reaction time, with discrete-stage and continuous-flow models as special cases," Psychological Review, 100, 702-715, 1993.

[14] J. Townsend and F. Ashby, "The Stochastic Modeling of Elementary Psychological Processes," Cambridge: Cambridge Univ. Press, 1983.

[15] R. Schweikert, "A critical path generalization of the additive factor methods: Analysis of a Stroop task," J. of Math. Psych., 18, pp.105-139, 1978.

[16] D. Fisher and W. Goldstein, "Stochastic PERT networks as models of cognition: Derivation of the mean, variance, and distribution of reaction time using Order-of-Processing (OP) diagrams," J. of Math. Psych., vol-27, 121-151, 1983.

[17] B. Simon and R. Foley, "Some results on sojourn times in acyclic Jackson networks," Management Science, vol-25, 1027-1034, 1979.

[18] F. Baccelli, et al., "Acyclic fork-join queueing networks," J. ACM, vol-36, pp.615-642, 1989.

[19] J. Walrand and P. Varaiya, "Sojourn times and the overtaking condition in Jacksonian networks," Adv. Appl. Prob., vol-12, 1000-1018, 1980.

[20] S. Baron, et al., (Eds.) "Quantitative Modeling of Human Performance in Complex, Dynamic Systems," Washington, DC.: National Academy Press, 1990.

[21] K. Pattipati, and D. Kleinman, "A review of the engineering models of information-processing and decision-making in multi-task supervisory control," In Damos, D. (Ed.), Multiple task performance. pp. 35-68, 1992.

[22] J. Senders, "The human operator as a monitor and controller of multidegree freedom systems," IEEE Trans. on HFE, HFE-5, 2-5, 1964.

[23] J. Carbonell, "A queueing model of many-instrument visual sampling," IEEE Trans. HFE-4, pp.157-164, 1966.

[24] J. Senders and M. Posner, "A queueing model of monitoring and supervisory behavior," in T. Sheridan and G. Johannsen (Eds.), Monitoring Behavior and Supervisory Control. New York: Plenum, 1976.

[25] D. Schmidt, "A queueing analysis of the air traffic controller's workload," IEEE Trans. SMC-8, pp.492-293, 1978.

[26] W. Rouse, "Human-computer interaction in multitask situations," IEEE Trans. on SMC-7, 384-391, 1977.

[27] Y. Chu and W. Rouse, "Adaptive allocation of decision making responsibility between human and computer in multi-tsk situations," IEEE Trans. SMC-9, pp.769-778, 1979.

[28] R. Walden and W. Rouse, "a queueing model of pilot decisionmaking in a multitask flight management situation," IEEE Trans. SMC-8, pp.867-875, 1978.

[29] J. Greenstein and W. Rouse, "A model of human deciison making in multiple process monitoring situations," IEEE Trans. SMC-12, pp.182-193, 1982.

[30] M. Adams, et al., "Strategic workload and the cognitive management of advanced multi-task systems," CSERIAC SOAR Report 91-6, 1991.

[31] C. Wickens, "Processing resources in attention," In Parasuraman, R. and D. Davis (Eds.) Varieties of attention. New York: Academic Press, 1984.

[32] K. Laughery, "Micro SAINT: A tool for modeling human performance in systems," in G. McMillan et al., (Eds.), "Applications of Human Performance Models to System Design," pp.219-230, New York: Plenum, 1989.

[33] R. North and V. Riley, "W/INDEX: A predictive model of operator workload," in G. McMillan et al., (Eds.), "Applications of Human Performance Models to System Design," pp.81-89, New York: Plenum, 1989.

[34] J. Jackson, "Jobshop-like queueing systems," Management Sci., vol-10, pp. 131-142, 1963.

[35] B. Schneiderman, "Designing the User Interface," 2nd Ed., New York: Addison-Wesley, 1992.