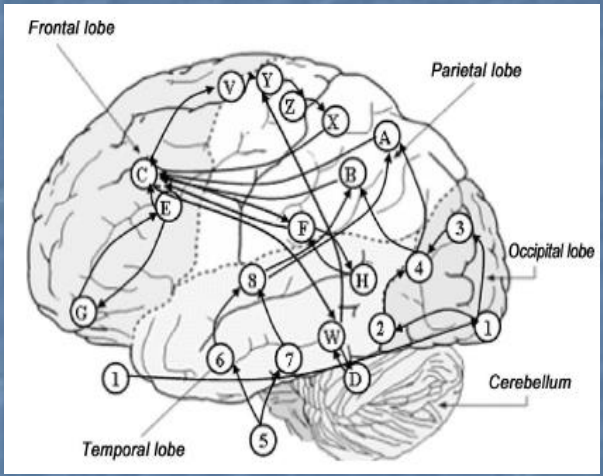
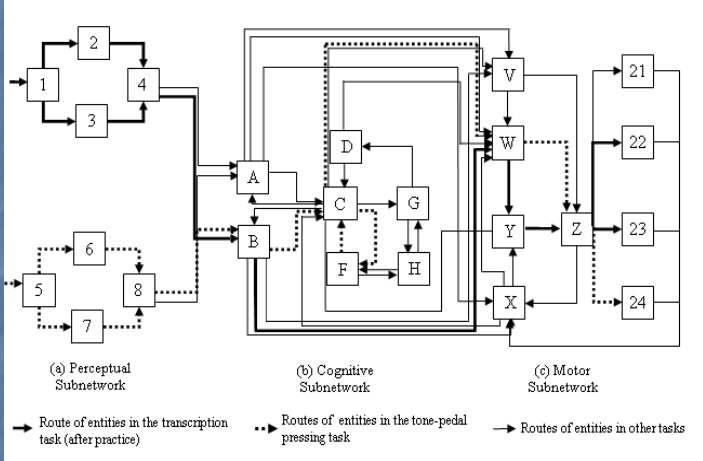
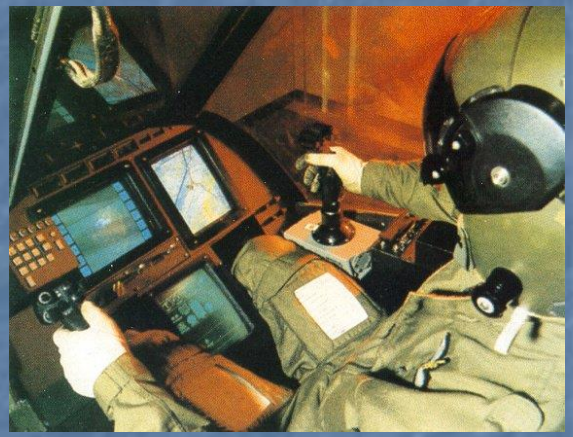
Integrative Modeling and Simulation of Human Behavior and Human-Machine Systems with Queuing Network (QN) Architecture







Queueing Network of Mental Architecture

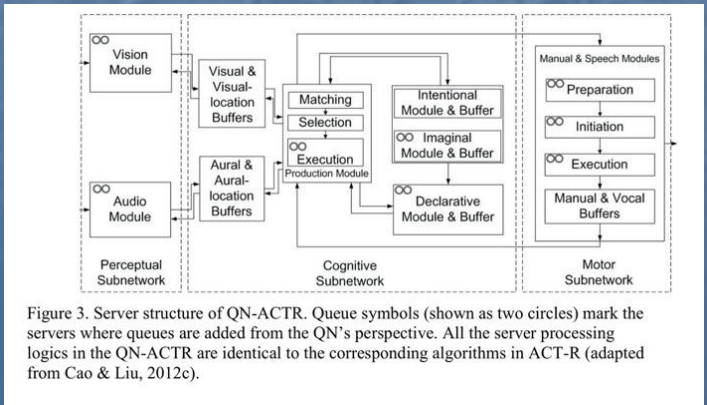


Figure 3. Server structure of QN-ACTR. Queue symbols (shown as two circles) mark the servers where queues are added from the QN's perspective. All the server processing logics in the QN-ACTR are identical to the corresponding algorithms in ACT-R (adapted from Cao & Liu, 2012c).

Yili Liu UM-IOE HFES-Aspire Workshop 2024

Queueing Network (QN)
Models of Human Behavior (MHB)
QN-MHB

1. RT: Reaction Time (QN-RT) and Mental Structure
2. RT and Accuracy (QN-RMD) (Mental Structure vs State of Mind)
3. Procedural Tasks (QN-MHP or QN-MHP-BE)
4. Complex Cognition Tasks (QN-ACTR)
5. Visual Attention tasks (QN-NSEEV) (QN-RLEM)
6. Manual or Continuous Control tasks (QN-Control: Classical/Modern)
7. Basic Body Motion tasks (QN-MTM)
8. Mind-Body System (QN-MBS)
9. Neural level (QN-Neural: Indexes and Neural Networks)
10. Nervous and Endocrine Systems (QN-NES)
11. Multi-Person Multi-Machine QN (QN-HMN)
12. Engineering Applications in various domains

Note: relations with **Task Network** Methods/Tools such as Micro Saint #, IMPRINT)

Psychological Review
1996, Vol. 103, No. 1, 116–136

0033-295X/96/$3.00

Queueing Network Modeling of Elementary Mental Processes

Yili Liu
University of Michigan

This article examines the use of reaction time (RT) to infer the possible configurations of mental systems and presents a class of queueing network models of elementary mental processes. The models consider the temporal issue of discrete versus continuous information transmission in conjunction with the architectural issue of serial versus network arrangement of mental processes. Five elementary but important types of queueing networks are described in detail with regard to their predictions for RT behavior, and they are used to re-examine existing models for psychological processes. As continuous-transmission networks in the general form, queueing network models include the existing discrete and continuous serial models and discrete network models as special cases, cover a broader range of temporal and architectural structures that mental processes might assume, and can be subjected to empirical tests.

Why is there a delay between stimulus presentation and response initiation? This has been one of the most enduring and fundamental puzzles for psychologists. The current belief of cognitive psychologists is that this delay is a reflection of the dynamic activities of an underlying mental structure that transforms stimulus into response. Most important, because 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 structure.

Theoretical models that use RT as the primary performance measure to infer the general structure of mental systems are often called models for RT. Of great interest to the present article are two issues that are central to RT modeling and theory in cognitive psychology. The two issues also define two dimensions along which RT models can be classified. One of the two is a temporal dimension distinguishing discrete-transmission models from continuous-transmission models, and the other is an architectural dimension distinguishing serial-stage models from network models. All of the models assume that the psychological activity that transforms stimulus into response is composed of a system of mental processes. Discrete-transmission models assume that a mental process transmits its processing output in an indivisible unit and will not make its output available to other processes until it is completed. Therefore, a process cannot begin until all of its preceding processes are completed. Continuous-transmission models, in contrast, assume that each process transmits its partial outputs to other processes continuously as soon as they are available rather than waiting for the full completion of processing, and thus a process can begin even though its preceding processes are still active. Serial-stage models assume a serial arrangement of mental processes, whereas network models assume a network configuration. The two dimensions jointly define four classes of models as shown in Table 1.

Although the distinction between the terms *serial* and *network* is usually quite standard in the literature, there exist some differences in the use of the terms *discrete* and *continuous* by different authors. As discussed by Miller (1988, 1990), the terms *discrete* and *continuous* have been used in at least four different senses in cognitive models: discrete versus continuous information representations, discrete versus continuous information transformation, discrete versus continuous information transmission, and discrete versus continuous variation in a priori state of an information processing stage (see Miller, 1988, 1990, for excellent discussions of this topic). The aim of the present article is to address the issue of discrete versus continuous information transmission in conjunction with the issue of serial versus network arrangements. Several important RT models are therefore not included in Table 1, primarily because their concerns were on discrete versus continuous information representation or transformation. Prominent among these models include the model developed by Meyer and his colleagues (Meyer, Irwin, Osman, & Kounios, 1988) and the stochastic diffusion model (Ratcliff, 1988).

It should also be noted here that, although the terms *continuous* and *discrete* have been used extensively in the literature to refer to models that do or do not allow partial output and temporal overlap of process durations, there is no intrinsic relationship between continuity of transmission and temporal overlapping of process activities. A process may continuously transmit its partial output to its successors, but processes could still be in strict temporal sequence if each process has to wait until it has accumulated all of the continuously arrived inputs before it starts. Similarly, although partial outputs transmitted in the form of a continuous flow will support overlapping process activities, those transmitted as "discrete packets" will do so as well, as long as the number of packets that can be separately transmitted is greater than 1. However, for lack of a better term and to be consistent with the common practice in RT modeling, I continue to use *continuous* and *discrete* transmission to distinguish whether a series of processes could be active concurrently.

Miller (1982, 1988) has suggested that discrete and continu-

Correspondence concerning this article should be addressed to Yili Liu, Department of Industrial and Operations Engineering, University of Michigan, 1205 Beal Avenue, Ann Arbor, Michigan 48109-2117. Electronic mail may be sent via the Internet to yililiu@umich.edu.

Table 1
Reaction Time Models Classified in Terms of Discrete Versus Continuous Transmission and Serial Versus Network Architectures

Temporal transmission	Arrangement of mental processes: Serial stages	Arrangement of mental processes: Network configuration
Discrete	Subtractive Additive factors General gamma	PERT (critical path) Network
Continuous	Cascade Queue series	Queueing network

Note. PERT = program evaluation and review technique.

ous transmissions be viewed as the extremes of a continuum defined by the extent to which the output of a stage can be divided and separately transmitted to other stages (the *grain size* of transmission). At one extreme, discrete transmission models have the largest possible grain size because the output must be transmitted as a whole unit. At the other extreme, outputs in *continuous-flow* models can be divided into an arbitrarily large number of small units. Intermediate models assume grain sizes between the two extremes, which are also called *nondiscrete models* (Miller, 1993). In this article, I use the term *continuous* to refer to both truly continuous flows and continuous transmission of discrete packets of partial outputs.

Historically, the modeling work covered in Table 1 started with the serial discrete-stage models shown in the top left cell. These models assume nonoverlapping durations of serially arranged processes or stages. The underlying models for the subtraction method developed by Donders (1868/1969), and the additive factor method developed by Sternberg (1969), belong to this class of models. Donders assumed that processes can be added or deleted from a chain of processes while leaving intact the rest of the chain (called the *assumption of pure insertion*). On the basis of this assumption, Donders proposed that the mean duration of an inserted or deleted process can be inferred by examining the difference between the mean duration of a task that does not include the process in question and one that does—a method known as the *subtraction method for mean RT analysis*. Because pure insertion appears to be a strong assumption, Sternberg tried to relax this assumption by addressing the issue of how experimental manipulations might change the durations of processes rather than insert or delete them. Sternberg assumed that the mean duration of a process depends on experimental manipulations that influence it, but not directly on the mean durations of other processes, and a change in the mean duration of a process will not produce indirect effects on the mean durations of other processes in the processing chain (called the *assumption of selective influence*). On the basis of this assumption, Sternberg proposed an additive factor method for mean RT analysis, according to which experimental factors that influence a common process will interact with each other in an analysis of variance of the RT data, whereas those influencing separate processes will be additive. The serial discrete-stage model and the additive factor method have been the fundamental basis of a large body of experimental literature.

Although the models underlying Donders's (1868/1969) and Sternberg's (1969) methods are models for mean RT, numerous authors have examined properties of RT at the distributional level, because much more can be obtained from examining RT distributions than from examining mean RTs alone. It has been shown that examining RT distributions could be critical in discriminating models that would demonstrate similar behavior at the mean level. When the durations of serial processes are independent of each other, RT distribution is the convolution of the process durations. McGill and Gibbon (1965) noted that RT in a serial discrete-stage model can be described by the general-gamma distribution, if the independent stage durations are exponentially distributed with different duration means. Several authors argued that the convolution of normal and exponential distributions provides a close approximation to observed RT distributions (Hockley, 1984; Hohle, 1965; Ratcliff & Murdock, 1976). Ashby and Townsend (1980; Ashby, 1982b; Townsend & Ashby, 1983) extended the assumptions of pure insertion and selective influence to the distributional level and proved a set of theorems that can be used to test these assumptions.

Models in the other cells of Table 1 try to relax the assumption of serial and nonoverlapping process activities adopted by serial discrete-stage models with an aim to generalize the class of RT models and broaden the range of possible mental structures for elementary psychological processes. In the bottom left cell of Table 1, we find the models that permit temporal overlap of sequentially arranged processes. Prominent among this class of RT models are McClelland's (1979) cascade model and, more recently, Miller's (1993) queue-series model.

The cascade 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 produce continuous output that is always available for processing at the next level. The heart of the cascade model is a cascade equation, which expresses the activation of a linear integrator at a processing level as a function of the asymptotic activation of the linear integrator that would result if the stimulus were left on indefinitely and the rate constants of the different processes in the system. McClelland (1979) examined the effects of manipulating rate constants and asymptotic levels on RT and derived a set of predictions for RT behavior. Similar to the additive factor method, the cascade model shows that experimental factors affecting the rate of the same process will tend to interact, whereas those affecting the rates of different processes are generally additive. However, the predictions become more complicated and start to diverge from those of the additive factor method when at least one of the experimental factors affects the asymptotic level of activation.

The queue-series model recently developed by Miller (1993) assumes that the cognitive system is composed of a series of stages, and the stimulus is regarded as consisting of a number, M, of distinct components. The important concept of grain size of transmission is mathematically represented by the parameter M. Discrete stages and cascade flows are treated as special cases of the queue series, corresponding to the cases of $M = 1$ and $M = \infty$, respectively. Other positive values of M represent inter-

Acta Psychologica 30 *Attention and Performance II (W. G. Koster, ed.)* 1969, 412–431

ON THE SPEED OF MENTAL PROCESSES

F. C. DONDERS [1]

While philosophy is occupied in the abstract with the contemplation of mental phenomena, physiology, having at its disposal the results of philosophy, has to investigate the relation between those phenomena and the action of the brain. In the domain of morphology that relation immediately leaps to the eye. Considering the known facts of comparative anatomy and anthropology, any doubt concerning the existence of such a relation is untenable. But physiology cannot be content with that general result. Along with disorders observed in the case of pathological changes, physiology tries to locate the various mental faculties as much as possible by experimentation, and especially to trace the nature of the action accompanying the mental phenomena. It therefore relates the study on chemical composition and the metabolism of its components with the investigation of the fine structure of the brain. It finds that with the loss of blood or suppressed action of the heart, consciousness is lost, it learns from this that the regular supply of blood is a necessary condition for mental processes, and concludes that metabolism is at the root of brain life. Further, it establishes that, as in all other organs, the blood undergoes a change as a consequence of the nourishment of the brain, and discovers in comparing the incoming and outflowing blood that oxygen has been consumed, that carbonic acid has been formed and that heat has been generated. It knows that the heat may have originated from other forms of energy, for instance from electromotive action that it may postulate in the brain, after proving

[1] This is a translation of Donders' article entitled 'Over de snelheid van psychische processen' which appeared in 'Onderzoekingen gedaan in het Physiologisch Laboratorium der Utrechtsche Hoogeschool, 1868—1869, Tweede reeks, II, 92—120'. An identical article appeared in: 'Nederlandsch Archief voor Genees- en Natuurkunde 1869, 4, 117—145'. The translation was made by W. G. Koster.
A translation into French entitled 'La vitesse des actes psychiques' appeared in 'Archives Néerlandaises, 1868, III, 269—317'.
A translation into German entitled 'Die Schnelligkeit psychischer Prozesse' appeared in 'Archiv für Anatomie und Physiologie und wissenschaftliche Medizin, 1868, 657—681'.

DONDERS INSTITUTE

Search

Donders Institute for Brain, Cognition and Behaviour

Students • Staff • Alumni

HOME | RESEARCH | Ssocietal impact | TALENT DEVELOPMENT | WORKING AT | AGENDA | ABOUT US

Donders Institute for Brain, Cognition and ... > About us > Biography F.C. Donders

- Donders Institute
- Research integrity advisors
- Diversity and Sustainable Science
- Preferred Institutional Partners
- Donders Awards
- Public Outreach
- Biography F.C. Donders
- Contact

Biography F.C. Donders

F.C. Donders, the father of mental chronometry

Franciscus Cornelis Donders was born on the 27th of May in 1818, in Tilburg, a manufacturing town in North Brabant, the Netherlands. He had eight older sisters - to have a son was a long deferred hope of his parents. Frans Donders went to seminaries in Tilburg and Boxmeer and subsequently attended Medical School in Utrecht. The rest of his academic career (from the age of 29 on) he held a position as Professor of Physiology at the University of Utrecht. (To read F.C. Donders acceptance speech (in Dutch) after obtaining his professorship, click here (pdf, 280 kB)).

Donders was one of the pioneers of ophthalmology. His major contributions were in the areas of refraction and astigmatism. In 1858 Donders established the first eye hospital in the Netherlands. In 1864 his influential work "On the anomalies of accommodation and refraction of the eye with a preliminary essay on physiologic dioptrics" was published in English. It describes a complete doctrine, both theory and practice, of the employment and prescription of corrective glasses.

Donders' interests included not only ocular physiology, eye movements (he discovered what came to be called the Law of Donders), color vision and color blindness, but also general physiology, evolution, and mental processes. His views are strikingly modern. For example, he investigated cerebral circulation, and was excited about his discoveries on the topic of oxygen metabolism in the brain:

"As in all organs, the blood undergoes a change as a consequence of the nourishment of the brain". One "discovers in comparing the incoming and outflowing blood that oxygen has been consumed" (Donders, 1868). This insight, together with a subtractive method designed by Donders, constitutes the basis of the two most widely used modern functional neuroimaging techniques, PET and fMRI.

Donders was also interested in the mechanisms underlying speech. In his monograph "De physiologie der spraakklanken, in het bijzonder van die der Nederlandsche taal" [The physiology of speech sounds, in particular those of the Dutch language] (Donders, 1870), he gave a detailed account of the acoustic and phonetic properties of (Dutch) speech sounds and how they are articulated.

Acta Psychologica
Volume 30, 1969, Pages 276-315

The discovery of processing stages: Extensions of Donders' method

Saul Sternberg [1]

https://doi.org/10.1016/0001-6918(69)90055-9 Get rights and content

Abstract

A new method is proposed for using reaction-time (RT) measurements to study stages of information processing. It overcomes limitations of Donders' and more recent methods, and permits the discovery of stages, assessment of their properties, and separate testing of the additivity and stochastic independence of stage durations. The main feature of the *additive-factor method* is the search for non-interacting effects of experimental factors on mean RT. The method is applied to several binary-classification experiments, where it leads to a four-stage model, and to an identification experiment, where it distinguishes two stages. The sets of stages inferred from both these and other data are shown to carry substantive implications. It is demonstrated that stage-durations may be additive without being stochastically independent, a result that is relevant to the formulation of mathematical models of RT.

Journal of Mathematical Psychology
Volume 18, Issue 2, October 1978, Pages 105-139

A critical path generalization of the additive factor method: Analysis of a stroop task

Richard Schweickert

https://doi.org/10.1016/0022-2496(78)90059-7 Get rights and content

Abstract

Sternberg's additive factor method was generalized to apply to tasks involving both serial and concurrent processing. The generalization is based on the critical path method of scheduling. The effects on reaction time of factors prolonging separate processes in a task are discussed; in general these effects are interactions of a simple form. Reaction times can be used to deduce, in part, the schedule of the mental processes in a task, including their order of execution. Bounds on process durations can be derived. Often there are redundant equations so the method can be easily rejected if it does not apply. A dual task experiment by Greenwald was analyzed. In the task subjects were presented with two stimuli and made a response to each under high and low compatibility conditions. Two bottlenecks in processing were located: (a) Subjects make only one decision at a time, in accordance with single channel theory, although the high compatibility condition may be an exception; (b) there is a mental process which takes longer when the stimuli conflict. The decisions about the two stimuli probably change places in the schedula when compatibility is changed.

JOURNAL OF MATHEMATICAL PSYCHOLOGY 18, 105–139 (1978)

A Critical Path Generalization of the Additive Factor Method: Analysis of a Stroop Task

RICHARD SCHWEICKERT

The University of Michigan

Sternberg's additive factor method was generalized to apply to tasks involving both serial and concurrent processing. The generalization is based on the critical path method of scheduling. The effects on reaction time of factors prolonging separate processes in a task are discussed; in general these effects are interactions of a simple form. Reaction times can be used to deduce, in part, the schedule of the mental processes in a task, including their order of execution. Bounds on process durations can be derived. Often there are redundant equations so the method can be easily rejected if it does not apply. A dual task experiment by Greenwald was analyzed. In the task subjects were presented with two stimuli and made a response to each under high and low compatibility conditions. Two bottlenecks in processing were located: (a) Subjects make only one decision at a time, in accordance with single channel theory, although the high compatibility condition may be an exception; (b) there is a mental process which takes longer when the stimuli conflict. The decisions about the two stimuli probably change places in the schedula when compatibility is changed.

We measure reaction times in order to make inferences about mental processes, such as perceiving, deciding, remembering, and thinking, which we cannot observe in subjects directly. Two methods often used to analyze reaction times, Donders' subtractive method (1968) and Sternberg's additive factor method (1969a), assume that the mental processes under consideration are performed in a sequence, one process beginning as soon as its predecessor has finished (Fig. 1). In this paper we consider more general arrangements of processes (Fig. 2).

FIG. 1. A sequence of processes arranged end to end.

In both the subtractive and the additive factor methods, the amount of time required to complete a sequence of processes is considered to be the sum of the durations of all the processes in the sequence. In the subtractive method various experimental manipulations are used to insert processes into the sequence of processes or to delete them. The duration of a process can be determined by subtracting the amount of time required to complete a sequence which does not include the process from the amount of time required to complete the sequence when the process is included.

0022-2496/78/0182-0105$02.00/0

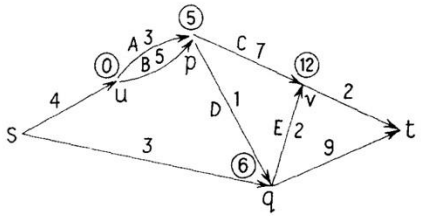


FIG. 2. A task network. Each arrow represents a process and associated with each arrow is a number giving the duration of the process. Circled numbers indicate the duration of the longest path from u to the point under the circled number.

A drawback of the subtractive method is that it cannot be used to determine the order of the processes in the sequence. A more important drawback of the method is that in most cases verification of its results must rest on evidence gathered by other techniques. To use the subtractive method to make two independent measurements of the duration of some process for the purpose of verification, the process must be embedded in at least two different sequences. But it may be difficult or impossible experimentally to construct two such different sequences of processes, each complete enough psychologically to allow a subject to respond to a stimulus and yield a reaction time.

In the additive factor method the sequence always consists of the same processes, but various experimental manipulations are used to prolong the processes of interest beyond their baseline durations. For example, a visual perception process may be prolonged by making the stimulus fuzzy; a decision process may be prolonged by increasing the number of choices available to the subject. These prolongations can be made singly or in combinations. If each of several manipulations prolongs a different process, the increase in reaction time (RT) produced by performing the manipulations concurrently would be the sum of the increases produced by performing them singly. Sternberg (1969a) says this additive rule is very likely to hold, although not inevitable. We see below that if all the processes are not arranged in a sequence, the effects on RT of manipulations prolonging separate processes can be interactive. Therefore, investigators should be cautious about interpreting an interaction between factors as an indication that they affect the same process.

The additive factor method generates falsifiable predictions since the effect of prolonging any combination of processes concurrently should be predictable from the effects of prolonging them individually. However, when all the processes are arranged in a sequence the method cannot be used to obtain their order nor their durations.

Not all psychological activities are suitably represented as a sequence of processes arranged end to end. For example, when a subject is given two tasks to perform simultaneously the time required to complete both tasks is usually less than the sum of the times required to perform them separately (Kantowitz, 1974). Evidentially, some of the processes involved in the two tasks can be performed simultaneously.

In general a task composed of several processes can be represented as a network in which each arrow represents a process (Fig. 2). Two special types of network are the serial

Psychological Review

VOLUME 86 NUMBER 4 JULY 1979

On the Time Relations of Mental Processes: An Examination of Systems of Processes in Cascade

James L. McClelland
University of California, San Diego

This article examines the possibility that the components of an information-processing system all operate continuously, passing information from one to the next as it becomes available. A model called the *cascade model* is presented, and it is shown to be compatible with the general form of the relation between time and accuracy in speed–accuracy trade-off experiments. In the model, experimental manipulations may have either or both of two effects on a processing level: They may alter the rate of response or the asymptotic quality of the output. The effects of such manipulations on the output of a system of processes are described. The model is then used to reexamine the subtraction and additive factors methods for analyzing the composition of systems of processes. The examination of the additive factors method yields particularly interesting results. Among them is the finding that factors that affect the rates of two different processes would be expected to have additive effects on reaction times under the cascade model, whereas factors that both affect the rate of the same process would tend to interact, just as in the case in which the manipulations affect the durations of discrete stages. On the other hand, factors that affect asymptotic output tend to interact whether they affect the same or different processes. In light of this observation, the conclusions drawn from several studies about the locus of perceptual and attentional effects on processing are reexamined. Finally, an outline is presented of a new method for analyzing processes in cascade. The method extends the additive factors method to an analysis of the parameters of the function relating response time and accuracy.

When we analyze performance in an information-processing task, we often proceed by assuming that performance may be decomposed into a set of separate subprocesses. Sternberg (1969a), following Donders (1868–1869), has noted that we can attempt to study the supposed component processes themselves using reaction-time data if we make some additional assumptions about their temporal relations. In Sternberg's formulation, the important assumptions are (a) that only one component process may be active at any one time, and (b) that the amount of time taken up by one component process does not influence the time required for another. In practice, these assumptions have usually been embodied in a model of performance in which the subprocesses are identified as successive temporal *stages*, each of which occupies a separate interval of time. I call this model the *discrete stage model*.

Explicitly or implicitly, the discrete stage model is the cornerstone of a huge experimental literature addressing itself to the nature and organization of mental processes. The logic underlying this literature is direct and compelling. If the discrete stage model is correct, and if we can find two tasks that differ in that a

 0033-295X/79/8604-0287$00.75

Psychological Review
1996, Vol. 103, No. 1, 116–136

0033-295X/96/$3.00

Queueing Network Modeling of Elementary Mental Processes

Yili Liu
University of Michigan

This article examines the use of reaction time (RT) to infer the possible configurations of mental systems and presents a class of queueing network models of elementary mental processes. The models consider the temporal issue of discrete versus continuous information transmission in conjunction with the architectural issue of serial versus network arrangement of mental processes. Five elementary but important types of queueing networks are described in detail with regard to their predictions for RT behavior, and they are used to re-examine existing models for psychological processes. As continuous-transmission networks in the general form, queueing network models include the existing discrete and continuous serial models and discrete network models as special cases, cover a broader range of temporal and architectural structures that mental processes might assume, and can be subjected to empirical tests.

Why is there a delay between stimulus presentation and response initiation? This has been one of the most enduring and fundamental puzzles for psychologists. The current belief of cognitive psychologists is that this delay is a reflection of the dynamic activities of an underlying mental structure that transforms stimulus into response. Most important, because 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 structure.

Theoretical models that use RT as the primary performance measure to infer the general structure of mental systems are often called models for RT. Of great interest to the present article are two issues that are central to RT modeling and theory in cognitive psychology. The two issues also define two dimensions along which RT models can be classified. One of the two is a temporal dimension distinguishing discrete-transmission models from continuous-transmission models, and the other is an architectural dimension distinguishing serial-stage models from network models. All of the models assume that the psychological activity that transforms stimulus into response is composed of a system of mental processes. Discrete-transmission models assume that a mental process transmits its processing output in an indivisible unit and will not make its output available to other processes until it is completed. Therefore, a process cannot begin until all of its preceding processes are completed. Continuous-transmission models, in contrast, assume that each process transmits its partial outputs to other processes continuously as soon as they are available rather than waiting for the full completion of processing, and thus a process can begin even though its preceding processes are still active. Serial-stage models assume a serial arrangement of mental processes, whereas network models assume a network configuration. The two dimensions jointly define four classes of models as shown in Table 1.

Although the distinction between the terms *serial* and *network* is usually quite standard in the literature, there exist some differences in the use of the terms *discrete* and *continuous* by different authors. As discussed by Miller (1988, 1990), the terms *discrete* and *continuous* have been used in at least four different senses in cognitive models: discrete versus continuous information representations, discrete versus continuous information transformation, discrete versus continuous information transmission, and discrete versus continuous variation in a priori state of an information processing stage (see Miller, 1988, 1990, for excellent discussions of this topic). The aim of the present article is to address the issue of discrete versus continuous information transmission in conjunction with the issue of serial versus network arrangements. Several important RT models are therefore not included in Table 1, primarily because their concerns were on discrete versus continuous information representation or transformation. Prominent among these models include the model developed by Meyer and his colleagues (Meyer, Irwin, Osman, & Kounios, 1988) and the stochastic diffusion model (Ratcliff, 1988).

It should also be noted here that, although the terms *continuous* and *discrete* have been used extensively in the literature to refer to models that do or do not allow partial output and temporal overlap of process durations, there is no intrinsic relationship between continuity of transmission and temporal overlapping of process activities. A process may continuously transmit its partial output to its successors, but processes could still be in strict temporal sequence if each process has to wait until it has accumulated all of the continuously arrived inputs before it starts. Similarly, although partial outputs transmitted in the form of a continuous flow will support overlapping process activities, those transmitted as "discrete packets" will do so as well, as long as the number of packets that can be separately transmitted is greater than 1. However, for lack of a better term and to be consistent with the common practice in RT modeling, I continue to use *continuous* and *discrete* transmission to distinguish whether a series of processes could be active concurrently.

Miller (1982, 1988) has suggested that discrete and continu-

Correspondence concerning this article should be addressed to Yili Liu, Department of Industrial and Operations Engineering, University of Michigan, 1205 Beal Avenue, Ann Arbor, Michigan 48109-2117. Electronic mail may be sent via the Internet to yililiu@umich.edu.

Table 1
Reaction Time Models Classified in Terms of Discrete Versus Continuous Transmission and Serial Versus Network Architectures

Temporal transmission	Arrangement of mental processes: Serial stages	Arrangement of mental processes: Network configuration
Discrete	Subtractive Additive factors General gamma	PERT (critical path) Network
Continuous	Cascade Queue series	Queueing network

Note. PERT = program evaluation and review technique.

ous transmissions be viewed as the extremes of a continuum defined by the extent to which the output of a stage can be divided and separately transmitted to other stages (the *grain size* of transmission). At one extreme, discrete transmission models have the largest possible grain size because the output must be transmitted as a whole unit. At the other extreme, outputs in *continuous-flow* models can be divided into an arbitrarily large number of small units. Intermediate models assume grain sizes between the two extremes, which are also called *nondiscrete models* (Miller, 1993). In this article, I use the term *continuous* to refer to both truly continuous flows and continuous transmission of discrete packets of partial outputs.

Historically, the modeling work covered in Table 1 started with the serial discrete-stage models shown in the top left cell. These models assume nonoverlapping durations of serially arranged processes or stages. The underlying models for the subtraction method developed by Donders (1868/1969), and the additive factor method developed by Sternberg (1969), belong to this class of models. Donders assumed that processes can be added or deleted from a chain of processes while leaving intact the rest of the chain (called the *assumption of pure insertion*). On the basis of this assumption, Donders proposed that the mean duration of an inserted or deleted process can be inferred by examining the difference between the mean duration of a task that does not include the process in question and one that does—a method known as the *subtraction method for mean RT analysis*. Because pure insertion appears to be a strong assumption, Sternberg tried to relax this assumption by addressing the issue of how experimental manipulations might change the durations of processes rather than insert or delete them. Sternberg assumed that the mean duration of a process depends on experimental manipulations that influence it, but not directly on the mean durations of other processes, and a change in the mean duration of a process will not produce indirect effects on the mean durations of other processes in the processing chain (called the *assumption of selective influence*). On the basis of this assumption, Sternberg proposed an additive factor method for mean RT analysis, according to which experimental factors that influence a common process will interact with each other in an analysis of variance of the RT data, whereas those influencing separate processes will be additive. The serial discrete-stage model and the additive factor method have been the fundamental basis of a large body of experimental literature.

Although the models underlying Donders's (1868/1969) and Sternberg's (1969) methods are models for mean RT, numerous authors have examined properties of RT at the distributional level, because much more can be obtained from examining RT distributions than from examining mean RTs alone. It has been shown that examining RT distributions could be critical in discriminating models that would demonstrate similar behavior at the mean level. When the durations of serial processes are independent of each other, RT distribution is the convolution of the process durations. McGill and Gibbon (1965) noted that RT in a serial discrete-stage model can be described by the general-gamma distribution, if the independent stage durations are exponentially distributed with different duration means. Several authors argued that the convolution of normal and exponential distributions provides a close approximation to observed RT distributions (Hockley, 1984; Hohle, 1965; Ratcliff & Murdock, 1976). Ashby and Townsend (1980; Ashby, 1982b; Townsend & Ashby, 1983) extended the assumptions of pure insertion and selective influence to the distributional level and proved a set of theorems that can be used to test these assumptions.

Models in the other cells of Table 1 try to relax the assumption of serial and nonoverlapping process activities adopted by serial discrete-stage models with an aim to generalize the class of RT models and broaden the range of possible mental structures for elementary psychological processes. In the bottom left cell of Table 1, we find the models that permit temporal overlap of sequentially arranged processes. Prominent among this class of RT models are McClelland's (1979) cascade model and, more recently, Miller's (1993) queue-series model.

The cascade 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 produce continuous output that is always available for processing at the next level. The heart of the cascade model is a cascade equation, which expresses the activation of a linear integrator at a processing level as a function of the asymptotic activation of the linear integrator that would result if the stimulus were left on indefinitely and the rate constants of the different processes in the system. McClelland (1979) examined the effects of manipulating rate constants and asymptotic levels on RT and derived a set of predictions for RT behavior. Similar to the additive factor method, the cascade model shows that experimental factors affecting the rate of the same process will tend to interact, whereas those affecting the rates of different processes are generally additive. However, the predictions become more complicated and start to diverge from those of the additive factor method when at least one of the experimental factors affects the asymptotic level of activation.

The queue-series model recently developed by Miller (1993) assumes that the cognitive system is composed of a series of stages, and the stimulus is regarded as consisting of a number, M, of distinct components. The important concept of grain size of transmission is mathematically represented by the parameter M. Discrete stages and cascade flows are treated as special cases of the queue series, corresponding to the cases of $M = 1$ and $M = \infty$, respectively. Other positive values of M represent inter-

Mathematical Models of **RT** and Mental **Structure** Classified in terms of Discrete versus Continuous Information Transmission and Serial versus Network Architecture

(from Liu, 1996, “Queueing network modeling of elementary mental processes,” Psychological Review, 103(1), pp. 116-136).

Temporal Transmission	Architectural arrangement of mental processes: **Serial Stages**	Network Configurations
Discrete	**Subtractive** **Additive factors** **General Gamma**	
Continuous		

Mathematical Models of **RT** and Mental **Structure** Classified in terms of Discrete versus Continuous Information Transmission and Serial versus Network Architecture

(from Liu, 1996, “Queueing network modeling of elementary mental processes,” Psychological Review, 103(1), pp. 116-136).

Temporal Transmission	Architectural arrangement of mental processes: Serial Stages	Architectural arrangement of mental processes: **Network Configurations**
Discrete	Subtractive Additive factors General Gamma	**Critical Path Network (PERT)**
Continuous		

Mathematical Models of **RT** and Mental **Structure** Classified in terms of Discrete versus Continuous Information Transmission and Serial versus Network Architecture

(from Liu, 1996, "Queueing network modeling of elementary mental processes," Psychological Review, 103(1), pp. 116-136).

	Architectural arrangement of mental processes	
Temporal Transmission	**Serial Stages**	Network Configurations
Discrete	Subtractive Additive factors General Gamma	Critical Path Network (PERT)
Continuous	**Cascade** **Queueing series**	

Mathematical Models of **RT** and Mental **Structure** Classified in terms of Discrete versus Continuous Information Transmission and Serial versus Network Architecture

(from Liu, 1996, "Queueing network modeling of elementary mental processes," Psychological Review, 103(1), pp. 116-136).

Temporal Transmission	Architectural arrangement of mental processes: Serial Stages	Architectural arrangement of mental processes: **Network Configurations**
Discrete	Subtractive Additive factors General Gamma	Critical Path Network (PERT)
Continuous	Cascade Queueing series	**Queueing Network (QN)**

- **Non-unidirectional flow**
- **Non-selective influence**
- **By-passing**
- **Fixed capacity**

where $E[T_i]$ is the expected value of the remaining network sojourn time of a customer at the instant when it arrives at Node i of a network with K nodes.

Apparently, if a customer visits Node 1 to Node J successively, without skipping any node or visiting any node more than once, then we have $p_{ij} = 1$ for $i = 1$ to $J - 1$ and $j = 2$ to J, and $p_{ij} = 0$ for all other values of i and j. In this case, Equation 6 specializes to Equation 5. Lemoine (1987) also derived the recursive relations for computing the second moment of network sojourn times, which involves more unknown variables than the number of equations. In general, exact computations of the second or higher moments of sojourn times in a product-form network are not possible without additional information about some characteristics of the network.

RT as Network Sojourn Time

To link customer sojourn time with RT, queueing network models for RT assume that a response is made when the response unit has accumulated M of the N stimulus components (M and N are usually defined arbitrarily and can be made arbitrarily close to each other). This assumption is similar to that in the accumulator model (see, for example, Pachella, 1974) and in Miller's (1993) queue-series model. According to this assumption, total RT is the time interval between the instant of stimulus presentation and the instant at which the Mth stimulus component arrives at the response unit.

The models assume that an input node in a discrete network has the function of accumulating all the independently arrived M components and then transmitting them as an "assembled package" to internal nodes. Components that arrive later than the Mth component are not allowed to enter the network while the current "package" is being processed by the network. Thus, in a discrete network, all nodes operate in strict sequence without temporal overlap of node activities. In contrast, an input node in a continuous-transmission network, like all internal nodes, transmits each customer immediately after it has received and processed it. Therefore, all nodes could operate concurrently.

To use an observable natural phenomenon as an analogy, we can imagine a discrete network as a special type of highway transportation system in which shipping materials arrive at the highway entrance independently, are assembled into one big package there, and are then shipped through an otherwise empty highway (empty except for the only package). Similarly, a continuous network can be imagined as a "normal" highway, where shipping materials arrive at and pass through the entrance independently and travel through the network individually as in a traffic-flow situation. As is discussed below, in some special classes of networks (discrete PERT networks or continuous fork-join networks), the shipping materials may be disassembled into parcels after having entered the network. Each parcel may take a separate path of the network, and then they are reassembled at the destination.

For discrete networks, RT (denoted as RT_d) is apparently the sum of the time required by the input node to accumulate M components (T_1) and the time for the assembled "package" to traverse the network (T_d), that is,

$$RT_d = T_1 + T_d. \tag{7}$$

For Poisson arrivals, the time interval between the instant of stimulus presentation and the Mth arrival to the input node (T_1) follows the ordinary gamma distribution with parameters M and λ (Ross, 1983; Townsend & Ashby, 1983) and is independent of T_d.

If the structure of a continuous-transmission network does not permit components to overtake each other, then the Mth component to depart from the network is also the Mth to arrive at the input node from the outside. Apparently, RT in this case (denoted as RT_c) is the sum of the time interval between the instant of stimulus presentation and the Mth arrival (T_2) and the Mth customer's network sojourn time (T_c), that is,

$$RT_c = T_2 + T_c. \tag{8}$$

It is easy to see that $T_1 = T_2$, because both can be described as the same ordinary gamma distribution with parameters (M, λ). For values of M that are not too small, this distribution approximates a normal distribution. Several authors have shown that the convolution of a normal and an exponential distribution provides a close approximation to experimental data (Ashby, 1982b; Hockley, 1984; Hohle, 1965; Ratcliff & Murdock, 1976). This finding may be borrowed as a tentative support of the role of T_1 and T_2.

Because $T_1 = T_2$ and they are independent of T_d and T_c, respectively, to compare the RT behavior of a discrete and a corresponding continuous network (RT_d and RT_c), it suffices to compare T_d and T_c. To continue to use the highway system analogy, Equations 7 and 8 tell us that to compare the time needed for shipping M pieces of materials in a discrete network (RT_d) and that in its continuous counterpart (RT_c), what is needed is to compare the sojourn time of a large package containing the M components in an empty network (T_d) with the sojourn time of one of the components (the Mth one to arrive at the network) in a crowded network (T_c).

With these general descriptions and assumptions at hand, I am ready to examine RT behavior in several interesting classes of queueing networks. I first compare T_c and T_d in the simplest network, called *series queues,* in which nodes are arranged in sequence. Then I examine fork-join queues to show that they include PERT networks as special cases. I also compare T_c and T_d in a simple feedback queueing system. In the last two sections, I discuss the characteristics of T_c in two classes of continuous-transmission queueing networks, the first of which allows one class of stimulus components to overtake another class of components; the second allows only a fixed number of stimulus components to exist in the system.

Series Queues as a Model for RT

Network Sojourn Time in Series Queues

The simplest type of queueing networks is series queues, also called *tandem queues,* in which the service stations form a series system with flows always in a single direction from the first node to the last node. As shown in Figure 1, customers may enter

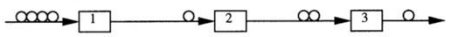


Figure 1. A series queueing network.

forked and joined in the same way as in the corresponding PERT network. A response is initiated when M components have departed the system. Similar to the acyclic characteristic of PERT networks, fork-join networks do not allow a customer to visit the same node more than once, and they are often called *acyclic fork-join queueing networks* (AFJQNs) in the literature.

The simplest instance of a nontrivial fork-join network is a parallel network consisting of a number, K, of parallel queueing systems, as shown in Figure 2. Customers arrive at the fork node as a Poisson flow with mean arrival rate λ and, on arrival, a customer forks into K offsprings. The ith offspring is assigned to the ith queueing system that consists of a single-channel FCFS service node and an infinite capacity queue. The service times of the nodes are independent and exponentially distributed with mean $1/\mu_i$ for Node i. A customer leaves the system as soon as all its K offsprings have completed their service and are merged at the join node. The network sojourn time, T_c, of any arbitrarily selected customer is the maximum of the sojourn times of its K offsprings, that is,

$$T_c = \max(T_1, T_2, \ldots, T_K), \tag{20}$$

where $T_j = S_j + W_j$ is the sojourn time of the jth offspring of the customer at Queue j ($j = 1, \ldots, K$), including both service time (S_j) and waiting time (W_j).

In the extreme case in which only one customer is allowed to enter the system through the fork node, we have a corresponding parallel PERT network. All the offspring of the admitted customer are processed immediately at the servers, and thus the T_js include only service times (i.e., $T_j = S_j$, for all j), which are usually referred to as *process durations* in PERT terms and are commonly assumed to be independent random variables. The problem of determining customer sojourn time in this discrete network, T_d, is that of finding the maximum of K independent random variables (referred to as *determining the length of the critical path*). Unfortunately, the problem becomes more difficult for fork-join networks, because the continuous nature of customer arrival makes it necessary to consider the queueing effects at the K service nodes. T_js are no longer service times but sojourn times—the sum of service times and waiting times. Determining the network sojourn time, T_c, becomes that of finding the maximum of K random variables that are not necessarily independent of each other.

Recently, Nelson and Tantawi (1988) proved that the T_js in this simple parallel fork-join system, $j = 1, \ldots, K$, are associated random variables. Random variables $T_1, T_2, \ldots, T_K$ are said to be associated if $\operatorname{cov}[f(T_1, T_2, \ldots, T_K), g(T_1, T_2, \ldots, T_K)] \geq 0$ for all pairs of nondecreasing functions f and g. The properties of associated random variables that are relevant to the present discussion are: All independent random variables are associated, but associated variables are not necessarily independent. If $T_1, T_2, \ldots, T_K$ are associated, then

$$P[\max_{1 \leq i \leq K} T_i > t] \leq 1 - \prod_{i=1}^{K} P[T_i \leq t], \tag{21}$$

and the expected value has an upper bound expressed as

$$E[\max_{1 \leq i \leq K} T_i < t] \leq \int_0^\infty (1 - \prod_{i=1}^{K} P[T_i \leq t])\,dt. \tag{22}$$

In Equations 21 and 22, equality holds if and only if $T_1, T_2, \ldots, T_K$ are independent. For a parallel system consisting of exponential servers, Equation 22 specializes to

$$E[T_c] = E[\max_{1 \leq i \leq K} T_i < t] \leq \int_0^\infty (1 - \prod_{i=1}^{K} (1 - e^{-(\mu_i - \lambda)t})\,dt. \tag{23}$$

In the case of K identical servers, Equation 23 becomes

$$E[T_c] = E[\max_{1 \leq i \leq K} T_i < t] \leq \int_0^\infty [1 - (1 - e^{-[\mu - \lambda]t})^K]\,dt$$

$$= \frac{1}{\mu - \lambda} H_K, \tag{24}$$

where

$$H_K \text{ is the harmonic series: } H_K = \sum_{i=1}^{K} \frac{1}{i}. \tag{25}$$

As Nelson and Tantawi (1988) pointed out, the lower bound for $E[T_c]$ is obtained by ignoring queueing effects. Let $\lambda = 0$; we have:

$$\frac{1}{\mu} H_K \leq E[T_c] \leq \frac{1}{\mu - \lambda} H_K. \tag{26}$$

Baccelli and Makowski (1990) later generalized this result to include customer arrivals that are not necessarily Poisson and showed that as long as the parallel servers are identical and exponential, the following expression holds:

$$\frac{1}{a} H_K \leq E[T_c] \leq \frac{1}{b} H_K, \tag{27}$$

where a and b are uniquely determined by the rate of exponential service (μ) and the probability distribution of customer arrival. For Poisson arrivals, $a = \mu$.

As pointed out by Baccelli and Makowski (1990), because both bounds grow at the same rate, H_K, $E[T_c]$ itself must grow at the same rate. An interesting property of the harmonic series is that H_K approximates $\log K$ for large K, which implies that mean customer sojourn time grows logarithmically in the number of parallel servers.

Because the extreme case of $\lambda = 0$ corresponds to a PERT

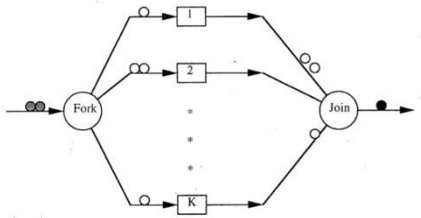


Figure 2. A parallel fork-join queueing network.

this article has raised more questions than it has answered. Future researchers may find it worthwhile to examine further the equivalence and the identifiability of the two classes of networks.

A Single-Server Feedback Queueing System

In queueing network literature the network in Figure 3 is called a *single server queueing system with instantaneous Bernoulli feedback*. Customers arrive at the system in accordance with a Poisson process with mean arrival rate of γ. The server is a single-channel FCFS exponential server with rate parameter μ and an infinite queue capacity. After receiving service, each customer may immediately return to the end of the queue in front of the server with probability p or depart the system with probability $q = 1 - p$. The feedback probability is independent of the state of the system. It should be noted here that use of the term *feedback* in queueing network literature does not correspond well to the standard notion of feedback in psychological modeling. In psychological modeling, *feedback* generally refers to a reverse influence of a later process on an earlier process, rather than a need to repeat a process. However, for lack of a better term to describe the queueing network in Figure 3, and to be consistent with standard queueing network literature, I continue to use *feedback queueing system* to describe this type of queueing system.

Because this queueing system satisfies all three Jacksonian assumptions for product-form networks, it belongs to the class of product-form networks—the joint probability distribution of the number of customers being in their first, second, . . . , and Kth loop has a product form. However, this feedback system is not overtake-free, and the order of customer arrival is not preserved in the order of their departure from the system. Because customers may overtake each other while traversing the system, the sojourn time of an arbitrary customer is influenced not only by the number of customers (and their remaining service requests) found on its arrival, but also by later arrivals. Thus, the sojourn times of a customer's successive visits at the server are not independent of each other.

Takacs (1963) was the first to examine this feedback system and derived an exact expression for the mean network sojourn time of a customer, $E[T_c]$, as follows:

$$E[T_c] = \frac{1}{q\mu - \gamma}. \tag{32}$$

This expression also can be derived from Equation 6 directly as follows (Lemoine, 1987). Because each customer is expected to visit the server $1/q$ times, the total arrival rate at the server, λ, is γ/q (including both external and feedback arrivals). Therefore, according to Equation 6, the expected network sojourn time can be computed as:

$$E[T] = \frac{1}{q}\frac{1}{\mu - \frac{\gamma}{q}} = \frac{1}{q\mu - \gamma}.$$

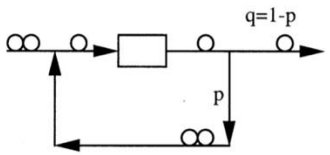


Figure 3. A queueing system with instantaneous Bernoulli feedback.

As described earlier, RT is characterized by the time for the response unit to accumulate M components. Of great interest to RT modeling, therefore, is the sojourn time of the customer who is the Mth to depart from the system. However, queueing network research investigates the issue of customer sojourn times from the perspective of arrivals: How long does it take for the Mth arrival rather than the Mth departure to traverse the system? This difference in perspective does not a pose a problem when a network is overtake-free, as in the case of series queues and fork-join networks discussed thus far, because the Mth departure is also the Mth arrival. This simple relation between arrival and departure does not hold for the feedback queueing system, however, because of customer overtaking. An important result in this regard is that of Whitt (1984), who proved that in this feedback system the expected number of customers that overtake a particular customer is the same as the expected number of customers that are overtaken by this customer. This result implies that although on a particular trial of an RT experiment the Mth stimulus component to arrive at the system is not necessarily the Mth to depart, over a large number of repeated trials the Mth arrival is still expected to be the Mth to depart. Therefore, in a typical RT experiment involving a large number of trials, the network sojourn time of the arbitrarily selected Mth arrival as expressed in Equation 32 can still be used to infer the RT behavior of this feedback queueing system.

What is particularly interesting about Equation 32 to RT modeling is that it tells us that, at the level of the mean RT, this feedback system is able to mimic a serial system with N identical exponential servers accurately. In Equation 32, if we let $q = 1/N$ and $\gamma = \lambda/N$, with N take integer values, we have

$$E[T] = \frac{1}{q\mu - \gamma} = \frac{N}{\mu - \lambda} = \sum_{i=1}^{N} \frac{1}{\mu - \lambda} = kN. \tag{33}$$

Clearly, $E[T] = \sum_{i=1}^{N} 1/(\mu - \lambda)$ in Equation 33 is the expression for the expected sojourn time of a customer in a system of a series of N identical exponential servers with parameters μ and λ for each server. Furthermore, $E[T] = kN$ in Equation 33 tells us that the detection of a linear relationship between mean RT and N in a set of RT data is not sufficient to distinguish whether the underlying mental system is consisted of a sequence of N identical servers or of a single server with departure probability of $\frac{1}{N}$.

In psychological experiments a linear relationship between mean RT and a discrete independent variable has traditionally been interpreted as evidence in support of a serial-stage model.

A classic example is Sternberg's (1969) memory scanning task, in which subjects are asked to remember a list of items (called *positive set*) and then to make a yes–no type of binary response about whether a displayed item is a member of the positive set. A large number of studies using this experimental paradigm have shown a robust linear relationship between mean RT and the size of the positive set. The slope of this linear relation is interpreted as the duration of a new stage inserted in the processing chain when the size of the positive set increases by 1. Townsend (1974) showed that a system of identical and independent parallel processes could predict the same linear relationship with arbitrary slope k by assuming that the processing rate of the parallel processes decreases as the number of items increases. More specifically, Townsend showed that because

$$E[T] = \frac{1}{\mu}\sum_{i=1}^{N}\frac{1}{i}$$

in parallel systems, $E[T] = kN$ could be obtained if

$$\mu = \frac{\sum_{i=1}^{N}\frac{1}{i}}{kN}.$$

Using the results discussed earlier in this section about fork-join networks, it is easy to see that a corresponding continuous-transmission parallel fork-join network could predict the same linear relation equally well. As Townsend pointed out, this interpretation based on parallel systems is not necessarily intuitive or natural, particularly considering the complicated relation between μ and N.

The single-node feedback system offers another plausible explanation of the linear RT relation. The effect of adding a new item to the positive set may very well be that of decreasing the departure probability q in this feedback system rather than that of inserting an additional stage. Because q is related to set size N in a simple reciprocal relation ($q = 1/N$), this interpretation does not appear to be unnatural. A single-node system with a feedback loop appears to be perhaps more parsimonious as a model for RT than a chain of N nodes, particularly when N is large.

If the feedback system is a discrete system, then it seems impossible to distinguish, even at the distributional level, whether a process visits the same node N times or visits N identical nodes in series, because both can be characterized by the same ordinary gamma distribution with parameter (N, μ). However, for continuous-transmission systems, the two classes of systems dissociate in their predictions of sojourn time at levels higher than the means. Takacs (1963) derived an exact expression for the Laplace–Stieltjes transform of the network sojourn time distribution, and its form is far more complex than that of the ordinary gamma distributions. Furthermore, the existence of customer overtaking in the feedback system tends to produce a greater variance in network sojourn time than the series system (Lemoine, 1987; Takacs, 1963). Therefore, detection of large RT variances in conjunction with a linear mean RT relationship appears to be evidence in favor of the continuous-transmission feedback model over a series model, although not necessarily definitive evidence. The larger the RT variances detected in conjunction with a linear mean RT, the more likely the underlying mental system is a continuous-transmission system.

Simon–Foley Network and Overadditive Factor Interactions

If a product-form network does not allow customers to overtake each other, then sojourn times of a customer at successive nodes are mutually independent and exponentially distributed random variables, and network sojourn time can be described as general-gamma distributions, which have been shown to play a central role in McGill and Gibbon's (1965) model, the cascade model, and the series queueing model. In this section I discuss a classic example of a non-overtake-free network, shown in Figure 4. This network is often referred to as a *Simon–Foley network,* which is a three-node product-form network with a single server at each node (Simon & Foley, 1979). Customers enter the system only at Node 1 and exit the system at Node 3. After visiting Node 1, a customer goes directly to Node 3 with probability $(1 - p)$, or goes to Node 2 and Node 3 in sequence with probability p. We may also think of this system as having two types of customers: Type 1 customers take the indirect route, whereas Type 2 customers take the direct route. The two types of customers have identical service requirements and priority level at the nodes they visit, and the value of p decides the proportion of Type 1 customers in the total customer population.

This network has an interesting property: The sojourn time in the first and the third queues (T_1 and T_3) are not independent for customers who go through the second queue, but they are independent for customers who go directly from Node 1 to Node 3. T_1 and T_2 are independent. T_2 and T_3 also are independent. Recent research has shown that T_3 is stochastically increasing in T_1 for a customer that goes through Node 2, that is, $P\{T_3 > t | T_1\}$ is increasing in T_1 (Foley & Kiessler, 1989). A result that is particularly useful for mean sojourn time analysis was derived by Walrand and Varaiya (1980), who showed that the expected value of T_3 increases as T_1 increases. That is,

$$E[T_3|T_1 = t'] > E[T_3|T_1 = t], \quad t' > t > 0, \qquad (34)$$

where $E[T]$ represents the mean of T.

This relationship has a quite intuitive interpretation. Let C be a customer who goes to Queue 3 via Queue 2 after leaving Queue 1. Some customers who arrived and departed from Queue 1 later than C may arrive at Queue 3 before C arrives there because they took a direct route. The longer C spent at Queue 1, the more likely it is that C had left a long queue waiting behind it there, and the more likely that many of these late arrivals would have arrived at Queue 3 earlier than C because they took the direct route. Thus, the longer C stayed at Queue 1, the

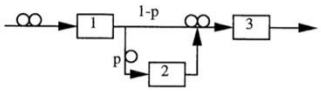


Figure 4. A Simon–Foley network in which noise components may overtake signal components.

point is the result reported by Miller (1976). Miller found that degrading by dots produced an overadditive interaction with the probability of stimulus occurrence, but degrading by contrast reduction had additive effects with the probability manipulation. In terms of the queueing network model, this result could be because degrading by dots creates "noise" customers that overtake signal components, whereas degrading by contrast reduction does not. Therefore, only degrading by dots will destroy the independence of T_1 and T_3 and produce an overadditive interaction between degrading by dots and occurrence probability.

The above discussion has suggested a plausible explanation to overadditive interactions discovered in psychological experiments such as the lexical decision tasks, in addition to that offered by Sternberg (1969) based on serial discrete stages and that by McClelland (1979) based on cascade processes. It should be noted here that the presence of an overadditive interaction does not confirm the presence of a network shown in Figure 4, just as it does not exclusively confirm Sternberg's or McClelland's hypotheses.

For the purpose of detecting the presence of a network arrangement of Figure 4, there does exist a test that is stronger than detecting the presence of interactions. This test is based on Equation 34 and is provided by directly measuring the durations of T_3 and T_1 if related measurement methods are assumed to be available. This assumption is not stronger than those for testing the validity of Schweikert's (1978) PERT methodology for RT analysis, which assumes that we are able to prolong the duration of a process of interest, and

> we may be able to record time at several points in the network. We may know the times at which various stimuli are presented and responses made, and we may also know the times at which various physiological events occur. (Schweikert, 1978, p. 123)

According to Equation 34, if in a task situation in which prolonging a process produces a corresponding increase in the duration of another process, but not vice versa, then there is a great possibility that the task situation involves a continuous network of mental processes shown in Figure 4, particularly if such a network also " 'makes sense' in terms of other knowledge" (Sternberg, 1969, p. 283).

It should be noted here that this relationship between T_1 and T_3 is different from the type of possible correlation of stage durations induced by factors such as motivation or preparation. Several authors (e.g., Ashby & Townsend, 1980; Sternberg, 1969; Townsend & Ashby, 1983) have pointed out that a subject-controlled factor (such as preparation or motivation) that either varies from trial to trial or is controlled by experimental manipulations such as reward magnitudes would induce a correlation of stage durations. For example, stage durations could both be short (or long) when the motivation is high (or low). This type of correlation would not destroy the additivity of factors that influence the two stages separately, because for any given level of the subject-controlled factor, stage durations would still be independent. For the Simon–Foley network, experimental manipulations that increase T_1 would be expected to produce a corresponding increase in T_3, but not vice versa. The dependence of T_3 on T_1 is not under the subject's control.

Closed Queueing Networks and Underadditive Factor Interactions

This section considers a special class of queueing networks that predict underadditive factor effects when they are used as models for RT. This class of networks are called *closed queueing networks.* A closed network can be viewed as an open network with a fixed capacity—the total number of customers that are allowed in the network is held fixed. All the networks discussed thus far in this article are open queueing networks, which are useful for modeling cognitive systems that have not reached full capacity. If, for some psychological tasks, the cognitive system has an upper limit in terms of the number of stimulus components or task components that it could hold in queues at once, then a closed network would appear to be an appropriate candidate for modeling the cognitive system when it functions at full capacity. As mentioned earlier, queueing network models for RT assume that a response is made when the response unit has accumulated M of the N stimulus components. For closed queueing networks, we assume $M < N$, which means that the tasks require subjects to process many or most, but not all, of the stimulus components by the time a response is made. This assumption is similar to that in existing models, such as the accumulator model (see, for example, Pachella, 1974). For closed queueing networks this assumption ensures that a network operating at full capacity would not lose its full-capacity status before a response is made.

A closed network could also constitute part of a larger network, which may be open or closed. For example, the virtually unlimited capacity of iconic storage could be modeled as an infinite-capacity server that is located in front of a closed network. The "iconic" server and the total system as a whole are open to the outside, but a portion of the system represented as a closed network is "closed" because of its limited capacity. This type of system is often called a *mixed* network or a *hybrid* network. In queueing network literature, the term *closed network* also is used to refer to a network in which the same customers circulate eternally through the network. This notion of a closed network does not appear to be very relevant for the purpose of RT modeling, however.

The simplest type of a closed queueing network is called a *cyclic queue,* which is essentially a series queue with a fixed number, C, of customers allowed in the system. The C customers can also be viewed as C "containers," each of which is able to carry one customer. When a customer departs from the last node, the container that carried it becomes empty and is immediately cycled back to the front of the system to admit a new customer. To simplify the discussion, let us consider the simplest cyclic queue with two nodes or stages, as shown in Figure 5 ($K =$

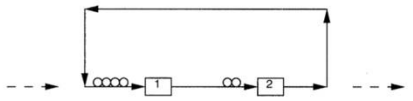


Figure 5. A cyclic queueing network that allows a fixed number of stimulus components to exist in the system.

Queueing Network (QN)
Models of Human Behavior (MHB)
QN-MHB

1. RT: Reaction Time (QN-RT) and Mental Structure
2. RT and Accuracy (QN-RMD) (Mental Structure vs State of Mind)
3. Procedural Tasks (QN-MHP or QN-MHP-BE)
4. Complex Cognition Tasks (QN-ACTR)
5. Visual Attention tasks (QN-NSEEV) (QN-RLEM)
6. Manual or Continuous Control tasks (QN-Control: Classical/Modern)
7. Basic Body Motion tasks (QN-MTM)
8. Mind-Body System (QN-MBS)
9. Neural level (QN-Neural: Indexes and Neural Networks)
10. Nervous and Endocrine Systems (QN-NES)
11. Multi-Person Multi-Machine QN (QN-HMN)
12. Engineering Applications in various domains

Note: relations with **Task Network** Methods/Tools such as Micro Saint #, IMPRINT)

Mathematical Models of RT and Mental Structure Classified in terms of Discrete versus Continuous Information Transmission and Serial versus Network Architecture

Mathematical Models of RT and Response Accuracy (sequential sampling models)

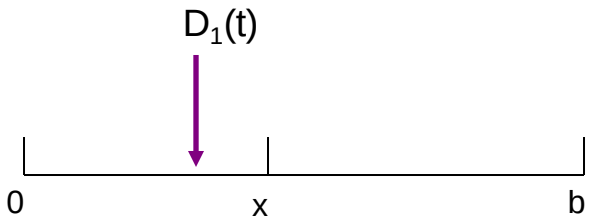
(from Liu, 1996, "Queueing network modeling of elementary mental processes," Psychological Review, 103(1), pp. 116-136).

Temporal Transmission	Architectural arrangement of mental processes: Serial Stages	Architectural arrangement of mental processes: Network Configurations
Discrete	Subtractive Additive factors General Gamma	Critical Path Network (PERT)
Continuous	Cascade Queueing series	Queueing Network (QN)

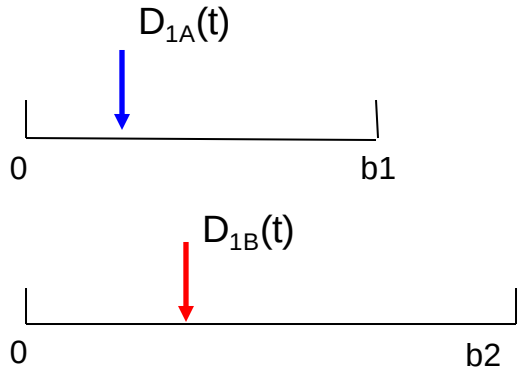
- Non-unidirectional flow
- Non-selective influence
- By-passing
- Fixed capacity

Counter/accumulator
Random-walk

Accumulator
Diffusion



1. Random-walk/Brownian-Motion/Diffusion Model



2. Counter/Accumulator Model

Diffusion and Accumulator Models of Speed-Accuracy Tradeoff (All 1-D)

Mathematical Models of RT and Mental Structure Classified in terms of Discrete versus Continuous Information Transmission and Serial versus Network Architecture

Mathematical Models of RT and Response Accuracy (sequential sampling models)

(from Liu, 1996, "Queueing network modeling of elementary mental processes," Psychological Review, 103(1), pp. 116-136).

Temporal Transmission	Architectural arrangement of mental processes: Serial Stages	Architectural arrangement of mental processes: Network Configurations
Discrete	Subtractive Additive factors General Gamma	Critical Path Network (PERT)
Continuous	Cascade Queueing series	Queueing Network (QN)

- Non-unidirectional flow
- Non-selective influence
- By-passing
- Fixed capacity

Counter/accumulator
Random-walk

Accumulator
Diffusion

Mathematical Models of RT and Mental Structure Classified in terms of Discrete versus Continuous Information Transmission and Serial versus Network Architecture

Mathematical Models of RT and Response Accuracy (sequential sampling models)

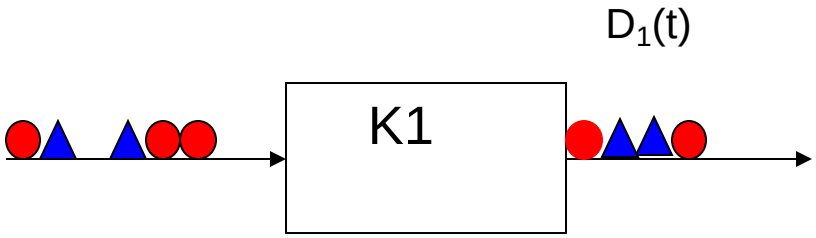
(from Liu, 1996, "Queueing network modeling of elementary mental processes," Psychological Review, 103(1), pp. 116-136).

	Architectural arrangement of mental processes		**State** transition
Temporal Transmission	Serial Stages	Network Configurations	
Discrete	Subtractive Additive factors General Gamma	Critical Path Network (PERT)	Counter/accumulator Random-walk
Continuous	Cascade Queueing series	**Queueing Network (QN)**	Accumulator Diffusion **Reflected Multidimensional Diffusions (RMD)**

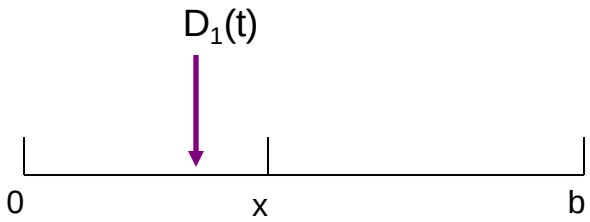
- **Non-unidirectional flow**
- **Non-selective influence**
- **By-passing**
- **Fixed capacity**

Queueing Network Mental Architecture, Response Time, and Response Accuracy:

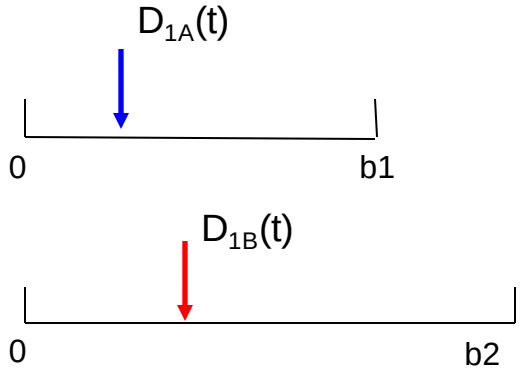
Reflected Multidimensional Diffusions



a). A single server system with two types of customers: type A (“triangles”) and type B (“circles”)

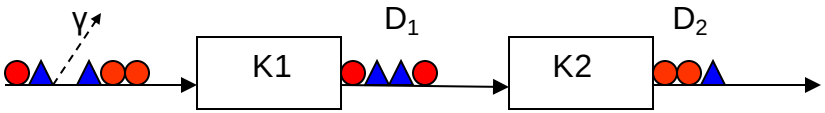


b). One 1-d diffusion with two absorbing barriers

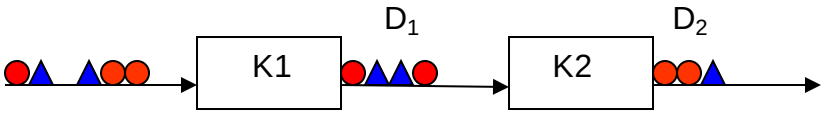


c). Two 1-d diffusions, each with one absorbing and one reflecting barrier

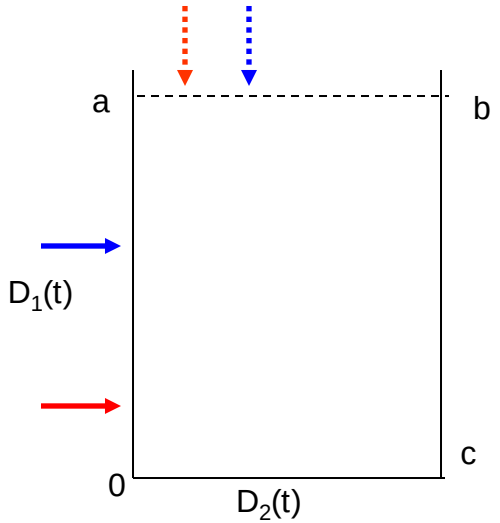
Figure 2: A single server queueing system and its alternative 1-d diffusion representations



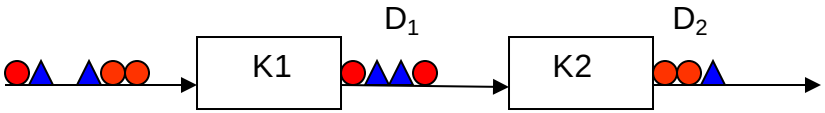
a). A tandem two-server system with two types of customers: type A (“triangles”) and type B (“circles”)



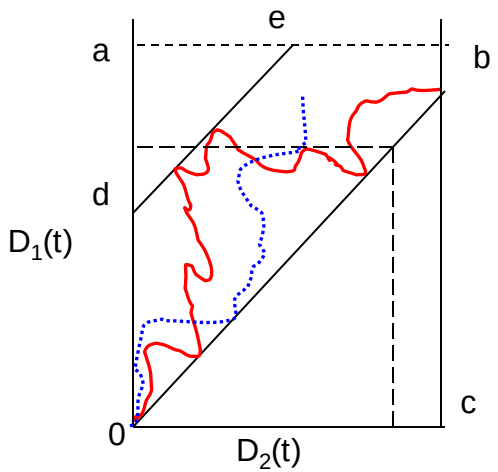
a). A tandem two-server system with two types of customers: type A ("triangles") and type B ("circles")



b). 2-D diffusion representation of $\{D_1(t), D_2(t)\}$ in a discrete tandem queue



a). A tandem two-server system with two types of customers: type A (“triangles”) and type B (“circles”)



b). 2-D diffusion representation of {$D_1(t)$, $D_2(t)$} in a continuous tandem queue

Queueing Network (QN)
Models of Human Behavior (MHB)
QN-MHB

1. RT: Reaction Time (QN-RT) and Mental Structure
2. RT and Accuracy (QN-RMD) (Mental Structure vs State of Mind)
3. Procedural Tasks (QN-MHP or QN-MHP-BE)
4. Complex Cognition Tasks (QN-ACTR)
5. Visual Attention tasks (QN-NSEEV) (QN-RLEM)
6. Manual or Continuous Control tasks (QN-Control: Classical/Modern)
7. Basic Body Motion tasks (QN-MTM)
8. Mind-Body System (QN-MBS)
9. Neural level (QN-Neural: Indexes and Neural Networks)
10. Nervous and Endocrine Systems (QN-NES)
11. Multi-Person Multi-Machine QN (QN-HMN)
12. Engineering Applications in various domains

Note: relations with **Task Network** Methods/Tools such as Micro Saint #, IMPRINT)

Queueing Network-Model Human Processor (QN-MHP): A Computational Architecture for Multitask Performance in Human-Machine Systems

YILI LIU, ROBERT FEYEN, and OMER TSIMHONI
University of Michigan

Queueing Network-Model Human Processor (QN-MHP) is a computational architecture that integrates two complementary approaches to cognitive modeling: the queueing network approach and the symbolic approach (exemplified by the MHP/GOMS family of models, ACT-R, EPIC, and SOAR). Queueing networks are particularly suited for modeling parallel activities and complex structures. Symbolic models have particular strength in generating a person's actions in specific task situations. By integrating the two approaches, QN-MHP offers an architecture for mathematical modeling and real-time generation of concurrent activities in a truly concurrent manner. QN-MHP expands the three discrete serial stages of MHP, of perceptual, cognitive, and motor processing, into three continuous-transmission subnetworks of servers, each performing distinct psychological functions specified with a GOMS-style language. Multitask performance emerges as the behavior of multiple streams of information flowing through a network, with no need to devise complex, task-specific procedures to either interleave production rules into a serial program (ACT-R), or for an executive process to interactively control task processes (EPIC). Using QN-MHP, a driver performance model was created and interfaced with a driving simulator to perform a vehicle steering, and a map reading task concurrently and in real time. The performance data of the model are similar to human subjects performing the same tasks.

Categories and Subject Descriptors: H.1.2 [**Models and Principles**]: User/Machine Systems—*Human information processing, human factors*; I.6.5 [**Simulation and Modeling**]: Model Development—*Modeling methodologies*

General Terms: Human Factors

Additional Key Words and Phrases: Cognitive model, human-computer interaction, cognition, user interfaces, human information processing

R. Feyen is currently at School of Industrial Engineering, Purdue University.
Authors' address: Y. Liu, O. Tsimhoni, Department of Industrial and Operations Engineering, University of Michigan, Ann Arbor, MI 48109; email: {yililiu,omert}@umich.edu; R. Feyen, School of Industrial Engineering, Purdue University, West Lafayette, IN 47907; email: rfeyen@purdue.edu.

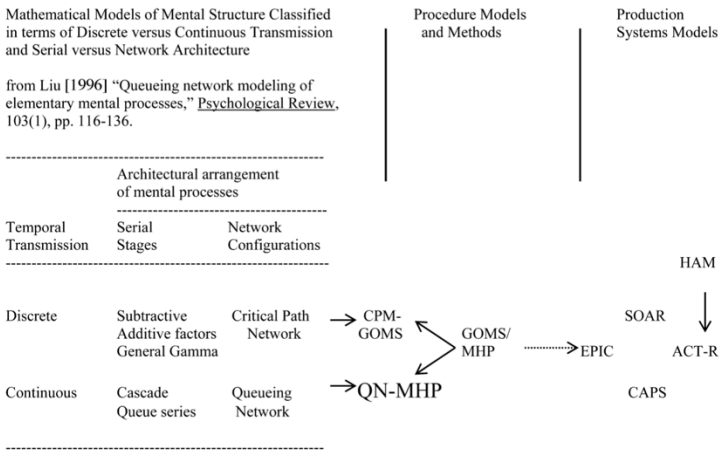
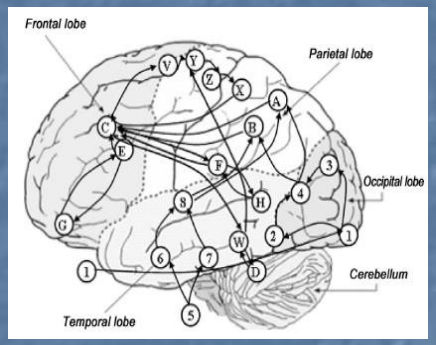
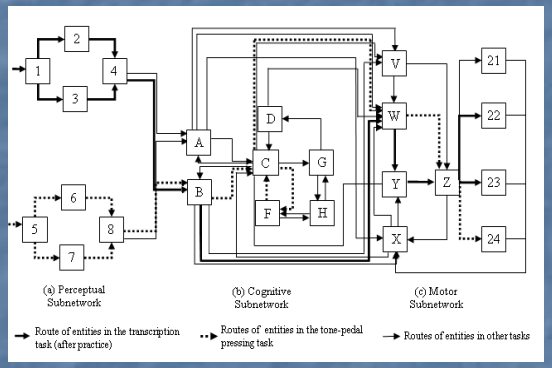


Fig. 1. Mathematical models of mental structure and procedure/production system models of cognitive architecture.

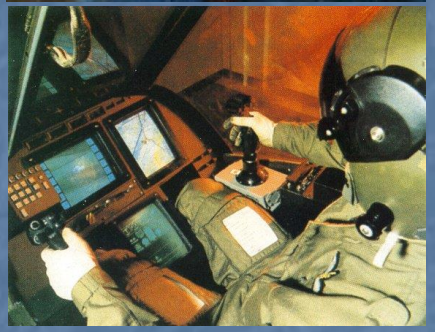
existing approaches. More specifically, we describe our current and proposed work in developing a complementary modeling approach that integrates the modeling philosophy and methods of the procedure-knowledge/production-systems models listed above and the mathematical/simulation theories and methods of queueing networks. Research on queueing networks is not only a major branch of mathematics and operations research but also one of the most commonly used methods for performance analysis of a large variety of real-world systems such as computer, communications, manufacturing, and transportation networks (e.g., Disney and Konig [1985]; Denning and Buzen [1978]; Boxma and Daduna [1990]). A large knowledge base on queueing networks exists, and some well-developed simulation and analysis software programs are widely used by engineers world-wide. Furthermore, from the psychological modeling perspective, as published in a *Psychological Review* article entitled "Queueing network modeling of elementary mental processes" [Liu 1996], we have successfully used queueing networks to integrate a large number of influential mathematical models of mental structure and psychological processes, such as Sternberg's serial stages model [Sternberg 1969], McClelland's cascade model [McClelland 1979], and Schweickert's critical path network model [Schweickert 1978] (see the left-half of Figure 1). From the systems engineering perspective, we have successfully used queueing networks to integrate single-channel one-server queueing models (e.g., Senders [1964]; Rouse [1980]) and the parallel processing models (e.g., Laughery [1989]; Wickens and Liu [1988]) as special cases [Liu 1994, 1997].

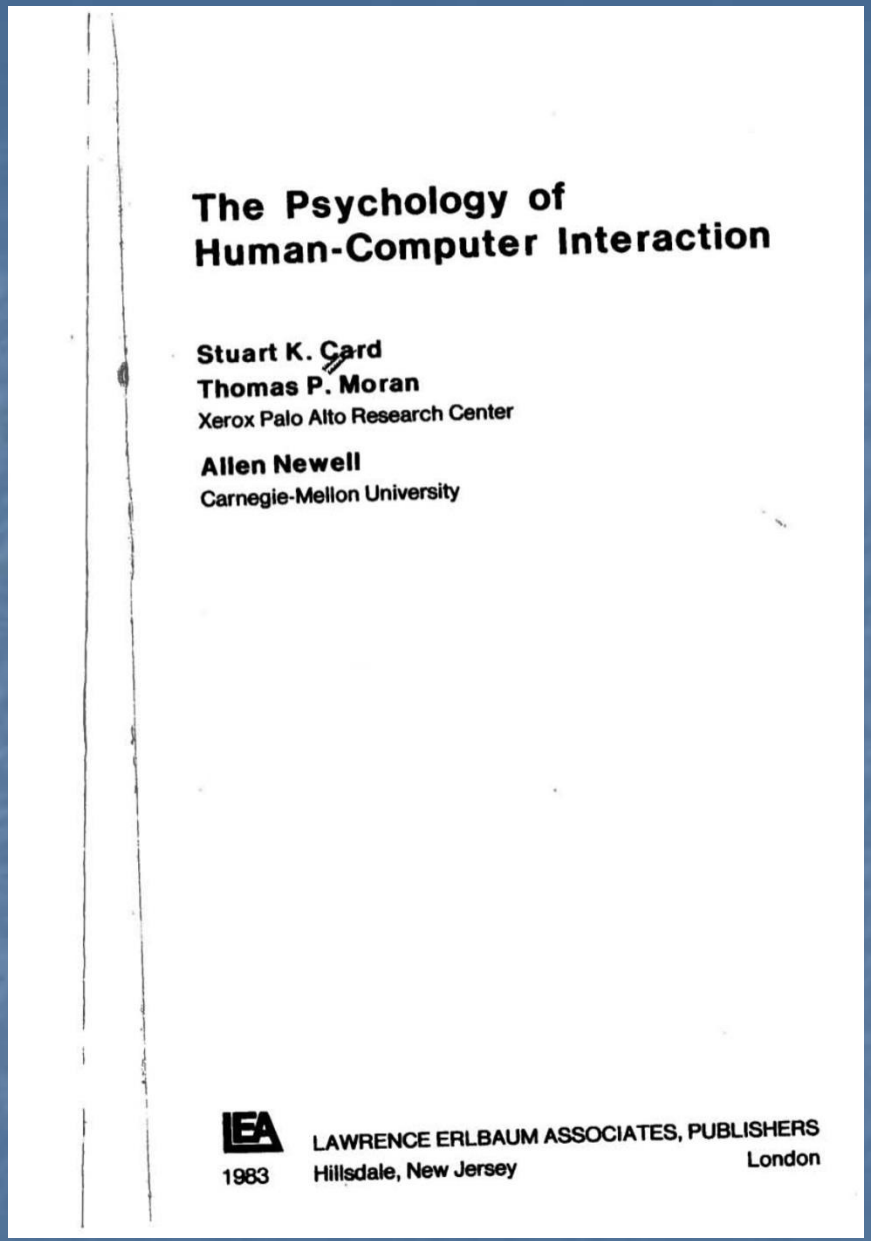
Queueing Network Modeling of Cognitive Architecture and Human-Machine Systems





Queueing Network of Mental Architecture




The Psychology of
Human-Computer Interaction
Stuart K. Card
Thomas P. Moran
Xerox Palo Alto Research Center
Allen Newell
Carnegie-Mellon University
LEA
LAWRENCE ERLBAUM ASSOCIATES, PUBLISHERS
1983
Hillsdale, New Jersey
London

2. The Human Information-Processor

Our purpose in this chapter is to convey a version of the existing psychological science base in a form suitable for analyzing human-computer interaction. To be practical to use and easy to grasp, the description must necessarily be an oversimplification of the complex and untidy state of present knowledge. Many current results are robust, but second-order phenomena are almost always known that reveal an underlying complexity; and alternative explanations usually exist for specific effects. An uncontroversial presentation in these circumstances would consist largely of purely experimental results. Such an approach would not only abandon the possibility of calculating parameters of human performance from the analysis of a task, but would also fail in the primary purpose of giving the reader knowledge in a form relatively easy to assimilate.

Our tack, therefore, is to organize the discussion around a specific, simple model. Though limited, this model allows us to give, insofar as possible, an integrated description of psychological knowledge about human performance as it is relevant to human-computer interaction.

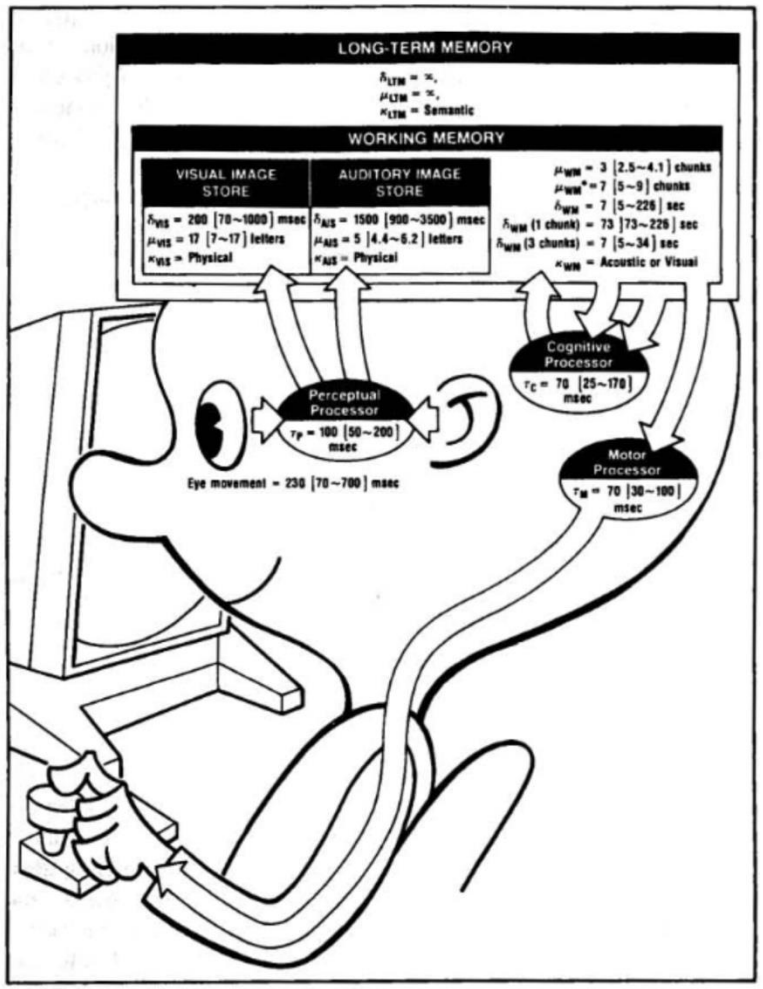


Figure 2.1. The Model Human Processor—memories and processors.

Sensory information flows into Working Memory through the Perceptual Processor. Working Memory consists of activated chunks in Long-Term Memory. The basic principle of operation of the Model Human Processor is the *Recognize-Act Cycle of the Cognitive Processor* (P0 in Figure 2.2). The Motor Processor is set in motion through activation of chunks in Working Memory.

Rate at which an item can be matched against Working Memory:		
Digits	33 [27~39] msec/item	Cavanaugh (1972)
Colors	38 msec/item	Cavanaugh (1972)
Letters	40 [24~65] msec/item	Cavanaugh (1972)
Words	47 [36~52] msec/item	Cavanaugh (1972)
Geometrical shapes	50 msec/item	Cavanaugh (1972)
Random forms	68 [42~93] msec/item	Cavanaugh (1972)
Nonsense syllables	73 msec/item	Cavanaugh (1972)
Range =	27~93 msec/item	
Rate at which four or fewer objects can be counted:		
Dot patterns	46 msec/item	Chi & Klahr (1975)
3-D shapes	94 [40~172] msec/item	Akin and Chase (1978)
Range =	40~172 msec/item	
Perceptual judgement:		
	92 msec/inspection	Welford (1973)
Choice reaction time:		
	92 msec/inspection	Welford (1973)
	153 msec/bit	Hyman (1953)
Silent counting rate:		
	167 msec/digit	Landauer (1962)

Figure 2.7. Cognitive processing rates.
Selected cycle times (msec/cycle) that might be identified with the Cognitive Processor cycle time.

principles of operation. The memories and processors are grouped into three main subsystems: a perceptual system, a cognitive system, and a motor system. The most salient characteristics of the memories and processors can be summarized by the values of a few parameters: processor cycle time τ, memory capacity μ, memory decay rate δ, and

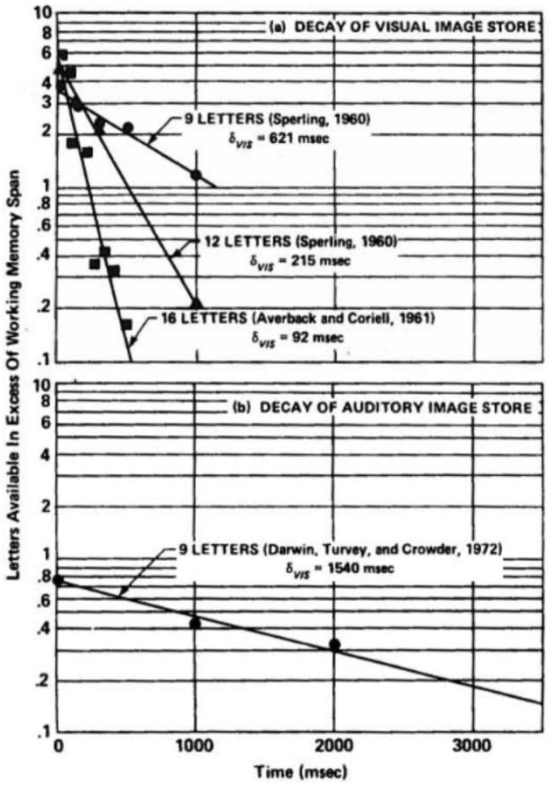


Figure 2.3. Time decay of Visual and Auditory Image Stores.
(a) Decay of the Visual Image Store. In each experiment, a matrix of letters was made observable tachistoscopically for 50 msec. In the case of the Sperling experiments, a tone sounded after the offset of the letters to indicate which row should be recalled. In the case of the Averbach and Coriell experiment, a bar appeared after the offset of the letters next to the letter to be identified. The percentage of indicated letters that could be recalled eventually asymptotes to μ_{WM}^*. The graph plots the percentage of letters reported correctly in excess of μ_{WM}^* as a function of time before the indicator.
(b) Decay of the Auditory Image Store. Nine letters were played to the observers over stereo earphones arranged so that three sequences of letters appear to come from each of three directions. A light lit after the offset of the letters to indicate which sequence should be recalled. The graph plots the percentage of the relevant 3-letter sequence in excess of μ_{WM}^* reported correctly as a function of time before the light was lit.

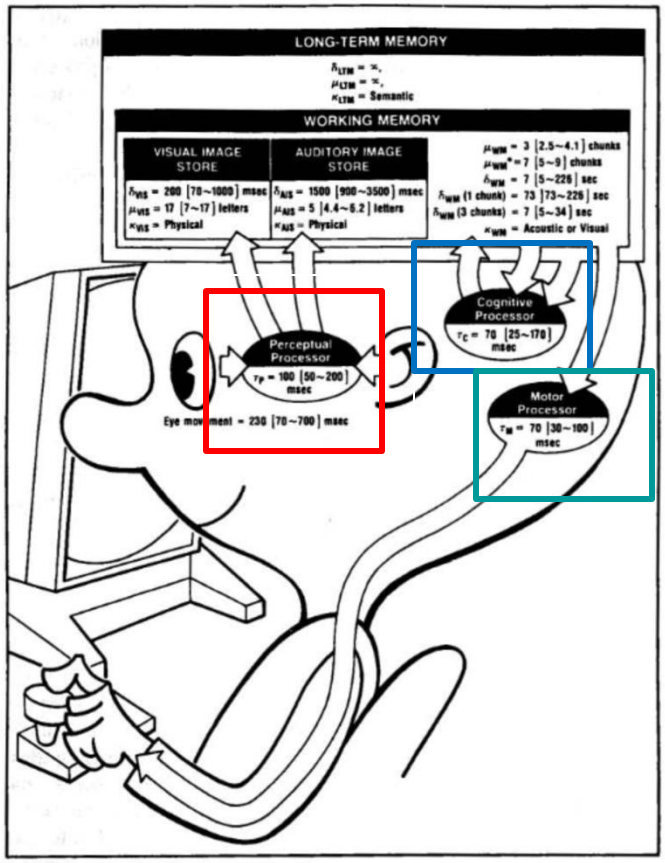
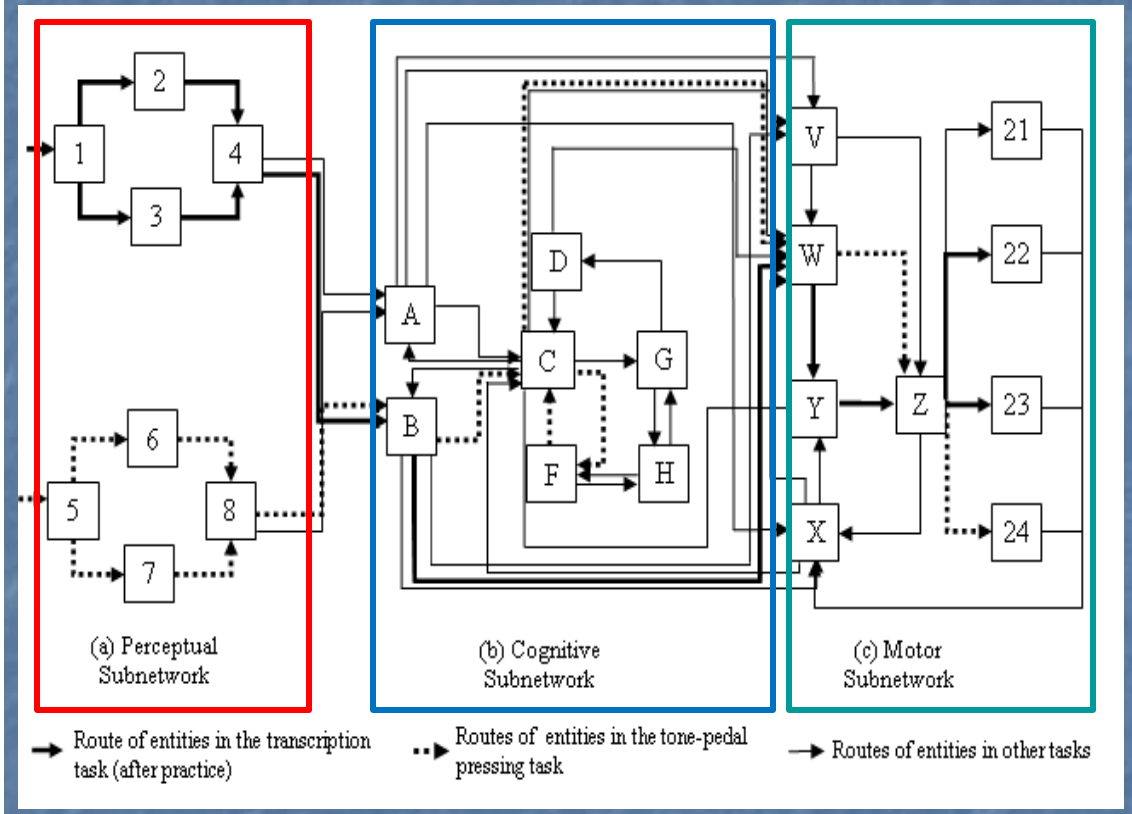
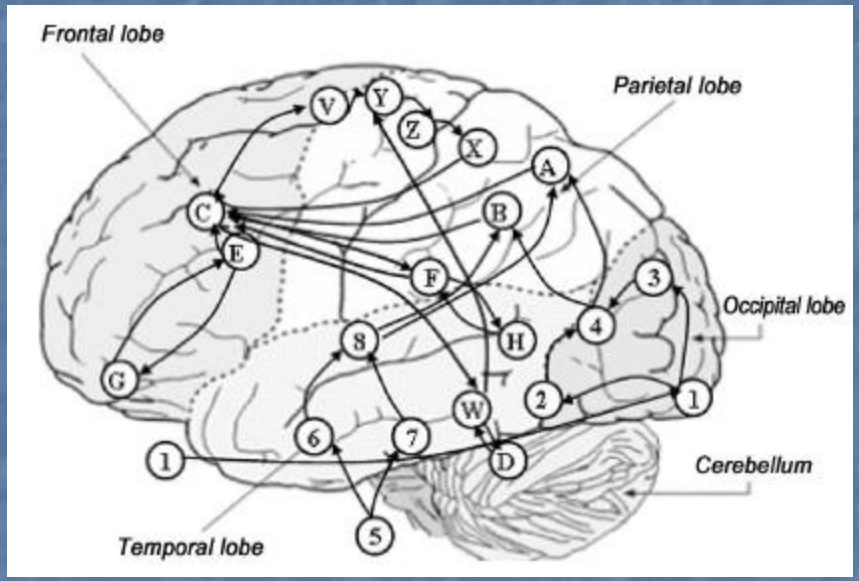


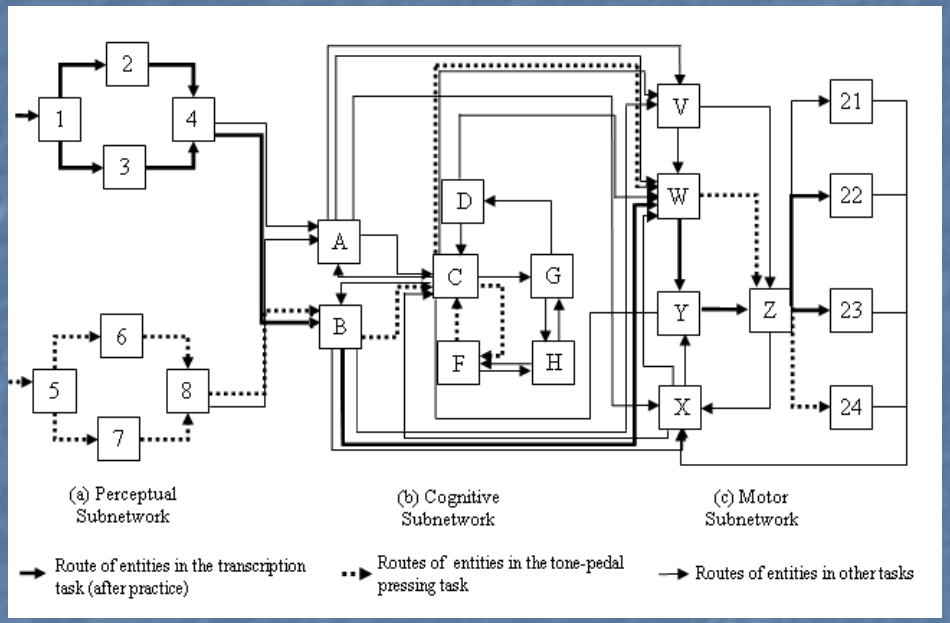
Figure 2.1. The Model Human Processor—memories and processors.

Sensory information flows into Working Memory through the Perceptual Processor. Working Memory consists of activated chunks in Long-Term Memory. The basic principle of operation of the Model Human Processor is the *Recognize-Act Cycle of the Cognitive Processor* (P0 in Figure 2.2). The Motor Processor is set in motion through activation of chunks in Working Memory.



Queueing Network - MHP





Human Brain

Queueing Network of Mental Architecture

QN-MHP-BE or QN-MHP

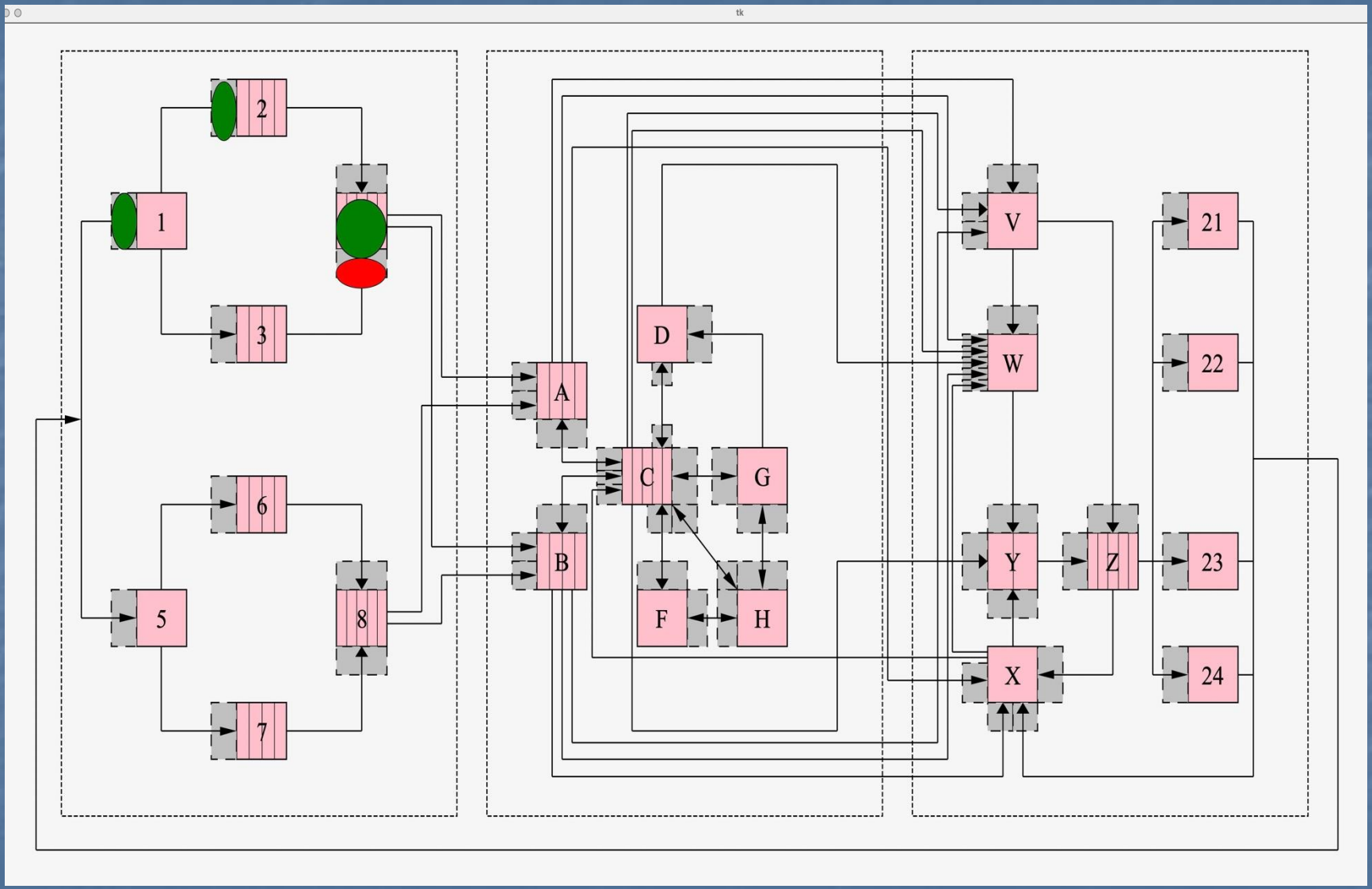
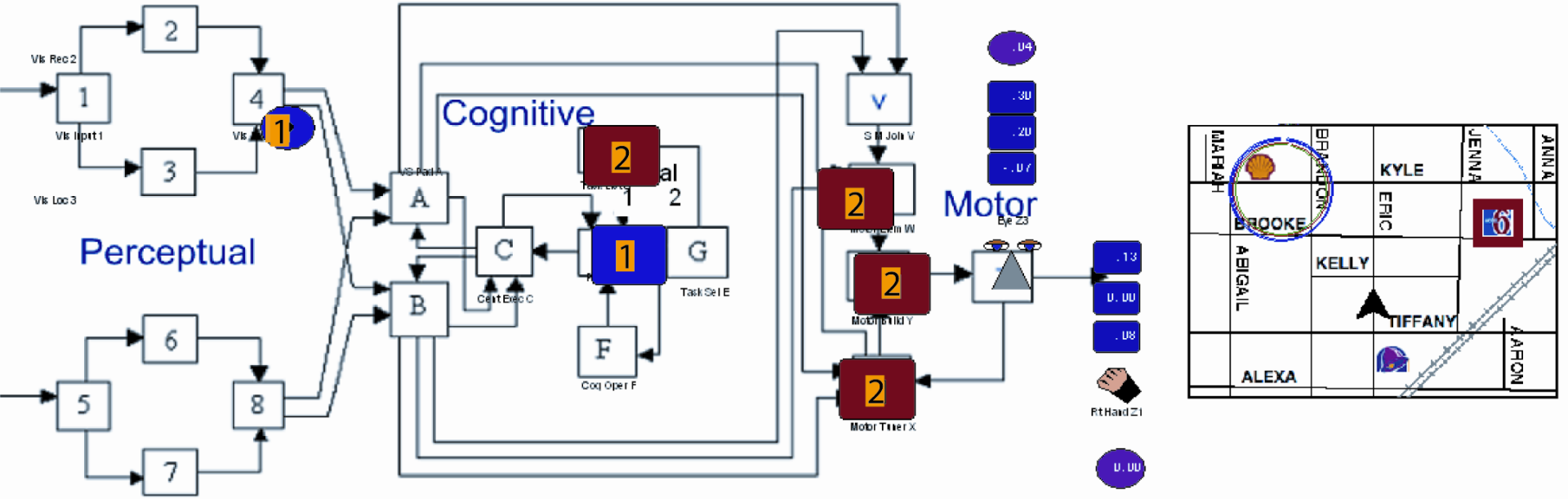
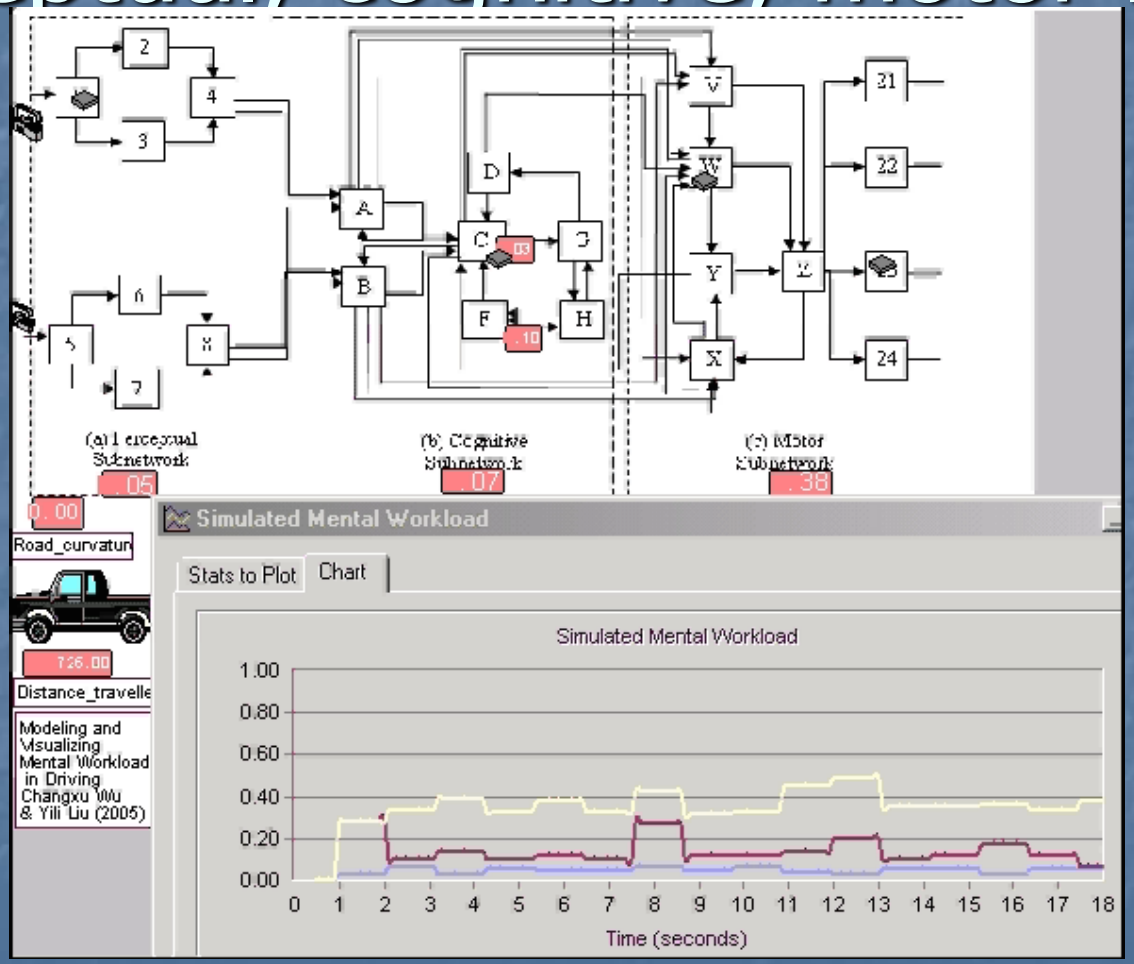


Fig. 10. Screenshot of QN-MHP in action. A visual entity is about to be processed by server A as two concurrent tasks are being processed in the cognitive subnetwork. The eye is looking at the map but a steering action is still underway.



Visualizing Mental Workload (perceptual, cognitive, motor loads)



QN-MHP can be used as

- A pure mathematical model (equations only)

and

- Mathematical and Simulational/Generative with Symbolic (rule-based) Processes

QN-MHP can be used as

- A pure mathematical model (equations only)

One example:

Wu and Liu (2008): "Queuing Network Modeling of the Psychological Refractory Period (PRP)," Psychological Review, 115(4), pp. 913-954

Psychological Review
2008, Vol. 115, No. 4, 913–954

0033-295X/08/$12.00 DOI: 10.1037/a0013123

Queuing Network Modeling of the Psychological Refractory Period (PRP)

Changxu Wu and Yili Liu
University of Michigan

The psychological refractory period (PRP) is a basic but important form of dual-task information processing. Existing serial or parallel processing models of PRP have successfully accounted for a variety of PRP phenomena; however, each also encounters at least 1 experimental counterexample to its predictions or modeling mechanisms. This article describes a queuing network-based mathematical model of PRP that is able to model various experimental findings in PRP with closed-form equations including all of the major counterexamples encountered by the existing models with fewer or equal numbers of free parameters. This modeling work also offers an alternative theoretical account for PRP and demonstrates the importance of the theoretical concepts of "queuing" and "hybrid cognitive networks" in understanding cognitive architecture and multitask performance.

Keywords: psychological refractory period (PRP), cognitive architecture, queuing network, multitask performance, serial and parallel processing

rently. The delay between the presentation of the stimuli of T1 and T2 is called the stimulus onset asynchrony (SOA). Two stimuli (S1 and S2) are presented to the participants in rapid succession, and each requires a quick response (R1 and R2). RT of each task (RT1 and RT2) is measured from the time when the stimulus is presented to the time when the corresponding response is made. In the basic PRP paradigm in which the tasks are choice RT tasks and participants do not receive extensive practice on the dual task, responses to S1 typically are unimpaired, but responses to S2 are slowed by 200–300 ms or more in the short SOA conditions. For instance, Figure 1 shows the experimental results of the basic PRP paradigm in Schumacher et al. (1999).

Schumacher et al.'s (1999) study consisted of a series of PRP experiments, of which their Experiments 3 and 4 are described here for illustration and modeling purposes. T1 was an auditory–manual task in Experiment 3 but an auditory–vocal task in Experiment 4. Participants heard either a low- or a high-pitched tone and responded by pressing the left middle or the left index finger on a keypad, respectively (Experiment 3), or making a corresponding vocal response (Experiment 4). T2 was always a visual–manual task involving compatible or incompatible situations. In each trial of T2, an *O* replaced one of four dashes in a horizontal row centered on the display monitor. In the compatible situation of T2, participants pressed the right-hand index, middle, ring, or little finger keys when the *O* appeared in the far left, middle left, middle right, or far right spatial positions, respectively. In the incompatible situation of T2, participants pressed the right-hand index, middle, ring, or little finger keys when the *O* appeared in the middle left, far right, far left, or middle right positions, respectively. The stimuli for T1 and T2 were separated by one of five SOAs: 50, 150, 250, 500, or 1,000 ms. The compatible situation of T2 in Experiment 4 is an example of the basic PRP experimental paradigm, whose results are shown in Figure 1: RT1 was not affected by T2, but RT2 was longer in the short SOA conditions than in the long SOA conditions. Similar experimental results can be found in many other PRP studies (e.g., Karlin & Kestenbaum, 1968). The importance of the other experimental conditions of Schumacher et al.'s study is described below.

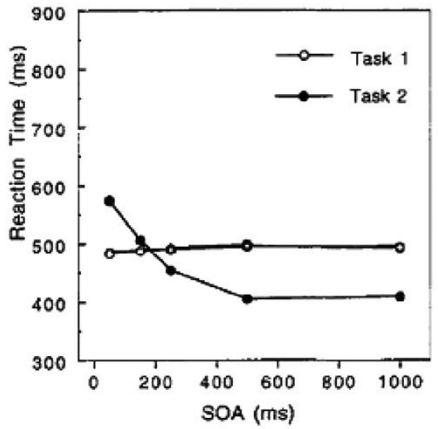


Figure 1. Typical experimental results in the basic psychological refractory period experiment paradigm. Adapted from "Concurrent Response-Selection Processes in Dual-Task Performance: Evidence for Adaptive Executive Control of Task Scheduling," by E. H. Schumacher et al., 1999, *Journal of Experimental Psychology: Human Perception and Performance, 25,* p. 809. Copyright 1999 by the American Psychological Association. SOA = stimulus onset asynchrony.

Subadditive Difficulty Effect

Several PRP experiments (Hawkins, Rodriguez, & Reicher, 1979; Karlin & Kestenbaum, 1968; Schumacher et al., 1999; Sommer et al., 2001) have found that if the difficulty level of T2 at its central processing stage (i.e., at the response-selection stage occurring after the perceptual stage and before the motor stage) is manipulated, the difference of RT2 between the easy and the hard T2s in the short SOA conditions is smaller than that in the long SOA conditions. This pattern or effect is called the subadditive difficulty effect.

Schumacher et al.'s (1999) Study

The subadditive difficulty effect appeared in Experiments 3 and 4 of Schumacher et al.'s (1999) study described above, in which the level of difficulty of T2 was manipulated via the degree of stimulus–response compatibility in T2. In both experiments, Schumacher et al. found a subadditive interaction between the SOA and the response-selection difficulty on the mean RTs of T2, showing the subadditive difficulty effect.

Hawkins, Rodriguez, and Reicher's (1979) Study

The subadditive difficulty effect can also be found in the experimental results of Hawkins et al.'s (1979) study, in which the difficulty level of T2 was manipulated by changing the number of stimuli in a category for making the same response. In the easy T2 condition, the stimuli were the digits *2* and *3,* and the responses were keypresses with the right-hand index and middle fingers, respectively. In the hard T2 condition, the stimuli were the digits *2–9*—four of them (*2, 5, 6,* and *9*) belonged to the first category, the other four digits belonged to the second category, and the participants were asked to press with the right-hand index or middle finger after they saw one of the four digits in the first or the second category.

Karlin and Kestenbaum's (1968) Study

Karlin and Kestenbaum (1968) found the subadditive difficulty effect by manipulating the difficulty level of T2 via changing the number of stimulus–response pairs—one was a simple reaction task, and the other was a two-choice reaction task. In their experiment, T1 was a visual–manual task: Participants were asked to respond to the digits (*1–5*) on a visual display by pressing their left-hand fingers corresponding to the digits. T2 was an auditory–manual task where participants used their right-hand index and middle fingers to respond to high- and low-pitched tones, respectively. Their experimental results clearly demonstrated the pattern of the subadditive difficulty effect: The difference of RT2 between the easy and the hard T2s in the short SOA condition is smaller than that in the long SOA conditions (see Figure 2).

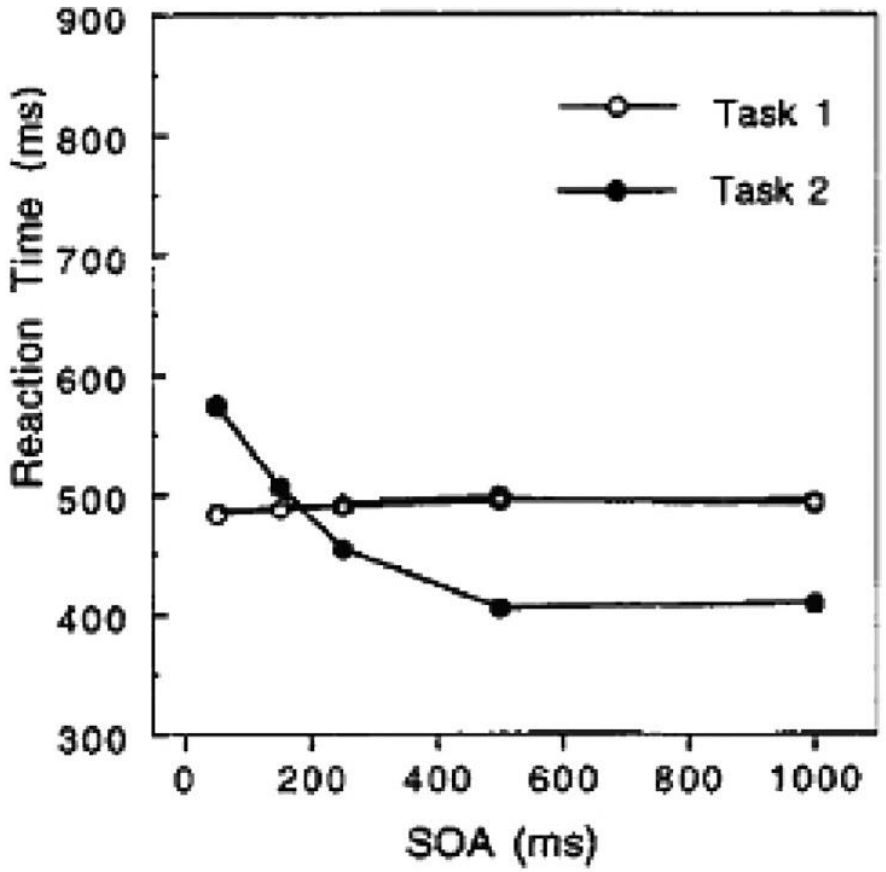


Figure 1. Typical experimental results in the basic psychological refractory period experiment paradigm. Adapted from "Concurrent Response-Selection Processes in Dual-Task Performance: Evidence for Adaptive Executive Control of Task Scheduling," by E. H. Schumacher et al., 1999, *Journal of Experimental Psychology: Human Perception and Performance, 25,* p. 809. Copyright 1999 by the American Psychological Association. SOA = stimulus onset asynchrony.

QN-MHP can be used as

- A pure mathematical model (equations only)

and

- Mathematical and Simulational/Generative with Symbolic (rule-based) Processes

Queueing Network-Model of Human Performance based on Behavior Elements (QN-MHP-BE)

In Chemistry, we have:

- Elements
- Compounds
- Mixtures

- Chemical structure (high-school/college)

In QN-MHP, we have:

- Behavior Elements (BE)
- Behavior Compounds
- Behavior Mixtures (Tasks/Behaviors)

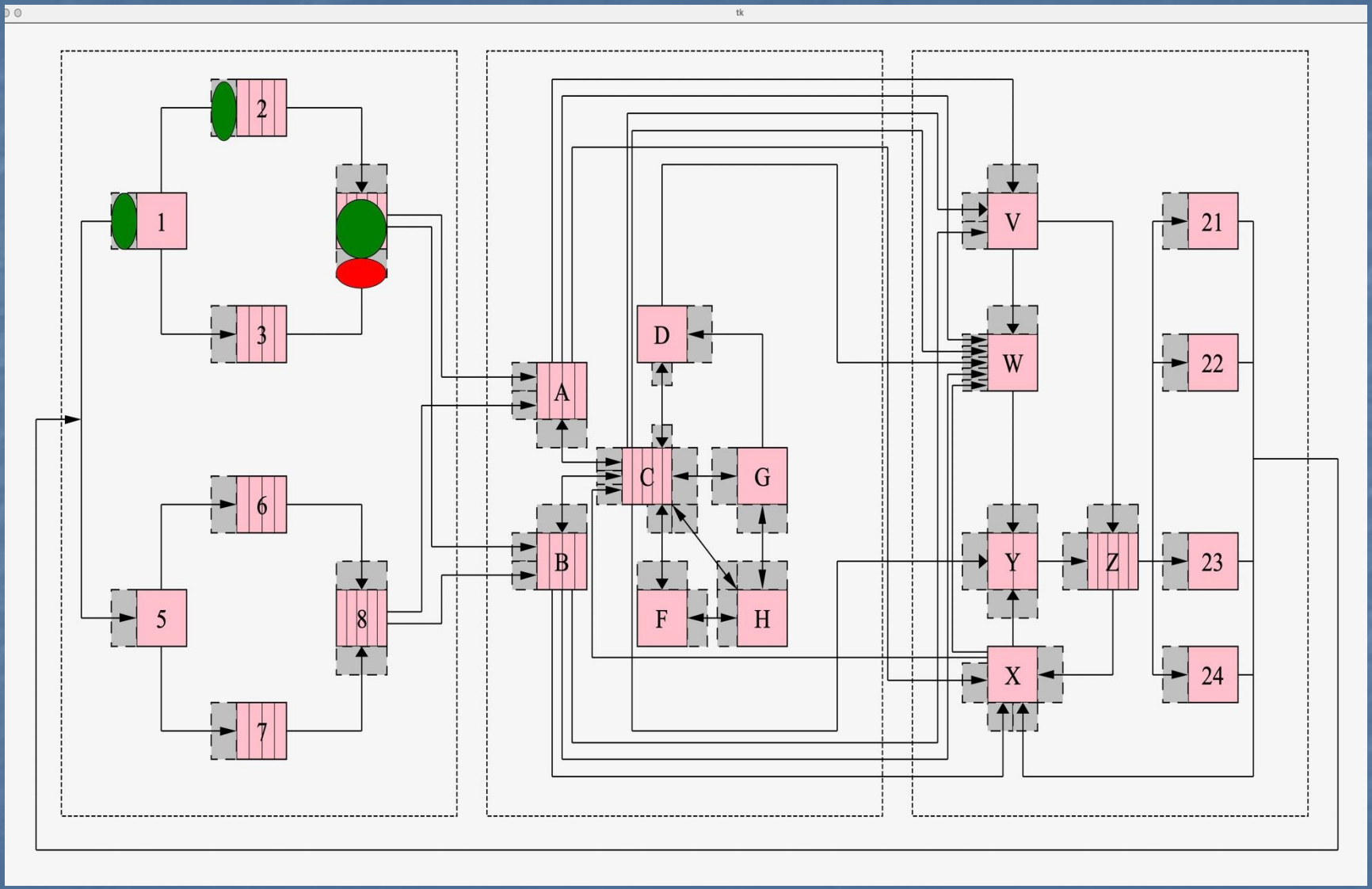
- Queueing Network Structure

Behavior Elements (BEs): Basic_BE
Most are done by one QN-MHP-BE Subnetwork

Examples:

1. Look_at (aka: eye_move)
2. Fixate
3. See
4. Hear
5. Store_to_WM
6. Retrieve_from_WM
7. Choice
8. Count
9. Basic_arithmetic (+/-/x//)
10. Judge_identity
11. Judge_Magnitude
12. Retrieve_from_LTM
13. Speak
14. Press_key
15. Move_Mouse
16. Drag_with_finger

QN-MHP-BE or QN-MHP



Behavior Elements: Basic_BE

Development of:

Drag_with_Finger (anthropometric data, angular direction)

INTERNATIONAL JOURNAL OF HUMAN–COMPUTER INTERACTION
2019, VOL. 35, NO. 6, 510–520
https://doi.org/10.1080/10447318.2018.1466858

Computational Modeling of Touchscreen Drag Gestures Using a Cognitive Architecture and Motion Tracking

Heejin Jeong and Yili Liu

Department of Industrial and Operations Engineering, University of Michigan, Ann Arbor, MI, USA

ABSTRACT

This article presents a computational model that predicts finger-drag gesture performance on touchscreen devices, by integrating the queueing network (QN) cognitive architecture and motion tracking. Specifically, the QN-based model was developed to predict two execution times: the finger movement time of drag-gesture (i.e., only the motion time of the finger touched and dragged on the surface of touchscreen) and the comprehensive process time of drag-gesture (i.e., the entire process time to complete the finger-drag task, including visual attention shift, memory storage and retrieval, and hand-finger movements). To develop predictive models for the finger movement time of drag-gesture, 11 participants' motion data were collected and a regression analysis with parameters of hand-finger anthropometric data and eight angular directions was conducted. Human subject data from our previous study (Jeong & Liu, 2017a) were used to evaluate the QN-based model, generating similar outputs (R^2 was more than 80% and root-mean square was less than 300 msec) for both execution times.

1. Introduction

As touchscreen technologies have rapidly developed over the last few decades, diverse interfaces that require a wide range of touchscreen gestures have become popular (Bhalla & Bhalla, 2010; Saffer, 2008). Although the definition varies with the research domain, touchscreen gestures can be generally categorized into two types, depending on the number of fingers used: single-touch and multi-touch gestures. Typical examples of a single-touch gesture include tap, swipe, and drag, whereas pinch and spread are examples of a multi-touch gesture.

Since the use of touchscreen gestures has become more prevalent in our daily lives, humans are likely to have more chances to interact with systems or environments using the touchscreen gestures. For instance, they perform touchscreen gestures while walking or driving to navigate to the destination. Thus, it is necessary to investigate human behaviors for touchscreen gesture tasks in the comprehensive process, using human brain and body segments, not just an index finger movement itself.

Many experimental studies have investigated finger gesture performance on touchscreens (e.g., Asakawa, Dennerlein, & Jindrich, 2017; Jeong & Liu, 2017a; Jorritsma, Prins, & Van Ooijen, 2015; Kim & Jo, 2015; Parhi, Karlson, & Bederson, 2006; Sasangohar, MacKenzie, & Scott, 2009). In addition to conducting experiments, several studies have developed prediction models using the empirical data obtained through their experiments (e.g., Epps, 1986; Bi, Li & Zhai, 2013; Ljubic, Glavinic, & Kukec, 2015). However, most of these models have several limitations. First, they relied on Fitts' law or its extension, treating movement time as a function of only the distance to and size of the target (Fitts, 1954). In other words, these previous models do not consider individual differences that affect finger gesture performance, such as human's anthropometry. Furthermore, the previous models focused on just finger movements, not considering human perceptual and cognitive processes; thus, these models cannot fully represent the details of comprehensive activities in human–computer interaction, such as visual attention shift, and memory storage and retrieval.

According to the literature, the movement time of an object (e.g., a fingertip or a mouse cursor) differs depending on which direction the object is heading to. Jagacinski and Monk (1985) compared the movement times when both joystick and head-mounted sights were used in two-dimension directions. In their study, horizontal and vertical movement times were slightly shorter than diagonal movement time, but finger movements on touchscreens were not investigated. In addition to movement direction, anthropometry is also regarded as a factor that affects movement time. Bergstrom-Lehtovirta and Oulasvirta (2014) found that hand size was one of the factors in predicting the functional area of the thumb on a touchscreen surface. Scott and Conzola (1997) examined the effect of finger size on touchscreen keying performance. They found that finger size had a significant effect on keying speed and duplication errors (i.e., a button on the touchscreen was

CONTACT Heejin Jeong heejinj@umich.edu
Color versions of one or more of the figures in the article can be found online at www.tandfonline.com/hihc.

Table 3. Summary of anthropometric data for the "*Drag-with-finger*" operator development (n = 11).

No.	Anthropometric Dimension (unit: millimeters)	*M*	*SD*	Min	Max	Population Percentiles (male/female) 5th	50th	95th
1	Stature (S)	1743	88	1622	1857	1651/1527	1755/1629	1868/1738
2	Finger spread (FS)	146	22	111	171	unknown	unknown	unknown
3	Thumb breadth (TB)	21	2	17	24	22/19	24/21	26/23
4	Index finger breadth (IB)	16	2	14	19	18/15	20/17	23/19
5	Short thumb length (STL)	64	8	55	74	62/56	70/63	78/72
6	Long thumb length (LTL)	128	10	109	145	124/112	138/125	153/141
7	Index finger length (IL)	73	6	63	84	67/62	75/70	84/77
8	Hand length (HL)	187	14	166	210	179/163	194/178	212/195
9	Hand breadth (HB)	84	7	73	95	86/76	95/83	105/90

participants had normal or corrected-to-normal vision and were right-handed. They reported no physical issues in using touchscreen display and used touchscreen devices (e.g., smartphones and tablets) for 7.5 years. Participants were paid for their time with #15 hourly rate in cash. Table 3 summarizes anthropometric data obtained from the participants, and population percentile data extracted from Greiner (1991) to compare with the participants' data.

3.2. Apparatus

A motion tracking system (OptoTrak® Certus™; Northern Digital Inc.) with two standing position sensors (three cameras on each sensor; 3.5 m away from each other) was used to record finger movements for finger-drag gestures. One marker was attached on the center of participants' right index fingernail and it was secured with Velcro® straps across the finger, wrist, and forearm, as shown in Figure 3. A touchscreen device (iPad; 1024 × 768; 132 ppi; 9.7-inch LED-backlit glossy widescreen multi-touch display) was mounted on the table (height = 95 cm). Participants were asked to find the most comfortable standing position so they do not feel any discomfort while performing the finger-drag gesture tasks.

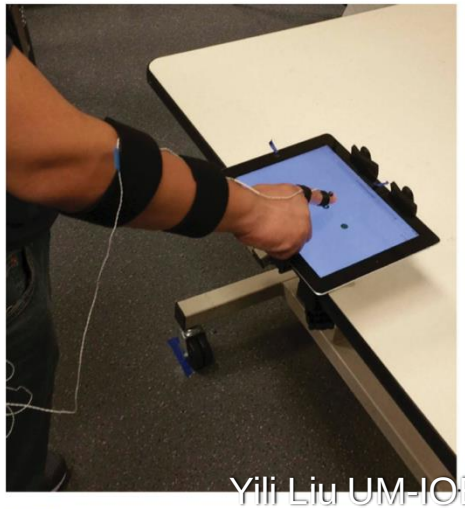


Figure 3. Experimental setup of the finger-drag gesture task.

3.3. Touchscreen gesture task and experimental design

The performance of finger-drag gestures was measured, using a touchscreen interface used in Jeong and Liu (2017a). As shown in Figure 4, nine circles (i.e., eight target circles around one center circle) were designed on a touchscreen display, but only two circles (i.e., one center circle and one of the eight target circles; colored in green – one with a hole, the other without a hole) were presented to the participants during the experiment. The distance between the center and the target circles was 40 mm, a fixed value. Participants were instructed to move their right index fingers from the center circle to one of the target circles; while the center circle was fixed and always presented, the target circle was randomly presented on the touchscreen display. The radius of both the center and target circles was identically 20 pixels (= 9 mm). Whenever the finger arrives from the air to the display surface and leaves from the display surface to the air, a 20-pixel-radius black circle was shown on the display, as a visual feedback. Only when the Euclidean distance between the center of the black and the green circles equals to or less than 20 pixels (also called a match of the black and green circles), it was defined as a success. The participants were asked to press "start" button on the center of the screen (then the button disappeared immediately), and then to complete the dragging task with two successes on the center and target circle matches (i.e., initial and final matches). In the current study, only the data of the success task trials were used and analyzed. In other words, we did not model the accuracy of the drag-gesture's performance. Instead, we used and analyzed time data, only for the success task trials.

A within-subject factorial design was used in this study. The two independent variables (including a subject variable) manipulated in this experiment were 8 different angular directions (i.e., 0°, 45°, 90°, 135°, 180°, 225°, 270°, and 315°) and the participants' 9 anthropometric parameters (i.e., S, FS, HB, IB, STL, LTL, IL, HL, and HB). Each of the 11 participants conducted three replications of the drag gesture to each angular direction. Each participant performed finger-drag gestures in two sessions: (1) horizontal (0° and 180°)/vertical (90° and 270°) and (2) diagonal direction (45°, 135°, 225°, and

drag movement time. Using stepwise regression analysis, we acquired the predictive regression models. The predictors of these regression models in each direction were nine anthropometric parameters: stature (S), finger spread (FS), thumb breadth (TB), index finger breadth (IB), short thumb length (STL), long thumb length (LTL), index finger length (IL), hand length (HL), and hand breadth (HB). Most of parameters were the hand and finger sizes related to the single-touch drag gesture. The definition of each hand-finger anthropometric parameter was adopted from the Eastman Kodak Company Publication (1986) and is illustrated in Figure 2. Stature (measured by a linear distance from the floor to the top of the skull) was also selected as an anthropometric parameter because we were to model finger-drag gestures in the assumption of standing position. Using the regression equations for the finger movement time of drag-gesture, we built a comprehensive QN-based model of touchscreen drag gestures.

2. Queueing-network modeling of human performance

The QN-MHP architecture is composed of three subnetworks: perceptual, cognitive, and motor. As shown in Figure 1, it is assumed that each subnetwork includes multiple servers (1–8; A-G; W-Z) and each server plays its own role, based upon findings from previous neuroscience and psychology studies. The examples of the process that happens in the perceptual subnetwork include to look at objects (i.e., visual perception) or to listen to sounds (i.e., auditory perception). In the cognitive subnetwork, the process includes remembering the objects or the sounds, and comparing perceived signals to expected signals. The examples occurred in the motor subnetwork are to reach with hands or to say words.

To implement human performance models in the QN-MHP architecture, it is required to (1) conduct a task analysis and (2) develop operators based upon the result of the task analysis (or develop new operators if they are needed but do not exist yet). The task analysis was conducted using NGOMSL (Natural Goals, Operators, Methods, and Selection rules Language)-style task description (Kieras, 1997). The operator refers to the most elementary component of the task. Each operator is set with several parameters for specifying the task.

2.1. Task analysis in the QN-MHP architecture

In the NGOMSL-style task analysis, task components (TCs) are used to describe each step to accomplish the whole task. Each TC is made with a predetermined operator that runs with one or multiple parameter(s). Table 1 shows the result of task analysis for finger-drag gesture task. In Table 1, "*Look-at*", "*Determine-hand-movement*", "*Drag-with-finger*" are the examples of the operator, whereas "<target type>", "<device id>", "<x, y>" are the examples of the parameter. The comprehensive finger-drag gesture task consisted of 16 TCs, describing as the following three main subtasks.

2.1.1. Visual information perception [TCs 1–5]

The first subtask is a process to perceive a visual information and to compare the information to the information expected. It includes perceiving a visual signal (or information) on a touchscreen device with its two-dimension location, and storing and retrieving the information of the visual signal, using short-term-memory. Then, a comparison process is performed to determine

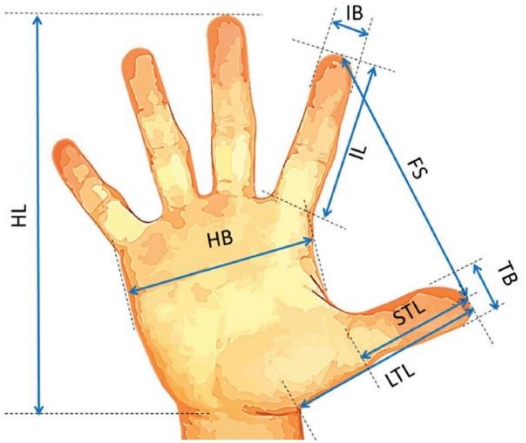


Figure 2. Hand-finger anthropometric dimensions.

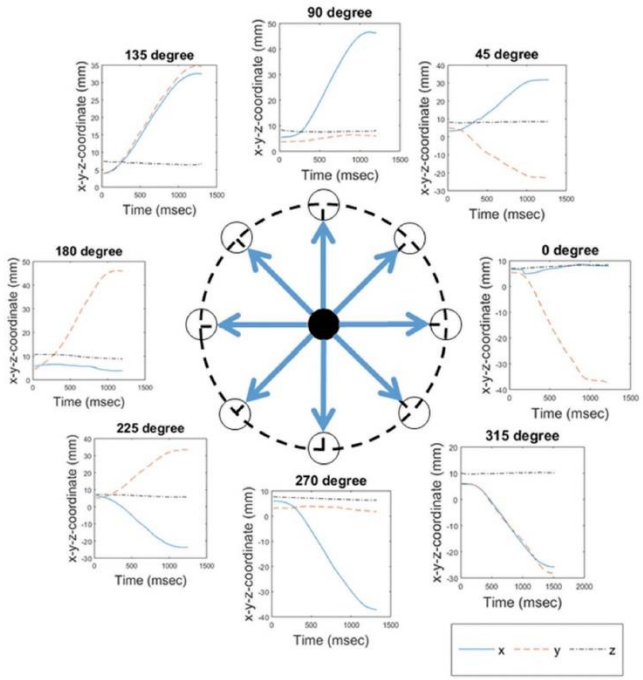


Figure 5. A sample motion data in eight different angular directions.

Table 4. Summary of anthropometric data for validation (n = 20).

No.	Anthropometric Dimension (unit: millimeters)	*M*	*SD*	Min	Max
1	Stature (S)	1729	98	1562	1869
2	Finger spread (FS)	149	21	111	185
3	Thumb breadth (TB)	20	2	17	24
4	Index finger breadth (IB)	16	1	14	19
5	Short thumb length (STL)	64	7	55	79
6	Long thumb length (LTL)	127	10	109	145
7	Index finger length (IL)	73	6	63	86
8	Hand length (HL)	186	13	165	210
9	Hand breadth (HB)	84	7	73	99

gesture. In addition to developing the QN-based model for the touchscreen finger-drag gesture, the model was evaluated with the data from 20 participants in our previous study (Jeong & Liu, 2017a). The model was able to generate similar outputs (R^2 was more than 80% and RMS was less than 300 msec.) for both execution times (i.e., the finger movement time of drag-gesture; the comprehensive process time of drag-gesture) with human performance data.

5.1. Implications

Since the QN-MHP framework was first developed, a wide range of human performance has been successfully modeled (e.g., map reading (Liu et al., 2006), transcription and numerical typing (Lin & Wu, 2012; Wu & Liu, 2008), and driving control (Bi, Gan, Shang, & Liu, 2012; Jeong, Feng, & Liu, 2017)). The application domains have been expanded to more practical fields, but there has been no attempt to model the touchscreen finger-drag gesture performance. In addition to the fact that this study is the first attempt to model the finger-drag gesture performance, it has two significant features, compared to other previous modeling studies of touchscreen gestures (e.g., Epps, 1986; Bi, Li & Zhai, 2013; Ljubic et al., 2015). First, this study considered the user's hand anthropometric data as a factor that affects the finger-drag performance. From the stepwise regression analysis, it turned out that the finger-drag movement time depends on its direction, and it is a function of IB, TB, STL, IL, and FS, shown in Table 2. Using the predictive regression models with the

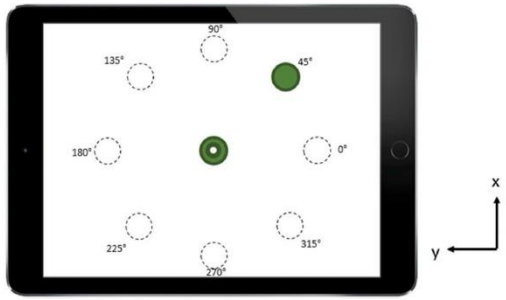


Figure 4. Angular directions for finger-drag gesture and the coordinate system (Note: This figure shows an example task when the direction is 45°; the dotted circles and the degree information were not presented to the participants).

315°). The order of conditions was balanced across participants.

3.4. Experimental procedures

After arrival in the laboratory, each subject was informed of the purpose and nature of the experimental task. They completed a consent form and filled out a brief survey about their demographic information (e.g., age and gender) and years of touchscreen usage. After each participant's anthropometric data (e.g., hand and finger size) were measured, a marker and motion-tracking system wires were attached to their right index finger and upper extremity. Prior to the actual finger-drag gesture task, they practiced sample trials to get familiar with the tasks for 10–15 min. Each of 11 participants conducted two sessions (i.e., horizontal/vertical and diagonal) for drag gesture tasks with their right index finger. At the end of the experiment, they completed a payment form and were paid for their times, taken for approximately less than an hour.

3.5. Development of finger-drag movement time regression models for "drag-with-finger" operator

Out of 264 motion data sets (i.e., 11 participants × 8 directions × 3 replications), there were only 6 missing data points caused by procedural and equipment malfunction (including non-success trials). Figure 5 shows a subject's sample motion data, depending on the angular directions. The time frames (in milliseconds) for each motion (in millimeters) were collected and they were regarded as the execution time of "*Drag-with-finger*" operator. SPSS Statistics 23 was used for stepwise regression analysis to model the relationship between the finger-drag movement time and potential predictors including all anthropometric parameters. The criteria were set as probability-of-F-to-enter ≤ .05 and probability-of-F-to-remove ≥ .10. In order to detect the multicollinearity problem (i.e., having highly correlated predictors in a regression model), variance inflation factors (VIFs) was used to check whether it is less than 10 (Kutner, Nachtsheim, & Neter, 2004; Montgomery, Peck, & Vining, 2015). High degree of multicollinearity was not present for all eight conditions. From the automated stepwise process, regression equations were obtained depending on each angular direction (see Table 2).

4. Model validation

To validate the model including (1) finger movement process of the drag-gesture (i.e., TC 15 only) and (2) its comprehensive process (i.e., all TCs 1–16), data from our previous study (Jeong & Liu, 2017a) were used. The data were collected from 20 (11 males and 9 females; age M = 23.8, SD = 3.3) right-handed students who had used the touchscreen devices for 6.7 years on average. Participants' anthropometric data are shown in Table 4 (note that the data were collected in Jeong and Liu (2017a), but not used in that study). The touchscreen software collected the finger movement time of drag-gesture (i.e., only the motion time of the finger touched and dragged on the surface of touchscreen) and the comprehensive process time (i.e., the entire process time to complete the finger-drag task, from when "start" button was pressed to when the task was completed).

The model performed 20 simulation runs with different angular directions in the MATLAB-Simulink software. Figure 6 shows the modeling results (i.e., solid lines) compared with experimental results (i.e., dashed lines) in both the finger movement time and the comprehensive process time of drag-gesture. In the comparison of finger-drag gesture process time, the R^2 of the model was .90 and the RMS = 92.2 msec; in the comprehensive process time, the R^2 of the model was .80 and the RMS = 256.4 msec.

5. Discussion

This article presented a computational model for finger-drag gestures on touchscreen devices, integrating the QN-MHP architecture and motion tracking: specifically, for finger movement and comprehensive process of the drag-

Behavior Compounds: Compound_BE

Examples:

1. Look_for

A Compound of Basic BEs:

Look_at
See
Judge_identity
Store_to_WM (or update_WM)

Look_at (next) can be sequential, random,…;

Loop till Target found or all Items seen (e.g.,)

Items may be seen multiple times or once (depending on WM)

Behavior Compounds: Compound_BE

Examples:

Scan_SEEV:

A Compound of Basic BEs:

Look_at

Fixate

Look_at (next) is determined by SEEV
(Salience, Effort, Expectancy, Value)

Or (more advanced/complex versions)

A Compound of Basic BEs:

Look_at

Fixate

[Retrieve from LTM,

update_WM] periodically

Behavior Compounds: Compound_BE

Examples:

Scan_RLEM:

A Compound of Basic BEs:

Look_at

Fixate

Look_at (next) is determined by RLEM
(Reinforcement Learning in Eye Movement)

QN-Reinforcement Learning-Eye Movement

Modeling the Influences of Cyclic Top-Down and Bottom-Up Processes for Reinforcement Learning in Eye Movements

Ji Hyoun Lim and Yili Liu, *Member, IEEE*

***Abstract*—Understanding and reproducing complex human oculomotor behaviors using computational models is a challenging task. In this paper, two studies are presented, which focus on the development and evaluation of a computational model to show the influences of cyclic top-down and bottom-up processes on eye movements. To explain these processes, reinforcement learning was used to control eye movements. The first study showed that, in a picture-viewing task, different policies obtained from different picture-viewing conditions produced different types of eye movement patterns. In another visual search task, the second study illustrated that feedback information from each saccadic eye movement could be used to update the model's eye movement policy, generating different patterns in the following saccade. These two studies demonstrate the value of an integrated reinforcement learning model in explaining both top-down and bottom-up processes of eye movements within one computational model.**

***Index Terms*—Cognitive model, eye movement, queueing network model, reinforcement learning.**

I. Introduction

EYE movements are essential to acquire information in visual interfaces and represent one of the most prevalent activities in human–machine systems. Eye movements have been studied to understand covert and overt attention [1]–[4], spatial attention [5], [6], pattern recognition [7], [8], and influences of cognitive processes on oculomotor behavior [9]–[11] and to also improve machine vision [12]. A major consensus of these studies is that eye movements are influenced by both the perceiver's expectations (called the top-down process) and the characteristics of the visual environment (called the bottom-up process).

As Neisser [13] proposed, perception is a cyclic process in which a mental model directs the eyes to explore and sample the environment, and the information obtained from the environment is used to modify the mental model. The top-down process is driven by the mental model, while the bottom-up process is driven by the environment. Forty years ago, Yarbus [11] showed how human eye movements are influenced by top-down cognitive processes. Yarbus' experiment was recently revisited [7], [8], and the studies supported the original findings. While Yarbus' experiment demonstrated the influence of top-down processing on eye movement patterns, the influence of bottom-up processing has been shown through studies on visual saccades, a much more basic type of eye movement. Findlay's extensive studies on saccades [14]–[16] and other researchers' related studies on the distractor-ratio effect [17], [18] have shown the influence of bottom-up processes on saccadic eye movements. The distractor-ratio effect is observed in a visual search task involving two types of distractors in which, even though the total number of items in a display remains the same, visual search performance changes with the ratio between the two types of distractors.

A single saccade is the basic component of eye movement patterns. Spatial attention is assumed to play a major role in the planning and the execution of saccades, and so, the relationship between spatial attention and saccadic eye movement has been studied in depth over the last few decades (for an overview, see [15]). There are several conceptual models of spatial attention, including the spotlight theory [3], the zoom lens theory [1], and the selection theory of attention [19], [20]. The theories explaining the relationship between spatial attention, or, more specifically, covert attention, and eye movements are the premotor theory of attention [3] and the sequential attentional model [21]. Studies on the neurophysiology of attention have shown strong support for a close coupling between covert attention and saccade generation in that both arise from the same basic neural processes [22], [23]. The sequential attentional model and the premotor theory both assume a close link between covert attention and saccades. The sequential attentional model assumes that a saccade is driven by a covert attention shift, while the premotor theory assumes that a covert attention shift is a by-product of the action of the oculomotor system.

In the field of human factors, eye transition patterns have been studied in various task domains [24]. In the statistical models of eye transition, the open-loop versus closed-loop control is one issue of debate [24]. The open-loop control strategy assumes repetitive information gathering independent of the previous eye fixation. This strategy emphasizes the top-down influence on eye movements. On the other hand, the closed-loop strategy assumes that an eye movement is dependent on information gathered from the previous fixation. The

Manuscript received November 17, 2006; revised November 28, 2007 and June 27, 2008. Current version published June 19, 2009. This work was supported by the National Science Foundation under Grant 0308000. Any opinions, findings, conclusions, or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of the National Science Foundation. This paper was recommended by Associate Editor Y. Xiao.

J. H. Lim is with the Design Team, Mobile Communication Division, Samsung Electronics, Seoul 100-742, Korea (e-mail: smilelim@gmail.com).

Y. Liu is with the Department of Industrial and Operations Engineering, University of Michigan, Ann Arbor, MI 48109 USA (e-mail: yililiu@umich.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSMCA.2009.2018635

QN-Reinforcement Learning-Eye Movement

after each selected action (a_t) in a given state (s_t) provides information about how good the action was, but it says nothing about whether the action was correct or incorrect. Therefore, a reward function has to be defined based on the goal and properties of the information given by the environment. In eye movement modeling, the reward is determined by whether the perceived information delivers what the agent is looking for or not.

The third feature of reinforcement learning is the Markov property of the learning process. The Markov property states that the probability distribution of the future states of the process ($\Pr\{s_{t+1} = s', r_{t+1} = r | s_t, a_t\}$) depends only upon the current state, and is conditionally independent of the past states (the path of the process) given the current state. As long as the learning process has the Markov property, the expected value of reward ($\mathrm{R}^a_{ss'} = \Pr\{r_{t+1} | s_t = s, a_t = a, s_{t+1} = s'\}$) can be calculated based on the transition matrix of probabilities between the states ($\mathrm{P}^a_{ss'} = \Pr\{s_{t+1} = s' | s_t = s, a_t = a\}$) and the reward function.

A policy (π) defines how the learning agent selects an action at a given state. The main interest in reinforcement learning is to find the policy that delivers the maximum reward. There are two major issues in finding the policy that delivers the maximum reward: 1) how to evaluate the candidate policies and 2) how to update policies. One way to evaluate policies is to assess the value of a state. The other way is to evaluate the value of an action. In this study, we used the value function $Q(s, a)$ to evaluate actions. The definition of $Q(s, a)$ is provided in the next section.

III. Eye Movements and Reinforcement Learning

Reichle and Laurent [31] used reinforcement learning to understand eye movements in reading by assuming an artificial reader being capable of learning to control its eye movements. Their model proposed nine variables defining the states of the reading agent, and a reward was given for each word identified. Sprague and Ballard [32] examined temporal scheduling of human eye movements based on reinforcement learning. The task in their study was to find multiple targets in a given environment. The model tried to schedule the temporal order of targets to be looked at to maximize the total rewards. The reward for each target was assigned by the researchers as varying amounts with a predefined unit. Their study focused on strategic scheduling in eye movements rather than explaining the underlying processes in human eye movements.

Our approach to modeling eye movements was to consider each eye movement as a Markov decision process and then find an underlying policy using the reinforcement learning method. The Markov process assumption is essential for finding the optimal policy (π^*). The goal of our paper, however, was not to find the optimal policy for eye movements but to focus on the differences in policies (π) under different visual task conditions. Therefore, the Markov process assumption was not crucial for this paper but helped to map eye movements in a reinforcement learning problem.

A Markov decision process has four attributes: S is the state space, A is the action space, $\mathrm{P}^a_{ss'}$ is the transition matrix that indicates the probability of arriving in state s' when action a is taken in state s, and $\mathrm{R}^a_{ss'}$ is the expected reward value. To select an action in a given state, the optimal value function $Q(s, a)$ is used in this study. The optimal value function $Q(s, a)$ represents the expected discounted return if action a is taken in state s and the optimal policy is followed thereafter. Among many algorithms [30] for updating the value of $Q(s, a)$, the online Q-learning update rule was used to select a feature for a saccadic eye movement

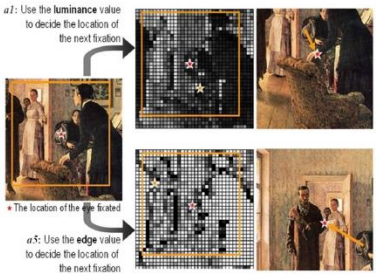


Fig. 1. Location selected for the next eye fixation depends on the action chosen.

$$Q(s, a) \leftarrow Q(s, a) + \alpha \left[r + \gamma \max_{a'} Q(s', a') - Q(s, a) \right].$$

Here, α is a learning rate parameter, and γ is a discounting factor of future reward. In this study, the initial value of the learning rate parameter α_0 was 0.3, and it decreased by following the rule

$$\alpha_n = 0.99\, \alpha_{n-1}.$$

The discounting factor of future reward γ was set at 0.5 to consider immediate reward, as well as future reward, through the learning process. The value of the discounting factor did not change so as to reduce computational complexity.

To map a reinforcement learning model to eye movements, the major components in reinforcement learning—agent, environment, states, action, reward, and policy—were defined as follows.

The ***Agent*** was defined as an eye with attended and unattended visual zones. In Fig. 1, the star indicates the attended visual zone, and the large square indicates the unattended visual zone. To simplify the calculation process during computational simulation, the square shape was used to represent the visual zones. The location of eye fixation ($x \in E$, where x is a 2-D vector) was determined as the location of the center of the unattended visual zone.

The ***Environment*** (E) was the visual stimulus (or the picture) shown to the agent to complete a given task. In Yarbus' picture-viewing task, the environment was I. E. Repin's painting, and in Findlay's visual search task, the 16 items in two concentric rings were presented as the environment.

QN-Reinforcement Learning-Eye Movement

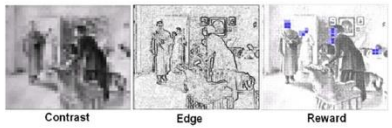


Fig. 3. Input data arrays of luminance, edge, and reward (give the ages of the people) values for I. E. Repin's painting "They Did Not Expect Him" (1884) used in the picture-viewing task.

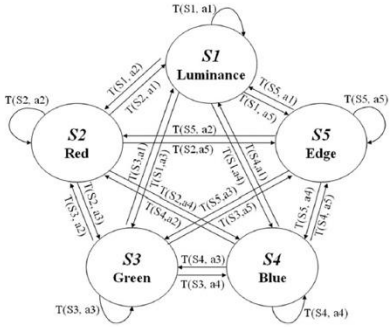


Fig. 4. Action space with 25 actions.

the location x on the original picture whose size is 0.5° of visual angle (estimated from the Yarbus' study). Fig. 3 shows the luminance and edge values used in this simulation study. The cell of the ith column and the jth row on the left array in Fig. 3 represents the amount of luminance value for the location $x(i, j)$ on the original picture. The range of values was from 0 to 255.

The expected value of ***Reward*** $\mathrm{R}^{a}_{ss'}$ was assumed to be the amount of information that the agent could retrieve at a given location x by taking the action a from the state s. In this case, the amount of information for each location x was assigned as High, Low, and None, depending on the question to be answered. For example, under the "give the ages of the people" condition shown in Fig. 3, the rewards were assumed to be distributed mainly at the faces in the picture, whereas under the "what family had been doing" condition, the rewards were assumed to be distributed mainly at the bodies of the people.

Fig. 4 shows the action space $A(s_t)$. The five states are the five alternative features (luminance, red, blue, green, and edge) currently used, and actions are staying in the current state or moving to one of the other states

$$\text{Reward per saccade} = \frac{\sum_{i=0}^{t} R_i}{t}$$

$$R_t = R^{a_t}_{s_t s_{t+1}}.$$

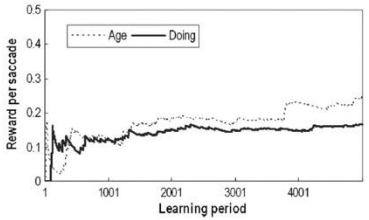


Fig. 5. Reward per saccade during the 5000 learning periods.

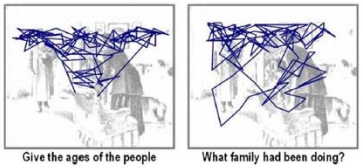


Fig. 6. Simulated eye movement patterns of 300 saccades with two different sets of rewards after 5000 learning trials.

Using 5000 randomly selected actions, the value of each $Q(s, a)$ was updated. Fig. 5 shows the changes in the amount of reward per saccade during the learning period. After the learning period, the amount of reward per saccade appeared to be stabilized, as well as the values of $Q(s, a)$ for all s's and a's. When the value of the reward per saccade reaches a plateau, it can be interpreted as the model starting to generate a certain pattern in eye movements.

As explained earlier, two different reward sets were used for the "Give the ages of the people" condition and the "What the family had been doing" condition. The two $F(s, a)$ matrices for these conditions are

$$F(s,a)_{\text{Age}} = \begin{bmatrix} 0.06 & 0.06 & 0.06 & 0.77 & 0.06 \\ 0.19 & 0.19 & 0.19 & 0.19 & 0.23 \\ 0.02 & 0.05 & 0.02 & 0.87 & 0.04 \\ 0.17 & 0.17 & 0.17 & 0.17 & 0.32 \\ 0.24 & 0.19 & 0.19 & 0.19 & 0.20 \end{bmatrix}$$

$$F(s,a)_{\text{Doing}} = \begin{bmatrix} 0.16 & 0.16 & 0.16 & 0.16 & 0.36 \\ 0.04 & 0.04 & 0.76 & 0.04 & 0.11 \\ 0.18 & 0.18 & 0.18 & 0.22 & 0.26 \\ 0.15 & 0.15 & 0.40 & 0.15 & 0.16 \\ 0.25 & 0.10 & 0.33 & 0.12 & 0.20 \end{bmatrix}.$$

Using the previous two policies, the viewing tasks were simulated. Fig. 6 shows the simulated eye movements of 300 saccades. The simulated eye movement patterns showed difference between the two different viewing conditions. The only difference in modeling the two conditions—"Give the ages" and "What the family had been doing"—was the *location of rewards*. All the other parameters, input data, and procedural rules for eye movements were exactly the same for both cases.

QN-Reinforcement Learning-Eye Movement

Investigation of Driver Performance With Night-Vision and Pedestrian-Detection Systems—Part 2: Queuing Network Human Performance Modeling

Ji Hyoun Lim, Yili Liu, *Member, IEEE*, and Omer Tsimhoni, *Member, IEEE*

***Abstract*—This paper introduces a queueing network-based computational model to explain driver performance in a pedestrian-detection task assisted with night-vision-enhancement systems. The computational cognitive model simulated the pedestrian-detection task using images displayed by two night-vision systems as input stimuli. The system equipped with a far-infrared (FIR) sensor generated less-cluttered images than the system equipped with a near-infrared (NIR) sensor. Using a reinforcement learning process, the model developed eye-movement strategies for each night-vision system. The differences in eye-movement strategies generated different eye-movement behaviors, in accord with the empirical findings.**

***Index Terms*—Cognitive model, human performance modeling, night vision, pedestrian detection, queueing network.**

I. Queueing Network Model of Pedestrian Detection and Driving

DRIVER performance is a critical factor in examining the effectiveness and efficiency of an intelligent transportation system (ITS) [1], [2]. Computational models of driver performance can help study driver behavior and assist system design. This study introduces an example of using a computational model to assess driver performance using ITSs (two night-vision-enhancement systems designed to assist in pedestrian-detection tasks). In this study, the near-infrared (NIR) night-vision-enhancement system and the far-infrared (FIR) system were investigated. The NIR system actively illuminates a scene in the NIR spectrum and captures the reflected radiation, whereas the FIR system generates images by passively detecting thermal emissions from objects in a scene of interest.

Manuscript received October 23, 2008; revised April 18, 2010; accepted April 27, 2010. Date of publication June 7, 2010; date of current version December 3, 2010. The Associate Editor for this paper was N. B. Sarter.

J. H. Lim was with the Department of Industrial and Operations Engineering, University of Michigan, Ann Arbor, MI 48109 USA. He is now with the Mobile Communication Division, Samsung Electronics, Seoul, 137-857 Korea (e-mail: smilelim@gmail.com).

Y. Liu is with the Department of Industrial and Operations Engineering, University of Michigan, Ann Arbor, MI 48109-2117 USA.

O. Tsimhoni was with the University of Michigan Transportation Research Institute and Department of Industrial and Operations Engineering, University of Michigan, Ann Arbor, MI 48109 USA. He is now with the General Motors Advanced Technical Center-Israel, Herzliya 46725, Israel (e-mail: omer.tsimhoni@gm.com).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TITS.2010.2049844

The two night-vision systems generate distinctively different images of the same scene, which may lead to different glance behaviors, such as different glance frequencies and search times to detect pedestrians during night driving. In this study, pedestrian detection with a concurrent driving task was simulated with a computational cognitive model, based on the queueing network architecture, to investigate the different glance behaviors associated with the different night vision systems.

As shown in Fig. 1, the cognitive agent (driver) processes information from the images displayed by the night-vision-enhancement systems and then generates glance behaviors (scans the scene for pedestrians). The relationship between the characteristics of a visual scene (input stimulus) and the glance behaviors has been examined in empirical studies [3]–[6]. The computational modeling approach allows us to investigate the possible underlying cognitive mechanisms of the glance behaviors.

Along the line of research on developing comprehensive human performance models with a unifying cognitive architecture, we have been making steady progress in developing a queuing network (QN) architecture for human performance modeling. Mathematical models based on QNs have successfully integrated a large number of models in response time [7] and in multitask performance [8] as special cases of QNs. A computational model based on the QN mental architecture called the queueing network—model human processor (QN-MHP) [9] has been developed to simulate and generate human performance. This architecture represents a human psychological system in the form of a queueing network with multiple servers that represent functional units in a human brain. Entities represent pieces of information to be processed by the servers. An entity travels on routes, which represent the flow of information in the system. The QN-MHP has successfully been applied to model various task domains including simple and choice reaction tasks [10]–[12], visual search tasks [10], [13], driving [14], [15], driving with a concurrent map reading task [7], and driver workload [16]. For a detailed description of the QN-MHP and its applications methodology, see [9].

Adopting the QN-MHP methodology, a QN menu search model was developed in our earlier study [13], which was significantly extended in the present study to establish a QN model of pedestrian detection. There are nine effective servers for the menu search model used in the previous study [13], which were kept in this pedestrian-detection model. A

QN-Reinforcement Learning-Eye Movement

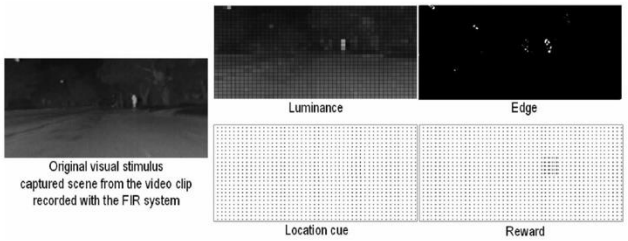


Fig. 3. Example of the input data for the pedestrian-detection task.

Fig. 4. Example of eye movements for the last three glances before response.

For reward ($f = 4$), a constant positive number was assigned to the location where the pedestrian appeared and zero to the rest. An example of the input data sets is shown in Fig. 3.

$\boldsymbol{InputSimulus}_{x,y,f,t}$
where

x horizontal location ($x = 1, 2, \ldots, 48$);
y vertical location ($y = 1, 2, \ldots, 25$);
f feature ($f = 1$ for the luminance, 2 for the edge, 3 for the location cue, and 4 for the reward);
t time ($t = 1, 2, \ldots, 20$).

When the eyes were fixated on a specific location, the two types of visual stimuli were encoded based on the data array $\boldsymbol{InputSimulus}_{x,y,f,t}$. One stimulus, representing attended visual information, carries information on all the three features (luminance, edge, and location cue). The other type of stimulus, representing preattended visual information, carries only information on one feature, which is currently selected as the hunt feature by the cognitive system.

The sets of visual stimuli were the scenes displayed by two different night-vision-enhancement systems—NIR and FIR—to assist pedestrian detection during night driving. Due to the different sensing mechanisms, the two night-vision systems generate distinctively different scenes, as shown in Fig. 4.

3) Eye-Movement Strategy for the Pedestrian Detection: To model the pedestrian-detection task, the hunt-feature rule and a reinforcement learning process were used as the procedure rule for eye movements. The hunt-feature rule explained the procedure of the varying length in a saccade phenomenon based on the attentional spotlight theory [21] and the feature synthesis theory [22]. The reinforcement learning process allowed the cognitive system to select a visual cue as a hunt feature based on the computed values of selecting the features.

Lim and Liu [23], [24] showed that eye-movement patterns changed according to the combinations of the probabilities of selecting each feature as a hunt feature. Therefore, the reinforcement learning process was integrated into the QN model to

Behavior Compounds: Compound_BE

Examples:

1. Tracking_1D
2. Tracking_2D or 3D
3. Tracking_1D on path
4. Tracking_2D or 3D on path
5. Tracing_1D
6. Tracing_2D or 3D

QN-Behavior Elements

- Available (in software for use, developed by QN developers)

  e.g., "See"

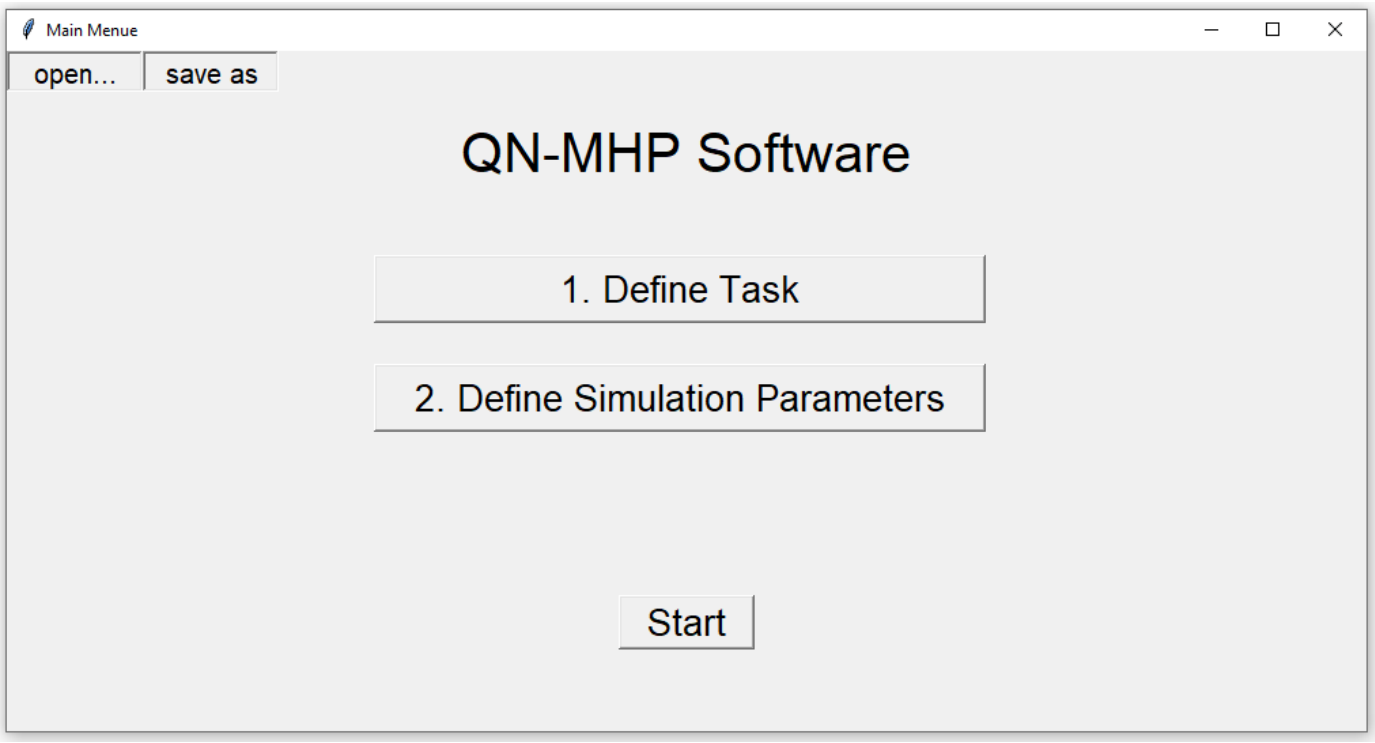
- Activated (by model users for a task)

  e.g., "visual choice RT task"

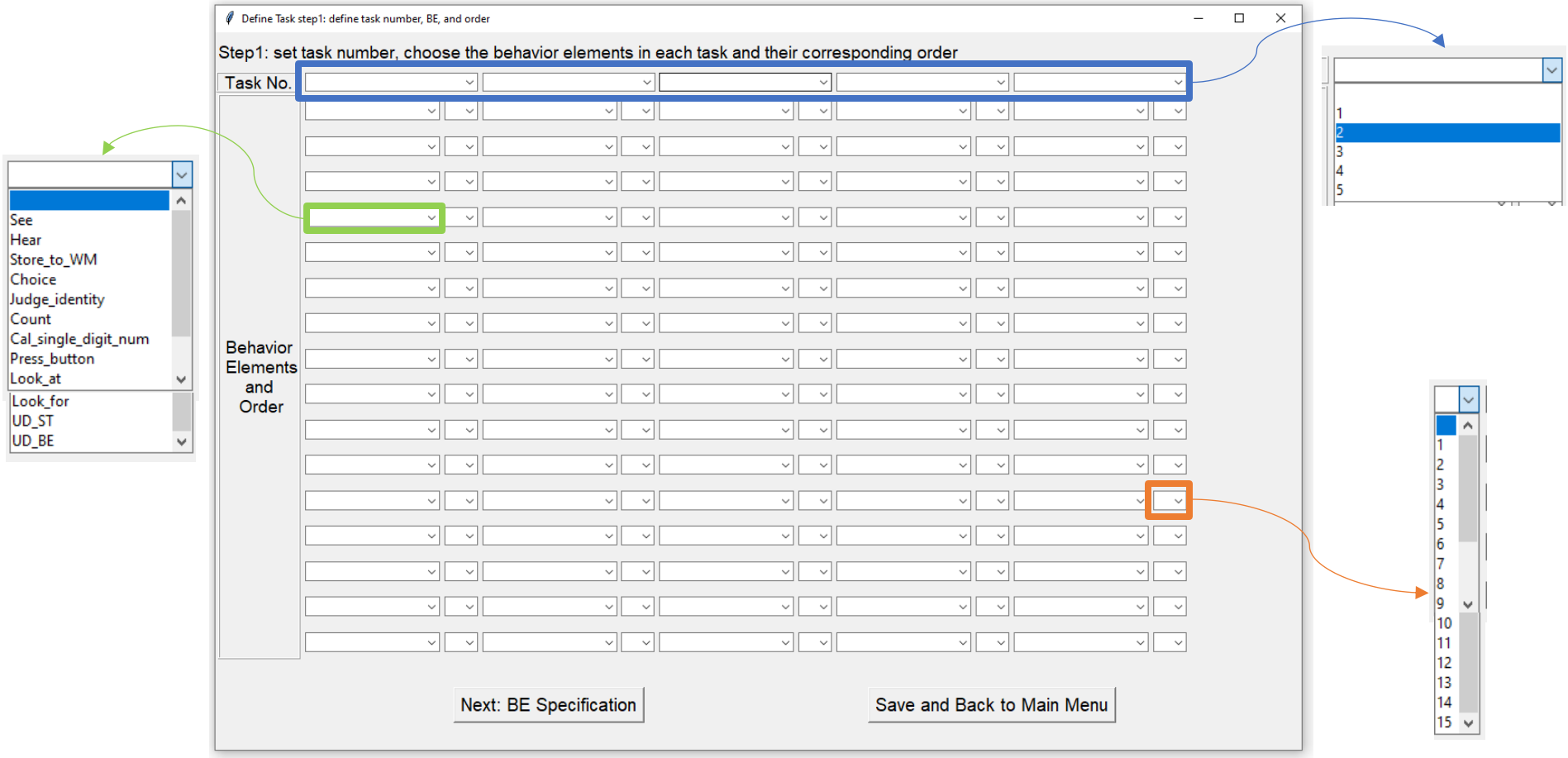
- Enabled (during simulation, when the BE's conditions are met)

  e.g., "when visual entity (e.g., color, text) arrives at Server 1"

Main Menu

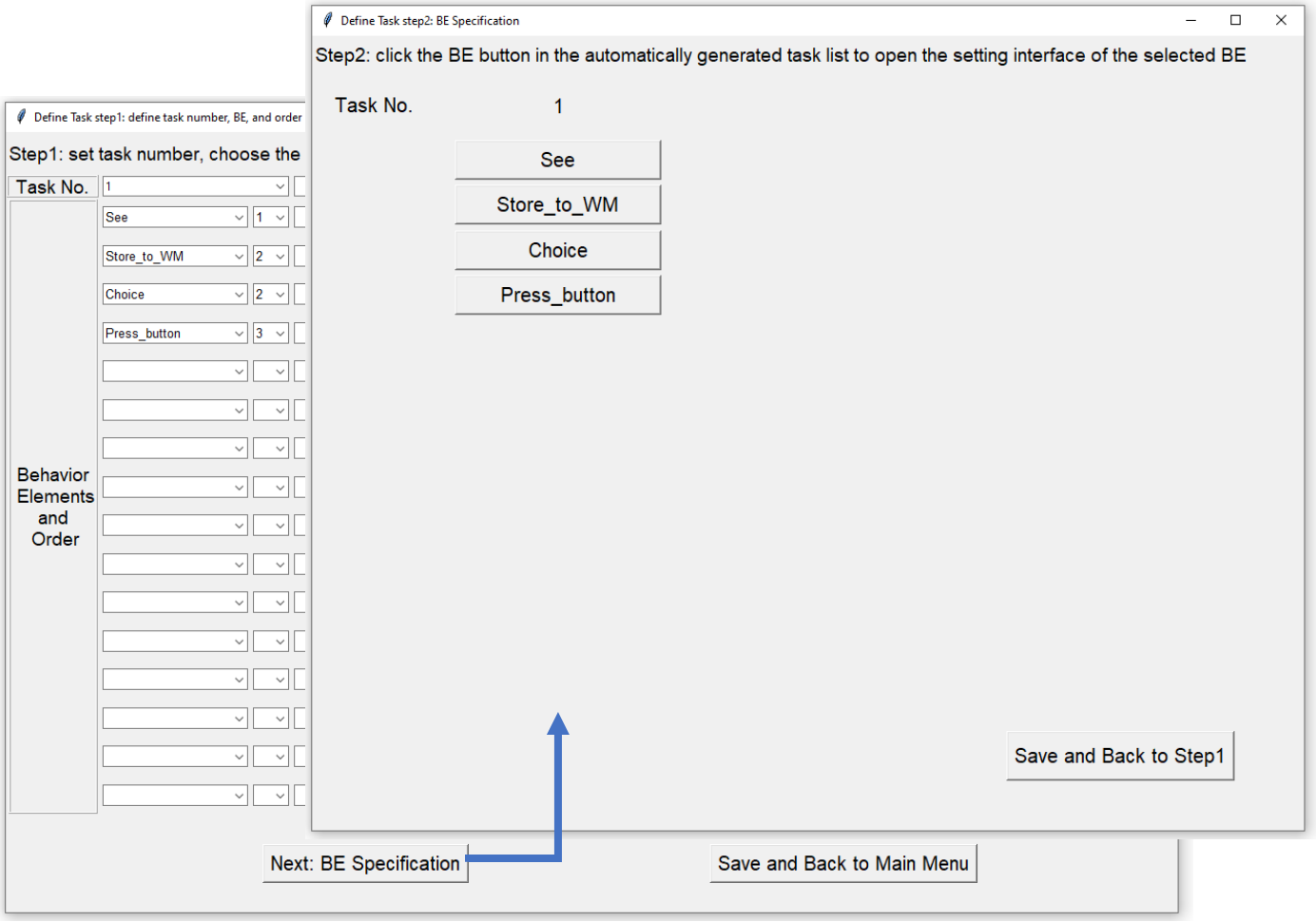

Main Menue
open...
save as
QN-MHP Software
1. Define Task
2. Define Simulation Parameters
Start

1. Define Task-step 1: define task No, BE and order

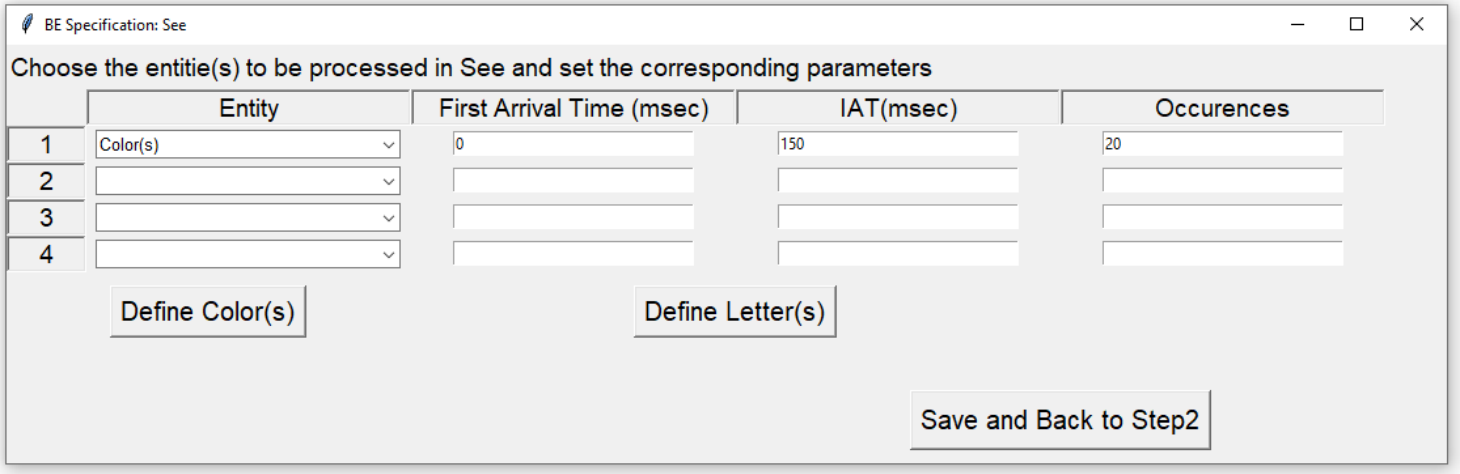


1. Define Task-step2: BE specification

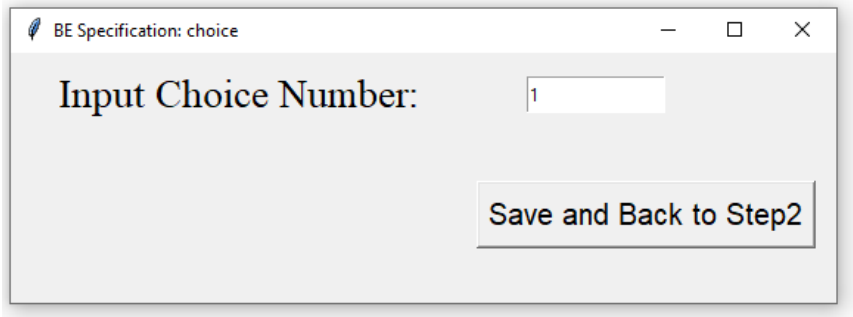
E.g. simple RT task



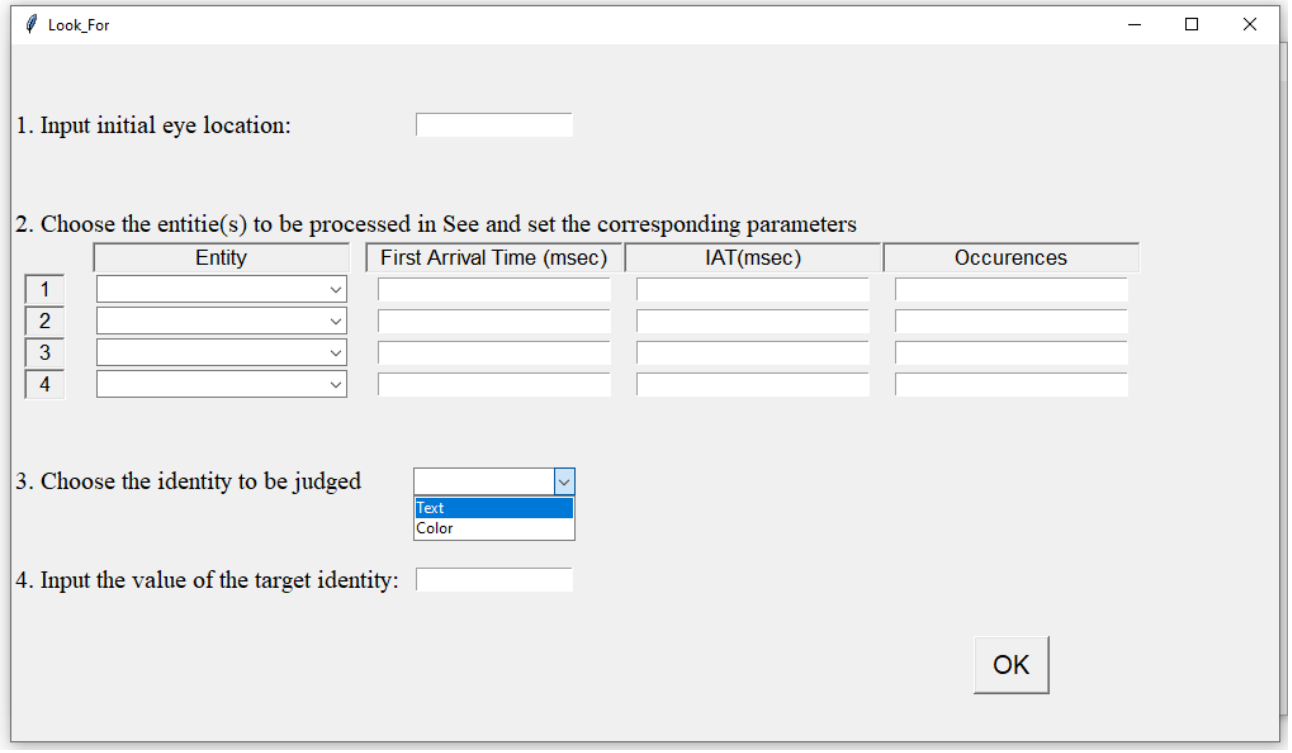
BE specification: see



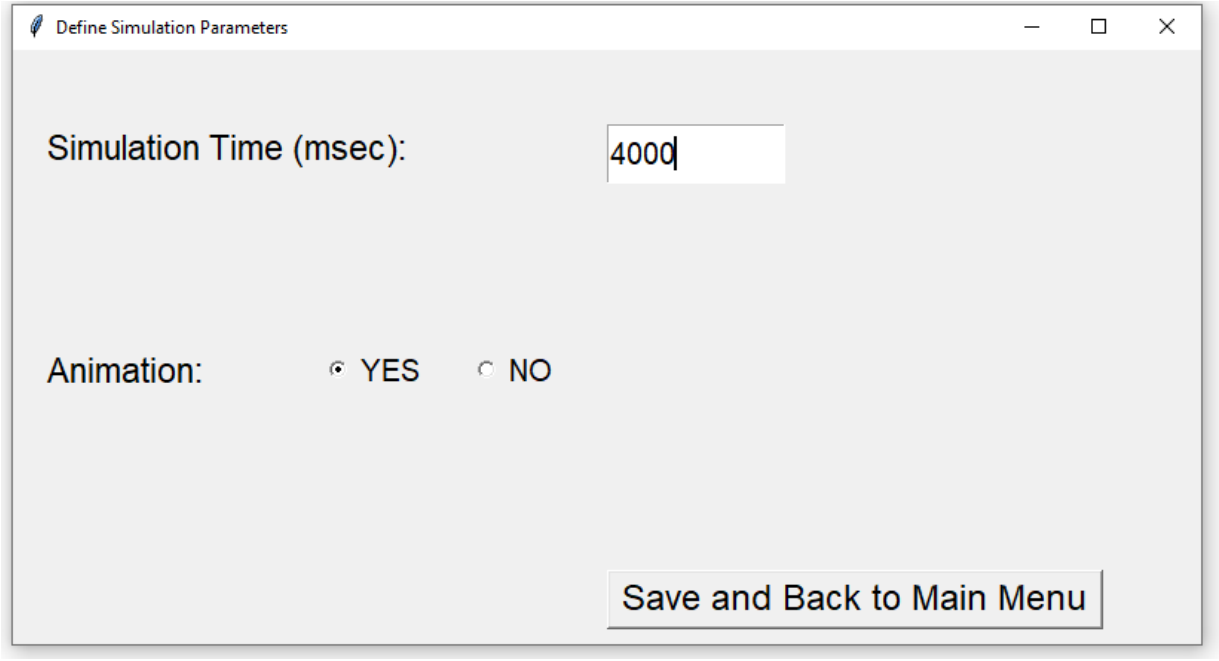
BE specification: choice



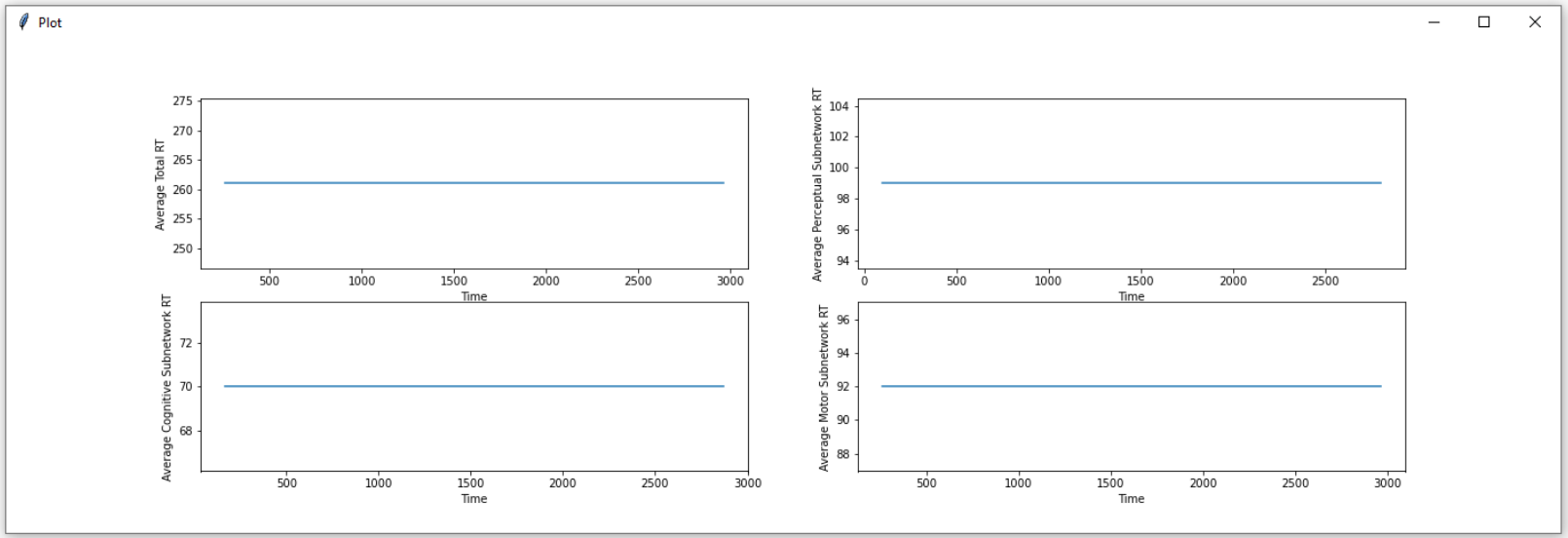
BE specification: Look for



2. Define Simulation Parameters



Plot for simpleRT task



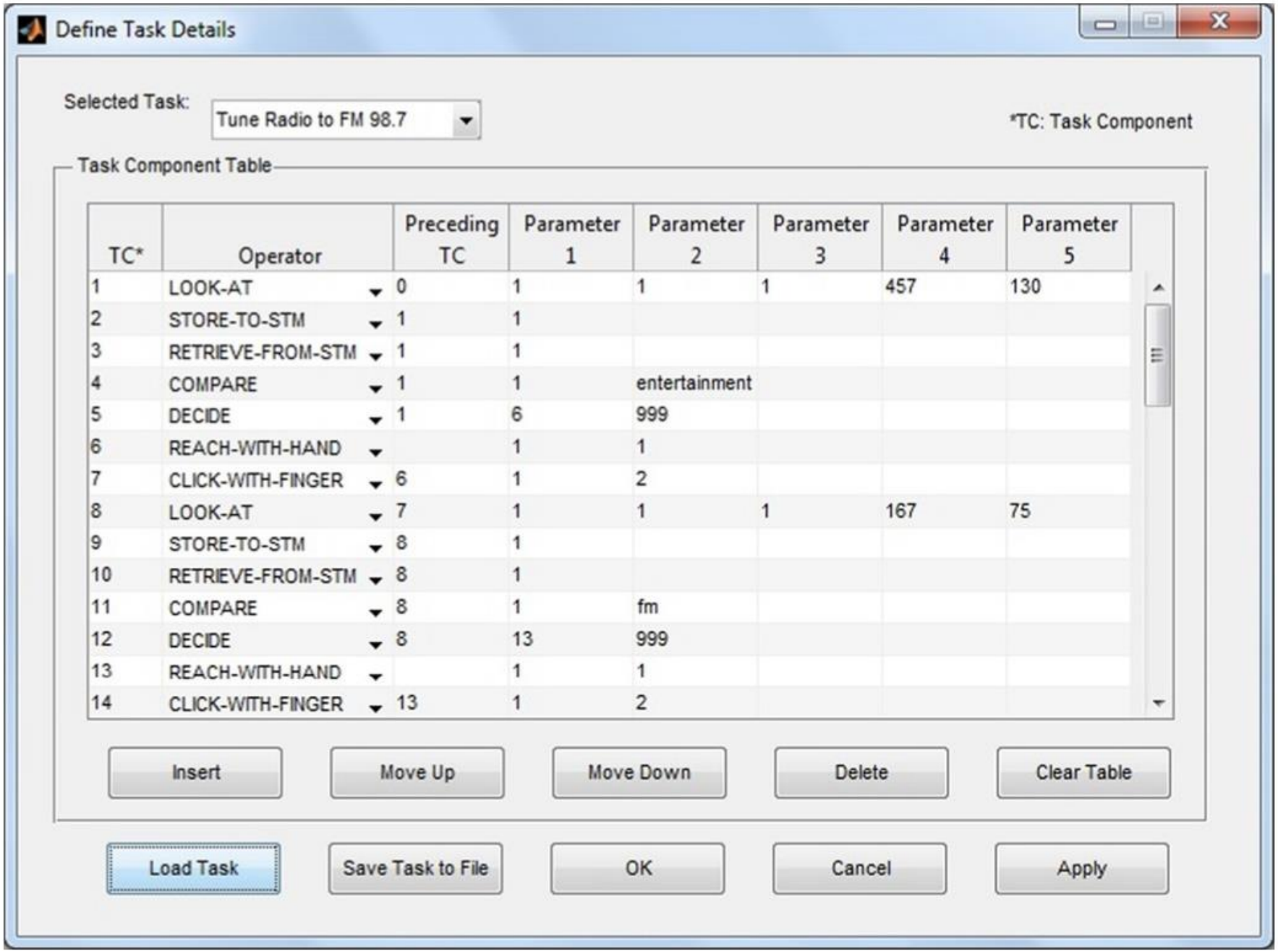


Fig. 6. Task specifications with QN-MHP task components.