Queueing Network-Model Human Processor (QN-MHP): A computational architecture for multitask performance in human-machine systems

Article *in* ACM Transactions on Computer-Human Interaction · March 2006

DOI: 10.1145/1143518.1143520 · Source: DBLP

Yili Liu
Ateneo de Manila University

Robert G. Feyen
University of Minnesota Duluth

Omer Tsimhoni
General Motors Company

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

Yili Liu
Robert Feyen
Omer Tsimhoni

Technical Report number 04-05
June 6, 2004

University of Michigan
Industrial and Operations Engineering
1205 Beal Avenue
Ann Arbor, MI 48109

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

YILI LIU, ROBERT FEYEN, OMER TSIMHONI

Technical Report 04-05

Department of Industrial and Operations Engineering

University of Michigan, Ann Arbor, MI, USA

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

YILI LIU
University of Michigan, Department of Industrial and Operations Engineering
and

ROBERT FEYEN
Purdue University, School of Industrial Engineering
and

OMER TSIMHONI
University of Michigan, Department of Industrial and Operations Engineering

---

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 generating in real-time concurrent activities in a truly concurrent manner. QN-MHP expands the three discrete serial stages of MHP 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.

---

1. INTRODUCTION

The increasing complexity and cost of advanced multitask, multi-modal, interactive human-computer systems make it necessary for system designers to consider human capabilities and limitations as early as possible in system design. In order to support quantitative tradeoff analysis and evaluation of design alternatives from the perspective of human performance, it is important to develop comprehensive and computational engineering models of human performance and human-computer interaction (HCI) (e.g., Adams, Tenney, and Pew, 1991; Elkind, Card, Hochberg, and Huey, 1990; Kieras and Meyer, 1997). Comprehensive models are not restricted to a specific aspect of human performance but attempt to model several aspects of human performance in a single, coherent structure. Computational models are expressed using computational methods or languages and support quantitative analyses or prediction of human performance and offer a rigorous understanding of the perceptual and cognitive processes underlying human performance.

Two publications three decades ago set the stage for this endeavor of developing comprehensive and computational models of human performance. In 1973, Allen Newell published a book chapter entitled “You Can’t Play 20 Questions with Nature and Win” (Newell, 1973), in which he advocated the development of unified theories of cognition (UTC) and made theoretical unification of micromodels and theoretical constructs an immediate and principal goal. In the same year, Anderson and Bower (1973) published the HAM theory of human memory, initiating an effort to create a theory of cognition that meets the twin goals of precision and complexity. Three decades of creative efforts of numerous researchers have led to the development of several important UTCs or harbingers to UTCs, including the Model Human Processor (MHP) and the GOMS family of models (Card, Moran, Newell, 1983; John and Kieras, 1996a, 1996b; Olson and Olson, 1990), ACT-R and ACT-R/PM (Anderson and Lebiere, 1998), SOAR (Newell, 1990; Laird, Newell, Rosenbloom, 1987), CAPS (Just and Carpenter, 1992), and EPIC (Meyer and Kieras, 1997a, 1997b) (see the right-half of Figure 1). These models have been successfully applied to modeling a large variety of tasks and demonstrated particular strengths in modeling and/or generating the detailed procedures and actions that a person might choose when interacting with an interface or system. ACT-R and SOAR have also enjoyed great success in modeling the learning process a person might go through. GOMS has also been used to model learning time and the difference between expert and novice performance (Kieras, 1988, 1999).

Continuing along this line of research, we describe in this article our modeling approach that is significantly different but complementary to the 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. 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 the 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, 1966; Rouse, 1980) and the parallel processing models (e.g., Laughery, 1989; Wickens and Liu, 1988) as special cases (Liu, 1994, 1997).

The rationale for us to develop this complementary approach is based on our analysis of the strengths and weaknesses of the procedure/production-systems approach and the queueing network approach. Although the procedure/production-system based models use mathematics in analyzing some of the specific aspects, mechanisms, or operations of their models (e.g., spreading activation and the 20 fundamental equations in ACT-R, algebraic formulation of the Psychological Refractory Period (PRP) in EPIC), they lack mathematical frameworks for representing their overall "architectures"—i.e., representing the interconnected arrangements of all the perceptual, cognitive, and motor components and their interaction patterns in one coherent mathematical structure. These models are symbolist or symbolic models (Card and Newell, 1990), not mathematical models. In contrast, although queueing networks are particularly suited for mathematically representing and analyzing these "architectural arrangement and network interaction" issues, if it is used as a pure mathematical theory, it cannot be used to generate detailed actions of a person in specific task situations. It does not represent the procedural knowledge a person may employ in accomplishing his/her specific goals. Clearly, the symbolist approach and the queueing network approach are complementary and the challenge is to develop a modeling approach that bridges the gap between the two approaches.

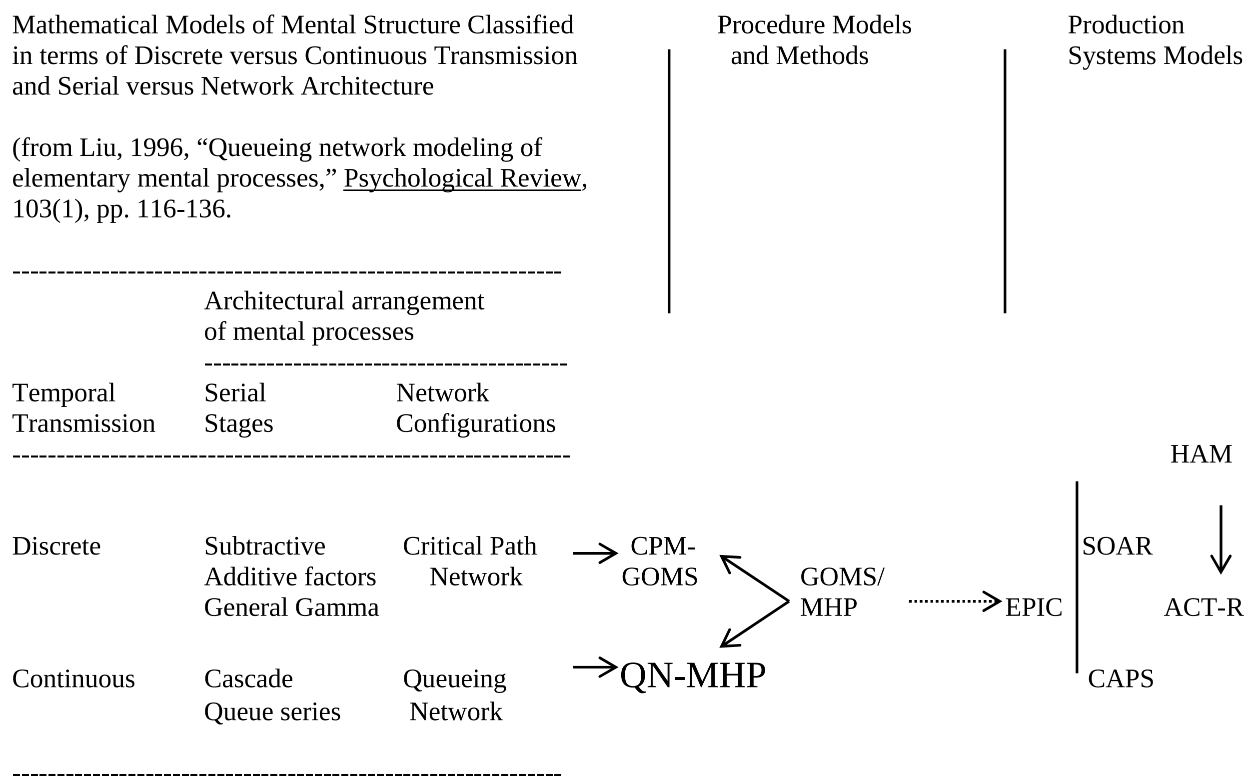

Figure 1: Mathematical models of mental structure and procedure/production system models of cognitive architecture.

In this article we describe our modeling approach that is called the Queuing Network-Model Human Processor (QN-MHP). As suggested by its name and as described in detail below, the QN-MHP is based on the modeling philosophy and methods of both the queueing networks approach and the procedure/production systems approach. Further, it offers a natural connection to the large body of human engineering modeling work in systems engineering and human-machine system simulation (Senders and Posner, 1976; Rouse, 1980; Sheridan, 1972; Laughery, 1989; Baron et al., 1980; Wherry, 1976).

Scientifically, the QN-MHP moves further along the direction of achieving Newell’s dream of unifying and “synthesizing what we know” (Newell, 1990; p. 16). By naturally integrating two powerful, complementary, but currently disjoint methods and modeling philosophies, we can achieve a broader and deeper understanding of human cognitive architecture and performance. For example, the QN-MHP offers another perspective from which we can examine some important theoretical issues such as whether the cognitive processor is serial (ACT-R’s view) or parallel (EPIC’s view), and whether there exists any simpler alternative approach to multitask modeling than interleaving production rules (in ACT-R) or specifying executive processes (in EPIC).

From the applications point of view, the QN-MHP is implemented in an easy-to-use general-purpose simulation software package (ProModel Optimization Software Suite, Harrell et al., 1995) that is widely used across the world, and our ProModel implementation can be transferred to any general-purpose simulation software. An easy-to-use Excel interface has also been developed to further facilitate the simulation. The QN-MHP simulation requires minimal learning time, and allows an analyst to visualize in real-time the QN-MHP internal network running processes, in addition to the final simulation statistical outcomes. These features are valuable not only for interface analysis but also for promoting teaching, training, and learning in cognitive analysis and modeling.

In this article we first briefly review the existing symbolist, mathematical, systems engineering, and connectionist models of human performance, which provide a broader background for our modeling work. Then we focus on a comparison between the symbolist models and the QN-MHP in multitask modeling. We then describe the structural components, assumptions, and methods of the QN-MHP, followed by a description of how the QN-MHP was used to model in a truly concurrent manner two concurrent activities of driver performance—steering and using secondary instrument panel displays. Empirical validation of the model is reported in the subsequent section.

The article concludes with a discussion of the implications of the queueing network modeling approach and a brief description of our related on-going and future research.

2. SYMBOLIST, MATHEMATICAL, SYSTEMS ENGINEERING, AND CONNECTIONIST MODELS OF HUMAN PERFORMANCE

2.1. Symbolist Models of Cognitive Architecture

MHP/GOMS family of models. The MHP was developed as an engineering model of human-computer interaction (HCI), which consists of a series of three processors—perceptual, cognitive, and motor processors, and some general purpose memory stores. Closely related to the MHP is the GOMS family of models and techniques, including the original formulation of Card Moran and Newell (1983), the Keystroke-Level Model, the CPM-GOMS, the NGOMSL and the recent developments of GOMSL and GLEAN3 (Kieras, 1999). John and Kieras provided a thorough and systematic analysis and comparison of the characteristics, strengths, objectives, and research needs for these methods (John and Kieras, 1996a, 1996b).

EPIC. EPIC is similar in spirit to the MHP, but EPIC incorporates many recent theoretical and empirical results about human performance in a computer simulation software framework. EPIC is a generative model in that it generates actions in simulated real time depending on task and environmental situations. EPIC consists of interconnected software modules for perceptual and motor processors and a cognitive processor that has a production rule interpreter and working memory, long term memory, and production memory. EPIC is a production based model that allows parallel processing at the cognitive stage. As indicated by its name, central to EPIC is the notion of "Executive Processes Interactively Controlling (EPIC)" the concurrent "task processes."

ACT-R. As stated by Newell, ACT*, the predecessor of ACT-R, "is the first unified theory of cognition" (Newell, 1990, p. 29). ACT-R distinguishes between declarative knowledge represented as chunks and procedural knowledge as production rules. The

ACT-R architecture consists of three modules: declarative memory, procedural memory, and goal stack. It is a fixed-attention architecture, in that at any point in time, it is focused on a single goal and a single production fires (Anderson and Lebiere, 1998). ACT-R 5.0 (formerly ACT-R/PM) incorporates an EPIC-like perceptual-motor system with ACT-R's original model of cognition (e.g., serial processing, conflict resolution and learning mechanisms, which EPIC does not have).

SOAR. Soar also uses a production system as the foundation of its architecture. Unlike ACT-R, it does not make a distinction between declarative and procedural memory. Its memory is entirely procedural and it learns by chunking (Newell, 1990). Like ACT-R, it is particularly strong in modeling problem solving and learning. However, "interactions between perception and motor control are not yet characterized sufficiently in SOAR. Nor does SOAR have much to say thus far about multiple task performance" (Meyer and Kieras, 1999; p. 22). Further, "In recent years the cognitive modeling use of SOAR has been receiving less attention" (Anderson and Lebiere, 1998; p. 447).

CAPS. CAPS is a production system based cognitive architecture for "vertical integration" (Newell, 1990) and it focuses on short-term memory, reading and language. Considering the focus of this article on multitask HCI modeling, we will not discuss CAPS further. Excellent and broad comparisons of the four models above can be found in, e.g., Anderson, John, Just, Carpenter, Kieras, and Meyer (1995, pp.9-12)**,** Anderson and Lebiere, (1998, pp.439-450), and Meyer and Kieras, (1999, pp. 20-22).

2.2 Mathematical Models of Mental Architecture

For over a hundred years since Donders's pioneering work of "subtractive method" (Donders, 1868/1969) and particularly since Sternberg's work of "additive factors method" (Sternberg, 1969), mathematical psychologists have been examining the issue and developing mathematical models of mental structure. These models use reaction time (RT) as the primary performance measure to infer the possible structure of mental systems. Two of the major issues of debate are whether the mental structure has a serial or network architecture and whether information transmission is discrete or continuous. The left half of Figure 1 shows the architectural dimension distinguishing serial stage models from network models and the temporal dimension distinguishing discrete-transmission models from continuous-transmission models.

As shown on the left half of Figure 1, Donders' "subtractive method", Sternberg's "additive factors" method, and "general gamma" models assume a serial stage architecture and discrete information transmission—a mental process cannot start until all of its preceding processes are completed. The cascade model (McClelland, 1978) and the queue-series model (Miller, 1993) preserve the serial stage architectural assumption, but consider continuous information transmission. The Critical Path Network model of Schweickert (Schweickert, 1978) preserves the discrete transmission assumption, but considers network arrangement of mental processes. The queueing network model (Liu, 1996) allows continuous transmission and network arrangement, treats discrete transmission and serial arrangement as special cases, mathematically integrates all the aforementioned models as special cases, and is able to model mental architectures such as "feedback" and "bypass" that cannot be modeled with previous models.

2.3. "Systems Engineering and Simulation" Models of Human-Machine Interaction

In parallel with the efforts of cognitive psychologists and mathematical psychologists, systems engineers have been developing engineering models of human performance for the primary purpose of engineering analysis and applications. Prominent among these are the single-channel one-server queueing models of human monitoring behavior and multitask sequencing and simulation models of complex task performance. The single-channel one-server queueing models postulate that the human is a single channel processor that quickly switches and allocates its processing capacity among a variety of tasks in a sequential and all-or-none fashion. These models include the instrument monitoring and visual sampling models (e.g., Senders, 1966; Carbonell, 1966; Senders and Posner, 1976; Schmidt, 1978), one-server queueing models of Rouse and his colleagues (Rouse, 1980; Chu and Rouse, 1979; Walden and Rouse, 1978, Greenstein and Rouse, 1982), and engineering models of attention allocation and task selection (Sheridan, 1972; Siegal and Wolf, 1969; Kleinman and Curry, 1977; Tulga and Sheridan, 1980; Pattipati, Kleinman and Ephrath; 1983). Several simulation models of human performance, e.g., the MICROSAINT model (Laughery, 1989; Laughery and Corker, 1997), the CRAWL model (Thompson and Bateman, 1986), and the WINDEX model (North and Riley, 1989), have started to accommodate some of the parallel processing assumptions of the multiple resources models. These models assume that the human has multiple pools of resources, which can be simultaneously allocated to multiple tasks in a graded fashion and allow parallel processing of simultaneous tasks as along as the needed resource(s) is available (e.g., Wickens, 1980; Wickens and Liu, 1988). “We must recognize that people in fact can do more than one thing at a time and normally do (Adams, Tenney and Pew, 1991; p.5).”

To model task performance that cannot be modeled or intuitively modeled by the serial and the parallel processing models and to model “the two most important features of macromodel: task selection and simultaneous execution” as identified by a report of the Committee on Human Factors of the National Research Council (Baron, et al., 1990), we have successfully employed a simple three-node queueing network to show that queueing networks can model intuitively and quantitatively a variety of tasks and that serial and parallel processing engineering models are special cases of the hybrid structure of queueing networks (Liu, 1994, 1997).

2.4. Connectionist Models

It is important to note that connectionism is a major approach to cognitive modeling (McClelland, Rumelhart, and the PDP Research Group, 1986). Connectionist models have shown great modeling capability in a large range of modeling areas including pattern recognition, motor behavior, language processing, and memory and learning. Anderson and Lebiere (2002) evaluated classical connectionism and ACT-R theory with regard to their strengths and weaknesses.

Because connectionism has been silent about multitask performance, which is the focus of the present article, we will not delve deep into connectionist models in this article. It is important to emphasize that although both connectionist models and queueing networks are network models, queueing network models adopt a small number of processing nodes (servers) that are at a much more macro level than the large number of neurons of the connectionist neural level. The relationship between the QN-MHP servers and the connectionist neurons is a very worthy topic for future research, although it is beyond the scope of the current article.

3. A COMPARISON OF THE SYMBOLIST MODELS AND THE QN-MHP IN MULTITASK MODELING

Although the MHP is very useful for engineering approximation to simple HCI tasks, its scope is seriously limited and “did not go much beyond the boxes-and-arrows phase of theory development” (Meyer and Kieras, 1999, p. 21). CPM-GOMS is developed to model parallel activities (Gray, John, and Atwood, 1993). However, CPM-GOMS is in a sequence form of schedule charts, not in an executable program form, and cannot be used to generate behavior. “Constructing the schedule charts required to analyze an interface design is quite labor-intensive…the CPM-GOMS method is recommended for predicting execution time only when there is a small number of benchmark tasks to be analyzed or when ‘task templates’ are available and can be reasonably assembled for task in question” (John and Kieras, 1996a; John and Gray, 1995; Kieras, Wood, and Meyer, 1997). This requirement, as well as its inability to generate behavior, severely restricts the applicability and usability of CPM-GOMS in modeling many complex multitask HCI situations. GLEAN3 is a further development along the GOMS approach. It is a procedure-based model in an executable form, but GLEAN3 currently assumes serial cognitive processing (Kieras, 1999).

EPIC represents a major advancement in multitask modeling. It is a production system model that allows parallel processing at the cognitive level. This assumption forms a sharp contrast to ACT-R’s serial cognitive processing assumption. “Because productions can fire in parallel, EPIC is left making predictions such as that a person can simultaneously solve a mental arithmetic problem and a mental multiplication problem (Byrne and Anderson, 2001)”. Byrne and Anderson reported a PRP experiment designed to expose EPIC’s problems in this regard and showed that humans may not be able to perform two complex cognitive tasks at once.

It should be noted here that from the theoretical point of view, although Byrne and Anderson’s experiment raises serious doubts about human ability to perform two *complex* cognitive tasks at once, it does not logically imply that human cognitive system cannot perform *any* two or more activities at once at all. From the practical application perspective, as shown in Salvucci and his collaborators’ work on driver interface modeling with ACT-R, ACT-R’s serial assumption makes it difficult or complicated to model multitask driving tasks such as in-car interface use and driver distraction

(Salvucci, Boer, and Liu, 2001; Salvucci, 2002). “Because of its implementation in the ACT-R architecture, the model is constrained to a serial line of cognitive processing (Salvucci, 2002).” To model concurrent tasks with a serial processor, their only method is perhaps to devise complicated and task-specific programs to interleave production rules of concurrent tasks into a serial program, as in Salvucci and Macuga (2001) and Salvucci (2002).

It should also be noted that although EPIC assumes parallel cognitive processing, its reliance on executive processes to “interactively control,” “supervise,” or “lock/unlock” the detailed progresses of each task may have theoretically assigned too much “intelligence” or “hard labor” to the executive processes and practically made it exceedingly difficult for a task analyst to model complex tasks. To model multitask performance with EPIC, one needs to write production rules not only for each task, but also for the executive processes, and they are extremely complicated even for simple multitask scenarios. “The actual control regimens that have been programmed in published EPIC models are very difficult to understand, and it stretches credibility to believe that such control structures could ever have been learned” (Anderson and Lebiere, 1998, p. 442).

As stated by Newell, “Often, as well, when Act* predictions are examined closely, much of the predictions depends on aspects of the Act* *program* rather than the Act* *architecture* itself” (Newell, 1990; p. 28). This comment of Newell becomes particularly relevant if a researcher has to write detailed, step-by-step, task-specific program codes, either in ACT-R for the serial cognitive processor to interleave production rules or in EPIC for the executive process(es) to “lock/unlock” task processes.

With these theoretical and methodological advancements and debates in mind, the QN-MHP attempts to make further advances along at least four dimensions for multitask HCI modeling: First, the QN-MHP offers a unique perspective from which we can examine the serial versus concurrent processing debate at the cognitive stage and the coordination among concurrent processes. As discussed in detail later, the cognitive processor in the QN-MHP is represented as a cognitive subnetwork of servers, each with a distinct cognitive function. It is logically possible that the separate servers in the “cognitive network” work concurrently, but certain servers within this network (such as the server responsible for complex cognitive functions such as mental arithmetic) can only process one “rule” at a time. In other words, the cognitive processor is a parallel and concurrent network of servers, some of which may (or may not) be serial.

Second, in the QN-MHP, multitask behavior emerges as an outcome of multiple tasks (represented as multiple streams of customers) traversing a network like trucks and cars traveling on the same highway, and much of the predictions will depend on aspects of the network architecture, as Newell advocated. Like ACT-R and EPIC, a modeler still needs to analyze each concurrent task; but unlike ACT-R and EPIC, one does not need to write separate and complicated codes to interleave or interactively control the tasks.

Third, although EPIC and ACT-R use mathematical methods to analyze specific aspects of their models (e.g., the algebraic formulation of PRP in EPIC and the conflict resolution mechanism and the 20 fundamental equations in ACT-R), they lack mathematical frameworks to represent their overall architectures, i.e., representing the interconnected arrangements of all the perceptual, cognitive, and motor components and their interaction patterns in one mathematical structure. In contrast, the QN-MHP represents its overall architecture as a queueing network—a major branch of mathematics and operations research, thus allowing comprehensive mathematical modeling. Further, since each of the QN-MHP servers is capable of performing procedural logic functions, the QN-MHP is able to generate detailed task actions and simulate real-time behavior, like EPIC and ACT-R.

Fourth, the QN-MHP is developed with a widely used general-purpose simulation package with sophisticated and easy-to-use simulation, visualization, and data analysis capabilities, and can be used by cognitive engineers with minimal special training, as mentioned earlier in this paper.

It should be noted that the primary goal of this article is to describe an integrative engineering model and an easy-to-use simulation technology for multitask HCI modeling. The focus is not on examining pure theoretical psychological issues or discriminating between pure psychological positions. The findings of our modeling work, however, do potentially offer interesting insights, possible alternative explanations, and preliminary supporting or opposing evidence to existing cognitive theories.

4. THE QN-MHP—ITS CURRENT STRUCTURE, ASSUMPTIONS, AND IMPLEMENTATION

The QN-MHP assumes that there is a close resemblance between a human cognitive system and a queueing network. The idea of a queueing network (QN) arises naturally when one thinks of a network of service stations (also called servers or nodes), each of which provides a service of some kind to the demanders for service (called customers), either immediately or after a delay. Each server has a waiting space for customers to wait if they cannot immediately receive their requested service, and thus multiple queues may exist simultaneously in the system. The servers are connected by routes thorough which customers flow from server to server in the network. Telecommunication systems, computer networks, and road traffic networks are examples of queueing networks.

As discussed earlier and shown in Figure 1, the Model Human Processor (MHP)/GOMS is the father of a large portion of the existing procedure/production system models of cognitive architecture. We started our efforts of integrating the QN approach with the procedure/production systems approach by adopting and modifying four major components of the MHP/GOMS: First, we decompose the three discrete stages of the MHP into three continuous subnetworks of QN servers based on neuroscience and psychological findings. Second, we adopt the corresponding MHP parameters such as decay rates and processing cycle times to define the performance characteristics of each QN server. Third, each server is defined with processing logics in a GOMS manner to perform certain procedural operations; and fourth, we use a GOMS-style method as the tool for task analysis. The resulting approach is thus called the QN-MHP. Its mathematical and simulation architecture is a queueing network, in which each server performs distinct procedural functions.

In order to decompose the three MHP processors into three QN-MHP subnetworks, we extensively reviewed neuroscience research findings to identify areas of the brain that might represent common cortical fields activated during the performance of a given task and to determine the primary connections between these fields (e.g., Carter, 1998; Frackiowiak, Friston, Frith, and Mazziota, 1997; Gilman and Newman, 1992; Martin, 1989; Pritchard and Alloway, 1999; Roland, 1993). Figure 2 shows the current structure of the QN-MHP. All models evolve; QN-MHP is not an exception. The structure of the QN-MHP may evolve as our research progresses and as new and relevant neuroscience and psychological evidences are discovered or identified. Most encouraging, as discussed later in this article, the current structure serves well in modeling a variety of tasks.

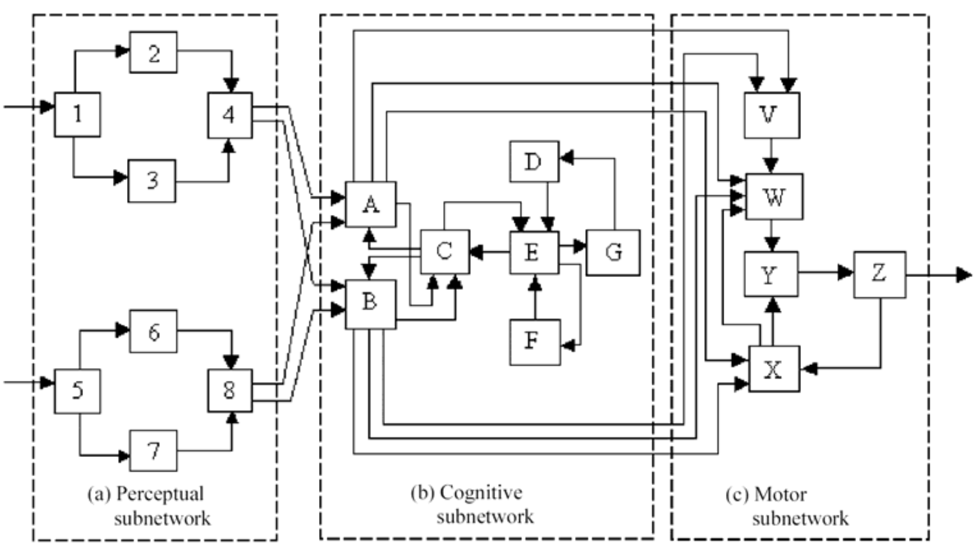

Figure 2. The Queuing Network-Model Human Processor (QN-MHP) (from Feyen and Liu, 2001). (a) Perceptual: 1=common visual processing; 2=visual recognition; 3=visual location; 4=location and recognition integrator; 5=sound localization; 6=linguistic processing; 7=processing of other sounds; 8=linguistic and other sounds integrator. (b) Cognitive: A=visuospatial sketchpad; B=phonological loop; C=central executor; D=goal procedures; E=performance monitoring; F=high-level cognitive operations; G=goal selection. (c) Motor: V=sensorimotor integration; W=motor element storage; X=movement tuning; Y=motor programming; Z=actuators.

Note: Although not shown in this figure, environmental and device servers receive output customers from server Z and supply input customers to server 1 and server 5.

4.1. Perceptual Subnetwork

The visual, auditory, and somatosensory systems are the typical sensory systems considered when modeling a person's perception of the surrounding environment. The current version of the QN-MHP considers visual and auditory perception only, each of which is represented as a subnetwork of four servers. Currently, only the visual network is implemented and tested.

Four servers are used in the visual system: common visual processing, visual recognition, visual location, and a visual integrator (integrating recognition and location information). Common visual processing (server 1 in Figure 2) is handled primarily by the eye itself, lateral geniculate nucleus (LGN), superior colliculus, the primary visual cortex (V1; Brodmann area 17), and the secondary visual cortex (V2; Brodmann areas 18 and 19). The visual field is broken into three regions of interest: foveal vision, parafoveal vision, and peripheral vision. At this server, the arriving visual entities (customers) are given different types of processing depending on the visual region in which they are located and are compared against other recent, similar entities for any obvious changes in state. If a change in state is identified, the entity is split and routed along two parallel paths. One path is used to recognize the object. Called the parvocellular stream (from its origination in the parvocellular region of the lateral geniculate), this path exits the parvocellular region of the secondary visual cortex (V2) and runs through the unimodal association area for color (V4; Brodmann's area 37) in the lower occipital lobe on its way to the inferior temporal cortex (Brodmann's areas 20 and 21) for form and pattern recognition. The QN-MHP assumes that various object characteristics such as labeling, color, size, shape, and orientation can be "unpacked" from the entity while traveling this path. The mechanisms that underlie this process (e.g., Biederman, 1987) are beyond the scope of current efforts.

The second path, the magnocellular stream, locates the object of interest in the field of view and detects any movement associated with the object (Courtney and Ungerleider, 1997). Entities following this path exit the magnocellular regions of V2 and pass through several unimodal association areas in the occipital and parietal lobes: V3 (depth and distance; Brodmann's area 19), and V5 (motion; medial temporal region of Brodmann's area 19). Once again, QN-MHP assumes that these characteristics – spatial coordinates, heading, and speed - can be extracted from the entity. One exception is that ascertaining heading and speed requires sampling of two or more subsequent entities by these same cortical fields to detect the temporal changes in depth and distance produced by motion.

Each of these paths comprises a server in QN-MHP: server 2 in Figure 2 represents the recognition path and server 3 represents the location and movement path. The last server, server 4, represents the lateral connections between the processing streams at different levels (e.g., connections between areas V3 and V4 as well as V4 and V5), which unify the various characteristics between the two split entities into a single stimulus. Depending on its characteristics, a customer at Server 4 may be routed either to the visuospatial sketchpad (server A in Figure 2) or the phonological loop (server B) in the cognitive subnetwork. Subsequent entities in the stimulus stream refresh the representative entity already in the QN-MHP so that, if the stimuli cease arriving such as when the eyes are closed for a period of time, the decay rate for the entity starts at the arrival time for the last entity in the stream. The decay rate, adopted from the MHP, is based on the half-life concept in which the time to decay, T, is exponentially distributed with parameter λ, with $\lambda = \ln(2)/t_{1/2}$ and $t_{1/2}$ = the decay half-life value. Entities in server 4 that are no longer being refreshed have a decay half-life of 200 milliseconds.

4.2. Cognitive Subnetwork

The cognitive subnetwork in the QN-MHP includes a working memory system and a goal execution system. Following Baddeley's model of working memory, the servers in the working memory system include a visuospatial sketchpad, phonological loop, and central executor. This tripartite division has been supported by neurophysiological evidence (see Baddeley, 1992, for a review). Functional magnetic imaging studies (fMRI) indicate that specific areas of the brain are activated when tasks associated with these three systems are performed (Baddeley, 1998).

Recent work suggests that the working memory system can be split even further (e.g., Roberts et al, 1996; Quinn and McConnell, 1996; Smith et al, 1996). Within the central executive, considerable evidence suggests the existence of a separate time-sharing executive subcomponent that may be crucial to switching attention between concurrent tasks. Episodic encoding (e.g., committing a list of words to memory) appears to involve separate subsystems for storage and retrieval, although the evidence for this is less convincing (as reviewed by Baddeley, 1998). One particularly interesting division, suggested by Duncan et al (1996), is a subsystem for maintaining memory of several different and simultaneous goals or tasks. When the various goals or tasks in a sequence can be recalled from long-term memory due to prior learning, performance apparently is not affected by this subsystem. However, when the performance tasks require novel problem solving or the recollection of several different instructions or subtasks (especially later instructions), impairments to this subsystem appear to affect task performance. Apparently, this system is crucial to certain types of goal-directed behavior by actively maintaining the memory of subsequent steps or tasks that cannot otherwise be recalled from long-term memory.

Along these lines, the QN-MHP employs separate servers to handle distinct goal execution functions such as goal prioritization, complex cognitive processing (e.g., search, mental rotation), performance monitoring, and procedure selection. Interestingly, these four servers correspond approximately to the four components of the GOMS concept. Evidence from fMRI studies indicate that, in most cases, distinctly separate regions of the prefrontal cortex appear to be activated by the various tasks that may be classified under each of these general functions. Encouragingly, this four-server goal execution network served well in our completed task simulations.

4.3. Motor Subnetwork

All motor responses exiting the cognitive subnetwork are routed via the working memory structures in order to gather any additional perceptual information necessary to carry out the response (e.g., Roland, 1993). For example, to define the hand movement required by a response, the target coordinates in extrapersonal or intrapersonal space must be retrieved from the visuospatial sketchpad (i.e., using the internal coordinate map thought to be contained within the visual association area V6) before the response entity can be routed to the motor subnetwork.

The motor subnetwork contains four servers and currently two actuators: sensorimotor integration, movement tuning, motor element recall, motor sequencing or programming, the right hand, and the eyes. Sensorimotor integration (server V in Figure 2) is based on research regarding the role of the premotor cortex (Brodmann's area 6) in motor movements (Passingham, 1987; Roland, 1993; Pritchard and Alloway, 1999). These studies indicate that the premotor area is a necessary motor structure in situations where perceptual information must be used to finalize a response action. Movement tuning (server X) is suggested by the role of the cerebellum in adjusting motor actions. Fine tuning and timing of movement appears to be modulated by the cerebellum and red nucleus, in both feedback and feedforward modes (e.g., Gilman and Newman, 1992; Pritchard and Alloway, 1999). Server W handles motor elements recall from two structures identified in the basal ganglia: the putamen and the caudate (Carter, 1998; Roland, 1993; Rosenbaum, 1991). Once the motor elements are recalled, they are passed to the supplementary motor area (SMA, Brodmann's area 6, server Y in the QN-MHP)), which puts motor elements in their proper sequence and queues them until the overall movement is triggered, prior motor elements are complete, and the actuator required is available. Once these conditions are met, the motor element is passed to the primary motor cortex (MI, Brodmann's area 4).

The muscles surrounding the joints of the actuators (the right hand and eyes) receive the signals from the MI and carry out response execution. Each of the actuators (although there are two actuators, only one server, Z, is shown in Figure 2) is limited to processing one entity at a time. Regression equations to determine hand movement times are adapted from data in Methods-Time Measurement (MTM) tables for reaches and moves (Niebel, 1993). If visually guided, hand movements can be specified as either simultaneous with the eye movements or requiring the eye to locate the target before the hand movement can be initiated. Eye movement times are estimated using a constant angular velocity of 4 msec/degree (Kieras and Meyer, 1997). Eye movements are saccadic between targets, but can use pursuit movements to track a mobile target of interest.

4.4. Processing Functions at Individual Servers

The QN-MHP is currently implemented in ProModel, but it can be implemented in any general-purpose simulation program. The psychological functions of each of the QN-MHP servers is first defined as “server services” or “processing function/logic” independent of any simulation software, and then implemented in ProModel. A detailed description of the processing services of each server and their ProModel implementation is beyond the scope of this article, but can be found in Feyen (2002). Here we use Server 1 as an illustration. Currently, Server 1 (visual input server) performs the following services for incoming “visual customers”: First, it checks whether saccadic suppression is in effect (no incoming customer can be serviced if the eyes are currently moving); Second, if the eyes are not moving, it determines the location zone of the “visual customer” (whether it is from the fovea, parafovea, or peripheral region), its intensity (whether it is intense enough), its status (whether it is a “new” or an “old” one—e.g., at an arrival rate of one customer per 50 msec, a red light of 1000 msec duration may generate a stream of 20 customers, among whom only the first “noticed” one is “new”); the server further splits any “qualified” “new or old” customer into two entities and sends them to servers 2 and 3. For example, a qualified new customer must be intense enough to receive further processing in the network. A qualified old customer enters the network to “refresh” its associated stream in case its earlier “sibling” of the same stream fades away while competing for service in the network. Consistent with the massively parallel feature of perceptual processing, these functions are performed in parallel at Server 1.

4.5. Server Processing Times

A server that simply routes information to another server is assumed to require no processing time. However, if a server must perform a more complicated service (such as performing an arithmetic calculation), time must elapse. Following the most commonly adopted assumption in performance analysis, we currently assume that the server processing times are stochastic and exponentially distributed with mean X and an axis shift of Y [hereafter written as E(X,Y)]. To be consistent with the MHP, the QN-MHP requires the minimum processing time of a QN-MHP subnetwork (time to traverse a subnetwork) to be equivalent to the MHP Fast processing time of the corresponding MHP stage, and the average processing time equivalent to the Middle or Typical processing time of the MHP stage. On the basis of this consideration, the processing time for the perceptual servers is established as exponentially distributed with a mean of 42 msec and a minimum of 25 msec; for the cognitive servers a mean of 18 msec and a minimum of 6 msec; and for the motor servers a mean of 24 msec and a minimum of 10 msec. These parameter values have worked well in our completed work of simulating a wide range of tasks.

4.6. NGOMSL-Style Task Analysis and its Implementation in the QN-MHP

One major issue to resolve in combining the queuing network approach with a procedure-based approach is devising a task analysis methodology for analyzing and representing the procedural aspects of a task with the QN-MHP. Rather than devising a new method for task analysis, the QN-MHP relies on cognitive task descriptions obtained using the procedural methods described by Kieras for an NGOMSL analysis (Kieras, 1988, 1999). In addition to improved consistency between analysts, other advantages of Kieras' task analysis method is that it is explicitly and thoroughly described in the literature, relatively easy to learn and understand, has a short learning curve, and thus has gained wide acceptance in HCI community. John and Kieras (1996) state, for example, that NGOMSL can be taught to undergraduate students in a few class sessions and homeworks; full day tutorials at professional conferences also seem sufficient to get individuals started in using the method successfully. For the QN-MHP, the overall learning and task description time should be shorter since only the description method is needed -- much of the computational analysis done by hand in NGOMSL is addressed and implemented within the processing function/logic of the various servers of the QN-MHP. The only significant difference in describing a task is that the predefined set of operators in the QN-MHP varies somewhat from the set proposed by Kieras, although analyst-defined operators are still available in the QN-MHP.

Currently we have defined a library of 24 NGOMSL-style QN-MHP task analysis operators independent of any simulation software. Each operator has also been implemented in our ProModel QN-MHP simulation program as a subroutine or block of processing logic accessed at a particular server in the QN-MHP network. The 24 operators include 7 basic motor operators, 5 perceptual processing operators, 4 complex cognitive function operators, 5 memory access operators, and 3 procedure flow operators.

Five of the 7 implemented motor operators (reach to target, move object to target, apply pressure to object, release object, delay movement for specified time) are defined and executed at the right hand server. The “single saccade” operator is executed at the eye server, and the “preloaded motor trigger” is at the response tuning server and the motor programming server. Perceptual “control” operators are executed in the central executive server to initiate goal-directed (rather than random) eye movement and to wait for necessary environmental customers to arrive. The five implemented operators are: glance at target, watch target until stimulus data, compare stimulus data to cognitive function, verify stimulus data, and trigger action given stimulus. Complex cognitive operators are executed within the high-level cognitive function server. In order to be processed by a complex cognitive operator, information must either result from prior cognitive operations, been recalled from a retained entity or have been merged into the executive entity from a perceptual entity carrying data from the external world. Four complex cognitive operators are implemented: select search target, decide, compute, and time check. Memory access operators are executed within the central executive server to control the storage and recall of information associated either with a customer or long term memory. Four memory access operators are implemented: recall information from working memory, retain entity in working memory, retrieve information from long term memory, and forget all retained entities. Three procedural flow operators are implemented in the procedural list server: accomplish goal, report goal accomplished, and go to step number; and they are responsible for the proper sequencing of steps when executing a goal expressed in NGOMSL-style task analysis. Each of these “context-free” operators is assigned a numeric ID, ready for use by interface analyst.

4.7. Concurrent Goal Processing

In the QN-MHP, multiple goal lists can be processed simultaneously and independent of one another, simulating the ability to think and perform more than one thing at a time. The processing procedure for each goal is the same as in single-goal task modeling, but there is no built-in limitation for a single goal to be processed at any given time. Each entity flowing through the queueing network is associated with one of the goals to allow potential competition between goals at the server level. Depending on the capacity and utilization level of each server, entities associated with different goals can either be processed in parallel, or wait until the server is available. Priority decisions are made in real time locally, at the server level, rather than centrally at an executive level, as required when only one goal at a time can be processed. It is the flow patterns of the entities and potential congestions at the various servers, not the limitations of a particular, pre-designated central executive that produces task interferences. Using the road network as an analogy, travelers navigate a road network do not need moment-to-moment commands from a traffic controller. Congestions may occur at any road segment due to different traffic flow situations, not necessarily at a pre-specified particular bottleneck or due to the limitations of a particular traffic controller.

4.8. How to Perform a QN-MHP simulation

Like GOMS, QN-MHP modeling starts with "task analysis" that identifies the user's goals and methods to accomplish the goals—a common skill of cognitive engineers (Diaper, 1989; Kirwan and Ainsworth, 1992). The result is expressed as an NGOMSL-style "task description" using the QN-MHP task analysis operators described above. The next step is to use the numeric operator IDs to convert the task description to an excel sheet called "task description sheet." A task is performed in response to its associated environmental and/or device "stimuli." This information is specified in a separate Excel sheet called "stimuli sheet."

For multitask modeling, each task and its associated environmental/device information are analyzed separately and recorded into separate sections of the corresponding Excel sheets. For example, analyzing two concurrent tasks (say, steering and button-pressing) will result in two sections in the task description sheet and two sections in the stimuli sheet—one for each task.

To perform a QN-MHP simulation, an analyst uses either one "task description" Excel section and its corresponding "stimuli" section (for single task performance analysis) or multiple task description and their corresponding stimuli sections concurrently (for multiple-task analysis) as simulation inputs. Potential task interference emerges when two or more streams of "customers" traverse the network concurrently and compete for service at the various servers. Using the visualization features of the ProModel such as color coding, one can visualize in real-time the travel patterns of the customers and potential network congestions, in addition to many advanced time-series and statistical results as standard features of the simulation package.

5. DRIVER PERFORMANCE MODELING WITH THE QN-MHP

The QN-MHP has been successfully applied to model a wide range of tasks including simple and choice reaction time tasks (Feyen and Liu, 2001a, 2001b, 2002), visual search tasks (Feyen and Liu, 2001a, 2002; Lim and Liu, 2004a, 2004b), the psychological refractory period (PRP) phenomenon (Wu and Liu, 2004a), transcription typing task (Wu and Liu, 2004b, 2004c), and driver performance (Tsimhoni and Liu, 2003a, 2003b).

To illustrate how the QN-MHP models multi-task performance, in this section we describe how the QN-MHP was used to model driver performance of two concurrent activities: steering a vehicle to keep it inside the lane boundaries and performing an in-vehicle map reading task to obtain the displayed information. We first describe how the QN-MHP was used to model the steering task alone and how the model was interfaced with a real-time driving simulator and used to steer the simulator in real time. We then describe how the QN-MHP models the concurrent steering and in-vehicle tasks. The model was tested with the driving simulator; the testing methods and results are described in the following section.

5.1. The QN-MHP Model of Vehicle Steering

The input, output, and processing logic of the QN-MHP vehicle steering model are summarized in Table 1. They are defined and developed on the basis of several major concepts and findings in the driver performance literature.

The inputs to the steering model consist of vehicle heading, vehicle lateral position, and road curvature. The use of these inputs for the model is based on the role of focal and ambient visual systems in driving and the concept of near-far dichotomy. The role of focal and ambient systems in driving was first identified by Mourant and Rockwell (1970) and Leibowitz and Owens (1977) and further studied and confirmed by subsequent studies (Owens and Tyrell, 1999; Summala, 1998). A near-far dichotomy was first proposed by Donges (1978) as an explanation to steering behavior. Steering consists of a guidance level, which involves anticipatory open-loop corrections and requires glances far from the vehicle, and a stabilization level, which involves closed-loop corrections and requires closer glances. The question of where drivers look when they negotiate curves was investigated by Land and Lee (1994), who reported that when entering curves drivers made anticipatory glances of 1-2 s and during the negotiation of curves they made glances at the tangent of the curve and moved their gaze longitudinally as they negotiated the curve. A similar notion of near and far is applied in the current model, in which most of the visual input for immediate processing is perceived by peripheral vision. Typical areas of visual input are in the lower periphery, around the lane markers and directly in front of the vehicle.

The model outputs coordinates of hand position as the hand moves the steering wheel and eye fixation point whenever the eye is moved. In the current model the hand moves the steering wheel with a movement that can be mathematically characterized as the first half of a sine wave, which was identified as the basic unit of steering without visual feedback in driving research (Wallis, Chatziastros, and Bulthoff, 2002).

Driving a vehicle can be described as a hierarchical combination of navigation, guidance, and vehicle control (McRuer, Allen, Weir, and Klein, 1977). This concept of hierarchical task structure is consistent with the GOMS-style task description and is adopted by the QN-MHP in defining its processing activities during driving. Following the hierarchical approach, the goal of steering the vehicle is a combination of subgoals for detecting the orientation parameters of the vehicle, selecting a steering strategy, and making the steering correction. It should be emphasized that these subgoals are accomplished concurrently as represented in QN-MHP, not serially.

The current model steers the vehicle at a fixed speed of 72 km/hr (45 mi/hr). Speed control and the effects of road geometry on speed selection (e.g., Levison et al., 2002) add a level of complexity that is not simulated at this point. Vestibular inputs have considerable effects on speed adjustments, especially when driving on curves (Reymond et al., 2001). These effects are not considered in the current model.

Table 1. Description of the steering model.

(References to QN-MHP servers correspond to Figure 2)

Inputs	
Vehicle heading	Vehicle heading relative to the road is retrieved as input to server 1 when fixating on a far point down the road, about 2-4 s in front of the driver.
Lateral position	Lateral position relative to the center of the lane is retrieved as input to server 1 when one of three near points (about 1 s in front of the driver, one at the lower center and two near each of the lane markers) is in the peripheral vision of the driver.
Road curvature	On curves and approaches to curves, road curvature is retrieved as input to server 1 when the eye is fixated on a far point about the point of tangency of the curve
Outputs	
Hand position	The model outputs coordinates of hand position as the hand moves the steering wheel. (To interface with the driving simulator, hand position is converted to corresponding steering angle by an intermediate module.)
Eye position	Coordinates of the eye fixation point in space are output whenever the eye (server Z) is moved (by server Y).
Processing Logic	The main goal of maintaining the lane consists of subgoals for detecting the orientation parameters of the vehicle, selecting a steering strategy, and steering the vehicle, correspondingly.
Detecting orientation	A 'watch for' cognitive command at server D directs the model's visual attention (Servers A and C issue commands to and wait for proper types of entities from servers 1-4) to a far point when about to retrieve heading and curvature and to a near point when about to retrieve lateral position. The eye is not moved if the near point information is accessible from peripheral vision. Otherwise, the eye is moved to that point using a saccade.
Selecting a steering strategy	Steering actions are selected based on the orientation of the vehicle within a look-ahead time (a parameter currently defined as 1 s) as calculated in server F using the following logic: If the vehicle's orientation within the look-ahead time is close to the center of the lane (±0.1 m), no action is taken. Otherwise, if it is within the lane boundaries, a normal steering action is initiated, or if it is outside the lane boundaries, an imminent steering action is initiated.
Steering action	A new steering angle is calculated at server F and executed by servers V, W, X, and Y as a function of the orientation at look-ahead time and the selected steering strategy (normal or imminent). Normal actions are characterized by a steering movement to the new steering angle, and then back to a neutral position. Imminent actions are characterized by a larger magnitude of steering angle and a shorter interval until the steering wheel is returned to its neutral position.

5.2. Interface with a Driving Simulator

To provide an off-the-shelf vehicle dynamics module that interacts with the steering model and is independent of it, and to examine the ability of the QN-MHP driving model to produce relevant steering actions in real-time, the QN-MHP driving model was interfaced with the DriveSafety Research Simulator, a high fidelity driving simulation system used for driving research and training. It utilizes a driving dynamics model that can be adjusted to simulate a variety of vehicle types. It keeps track of numerous state-variables and can output them to external devices. For communication with external devices, the driving simulator uses TCP/IP protocol. Although normally operated via a steering wheel and pedals installed in a simulated car, the driving simulator can also be controlled externally by digital inputs.

Communication between the QN-MHP driving model and the driving simulator (Figure 3) was implemented via a TCP/IP host, linked to the Promodel program (as a DLL). The QN-MHP driving model sent and received data as function calls directly to the DLL. The driving simulator sent and received data by a TCP/IP client that communicated with the host.

Whenever the QN-MHP model made a glance to a specific position in the road scene and information was assumed to be available to it, corresponding information was retrieved from the communication thread. Whenever a hand or eye movement was made by the model, the intermediate or final steering wheel position and the area of fixation were output to the driving simulator via the communication thread. The driving simulator retrieved steering angle and eye position continuously to keep the virtual steering wheel at the desired position and show the area of fixation overlaid on the road scene.

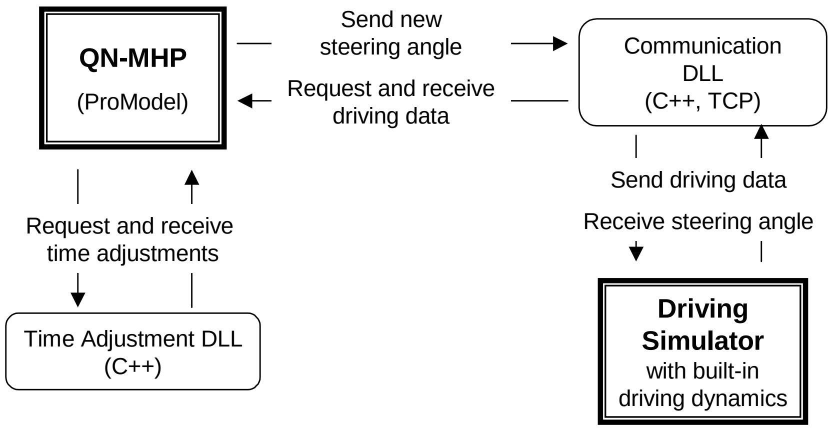

Figure 3. Block diagram of the interface between QN-MHP and the driving simulator.

QN-MHP was successful in steering the driving simulator on a test course and demonstrated realistic steering behavior. Figure 4 shows the physical layout. The simulated vehicle remained within the lane boundaries of straight sections and curves of varying radii. Transfer of information between the software modules of the system was smooth, and timing delays were short. A short movie clip can be seen on the website: http://www-personal.engin.umich.edu/~yililiu/cogmodel.html. Quantitative validation of the steering model is reported later in the empirical validation section of this article.

Figure 4. QN-MHP steering the driving simulator in real time.
A short movie clip can be seen on the website:
http://www-personal.engin.umich.edu/~yililiu/cogmodel.html.

5.3. In-vehicle Task

The in-vehicle task was a map viewing task, as illustrated in Figure 5. The map viewing experiment was conducted by Tsimhoni and Green (2001) to examine the effects of map viewing on driver performance. The task requires looking for a street name on which an icon is located. The icon is referred to by a category to which it belongs. Three categories of icons (fast food restaurant, gas-station, and hotel) commonly found on real maps were used in their experiment, in which 120 distinct maps were used and no map was presented to a subject more than once. Each map consisted of 12 streets (six horizontal and six vertical), one river, and one railroad track, based on what is typical for U.S. maps (George-Maletta, Hunter, Brooks, Lenneman, and Green, 1998). The street names were taken from a list of 100 most popular baby names (www.babycenter.com/babyname/names98.html) updated to 1998. The maps were 15.9 cm (6.25 inch) diagonal (4:3 aspect ratio), approximately the size of contemporary in-vehicle displays.

Map Reading Task

Question: What street is the fast food restaurant on?

Answer: CODY

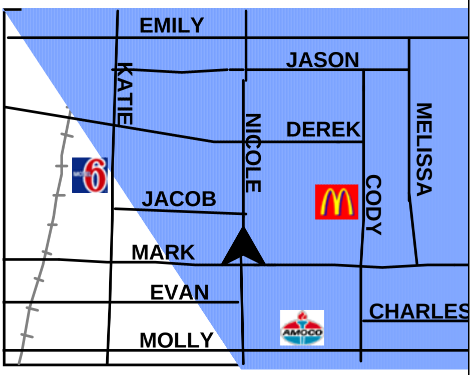

Figure 5. Sample map for the in-vehicle map reading task.

Subjects were asked to perform the map reading task while performing a steering task, which was designed to include four levels of driving demand: on a straight road or on curves of 3, 6, or 9 degrees of curvature (curve radii of 582 m, 291m, and 194 m, respectively). Subjects also performed the map reading task alone in the baseline condition in which the vehicle was parked (Tsimhoni and Green, 2001). The data from this experiment were used to compare with the performance of the QN-MHP driving model.

To model this task with the QN-MHP, a GOMS-style description of the map task was devised. It should be noted that the focus of this model was not on the details of the in-vehicle task, but rather on testing the feasibility of the concurrent processing in QN-MHP. The GOMS-style analysis appears in Table 2. The coordinates and orientation of all icons and street names on the map, and the category of the requested icon were loaded into the environment arrays representing the task situation.

Table 2. GOMS-style task description of in-vehicle map reading

```
GOAL: Do medium duration task

Method for GOAL: Do medium duration task
        Step 1. Find a restaurant
        Step 2. Find the street name
        Step 3. Cease //task completed

Method for GOAL: find <category>
        Step 1. Accomplish goal: Is icon 1 a restaurant
        Step 2. Decide: If location of restaurant in memory then,
                Move to step 7
        Step 3. Accomplish goal: Is icon 2 a restaurant
        Step 4. Decide: If location of restaurant in memory then,
                Move to step 7

        Step 5. Accomplish goal: Is icon 3 a restaurant
        Step 6. Decide: If location of restaurant in memory then,
                Move to step 7
        Step 7. Return with goal accomplished

Method for GOAL: Is icon <number> a <category>
        Step 1. Watch for icon <number> on the map
        Step 2. Compare icon with <category> icons in long-term memory
        Step 3. Decide: If Match, (icon represents a known <category>)
                Then retain location
        Step 4. Return with goal accomplished

Method for GOAL: find street name
        Step 1. Watch for the street name next to location of icon
        Step 2. Read the street name
        Step 3. Retain the street name
        Step 4. Return with goal accomplished
```

5.4. Dual Tasks

The steering task and the in-vehicle map reading tasks were implemented as two independent goals as described above. The steering goal was performed continuously as long as the vehicle was in motion. The in-vehicle task was initiated at intervals of 30 s and stopped after completion. Both goals were processed simultaneously except when there was a conflict at the server level. The most notable conflict for these two goals was over visual resources. The conflict occurred when each goal instructed to move the eye to a different location. When such conflicts occurred, priority was given to one of the goals based on the driving situation and the involvement in the in-vehicle task. More specifically, a satisficing rule based on time to line crossing (TLC) was generated such that the eyes were not diverted away from the road unless TLC was above a certain threshold (4 s). TLC represents the time available for a driver until the moment at which any part of the vehicle reaches one of the lane boundaries (Godthelp, Milgram, and Blaauw, 1984). The calculation used in the model was a simplified version of TLC calculation based on the lateral position and its derivatives, which has been shown to be an acceptable predictor of lane departures due to inattention (van Winsum, Brookhuis, and de Waard, 2000). When the eyes were already directed at the in-vehicle display, TLC was estimated by the steering task, and the eyes glanced back to the road when TLC reached the threshold.

6. EMPIRICAL VALIDATION OF THE QN-MHP DRIVING MODEL

6.1 Steering Performance

The model demonstrated realistic steering behavior. It steered a simulated vehicle within the lane boundaries on a straight section and on curves, which represented increasing difficulties of steering. Four dependent measures of driving performance (the mean and the standard deviation of steering angle, and the mean and the standard deviation of the vehicle lateral position) were used for a quantitative comparison of the model and the empirical data. Table 3 shows the driving parameters produced by the steering model in comparison to the same parameters from the validation experiment. Analysis of variance was performed for each of the four dependent measures using three levels of road curvature and two sources of data (model simulation versus human subject). Overall, differences between the model and human subjects were small and the trends due to road curvature were similar.

Mean steering angle in the simulation was up to 0.3 degrees lower than in the empirical data, $F(1,28) = 7.3$, $p<.05$, but the interaction between model-validation and curvature was not significant, $F(2,56) = 3.3$. The Standard deviation (SD) of steering wheel angle in the simulation was about half a degree less than in the empirical data, $F(1,28) = 5.1$, $p<.05$, but the interaction between model-validation and road was not significant, $F(2,56) = .3$. These results suggest that there was less variability in the steering angle in the simulation model, but the increase in variability due to road curvature was the same as in the human subject data.

The difference between mean lateral position in the simulation and in the empirical data was not significant, $F(1,28) = .7$, and neither was the interaction with road curvature, $F(2,56) = 1.9$. However, the mean lateral position in the simulation on the sharp curve was slightly higher than in the empirical data. Finally, the difference between SD of lateral position in the simulation and in the empirical data was not significant, $F(1,28) = .01$, but there was an interaction with road curvature, in which the SD on the sharp curve was higher in the simulation than in the empirical data, $F(2,56) = 7.2$, $p < .01$. These results suggest that the model was able to maintain lateral position but was less effective doing so on the sharp curve.

Figure 6 illustrates the resulting driving performance of the model. The mean lateral position on a 400 m radius right curve is shown, as predicted by the model across eight runs. The vehicle remains within the lane boundaries and there is more variability on the curve. Figure 7illustrates driving performance of human subjects on the same road in the driving simulator. Several differences are noted: (1) the model is generally less consistent across time, (2) on curvature change, the model's timing is not as good as that of real subjects, and (3) the variability between runs is smaller than the variability between subjects.

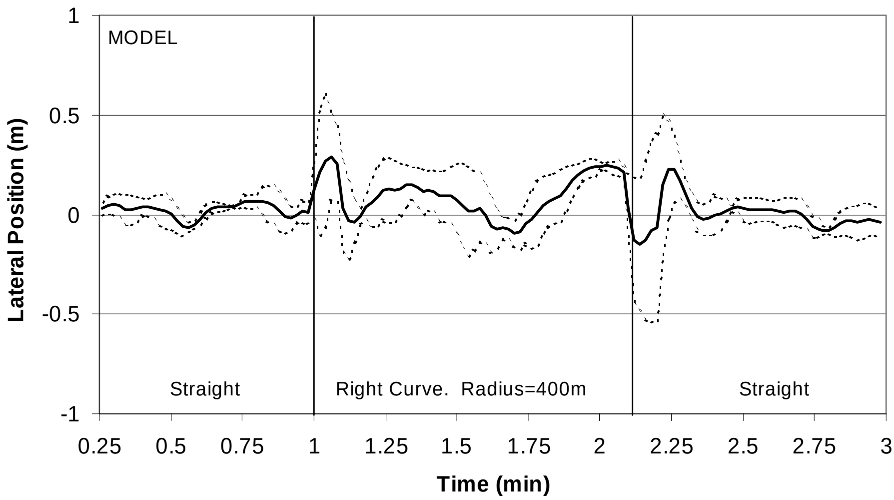

Figure 6. Lateral position as predicted by simulation model. Solid line – mean of eight runs, dashed lines – range of ±1 SD between runs

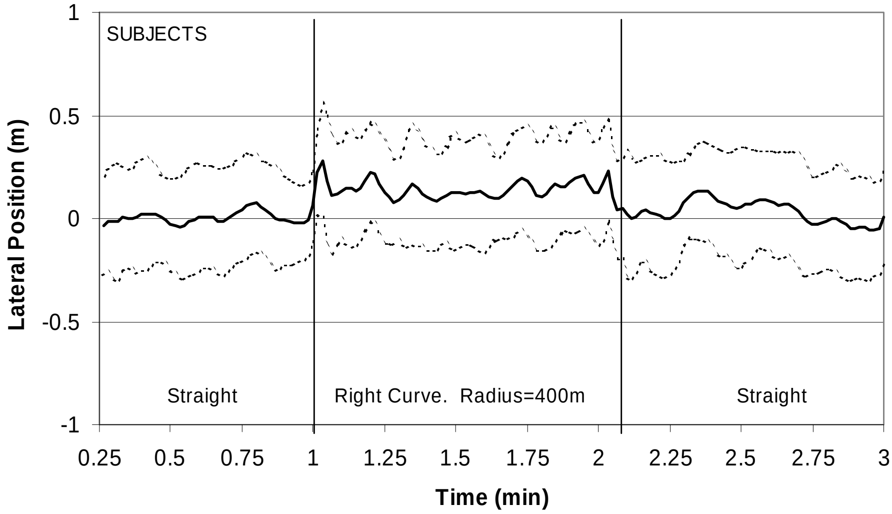

Figure 7. Lateral position as measured in driving simulator experiment with real subjects. Solid line – mean of 24 subjects, dashed lines – range of ±1 SD between subjects

Table 3. Driving performance parameters as predicted by the steering model and as measured in a driving simulator experiment.

	Steering model			Empirical validation		
Road (Curve Radius [m])	Straight	400	200	Straight	400	200
Steering Angle [deg]						
Mean	.01±.02	17.4±.05	34.8±.6	0±0	17.5±.6	35.1±.6
SD	.36±.04	1.06±.28	3.3±1.5	1.1±.5	1.7±.6	3.7±1.4
Lateral Position [m]						
Mean	.01±.03	.12±.10	0.44±.10	.01+.22	.13±.20	.27±.22
SD	.07±.01	.11±.06	0.26±.12	.11±.04	.15±.05	.19±.05

6.2 In-vehicle Task Performance

In-vehicle task performance was validated for the parked condition in which the subject performed the task without driving and the model had only the in-vehicle task goal active. The purpose of this validation was to confirm the GOMS-style analysis for in-vehicle task performance presented earlier (Table 2).

Figure 8 shows total task time for 8 maps in the order they were performed by the model and by subjects in the experiment. Generally, there was agreement between the model and the empirical results. Maps that subjects took longer to complete, took longer for the model to complete as well, and relative time differences between maps were maintained. Some empirical observations, however, remained unaccounted for. While the trends (slopes between maps) for younger subjects were predicted rather well by the model, the trends for older subjects were not. Older subjects performed maps 2, 7, and 8 more slowly than the other maps, but the model did not account for this interaction. This is not surprising since the current model has not reached the stage of addressing age differences. Nevertheless, it made reasonable predictions of overall total task time.

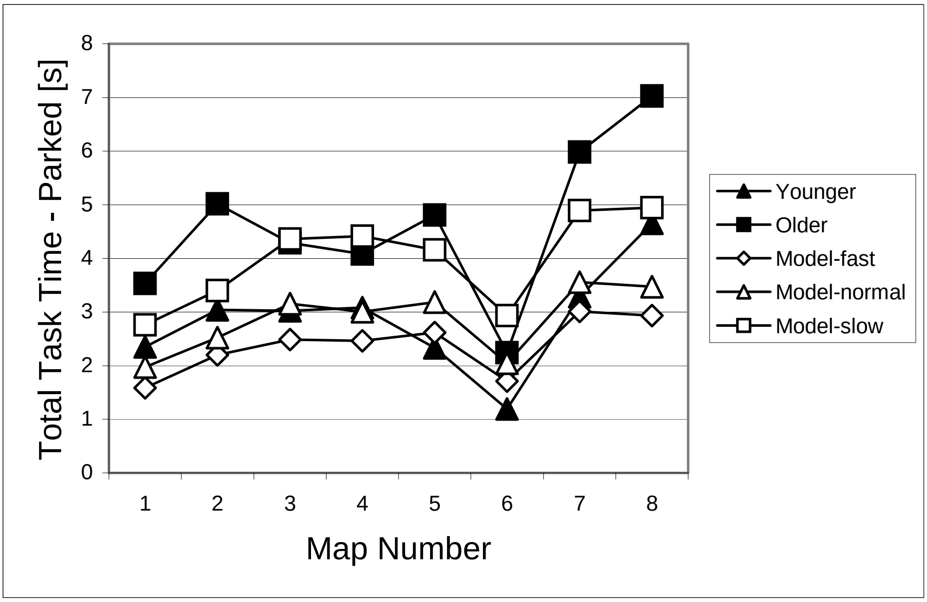

Figure 8. Comparison of model and actual total task time while parked, by map number

6.3. Dual Task Performance

Dual task performance was validated for driving on a straight section and on curves of three increasing radii. The algorithm for in-vehicle task performance, which was validated earlier for the parked condition, was performed concurrent to steering. The data were analyzed using repeated measures ANOVA with subjects/runs nested within a model versus experiment variable and block and map as within subject/run factors.

Figure 9 A shows total task time for the driving conditions as compared to the parked condition. There was no significant difference between the model and actual data $F(1,28) = .41$ and the interaction with road curvature was insignificant as well $F(5,140) = 1.4$. Total task time matched well when driving was easy. On the straight section, total task time for the model and actual was 3.96 and 3.95 s, respectively and on the moderate curve it was 4.44 and 4.22 s, respectively.

Figure 9 B shows the number of glances for the driving conditions compared to the parked condition. There was no significant difference between the model and actual data F(1,35) = .02 and the interaction with road curvature was insignificant as well F(5,175) = .20. The mean number of glances on the straight road was 1.7 for both the model and the actual setting. The actual number of glances on curves was higher by about 0.2 glances on all curves. The model had a gradual increase in the number of glances as curvature increased. Overall, the number of glances was predicted reasonably well by the model.

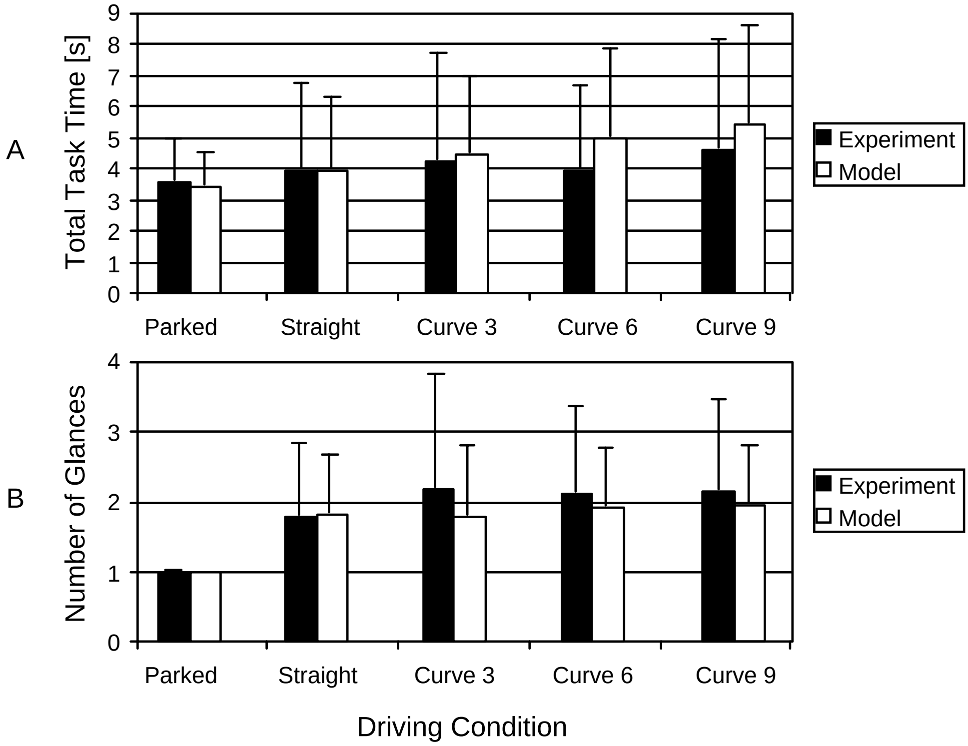

Figure 9. Model predictions of task timing by driving condition. (Curve number denotes degree of curvature. 3=moderate, 9=sharp).

7. DISCUSSION AND CONCLUSIONS

As shown in this article, our modeling approach, called the Queueing Network – Model Human Processor (QN-MHP) is significantly different but very complementary to the existing approaches. As a model of multi-task performance, a unique characteristic of the QN-MHP is its ability to model concurrent activities in an underlying, context-free, modeling architecture without limiting or predefining their order of occurrence. In the QN-MHP, multitask behavior emerges as an outcome of multiple tasks (represented as multiple streams of customers) traversing a network like trucks and cars traveling on the same highway, and much of the performance predictions depend on aspects of the network architecture, as Newell advocated. Like ACT-R and EPIC, a modeler still needs to analyze each concurrent task; but unlike ACT-R and EPIC, one does not need to write separate, step-by-step, task specific and potentially complicated codes to either interleave the production rules or interactively control (lock/unlock) the task processes.

In this article, we described the current structure, assumptions, implementation, and methods of the QN-MHP, and showed how it was used to model two single task conditions: steering a simulated vehicle, and reading an electronic map, and a dual task condition that consists of performing both the vehicle steering and the map reading task together. The successful interfacing of the QN-MHP with a driving simulator allowed the model to perform the tasks in real time. The model was able to perform all of the single and the dual task conditions well. The performance data of the model are similar to human subjects performing the same tasks.

In our related modeling efforts, we have successfully employed the QN-MHP in modeling a variety of other representative human performance tasks including the menu search task, the psychological refractory period (PRP) phenomenon, and the transcription typing task. In all these modeling work, the behavioral phenomena under investigation emerged as outcomes of the natural operations of the underlying context-free human information processing queueing network. Each of the modeling work offered unique insights into the task domain and cognitive modeling from the queueing network perspective. For example, the queueing network model was able to account for data patterns of menu search (Nilsen, 1991) examined by either ACT-R (Anderson, Matessa, and Lebiere, 1997; Byrne, 2001) or the EPIC-based *dual strategy varying distance hybrid* (DSVDH) model (Hornof, 1999) by using only one task strategy that was already employed in ACT-R (the “hunt feature” production of ACT-R). In contrast, the DSVDH model relied on four task strategies to account for the same data, whereas ACT-R was unable to account for a well established menu search phenomenon called “eye overshooting”—the eye moves further than the location of target (Lim and Liu, 2004a, 2004b).

The queuing network model also successfully modeled PRP without the need of setting up complex lock/unlock performance strategies employed in the EPIC model of PRP or drawing complex scheduling charts employed in the ACT-R/PM model of PRP. Further, by integrating queuing networks with reinforcement learning algorithms, the queueing network model of PRP (Wu and Liu, 2004a) successfully simulated the practice effect on PRP (Van Selst, Ruthruff, and Johnston, 1999), which has not been modeled in existing PRP models.

Lastly, the queueing network model has successfully accounted for 27 of the 29 behavioral regularities of transcription typing identified by Salthouse (1986), including all the 19 regularities modeled by the CPM-GOMS based model of transcription typing called TYPIST (John, 1996), as well as two neuroscience phenomena of transcription typing, involving different activation patterns of brain areas, discovered in recent fMRI and PET studies.

The success of modeling concurrent perceptual, cognitive, and motor activities of human performance as the behavioral outcomes of the natural operations of a truly concurrent, context-free, underlying queueing network, as illustrated in the current study as well as our related modeling work, provides another perspective from which we can examine and model human performance, in addition to existing ones. It also opens a range of possible areas of research and application for this modeling approach. Our on-going research builds upon the current work and expands it in several aspects, of which the following is a partial list: (1) The driving task will be expanded to include speed control and to alter behavior based on traffic; (2) Other perceptual modalities will be added to the QN-MHP architecture (e.g., vestibular, auditory); their addition to the driving task, and their effects on it, will be investigated; (3) The mathematics of queueing networks is being applied to analyze the performance of the QN-MHP, so that the full capacity of queueing networks as both a mathematical and a simulation method can be utilized; (4) The value and the methods of adding production systems capabilities to the QN-MHP will be explored, beyond the current procedure functions.

The modeling work described in this article represents the beginning, rather than the end, of our efforts in integrating the queueing network approach with the symbolic approach in cognitive modeling. The work is an important step toward our goals of contributing to the realization of Newell's dream of theoretical and methodological unification in cognitive modeling and of developing easy-to-use HCI simulation technology.

REFERENCES

ADAMS, M., TENNEY, Y. AND PEW, R. 1991. Strategic workload and the cognitive management of advanced multi-task systems, CSERIAC SOAR Report 91-6.

ANDERSON, J.R. 1993. *Rules of the Mind*. Hillsdale, NJ: Lawrence Erlbaum Associates.

ANDERSON, J.R., AND BOWER, G. H. 1973. Human associative memory. Mahwah, NJ: Lawrence Erlbaum Associates.

ANDERSON, J.R., JOHN, B.E., JUST, M.A., CARPENTER, P.A., KIERAS, D.E., AND MEYER, D.E. 1995. Production system models of complex cognition. In *Proceedings of the 17th Annual Conference of the Cognitive Science Society* (9-12). Hillsdale, NJ: Lawrence Erlbaum Associates.

ANDERSON, J.R. AND LEBIERE, C. Eds. 1998. *Atomic components of thought.* Hillsdale, NJ: Lawrence Erlbaum Associates.

ANDERSON, J.R., AND LEBIERE, C. 2002. The Newell test for a theory of mind. *Behavioral and Brain Sciences*.

ANDERSON, J.R., MATESSA, M., and LEBIERE, C. (1997). ACT-R: A theory of higher level cognition and its relation to visual attention. *Human-Computer Interaction*, *12*, 439-462.

BADDELEY, A. 1992. Working memory. *Science, 255,* 556-559.

BADDELEY, A. 1998. Recent developments in working memory. *Current Opinion in Neurobiology, 8,* 234-238.

BARON, S., KRUSER, D., AND HUEY, B. Eds. 1990. *Quantitative modeling of human performance in complex, dynamic systems*. Washington, D.C.: National Academy Press.

BARON, S., ZACHARIAS, G., MURALIDHARAN, R., AND LANCRAFT, R. 1980. *PROCRU: A model for analyzing flight crew procedures in approach to landing*. NASA-CR-1523397.

BOXMA, O. AND DADUNA, H. 1990. Sojourn times in queueing networks. In H. TAKAGI Ed., *Stochastic analysis of computer and communications systems*, 401-150. North Holland.

BYRNE, M., D. (2001). ACT-R/PM and menu selection: applying a cognitive architecture to HCI. International *Journal of Human-Computer Studies*, 55

BYRNE, M., AND ANDERSON, J.R. 2001. Serial modules in parallel: The psychological refractory period and perfect time-sharing. *Psychological Review*, 108, 847-868.

BUZEN, J. 1976. Fundamental operational laws of computer system performance. *Acta Informatica 7,* 167-182.

CARBONELL, J. 1966. A queueing model of many-instrument visual sampling. *IEEE Trans. HFE-4*, 157-164.

CARD, S., MORAN, T.P., AND NEWELL, A. 1983. *The psychology of human-computer interaction.* Hillsdale, NJ: Lawrence Erlbaum Associates.

CARD, S., AND NEWELL, A. 1990. Cognitive architectures. In BARON, S., KRUSER, D., AND HUEY, B. Eds. 1990. *Quantitative modeling of human performance in complex, dynamic systems*. Washington, D.C.: National Academy Press.

CARTER, R. 1998. *Mapping the mind*. Berkeley, CA: The University of California Press.

CHU, Y. AND ROUSE, W. 1979. Adaptive allocation of decision making responsibility between human and computer in multi-task situations. *IEEE Trans. SMC-9*, 769-778.

COURTNEY, S. AND UNGERLEIDER, L. 1997. What fMRI has taught us about human vision. *Current Opinion in Neurobiology, 7,* 554-561.

DENNING, P. AND BUZEN, J. 1978. The operational analysis of queueing network models. *Computing Surveys 10,* 225-261.

DISNEY, R. AND KONIG, D. 1985. Queueing networks: A survey of their random processes. *SIAM Review*, vol-27, 335-403.

DONDERS, F. C. 1868/1969. Over de snelheid van psychische processen [On the speed of mental processes]. In W. G. KOSTER Ed., *Attention and Performance 2* 412-431). Amsterdam: North-Holland. (Original work published in 1868).

DONGES, E. 1978. A two-level model of driver steering behavior. *Human Factors*, *20* 691-707

ELKIND, J.I., CARD, S.K., HOCHBERG, J., AND HUEY, B.M. Eds. 1990. *Human performance models for computer-aided engineering.* San Diego: Academic Press, Inc.

FEYEN, R. 2002. Modeling Human Performance using the Queueing Network – Model Human Processor (QN-MHP). Doctoral Dissertation. Department of Industrial and Operations Engineering, University of Michigan.

FEYEN, R. AND LIU, Y. 2001a. Modeling Task Performance Using the Queuing Network- Model Human Processor (QN-MHP). In E.M. ALTMANN, A. CLEERMANS, C.D. SCHUNN, AND W.D. GRAY Eds., *Proceedings of the Fourth International Conference on Cognitive Modeling.* Mahwah, NJ: Lawrence Erlbaum Associates.

FEYEN, R. AND LIU, Y. 2001b. The Queuing Network-Model Human Processor (QN-MHP): An engineering approach for modeling cognitive performance. In *Proceedings of the 45th Annual Meeting of the Human Factors and Ergonomics Society, San Diego, CA.*

GILMAN, S. AND NEWMAN, S. 1992. *Manter and Gatz's essentials of clinical neuroanatomy and neurophysiology (8^{th} edition)*. Philadelphia: F.A. Davis.

GRAY, W., JOHN, B., AND ATWOOD, M. 1993. Project Ernestine: Validating a GOMS analysis for predicting and explaining real-world performance. *Human Computer Interaction, 8(3),* 237-309.

GREENSTEIN, J. AND ROUSE, W. 1982. A model of human decision making in multiple process monitoring situations. *IEEE Trans. SMC-12*, 182-193.

HARRELL, C.R., BATEMAN, R.E., GOGG, T.J., AND MOTT, J.R.A. 1995. *System improvement using simulation* 3^{rd} Ed. Orem, Utah: PROMODEL Corporation.

HILDRETH, E. C., BEUSMANS, J. M. H., BOER, E. R., AND ROYDEN, C. S. 2000. From vision to action: Experiments AND models of steering control during driving. *Journal of Experimental Psychology: Human Perception and Performance*, *26*(3), 1106-1132.

HORNOF, A. J. 1999. Computational models of the perceptual, cognitive, and motor processes involved in the visual search of pull-down menus and computer screens. Doctoral dissertation. University of Michigan.

JOHN, B.E. 1996. TYPIST: A theory of performance in skilled typing. *Human Computer Interaction, 11*, 321-355.

JOHN, B.E. AND GRAY, W. 1995. GOMS analysis of parallel activities. Tutorial notes, CHI. Denver Colorado, May 7-11, 1995.

JOHN, B.E. AND KIERAS, D.E. 1996a. Using GOMS for user interface design and evaluation: Which technique? *ACM Transactions on Human-Computer Interaction, 3(4),* 287-319.

JOHN, B.E. AND KIERAS, D.E. 1996b. The GOMS family of user interface analysis techniques: comparison and contrast. *ACM Transactions on Human-Computer Interaction, 3(4),* 320-351.

JUST, M.A., AND CARPENTER, P.N. 1992. A capacity theory of comprehension: individual differences in working memory. *Psychological Review*, 99, 122-149.

KIERAS, D. 1999. A guide to GOMS usability evaluation using GOMSL and GLEAN3. University of Michigan Department of Electrical Engineering and Computer Science On-line publication.

KIERAS, D.E. 1988. Towards a practical GOMS model methodology for user interface design. In M. HELANDER Ed., *The handbook of human-computer interaction*. Amsterdam: North-Holland Elsevier.

KIERAS, D.E. AND MEYER, D.E. 1997. An overview of the EPIC architecture for cognition and performance with application to human-computer interaction. *Human-Computer Interaction, 12(4),* 391-438.

KIERAS, D.E., WOOD, S.D., AND MEYER, D.E. 1997. Predictive engineering models based on the EPIC architecture for a multimodal high-performance human-computer interaction task. *ACM Transactions on Computer-Human Interaction*, 4, 230-275.

KLEINMAN, D. L. AND CURRY, R. E. 1977. Some new control theoretic models for human operator display monitoring. *IEEE Trans. SMC-7*, 778-784.

LAIRD, J.E., NEWELL, A., AND ROSENBLOOM, P.S. 1987. Soar: An architecture for general intelligence. *Artificial Intelligence*, 33, 1-64.

LAND, M. F., AND LEE, D. N. 1994. Where we look when we steer. *Nature*, 369, 742-744.

LAUGHERY, K. 1989. Micro SAINT: A tool for modeling human performance in systems. In G. MCMILLAN et al., Eds., *Applications of Human Performance Models to System Design*, 219-230, New York: Plenum.

LAUGHERY, K., AND CORKER, K. 1997. Computer modeling and simulation. In SALVENDY, G Ed., *Handbook of Human Factors and Ergonomics (2nd edition).* New York: Wiley.

LEIBOWITZ, H. W., AND OWENS, D. A. 1977. Nighttime driving accidents and selective visual degradation. *Science, 197*, 422-423.

LIM, J., AND LIU, Y. 2004a. A queueing network model of menu selection and visual search. In *Proceedings of the 47 Annual Conference of the Human Factors and Ergonomics Society*.

LIM, J., AND LIU, Y. 2004b. A queueing network model for eye movement. In *Proceedings of the 2004 International Conference on Cognitive Modeling*. Pittsburg, PA.

LIU, Y. 1994. Queuing networks as models of human performance and human-computer systems. *Proceedings of the '94 Symposium on Human Interaction with Complex Systems*, 256-270.

LIU, Y. 1996. Queuing network modeling of elementary mental processes. *Psychological Review, 103*, 116-136.

LIU, Y. 1997. Queuing network modeling of human performance of concurrent spatial or verbal tasks. *IEEE Transactions on Systems, Man, and Cybernetics, 27*, 195-207.

MCCLELLAND, J. 1979. On the time relations of mental processes: An examination of systems of processes in cascade. *Psychological Review, 86*, 287-330.

MCCLELLAND, J.L., AND RUMELHART, D.E. 1986. Parallel distributed processing: Explorations in the microstructure of cognition (vol. 2). Cambridge, MA: MIT Press.

MCRUER, D. T., ALLEN, R. W., WEIR, D. H., AND KLEIN, R. H. 1977. New results in driver steering control models. *Human Factors, 19*(4), 381-397

MEYER, D.E., AND KIERAS, D.E. 1997a. A computational theory of executive cognitive processes and multiple-task performance: Part 1. Basic mechanisms. *Psychological Review 104*, 3-65.

MEYER, D.E., AND KIERAS, D.E. 1997b. A computational theory of executive cognitive processes and multiple-task performance: Part 2. Accounts of psychological refractory-period phenomena. *Psychological Review 104*, 749-791.

MEYER, D.E., AND KIERAS, D.E. 1999. Précis to a practical unified theory of cognition and action: Some lessons from EPIC computational models of human multiple-task performance. In D. GOPHER AND A. KORIAT Eds., *Attention and performance XVII. Cognitive regulation of performance: Interaction of theory and application*. Cambridge, 17-88, MA: MIT Press.

MILLER, J. O. 1993. A queue-series model for reaction time, with discrete-stage and continuous-flow models as special cases. *Psychological Review, 100*, 702-715.

MOURANT, R. R., AND ROCKWELL, T. H. 1970. Mapping eye-movement patterns to the visual scene in driving: An exploratory study. *Human Factors, 12*, 81-87.

NEWELL, A. 1973. You can't play 20 questions with nature and win: Projective comments on the papers of this symposium. In W. G. CHASE Ed., *Visual information processing*. New York: Plenum Press.

NEWELL, A. 1990. *Unified theories of cognition*. Cambridge, MA: Harvard University Press.

NIEBEL, B. 1993. *Motion and Time Study* (9th ed.). Boston: Irwin.

NILSEN, E. L. 1991. Perceptual-motor control in human-computer integration. Doctoral dissertation. Department of Psychology, University of Michigan.

NORTH, R. AND RILEY, V. 1989. W/INDEX: A predictive model of operator workload. In G. MCMILLAN et al., Eds., *Applications of Human Performance Models to System Design*, 81-89, New York: Plenum.

OLSON, J.R. AND OLSON, G.M. 1990. The growth of cognitive modeling in human-computer interaction since GOMS. *Human-Computer Interaction, 5,* 221-265.

OWENS, D. A., AND TYRRELL, R. A. 1999. Effects of luminance, blur, and age on nighttime visual guidance: A test of the selective degradation hypothesis. *Journal of Experimental Psychology: Applied, 5,* 115-128

PATTIPATI, K. R., KLEINMAN, D. L., AND EPHRATH, A. R. 1983. A dynamic decision model of human task selection performance. *IEEE Trans. SMC-13,* 145-166.

PEW, R.W. AND BARON, S. 1983. Perspectives on human performance modeling. *Automatica, 19(6)*, 663-676.

PRITCHARD, T.C. AND ALLOWAY, K.D. 1999. *Medical neuroscience*. Madison, CT: Fence Creek Publishing.

REYMOND G, KEMENY A, DROULEZ J, AND BERTHOZ A. 2001. Role of lateral acceleration in driving: experiments on a real vehicle and a driving simulator. *Human Factors, 43*(3), 483-495

ROLAND, P. 1993. *Brain activation*. New York: Wiley.

ROSENBAUM, D.A. 1991. *Human motor control*. New York: Academic Press.

ROUSE, W. B. 1980. *Systems engineering models of human-machine interaction,* New York: Elsevier North-Holland.

SALTHOUSE, T.A 1986. Perceptual, cognitive, and motoric aspects of transcription typing. *Psychology Bulletin, 99* (3), 303-31

SALVUCCI, D. D., BOER, E. R., AND LIU, A. 2001. Toward an integrated model of driver behavior in a cognitive architecture. *Transportation Research Record*, 1779.

SALVUCCI, D. D. AND MACUGA, K. L. 2001. Predicting the effects of cell-phone dialing on driver performance. In E.M. ALTMANN, A. CLEERMANS, C.D. SCHUNN, AND W.D. GRAY Eds., *Proceedings of the Fourth International Conference on Cognitive Modeling.* Mahwah, NJ: Lawrence Erlbaum Associates.

SALVUCCI, D. D. 2002. Modeling driver distraction from cognitive tasks. In Proceedings of the 24th Annual Conference of the Cognitive Science Society. Hillsdale, NJ: Lawrence Erlbaum Associates.

van SELST, M., RUTHRUFF, E. AND JOHNSTON, J.C. 1999. Can practice eliminate the psychological refractory period effect? *Journal of Experimental Psychology: Human Perception and Performance 25*(5), 1268-1283.

SCHMIDT, D. 1978. A queueing analysis of the air traffic controller's workload. *IEEE Trans. SMC-8*, 492-493.

SCHWEICKERT, R. 1978. A critical path generalization of the additive factor method: Analysis of a Stroop task. *Journal of Mathematical Psychology, 18,* 105-139.

SENDERS, J. W. 1964. The human operator as a monitor and controller of multidegree freedom systems. *IEEE Trans. on HFE, HFE-5,* 2-5.

SENDERS, J. W. AND POSNER, M. 1976. A queueing model of monitoring and supervisory behavior. In T. SHERIDAN AND G. JOHANNSEN Eds., *Monitoring Behavior and Supervisory Control,* New York: Plenum.

SHERIDAN, T. 1972. On how often the supervisor should sample. *IEEE Trans., SSS-6*, 140-145.

SIEGAL, A. I. AND Wolf, J. J. 1969. *Man-machine simulation models*, New York: Wiley.

STERNBERG, S. 1969. The discovery of processing stages: Extensions of Donders's method. *Acta Psychologica,* 30, p.276-235.

SUMMALA, H. 1998. Forced peripheral vision driving paradigm: Evidence for the hypothesis that car drivers learn to keep in lane with peripheral vision. In A.G. GALE et al. Eds., *Vision in Vehicles VI* (51-60). Amsterdam: Elsevier Science Publishers.

TSIMHONI, O. 2004. Visual sampling of in-vehicle displays while driving: Empirical findings and a queueing network cognitive model. Doctoral Dissertation. Department of Industrial and Operations Engineering. University of Michigan.

TSIMHONI, O., AND LIU, Y. 2003a. Modeling steering using the queueing network model human processor (QN-MHP). *Proceedings of the Human Factors and Ergonomics Society 47th annual meeting.*

TSIMHONI, O., AND LIU, Y. 2003b. Steering a driving simulator using the queueing network-model human processor (QN-MHP). In D. V. MCGEHEE, J. D. LEE, AND M. RIZZO Eds. *Driving Assessment 2003: International Symposium on Human Factors in Driver Assessment, Training, and Vehicle Design.*

TULGA, M. K. AND SHERIDAN, T. Dynamic decisions and workload in multitask supervisory control. *IEEE Trans., SMC-10*, 217-232.

WALDEN, R. AND ROUSE, W. 1978. A queueing model of pilot decision making in a multitask flight management situation. *IEEE Trans. SMC-8,* 867-875.

WALLIS, G., CHATZIASTROS, A., AND BULTHOFF, H. 2002. An unexpected role for visual feedback in vehicle steering control. *Current Biology, 12*, 295-299.

WHERRY, R. J. 1976. The human operator simulator—HOS. In T. B. SHERIDAN AND G. JOHANNSEN Eds., *Monitoring Behavior and Supervisory Control*. New York: Plenum Press.

WICKENS, C. D. 1984. Processing resources in attention. In PARASURAMAN, R. AND D. DAVIS Eds., *Varieties of attention*, New York: Academic Press.

WICKENS, C.D. AND LIU, Y. 1988. Codes and modalities in multiple resources: A success and a qualification. *Human Factors, 30*, 599-616.

WU, C., AND LIU, Y. 2004a. Modeling psychological refractory period (PRP) and practice effect on PRP with queuing networks and reinforcement learning algorithms. In *Proceedings of 2004 International Conference on Cognitive Modeling*. Pittsburg, PA.

WU, C., AND LIU, Y. 2004b. Modeling behavioral and brain imaging phenomena in transcription typing with queuing networks and reinforcement learning algorithms. In *Proceedings of 2004 International Conference on Cognitive Modeling*. Pittsburg, PA.

WU, C., AND LIU, Y. 2004c. Queueing network modeling of transcription typing. In *Proceedings of the 47 Annual Conference of the Human Factors and Ergonomics Society*.