STEERING A DRIVING SIMULATOR USING THE
QUEUEING NETWORK-MODEL HUMAN PROCESSOR (QN-MHP)
Omer Tsimhoni and Yili Liu
Department of Industrial and Operations Engineering
University of Michigan

Ann Arbor, Michigan, USA
E-mail: omert@umich.edu

Summary: The Queueing Network-Model Human Processor (QN-MHP) is a computational architecture that combines the mathematical theories and simulation methods of queueing networks (QN) with the symbolic and procedure methods of GOMS analysis and the Model Human Processor (MHP). QN-MHP has been successfully used to model reaction time tasks and visual search tasks (Feyen and Liu, 2001a,b). This paper describes our work of using QN-MHP to model vehicle steering and to steer a driving simulator as a step toward modeling more complex driving scenarios. The steering model was implemented in Promodel, a commercially available simulation program. A network of 20 servers represents different functional modules of the human perceptual, cognitive, and motor information processing system. Entities carrying information on vehicle location and orientation arrive at and flow through the visual, cognitive and motor sub-networks of the system and are processed independently and concurrently by the servers.

The QN-MHP steering model was interfaced with a driving simulator (DriveSafety) using an Ethernet protocol and several custom-built software modules. Heading and location information was received in real-time from the simulator and processed through the servers. Whenever the model made a hand movement, the corresponding position of the steering wheel was transferred to the simulator, thus steering the simulated vehicle. The model demonstrated realistic steering behavior. It steered the driving simulator within the lane boundaries of straight sections and curves of varying curvature. This work showed the potential strength of QN-MHP as a model of driving behavior. Ongoing work will further develop the model by expanding the scope of the driving task and by adding secondary in-vehicle tasks.

INTRODUCTION

Computational cognitive models can contribute considerably to driving-related human factors research. Computational models can make quantitative predictions for scenarios that have not been tested, and provide a precise common language for description of phenomena of interest. Further, computational models have a symbiotic relationship with empirical research. Empirical findings can be integrated into models to strengthen their validity and expand their scope. In turn, models can identify gaps in the empirical literature and point to new directions of research.

In this paper we describe our effort toward modeling driving using a novel computational model, called the QN-MHP (Queueing Network – Model Human Processor). The QN-MHP is a computational architecture that combines the mathematical theories and simulation methods of queueing networks with the symbolic and procedure methods of GOMS analysis and the Model Human Processor (MHP). As a network architecture, queueing networks are particularly suited for modeling parallel activities and complex mental architectures. Symbolic models have particular strength in generating a person's actions in specific task situations. By integrating the two complementary approaches, the QN-MHP offers a modeling and simulation architecture for generating in real-time and mathematical modeling of parallel and complex activities. QN-MHP has been successfully used to model reaction time tasks and visual search tasks (Feyen and Liu, 2001a, b). In this paper, we describe our work of modeling steering of a driving simulator using the QN-MHP.

QN-MHP AND PROMODEL

QN-MHP is implemented in ProModel (ProModel solutions, version 2001), a simulation-based software that is widely used for manufacturing and operational applications and provides a natural programming environment for queueing network simulation. In addition, it has built-in analysis tools and strong visualization capabilities. In QN-MHP, 20 servers represent different functional modules of the human perceptual, cognitive, and motor information processing system (Figure 1). (See Feyen and Liu, 2001a, for more details.) Customers enter the perceptual subnetwork carrying perceptual information, which is then processed by the cognitive subnetwork and converted into actions, carried out by the motor subnetwork. Flow of customers through the network can be visualized in real-time to provide an assessment of the utility of servers and the progress of actions. An output data file, documenting overt actions (e.g., hand movements and eye movements) and variable status (e.g., perceived vehicle information) is produced for post-simulation analysis.

Perceptual subnetwork
1 Common visual processing
2 Visual recognition
3 Visual Localization
4 Visual integration

Cognitive subnetwork
A Visuospatial sketchpad
B Phonological loop
C Central executive
D Task (procedure) selection
E Performance monitor
F High Cognitive operators
G Goal prioritization

Motor subnetwork
V Sensorimotor join
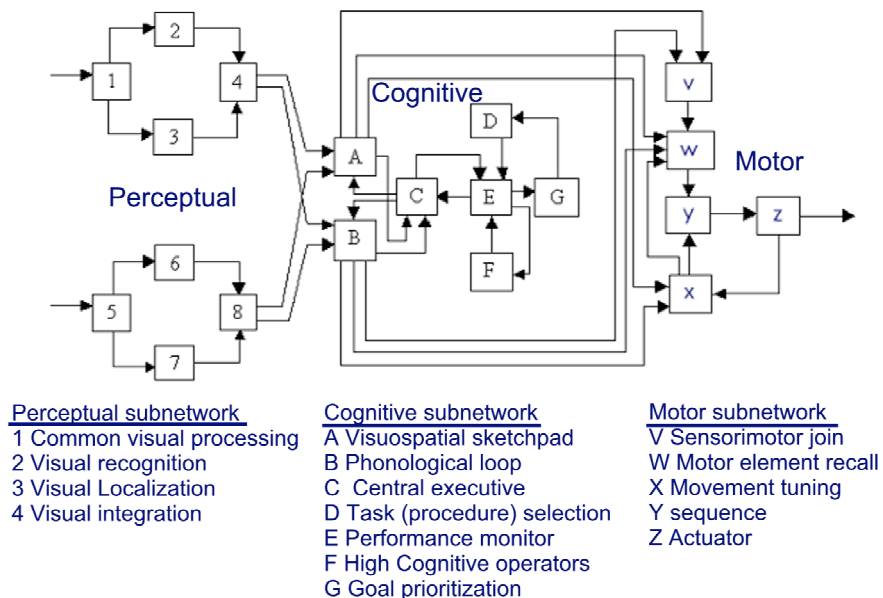W Motor element recall
X Movement tuning
Y sequence
Z Actuator

Figure 1. Layout of the servers in QN-MHP and the flow of information between them.

The front-end of the model is an MS-Excel file with data about the environment (stimuli and object description), the actuators and actions in use, parameters available in long-term memory, and a goal list. The goal list is based on a GOMS task-analysis using the defined actions and actuators as they interact with the environment. One of the servers in the network, the task selection server, scans this list step by step to provide the next step that needs to be performed based on the current goal and method processed. Additional elements such as if-then statements and choice probabilities enhance the range of scenarios that can be modeled.

THE STEERING MODEL

The steering model follows a goal-oriented analysis. The main goal of maintaining the lane consists of subgoals for detecting the orientation of the vehicle, selecting a steering action, and performing a hand movement correspondingly. In order to detect the orientation of the vehicle, a motor action is produced for moving the eyes to the road scene, and a request for information from the visual system is made. Information is continuously perceived from the road scene, except when the eyes are moving (saccadic suppression). As a result, steering actions are triggered by the cognitive subnetwork in response to the analysis of the current state of the vehicle in comparison to the desired state. These steering actions can be normal steering actions or, when about to depart the lane, imminent steering actions. The process of maintaining the lane is continuous, and once an action is initiated, it flows in the network independently and concurrent to other actions. The steering model combines several concepts based on current literature (Table 1). (For further discussion of the steering model, see Tsimhoni and Liu, 2003.)

Table 1. Concepts used in the steering model.

Concept	Description
Hierarchical task analysis	Driving a vehicle is described as a hierarchical combination of tasks.
Availability of visual input	Image processing is not performed explicitly. Rather, estimated processing time is added and the extracted data are retrieved directly.
Roles of focal and ambient visual systems	Most of the visual input for steering is perceived by the ambient visual system, around the lane markers in front of the vehicle.
Concurrent cognitive processing	Eye movements, information analysis, and motor actions are performed concurrently.
Limited speed control	The model currently operates at fixed speeds.
Steering movements	The steering wheel is moved in single-phase open-loop corrections followed by closed-loop adjustments.

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 steering model to produce relevant steering actions in real-time, Promodel was interfaced with the DriveSafety Research Simulator, a high fidelity driving simulation system used for driving research and training. It utilizes a 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, it can also be controlled externally by receiving inputs digitally.

Communication between Promodel and the driving simulator (Figure 2) was implemented via a TCP/IP host, created as an independent process by a communication DLL on the Promodel computer. Promodel sent and received data as function calls directly to the DLL. The driving simulator sent and received data by a TCP client that was connected to the host.

The communication protocol accommodated the event-based approach of Promodel and the time-based approach of the driving simulator. 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.
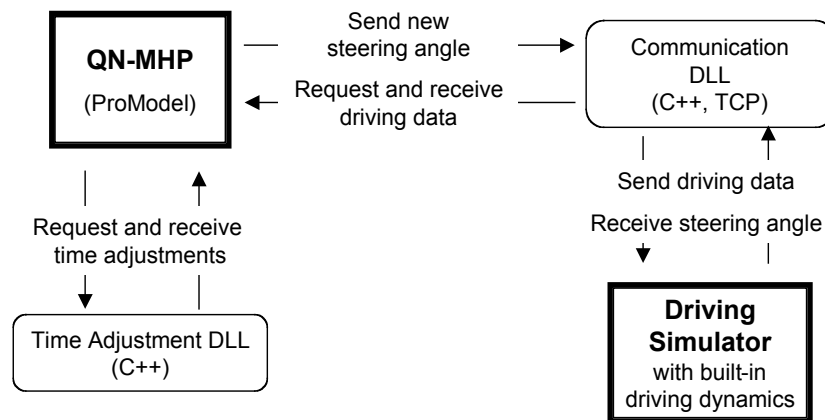

Figure 2. Block diagram of the interface between QN-MHP and the driving simulator.

Since Promodel is an event-based simulation program, its speed of operation varies as a function of the number of events fired at any given moment. Thus, the simulated time progresses at an irregular speed and is different from the actual time. To overcome this problem, an external time adjustment function was created and integrated with Promodel via a DLL (dynamic link library). This function was called by Promodel every 10 ms to adjust with its internal clock with the actual clock. It was assumed (and verified) that the computer running Promodel was always fast enough to perform all the required events faster than the real time equivalent.

TEST DRIVE

QN-MHP was successful in steering the driving simulator on a test course. Figure 3 shows the physical layout. The driving simulator demonstrated realistic steering behavior. It 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. Most important, the continuous flow of information through the QN-MHP servers appeared to represent the workload of steering a vehicle in its lane.

Figure 3. QN-MHP steering the driving simulator.

DISCUSSION

The work presented here illustrates the ability of QN-MHP to model the concurrent perceptual, cognitive, and motor activities of steering. Using a straightforward interface and a GOMS like task analysis, the model processed external information and created steering actions to maintain the vehicle in its lane. Since the structure of QN-MHP is context free, the steering model did not require manipulation of the architecture of the model. The successful interfacing of ProModel with a driving simulator opens a range of possible areas of research and application for the current model. For example, it allows for visible and real-time demonstration of the steering strategy implemented in the model. It may also serve as an "autopilot" for the simulator.

The potential strength of QN-MHP as a model of driving behavior is in its ability to add concurrent activities without limiting or predefining their order of occurrence. The success of modeling concurrent perceptual, cognitive, and motor activities of steering in a truly concurrent network architecture opens the door for modeling other concurrent activities. Our on-going research builds upon the current work and expands it in two aspects: (1) The driving task will be expanded to include speed control and to alter behavior based on traffic. (2) A secondary in-vehicle task will be added as a parallel activity. As other perceptual modalities are added to the QN-MHP architecture (e.g., vestibular, auditory), their addition to the driving task, and their effects on it, will be investigated.

REFERENCES

Feyen, R. G., & Liu, Y. (2001a). Modeling task performance using the queueing network - model human processor (QN-MHP). *Proceedings of the 4th International Conference on Cognitive Modeling*.

Feyen, R. G., & Liu, Y. (2001b). The queueing network-model human processor (QN-MHP): An engineering approach for modeling cognitive performance. *Proceedings of the Human Factors and Ergonomics Society 45th annual meeting*.

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