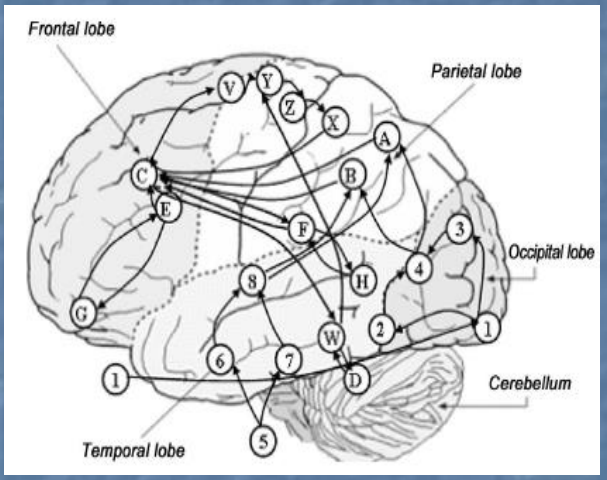
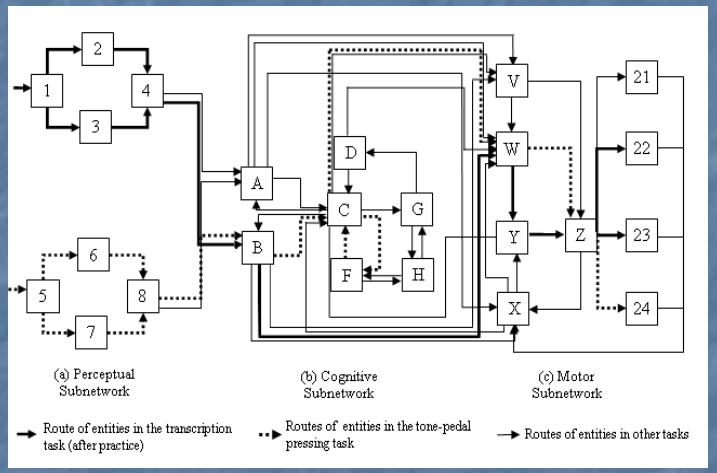
Integrative Modeling and Simulation of Human Behavior and Human-Machine Systems with Queuing Network (QN) Architecture





Queueing Network of Mental Architecture

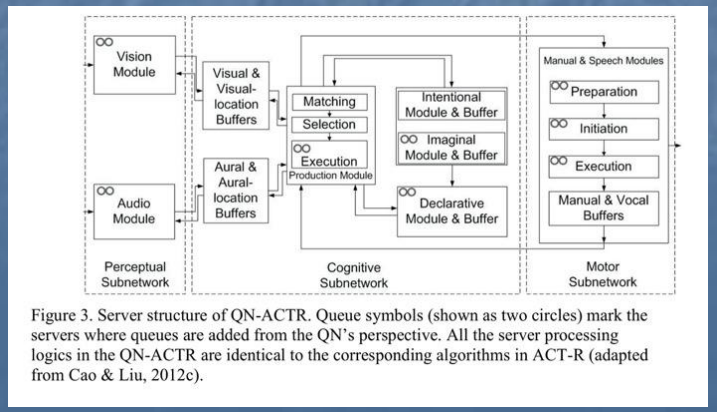
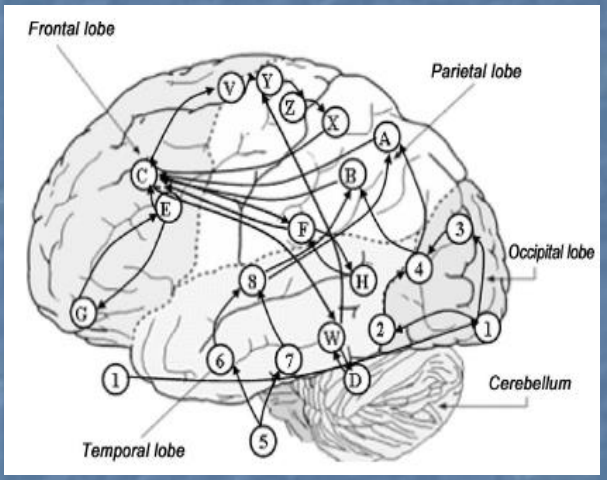
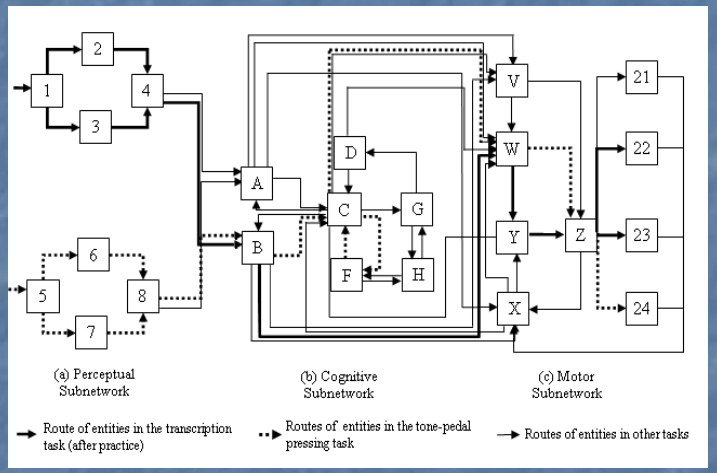


Figure 3. Server structure of QN-ACTR. Queue symbols (shown as two circles) mark the servers where queues are added from the QN's perspective. All the server processing logics in the QN-ACTR are identical to the corresponding algorithms in ACT-R (adapted from Cao & Liu, 2012c).

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





Queueing Network of Mental Architecture

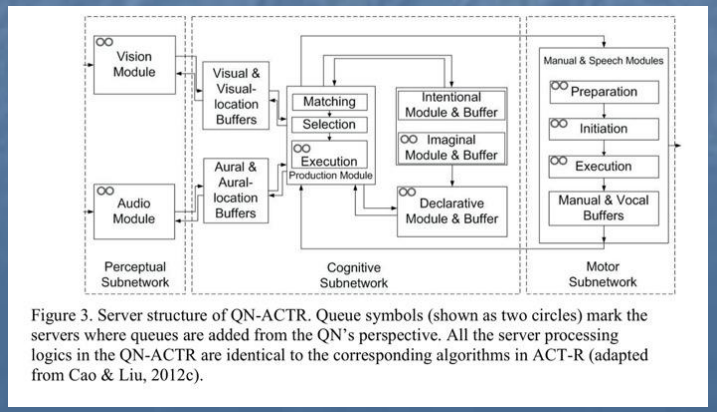


Figure 3. Server structure of QN-ACTR. Queue symbols (shown as two circles) mark the servers where queues are added from the QN's perspective. All the server processing logics in the QN-ACTR are identical to the corresponding algorithms in ACT-R (adapted from Cao & Liu, 2012c).

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

1. RT: Reaction Time (QN-RT) and Mental Structure
2. RT and Accuracy (QN-RMD) (Mental Structure vs State of Mind)
3. Procedural Tasks (QN-MHP or QN-MHP-BE)
4. Complex Cognition Tasks (QN-ACTR)
5. Visual Attention tasks (QN-NSEEV)
6. Manual or Continuous Control tasks (QN-Control)
7. Basic Body Motion tasks (QN-MTM)
8. Mind-Body System (QN-MBS)
9. Neural level (QN-Neural)
10. Nervous and Endocrine Systems (QN-NES)
11. Multi-Person Multi-Machine QN (QN-HMN)
12. Engineering Applications (relations with **Task Network** Methods such as MicroSaint#, IMPRINT)

Queueing Networks (QN) are everywhere (as phenomena)

- Airports
- Hotels/restaurants/conference rooms
- Manufacturing facilities
- Traffic networks (road, air, rail,…)
- Computer networks
- Inside computer (chips, ,memory, CPUs)
- Shopping malls
- You name it…
- *** ***
- Inside our mind and body!

--The theme of this Workshop

Queueing Networks (QN) are powerful (as a method)

- Intuitive
- Mathematical
- Computational
- Simulational
- "Generative"
- Versatile
- Widely used
- Integrative

Queueing Networks (QN) are easy-to-understand (the concepts)

- Servers

  (e.g., service time, capacity, busy time)

- Customers (Entities)

  (e.g., arrival time/rate, departure time, sojourn time, "balk rate")

- Routes/Paths and Routing Probabilities

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

1. RT: Reaction Time (QN-RT) and Mental Structure
2. RT and Accuracy (QN-RMD) (Mental Structure vs State of Mind)
3. Procedural Tasks (QN-MHP or QN-MHP-BE)
4. Complex Cognition Tasks (QN-ACTR)
5. Visual Attention tasks (QN-NSEEV)
6. Manual or Continuous Control tasks (QN-Control)
7. Basic Body Motion tasks (QN-MTM)
8. Mind-Body System (QN-MBS)
9. Neural level (QN-Neural)
10. Nervous and Endocrine Systems (QN-NES)
11. Multi-Person Multi-Machine QN (QN-HMN)
12. Engineering Applications (relations with **Task Network** Methods such as MicroSaint#, IMPRINT)

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

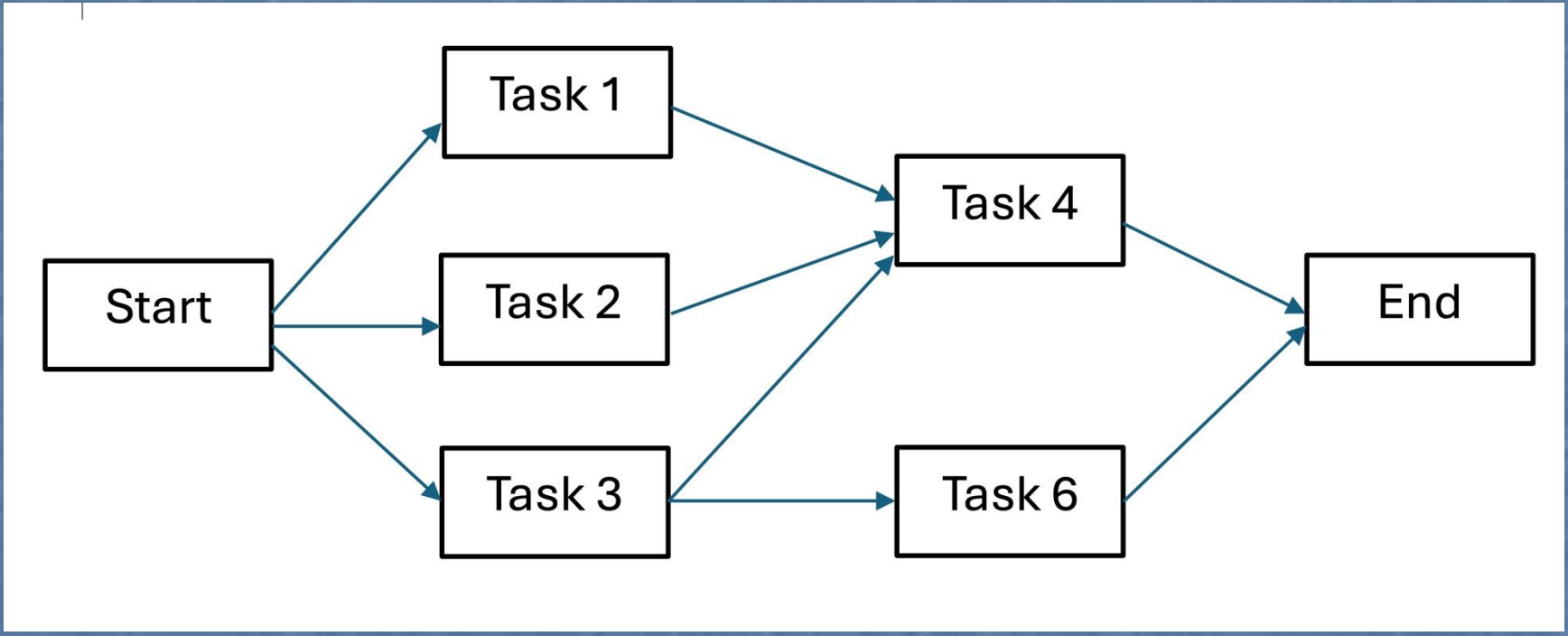
Mathematical Models of RT and Response Accuracy (sequential sampling models)

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

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

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

Task Network
(Project network, PERT, CPN, Mission Network, etc.)



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

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

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

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

General Terms: Human Factors

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

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

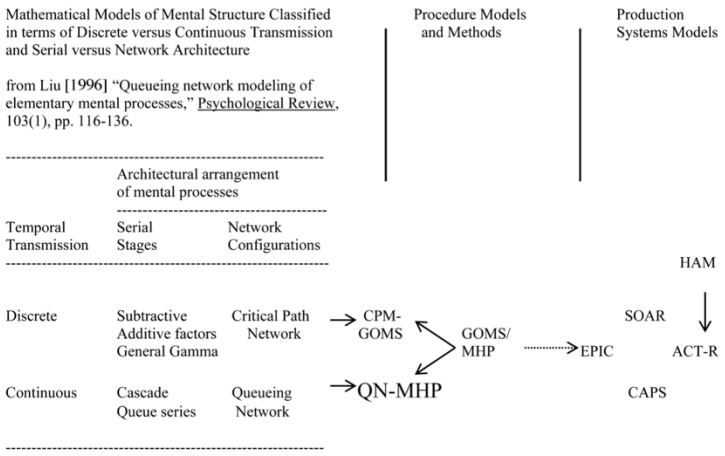
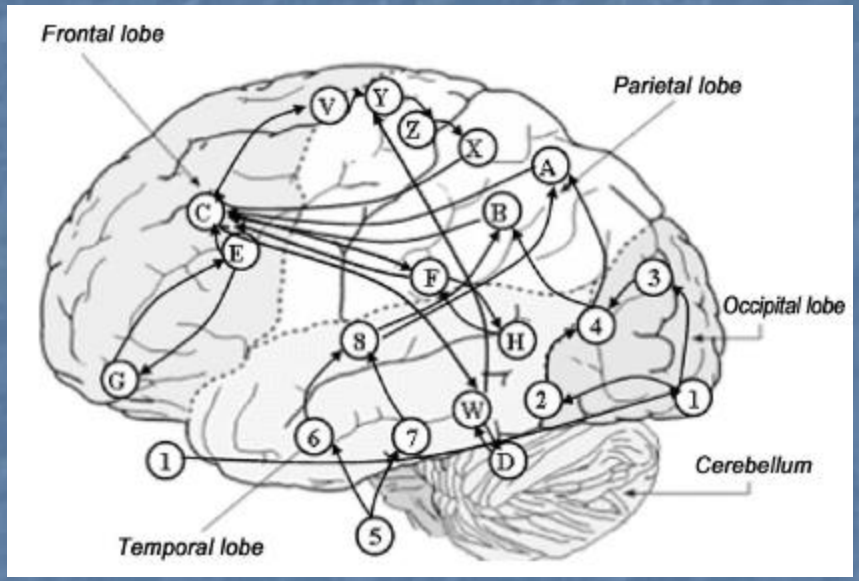
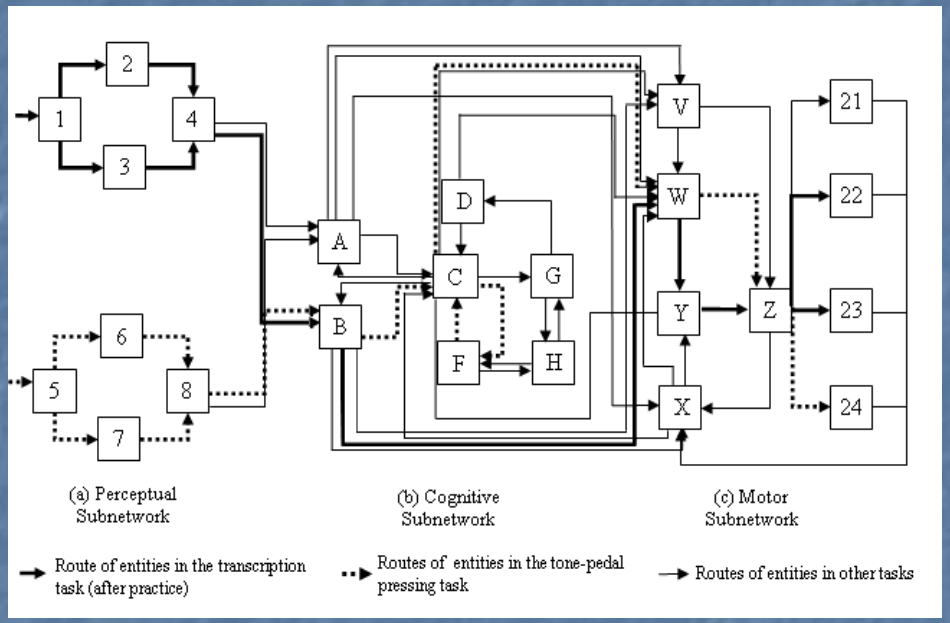


Fig. 1. Mathematical models of mental structure and procedure/production system models of cognitive architecture.

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

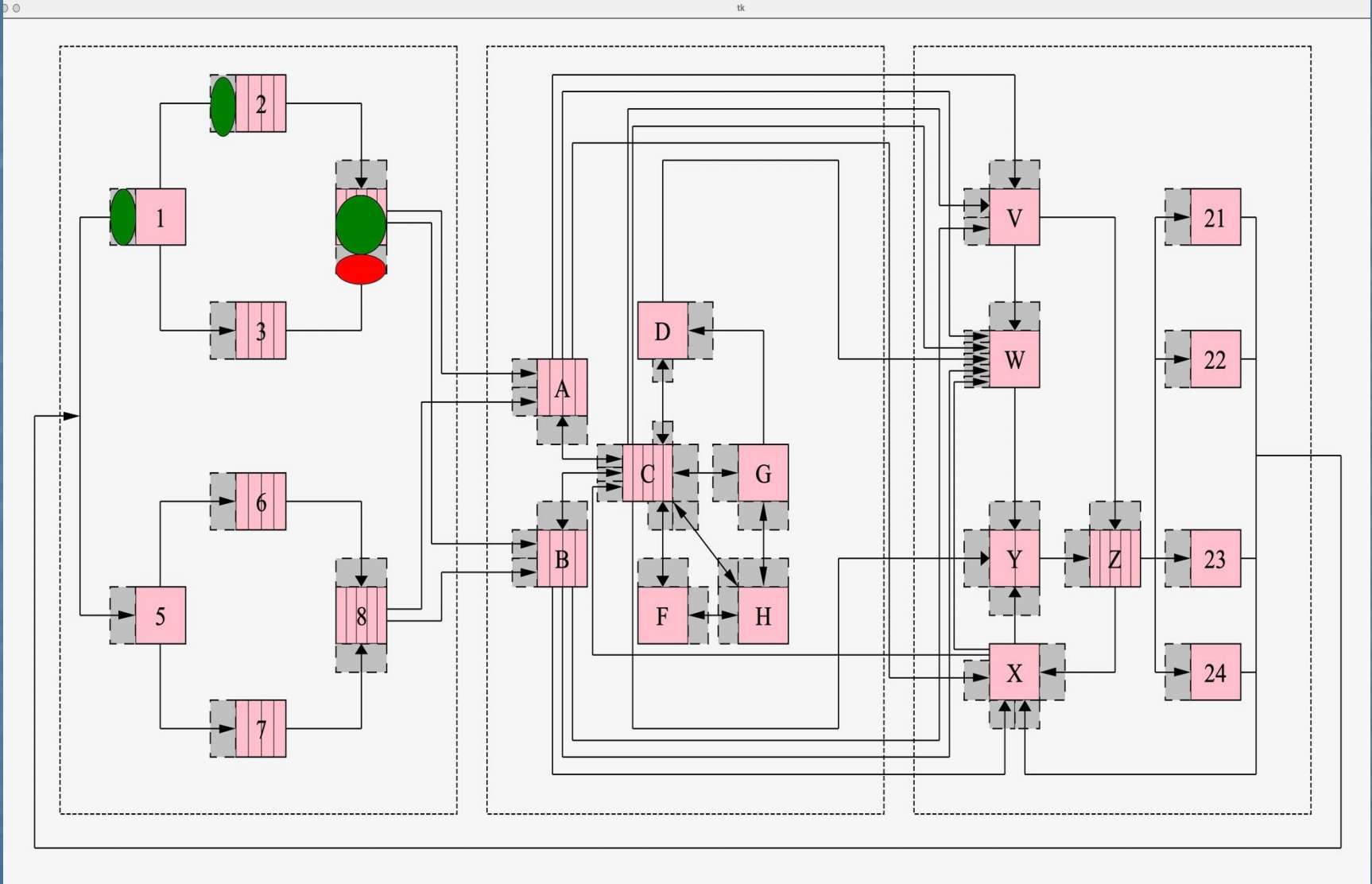




Human Brain

Queueing Network of Mental Architecture

QN-MHP-BE or QN-MHP



INTL. JOURNAL OF HUMAN–COMPUTER INTERACTION, *25*(6), 554–581, 2009

ISSN: 1044-7318 print / 1532-7590 online
DOI: 10.1080/10447310902973182

Taylor & Francis
Taylor & Francis Group

QN-ACES: Integrating Queueing Network and ACT-R, CAPS, EPIC, and Soar Architectures for Multitask Cognitive Modeling

Yili Liu
The University of Michigan

Comprehensive and computational models of human performance have both scientific and practical importance to human–machine system design and human-centered computing. This article describes QN-ACES, a cognitive architecture that aims to integrate two complementary classes of cognitive architectures: Queueing network (QN) mathematical architecture and ACT–R, CAPS, EPIC, and Soar (ACES) symbolic architectures. QN-ACES represents the fourth major step along the QN architecture development for theoretical and methodological unification in cognitive and human–computer interaction modeling. The first three steps—QN architecture for response time, QN-RMD (Reflected Multidimensional Diffusions) for response time, response accuracy, and mental architecture, and QN-MHP (Model Human Processor) for mathematical analysis and real time simulation of procedural tasks—are summarized first, followed by a discussion of the rationale, importance and specific research issues of QN-ACES.

1. INTRODUCTION

The increasing complexity of advanced human–machine systems makes it necessary for system designers to consider human capabilities and limitations as early as possible in system design. In order to reduce risks associated with poor task design with appropriate tools and methods for task analysis and function allocation, it is important to develop models of human performance and human-system interaction that are comprehensive, computational, science-driven, and application-relevant.

Models of human performance and human–system interaction should be comprehensive to capture the whole range of concurrent perceptual, cognitive, motor, and communication activities of human-system performance. These models should be computational and computerized to allow quantitative and rigorous simulation and analysis of design alternatives and scenarios. These models should be science driven, with deep roots in and strong connections with cognitive science

Correspondence should be addressed to Yili Liu, Department of Industrial & Operations Engineering, The University of Michigan, 1205 Beal Avenue, Ann Arbor, MI 48109-2117. Email: yililiu@umich.edu

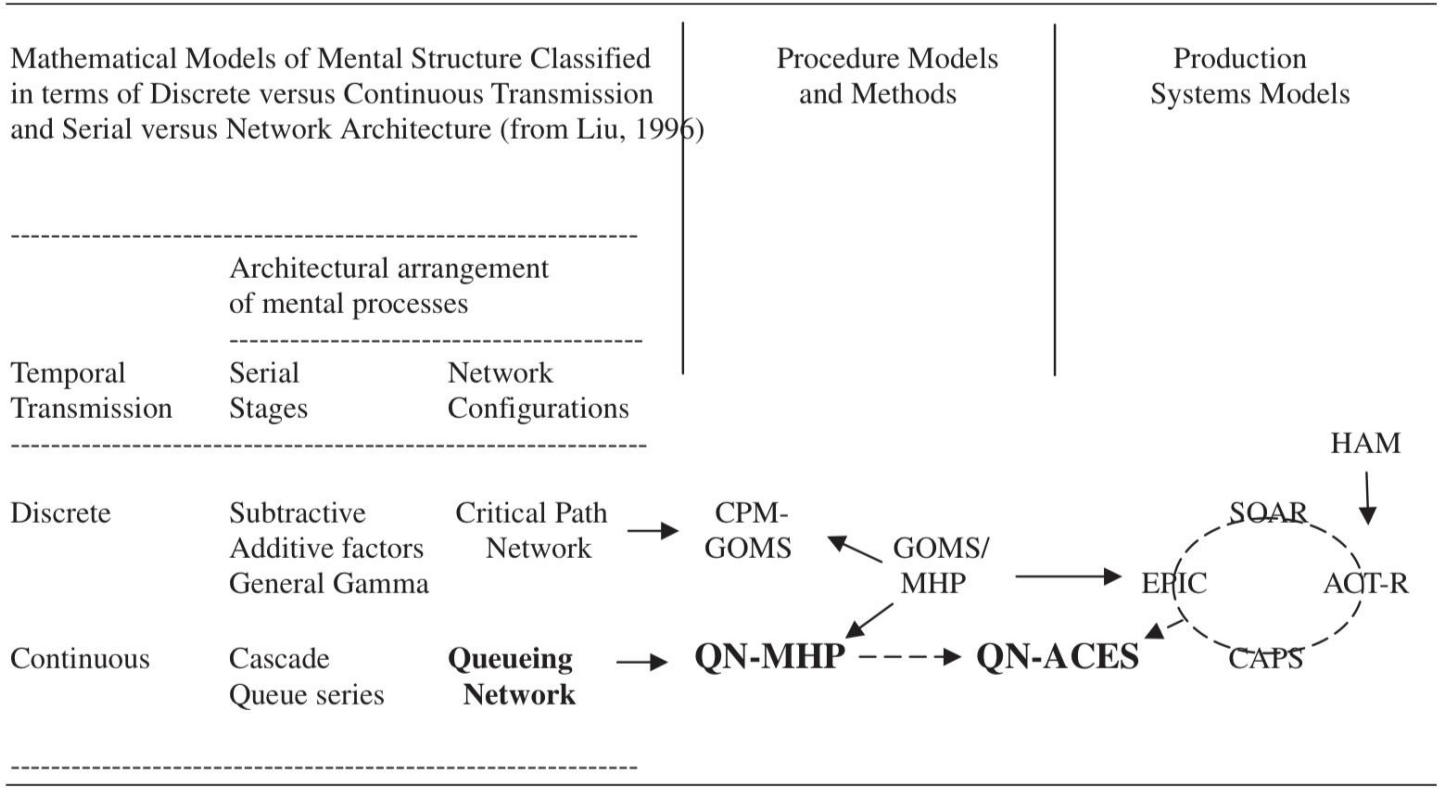
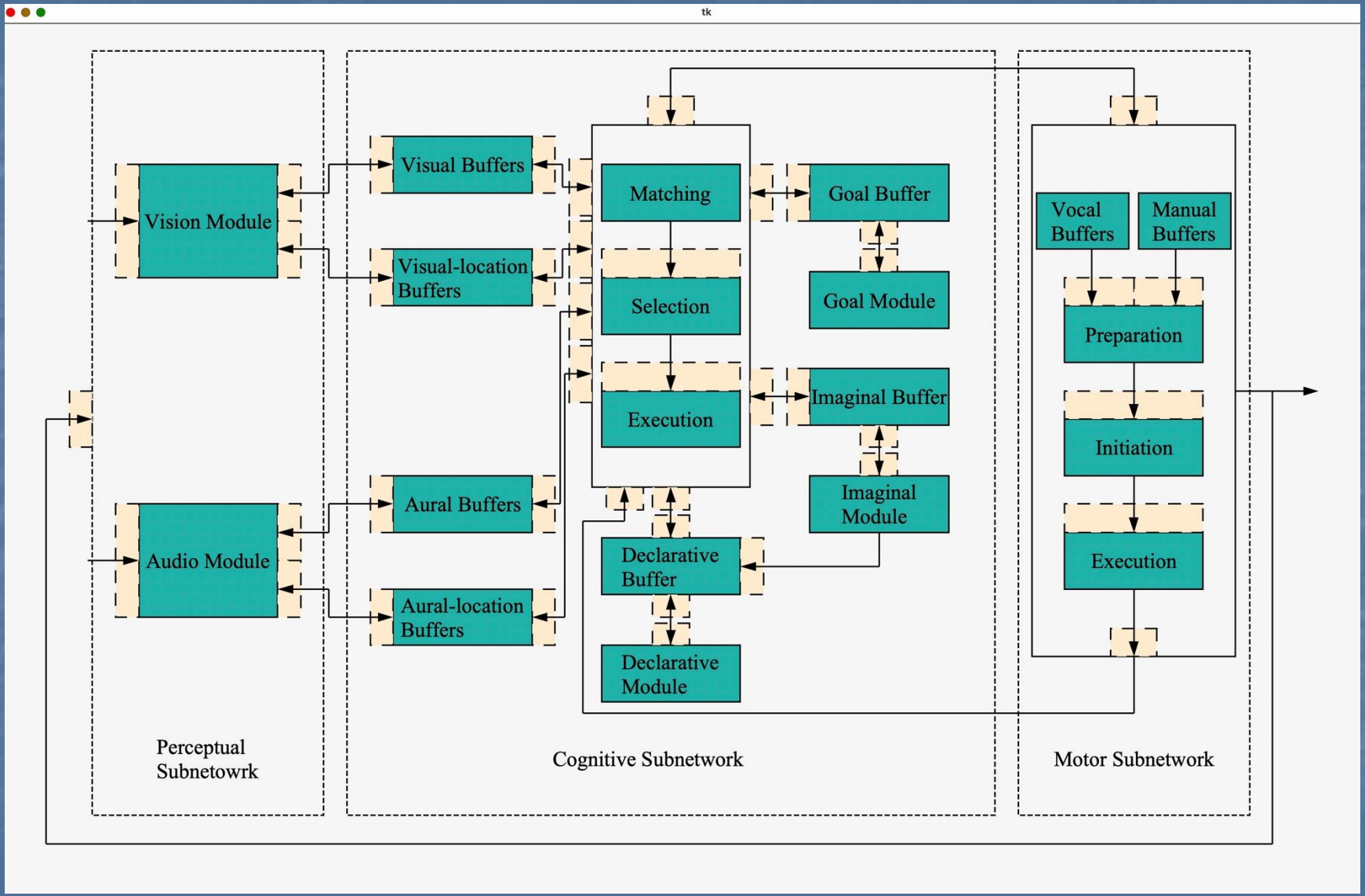
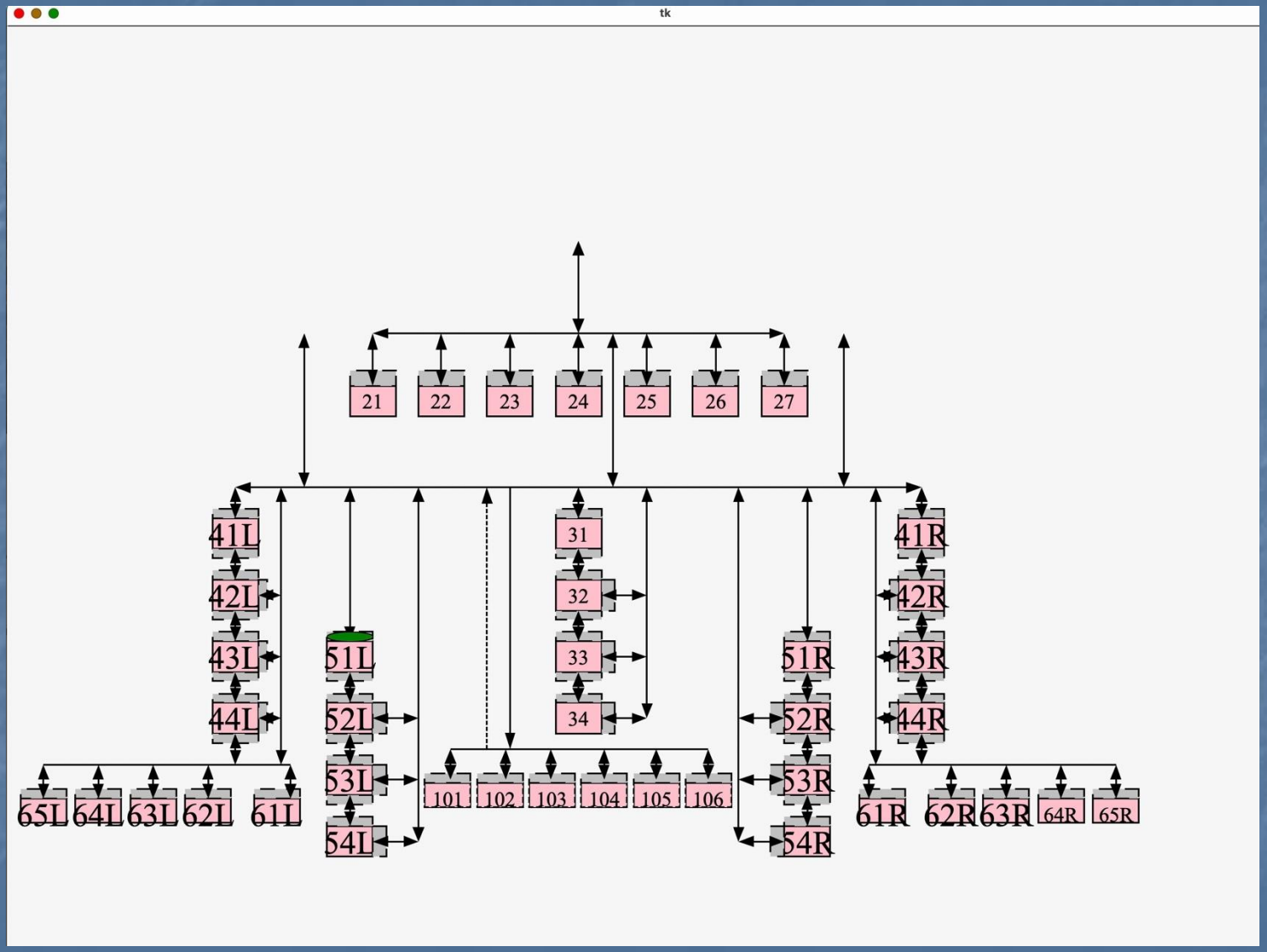


FIGURE 2 Mathematical and symbolic models of mental architecture (Liu, 2006; Liu et al, 2006) showing the relationship between QN, QN-MHP, QN-ACES and a sample of influential cognitive architectures. *Note*: By integrating the complementary schools of mathematical (left half of the figure) and symbolic (right half) models, QN-MHP supports both precise mathematical analysis and real-time generation of behavior, thus capitalizing on the strengths and overcoming the weaknesses of either mathematical or symbolic modeling alone.

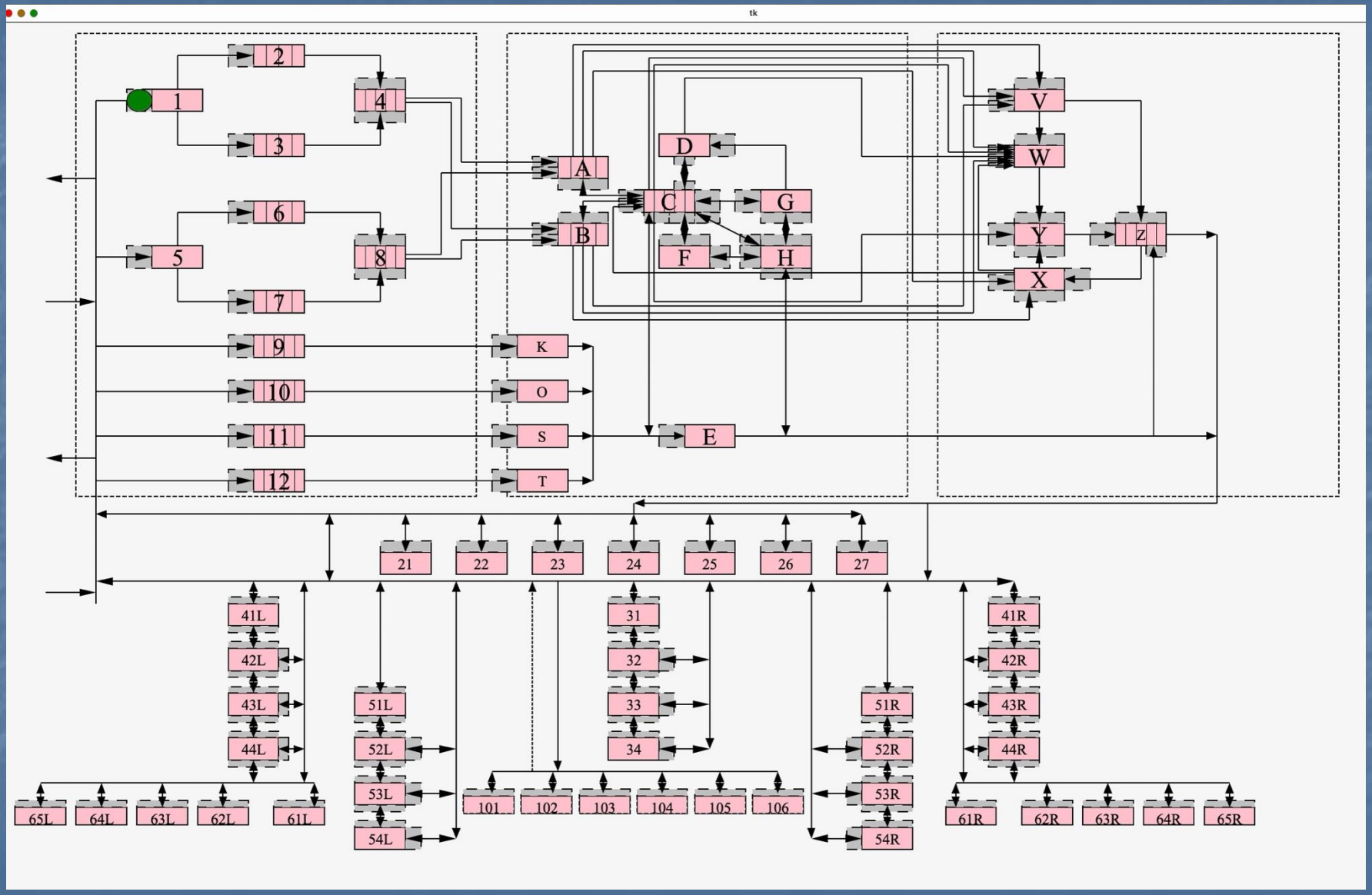
QN-ACTR


tk
Vision Module
Audio Module
Visual Buffers
Visual-location Buffers
Aural Buffers
Aural-location Buffers
Matching
Selection
Execution
Goal Buffer
Goal Module
Imaginal Buffer
Imaginal Module
Declarative Buffer
Declarative Module
Vocal Buffers
Manual Buffers
Preparation
Initiation
Execution
Perceptual Subnetowrk
Cognitive Subnetwork
Motor Subnetwork

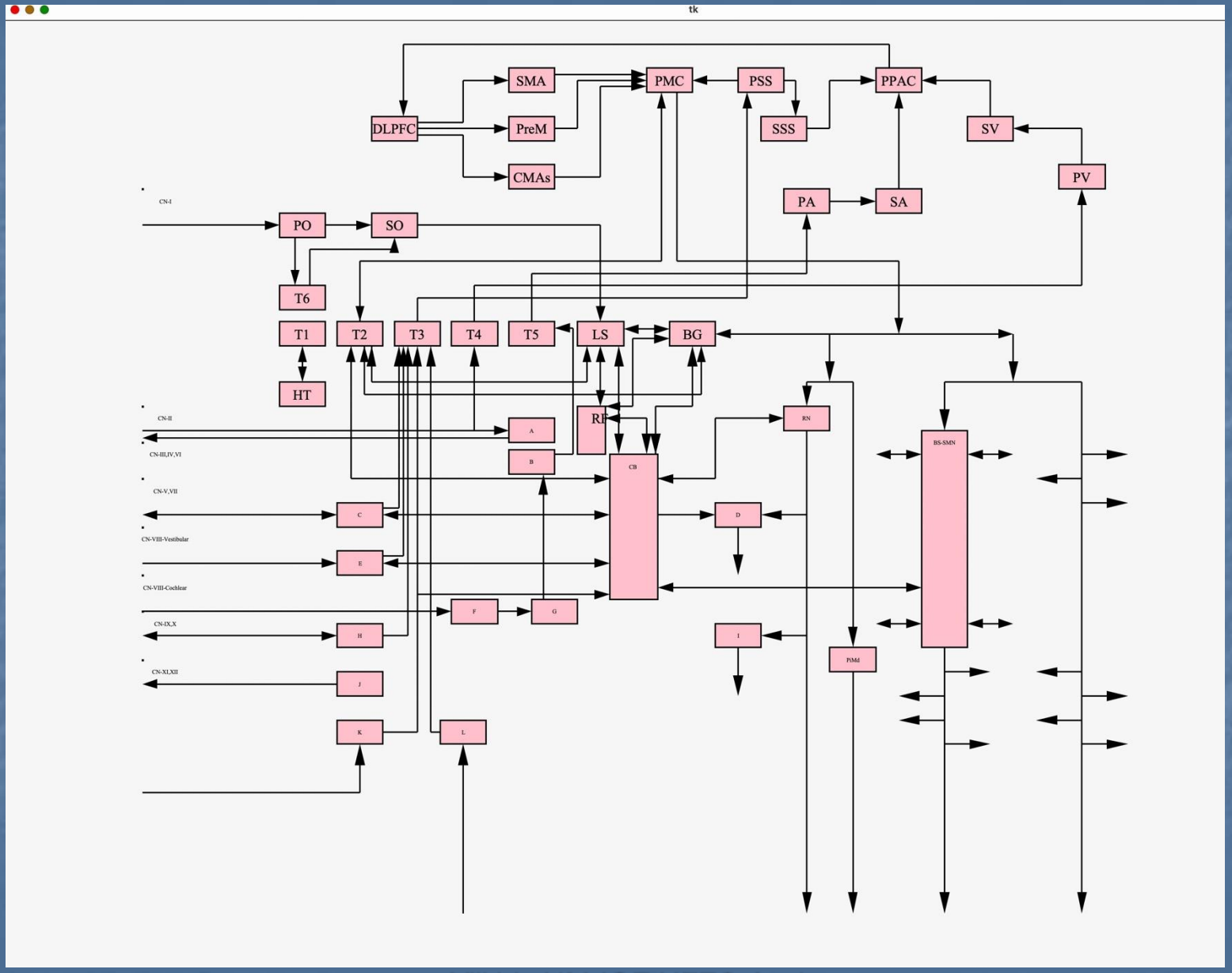
QN-BDS


tk
21
22
23
24
25
26
27
41L
42L
43L
44L
31
32
33
34
41R
42R
43R
44R
51L
52L
53L
54L
51R
52R
53R
54R
101
102
103
104
105
106
65L
64L
63L
62L
61L
61R
62R
63R
64R
65R

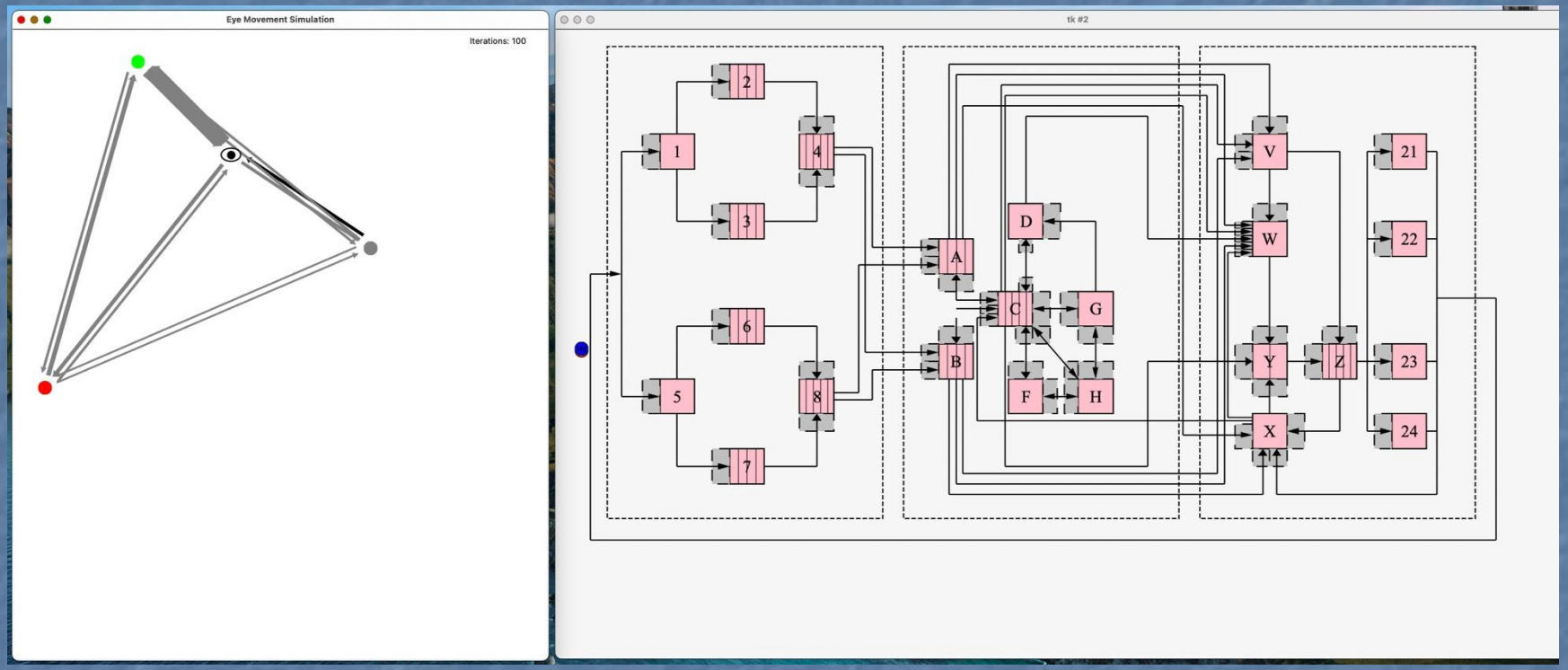
QN-MBS


tk
1
2
3
4
5
6
7
8
9
10
11
12
A
B
C
D
E
F
G
H
K
O
S
T
V
W
X
Y
Z
21
22
23
24
25
26
27
41L
42L
43L
44L
51L
52L
53L
54L
31
32
33
34
101
102
103
104
105
106
51R
52R
53R
54R
41R
42R
43R
44R
65L
64L
63L
62L
61L
61R
62R
63R
64R
65R

QN-NES


tk
SMA
PMC
PSS
PPAC
DLPFC
PreM
SSS
SV
CMAs
PV
PA
SA
CN-I
PO
SO
T6
T1
T2
T3
T4
T5
LS
BG
HT
CN-II
RF
RN
A
BS-SMN
CN-III,IV,VI
B
CB
CN-V,VII
C
D
CN-VIII-Vestibular
E
CN-VIII-Cochlear
F
G
CN-IX,X
H
I
PtMd
CN-XI,XII
J
K
L

QN-SEEV


Eye Movement Simulation
Iterations: 100
tk #2
1
2
3
4
5
6
7
8
A
B
C
D
F
G
H
V
W
X
Y
Z
21
22
23
24

QN-RL-EM

QN-Control (Classical)

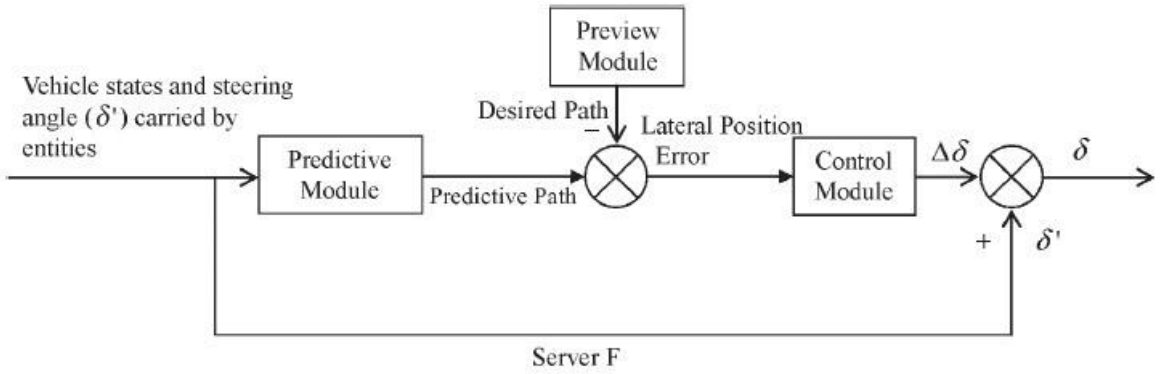


Fig. 2. Schematic of driver lateral control implemented in Server F.

The preview module previews the desired path for a preview interval to obtain information of the desired path. This paper

QN-Control (Classical)

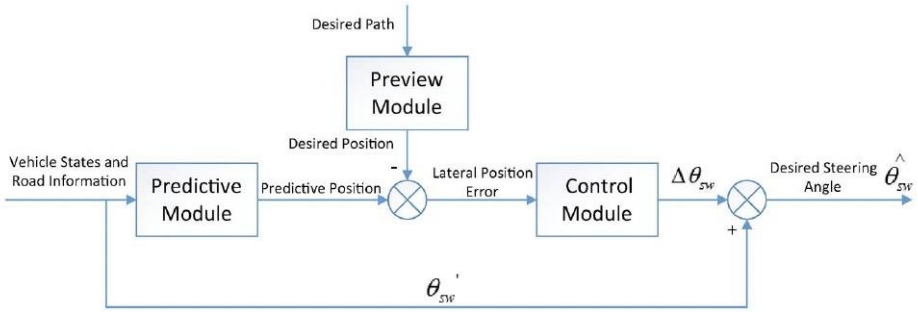


Fig. 3. Block diagram of driver preview model implemented in Server F.

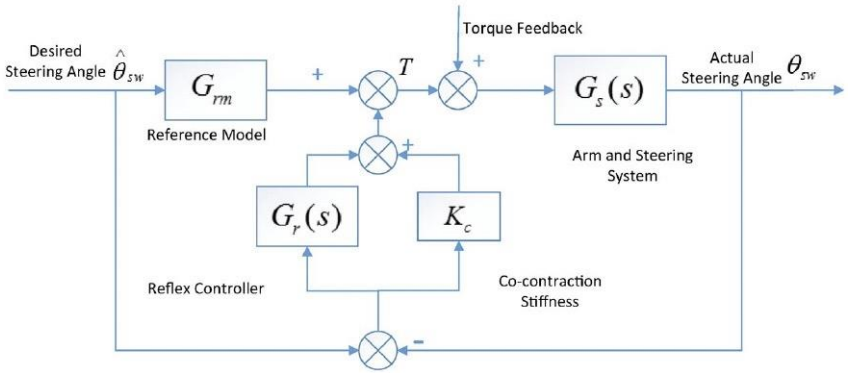


Fig. 4. The structure of the neuromuscular system.

QN-Control (Classical)

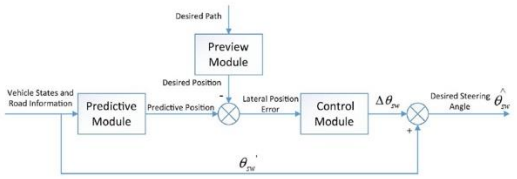


Fig. 3. Block diagram of driver preview model implemented in Server F.

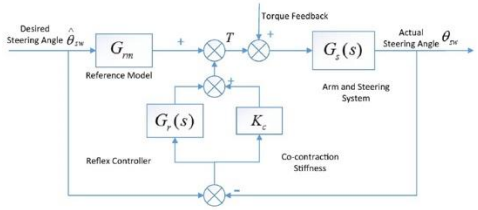


Fig. 4. The structure of the neuromuscular system.

As shown in Fig. 2, entities carrying the inputs of the entire model first enter the visual perceptual subnetwork (Servers 1 (visual input) → 2/3 (visual recognition/visual location) → 4 (perceptual integrator)). Via Server 4 (Perceptual-Integrator server), the entities are routed to the cognitive subnetwork, including Servers A (visuospatial sketchpad), C (central executor), and F (complex cognitive function), where the desired steering angle is computed according to the preview model, as described in Fig. 3. Then entities carrying the desired steering angle $\hat{\theta}_{sw}$ travel to the motor subnetwork via Server C, where the desired steering angle carried by entities is transported into the neuromuscular system via Server Z.

As shown in Fig. 3, the driver preview model implemented at Server F of the QN architecture contains three main modules: preview module, predictive module, and control module. The preview module previews the desired path for a preview time to obtain the information of the desired path (note that the desired path was predetermined). The predictive module predicts the vehicle response within the preview time by using an internal model (i.e., 3 degree-of-freedom (DOF) vehicle dynamics model, including longitudinal, lateral, and yaw movement).

The control module computes the desired control input (i.e., the increment of steering angle $\Delta\theta_{sw}$) to make a vehicle track the desired path. The calculation proceeds as follows [1]. The desired acceleration a_y (assuming that it is a constant in the preview time) is first calculated by

$$a_y = \frac{2 \times (\Delta E - v \times t_p)}{t_p^2}, \tag{1}$$

where ΔE is the error between the desired lateral position obtained with the desired path and predictive lateral position computed with the internal vehicle dynamics model, v is the current velocity, and t_p is the preview time.

The changed steering angle $\Delta\theta_{sw}$ is then calculated with a proportional derivative (PD) controller of acceleration:

$$\Delta\theta_{sw} = k_p \times a_y + k_d \times a'_y, \tag{2}$$

where k_p and k_d are the coefficients of the PD controller, and a'_y is the first derivative of the acceleration.

Finally, a new steering angle $\hat{\theta}_{sw}$ is computed by

$$\hat{\theta}_{sw} = \Delta\theta_{sw} + \theta'_{sw}, \tag{3}$$

where θ'_{sw} is the steering angle of the last cycle. More details on the QN-based driver lateral control can be found in our previous work [1].

C. Neuromuscular Dynamics

The neuromuscular system (NMS) was developed according to [6] and implemented outside of the QN architecture. The difference between the neuromuscular system in this paper and that of [6] was in the parameter values. The model was used to produce an actual steering angle given the reference input (i.e., the desired steering angle). It can reject any disturbance torque on the steering wheel, to which an external disturbance leads. The structure of NMS used is shown in Fig. 4 [6].

The proposed model includes feedforward module G_{rm}, the co-contraction stiffness K_c, the stretch reflex controller $G_r(S)$, and the arm and steering system $G_s(s)$. G_{rm} represents the angle-torque stiffness, which can provide a steering torque proportional to the desired angle $\hat{\theta}_{sw}$. K_c denotes the increased

QN-Control (Modern)

content may change prior to final publication. Citation information: DOI 10.1109/TVT.2023.3342151

output discrete driving commands with limited numbers in the brain-control driving, while they send continuous driving signals in the manual-control driving; 2) to execute their driving intentions, drivers perform brain-control operation via brain signals in the brain-control driving, while they conduct limb-control operation via muscles and peripheral nerves in the manual-control driving.

Since the car-following behavior of the brain-control driving consists of the car-following decision and brain-control operation sub-behaviors, we model it by fusing the models of the two sub-behaviors. As shown in Fig. 2, we first design a car-following decision model to simulate the car-following decision of the brain-control drivers. Then, the brain-control behavior and BCI performance models are applied to simulate the brain-control operation of the users. After that, we build two brain-control car-following models (i.e., CF-BCB and CF-BCI models) by combining the car-following decision model with the brain-control behavior and BCI performance models in the QN architecture, respectively. The QN architecture, including perceptual and cognitive subnetworks, is employed to denote human information processing in performing the brain-control driving.

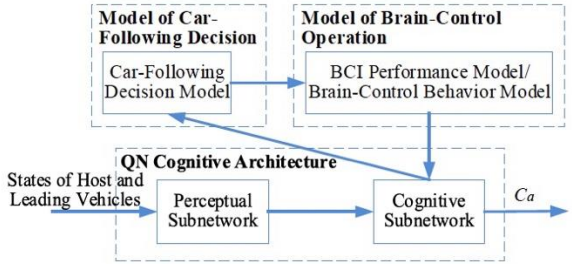


Fig. 2. Architecture of the brain-control car-following models.

MPC method, which performs well in the modeling of driving behaviors [13], [17], [18]. In the MPC algorithm, we use the longitudinal models of the host and leading vehicles as prediction models. They enable the car-following decision model to predict the future states of the two vehicles. The cost function incorporates a car-following and a smooth control objective.

1) Vehicle Model

The host vehicle is expected to imperfectly track its desired acceleration. Then, we can describe its longitudinal model with a first-order lag as

$$\tau \dot{a} + a = u \tag{1}$$

where τ denotes the time lag [33], and a and u represent the actual and desired accelerations of the host vehicle, respectively.

Integrating forward difference approximations, the dynamic model in (1) can be rewritten as

$$x(k+1) = Ax(k) + Bu(k),$$
$$A = \begin{bmatrix} 1 & T & 0 \\ 0 & 1 & T \\ 0 & 0 & 1-T/\tau \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 0 \\ T/\tau \end{bmatrix}. \tag{2}$$

where T denotes the sampling time and $x(k) = [s(k), v(k), a(k)]^T$ represents the state variable of the host vehicle. s and v are the absolute position and longitudinal velocity of the host vehicle, respectively. The users were allowed to control the host vehicle within a certain velocity range. The velocity constraint can be expressed as

$$v_{\min} \le v \le v_{\max} \tag{3}$$

where $v_{\min}$ and $v_{\max}$ are the limits of the longitudinal velocity.

QN-HMN

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

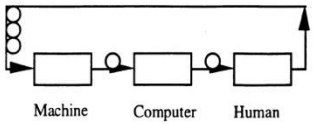


Fig.6. A closed queueing network model of human-computer interaction in a failure management system described in the text

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

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

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

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

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

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

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

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

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

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

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

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

p24 = p35 = p46 = p60 = p01 = 1

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

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

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

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

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

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

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

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

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

QN-HMN

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

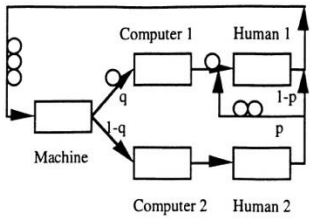


Fig. 7. A queueing network model of a human-computer network in the failure management system described in the text

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

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

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

REFERENCES

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

1. RT: Reaction Time (QN-RT) and Mental Structure
2. RT and Accuracy (QN-RMD) (Mental Structure vs State of Mind)
3. Procedural Tasks (QN-MHP or QN-MHP-BE)
4. Complex Cognition Tasks (QN-ACTR)
5. Visual Attention tasks (QN-NSEEV)
6. Manual or Continuous Control tasks (QN-Control)
7. Basic Body Motion tasks (QN-MTM)
8. Mind-Body System (QN-MBS)
9. Neural level (QN-Neural)
10. Nervous and Endocrine Systems (QN-NES)
11. Multi-Person Multi-Machine QN (QN-HMN)
12. Engineering Applications (relations with **Task Network** Methods such as MicroSaint#, IMPRINT)

(a) Tuning radio using the knob

(b) A zoom-in view of the physical panel

Figure 31. Task Procedure using the knob: (1) press the power button, (2) press the AM/FM button, (3) turn the knob to decrease or increase the frequency shown on the display ("590" as in the picture)

(a) Tuning radio on the touch screen

(b) Click the "Entertainment" button

(c) Click the "FM" then "Direct Tune" button

(d) Enter the radio frequency

Figure 32. Task procedure using the virtual buttons

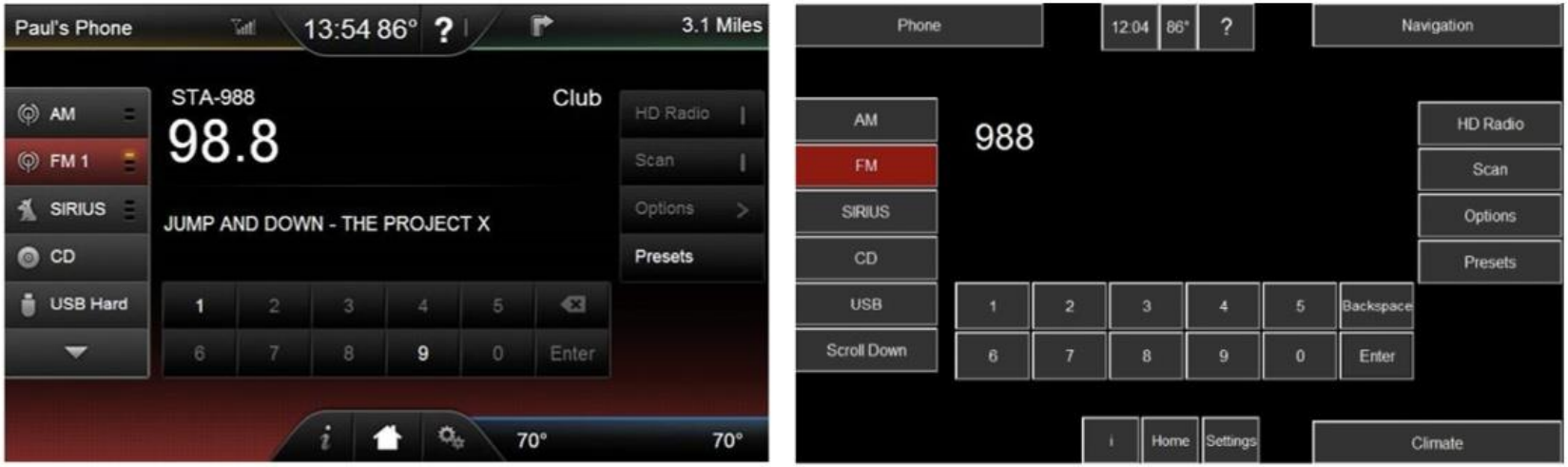


Fig. 9. Touch screen UI used in the experiment (left) and its digital mockup (right).

Fig. 10. Physical panel used in the experiment (left) and its digital mockup (right).

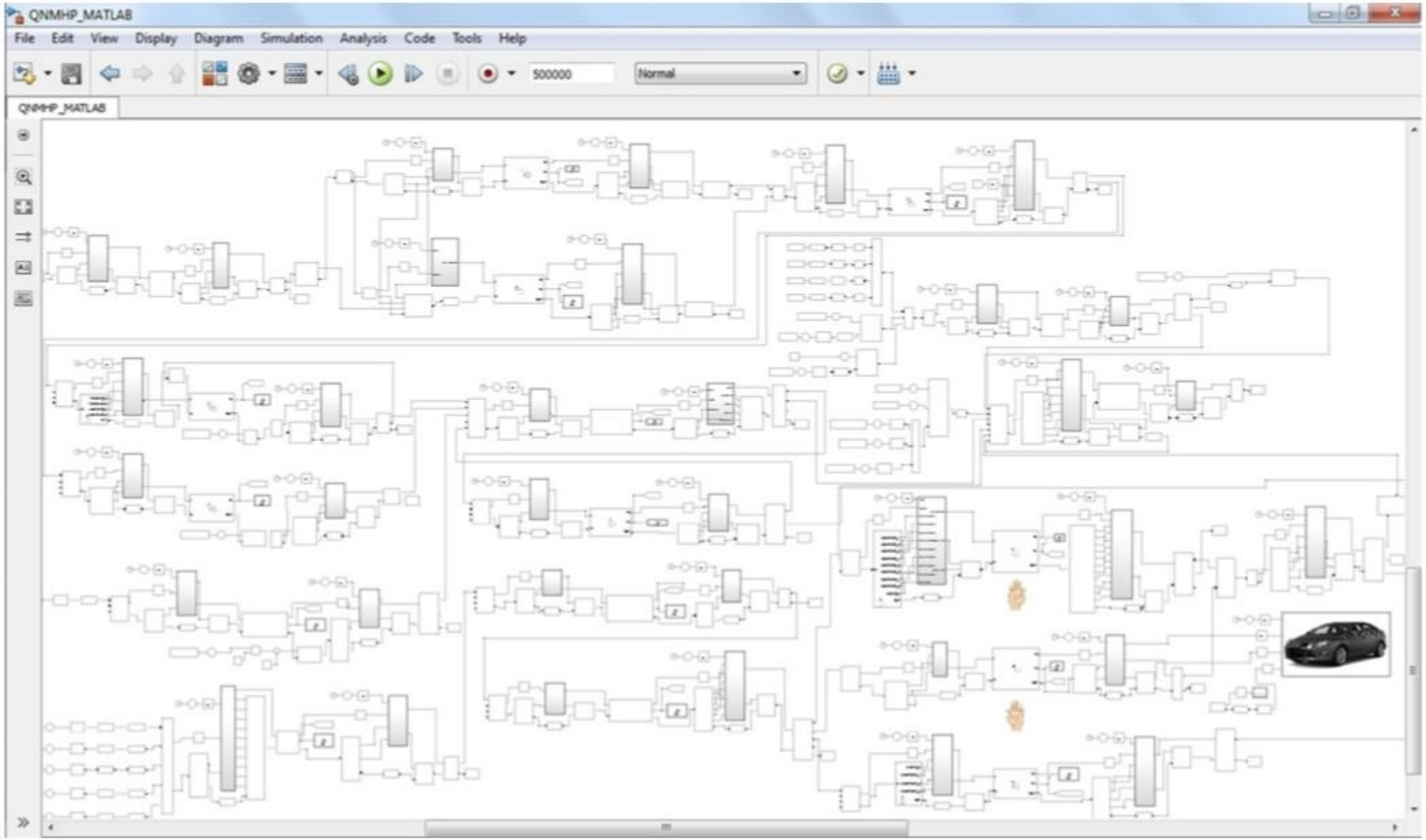


Fig. 2. A screenshot of the QN-MHP model implementation in MATLAB/Simulink.

Visualizing Mental Workload (perceptual, cognitive, motor loads)

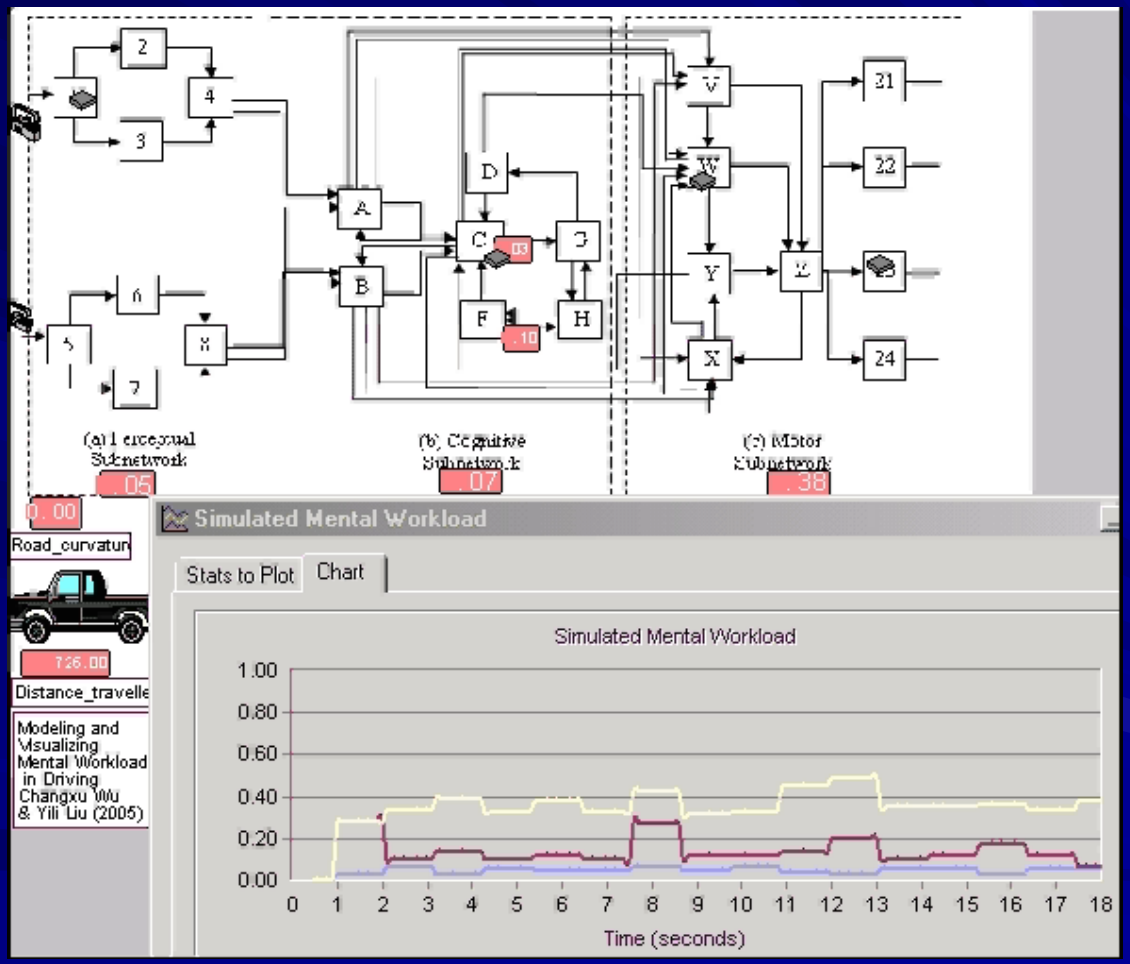


Table 3. Summary of anthropometric data for the "*Drag-with-finger*" operator development (n = 11).

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

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

3.2. Apparatus

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

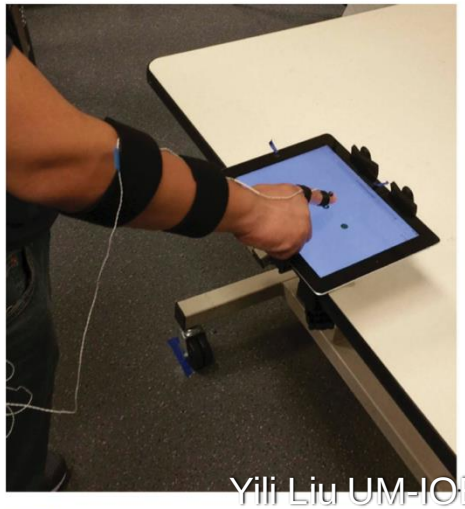


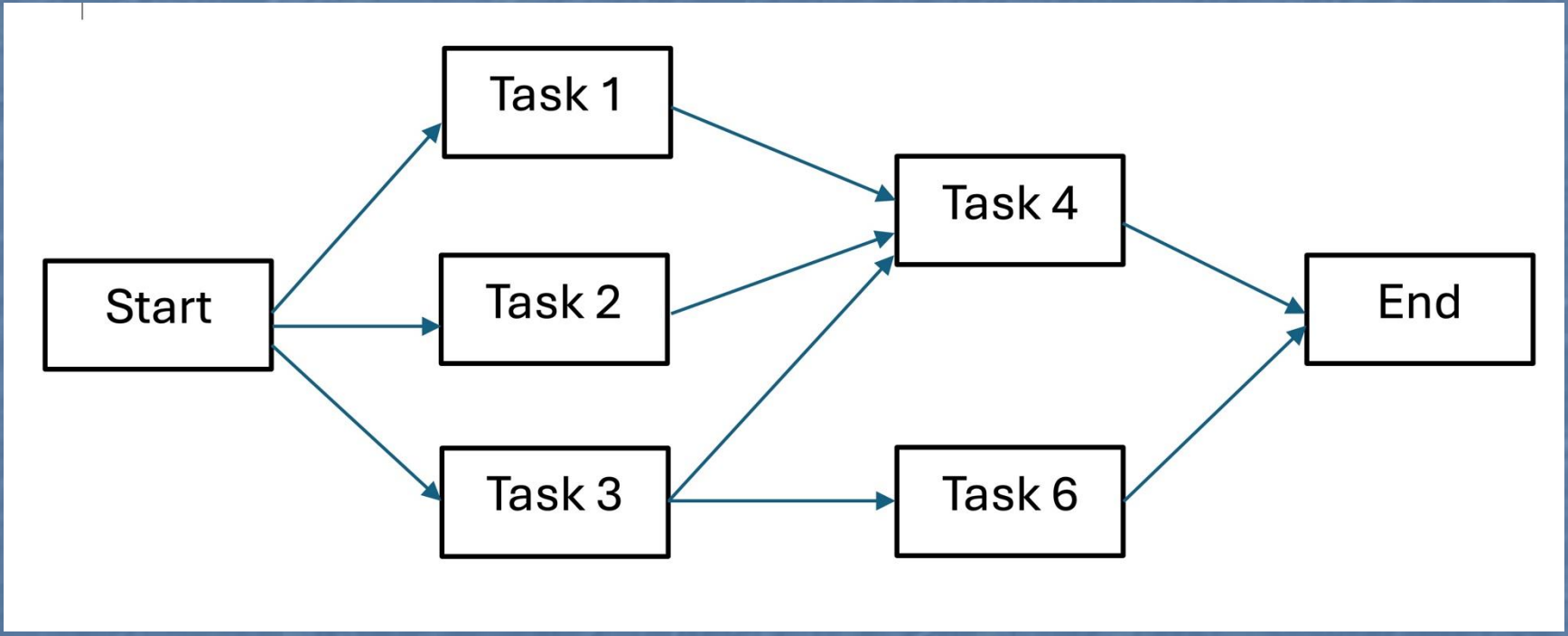
Figure 3. Experimental setup of the finger-drag gesture task.

3.3. Touchscreen gesture task and experimental design

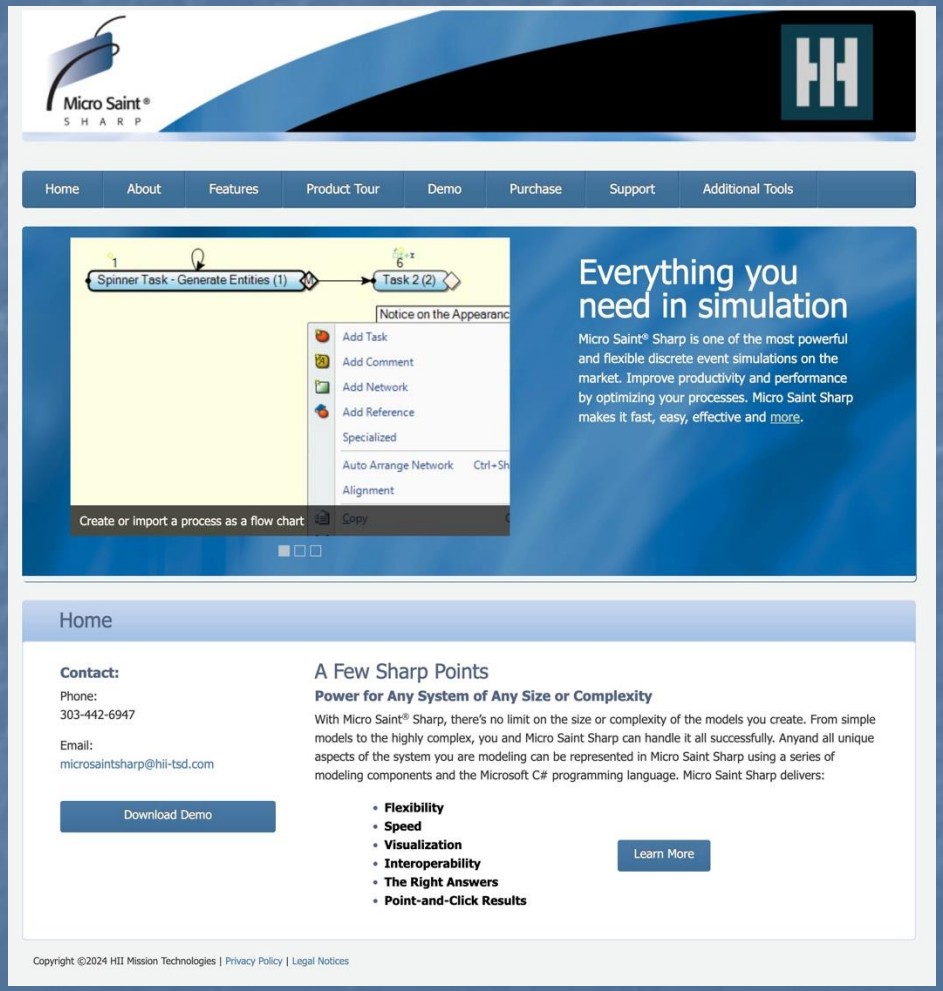
The performance of finger-drag gestures was measured, using a touchscreen interface used in Jeong and Liu (2017a). As shown in Figure 4, nine circles (i.e., eight target circles around one center circle) were designed on a touchscreen display, but only two circles (i.e., one center circle and one of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

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

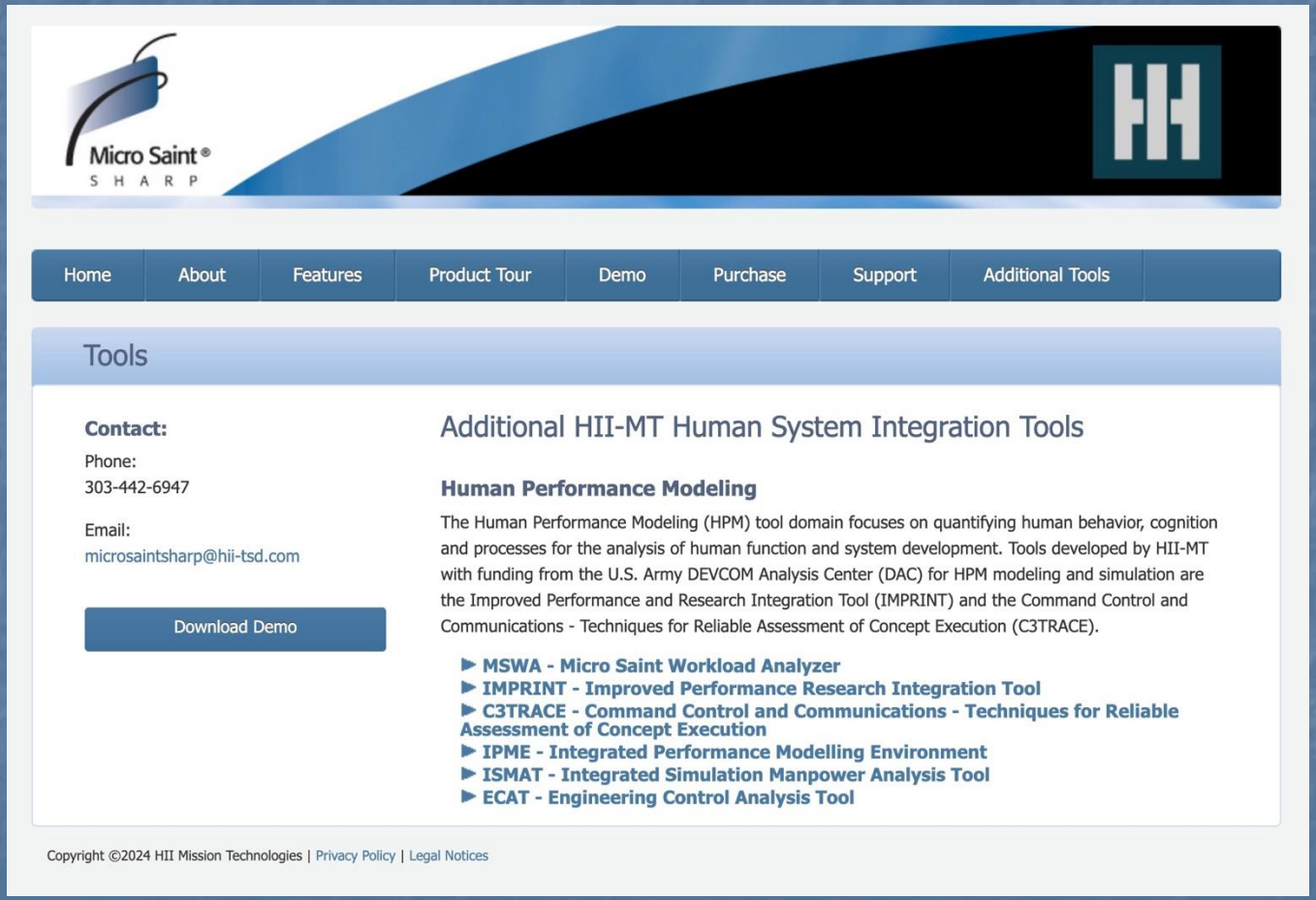
Task Network
(Project network, PERT, CPN, Mission Network, etc.)

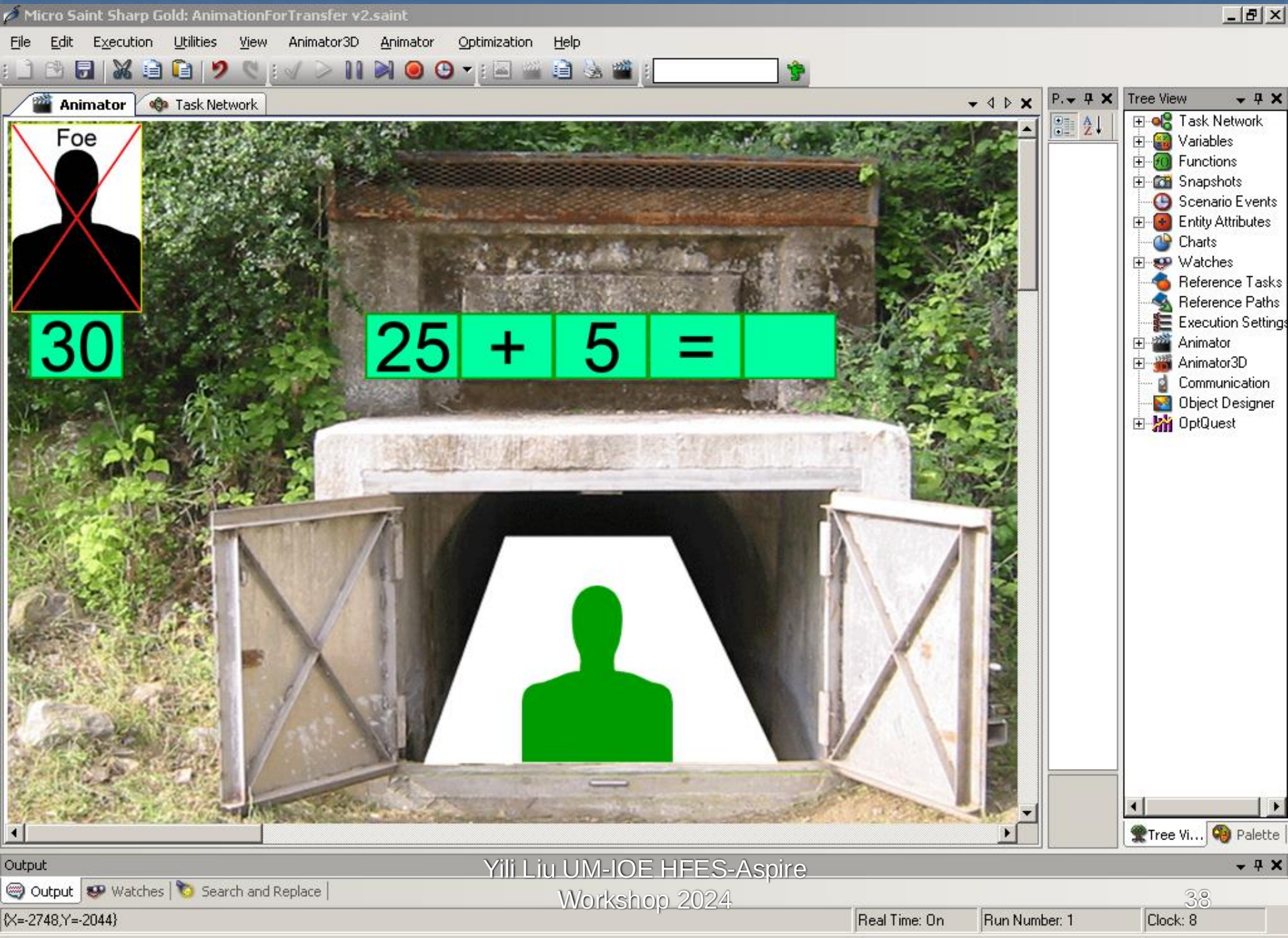


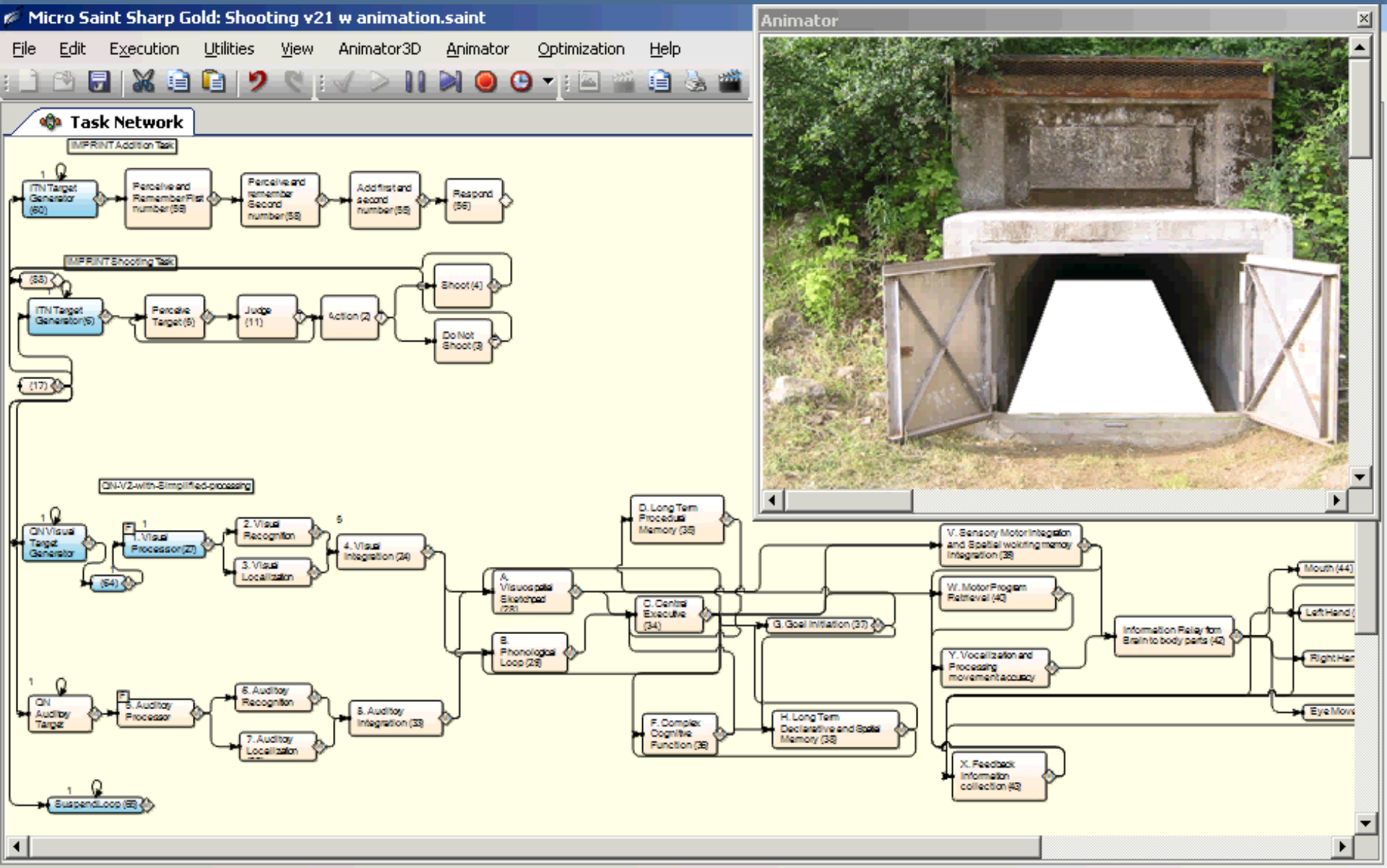
Relationship between QN and Task Network Modeling



Relationship between QN and Task Network Modeling




Micro Saint Sharp Gold: AnimationForTransfer v2.saint
File Edit Execution Utilities View Animator3D Animator Optimization Help
Animator
Task Network
Foe
30
25 + 5 =
Tree View
Task Network
Variables
Functions
Snapshots
Scenario Events
Entity Attributes
Charts
Watches
Reference Tasks
Reference Paths
Execution Settings
Animator
Animator3D
Communication
Object Designer
OptQuest
Tree Vi...
Palette
Output
Output
Watches
Search and Replace
{X=-2748,Y=-2044}
Real Time: On
Run Number: 1
Clock: 8


Micro Saint Sharp Gold: Shooting v21 w animation.saint
File
Edit
Execution
Utilities
View
Animator3D
Animator
Optimization
Help
Task Network
IMPRINT Addition Task
ITN Target Generator (60)
Perceive and Remember First number (58)
Perceive and remember Second number (58)
Add first and second number (58)
Respond (56)
IMPRINT Shooting Task
(88)
ITN Target Generator (5)
Perceive Target (5)
Judge (11)
Action (2)
Shoot (4)
Do Not Shoot (3)
(17)
ON-V2-with-Simplified-processing
ON Visual Target Generator
1. Visual Processor (27)
2. Visual Recognition
3. Visual Localization
4. Visual Integration (24)
(84)
ON Auditory Target
5. Auditory Processor
6. Auditory Recognition
7. Auditory Localization
8. Auditory Integration (33)
SuspendLoop (88)
A. Visuospatial Sketchpad (28)
B. Phonological Loop (29)
C. Central Executive (34)
D. Long Term Procedural Memory (35)
F. Complex Cognitive Function (36)
G. Goal Initiation (37)
H. Long Term Declarative and Spatial Memory (38)
V. Sensory Motor Integration and Spatial working memory Integration (39)
W. Motor Program Retrieval (40)
Y. Vocalization and Processing movement accuracy
X. Feedback Information collection (43)
Information Relay fom Brain to body parts (42)
Mouth (44)
Left Hand
Right Hand
Eye Move
Animator

An Integrated Cognitive Architecture for Cognitive Engineering Applications

Shi Cao and Yili Liu
University of Michigan
Ann Arbor, Michigan 48109 USA

The increasing complexity of computational cognitive architectures may increase both their modeling capabilities and their difficulty to learn and use as cognitive engineering tools. This paper reports our work dedicated to enhance the usability and the cognitive engineering applicability of a complex computational cognitive architecture called QN-ACTR, which integrates two complementary architectures Queueing Network and Adaptive Control of Thought-Rational. The aim is to provide an easy-to-use interface and intuitive modeling that support both inexperienced and experienced users in using this complex and powerful architecture. The process of model development is greatly simplified with improved visualization and validation methods. The results were examined using heuristic evaluation. The benefits and practice implications are discussed.

INTRODUCTION

Cognitive models can be used to support cognitive engineering. Compared with other forms of cognitive models such as verbal frameworks and pure mathematical models, cognitive architectures are particularly useful for complex cognitive engineering applications, because they unify a wide range of cognitive theories (Newell, 1990) and can computationally simulate human-machine interactions (Byrne & Pew, 2009; Schunn & Gray, 2002). For example, Adaptive Control of Thought-Rational (ACT-R, Anderson et al., 2004), a cognitive architecture that has incorporated many of the theoretical advances of cognitive science over the past decades, has been applied to cognitive engineering analyses of human-machine interactions including airport runway navigation, driving performance, and human-computer interactions (for a review, see Gray, 2008).

In the recent years, cognitive architectures are becoming increasingly integrated and complex in terms of having more components and interactions between components, requiring the use of knowledge description languages, and involving a large number of parameters. This complexity may increase both modeling capabilities and the difficulty to learn and use them as cognitive engineering tools. For example, building useful models in ACT-R requires a considerable amount of training and practice. The basic theory and syntaxes of ACT-R can be learned by reading a seven-unit tutorial and practicing with examples, which are often covered in a seven-day short course. The model description of displays and controls is written in the Lisp language, and therefore a modeler must also gain reasonable Lisp programming skills. Adjusting model parameters could also be very difficult, because the effect of changing a parameter may be buried deep in the text-based output traces. Currently, most users of cognitive architectures are expert researchers of cognitive modeling. The usage among cognitive engineers in the industry is very limited.

To emphasize the need for the usability development of cognitive architectures for cognitive engineering, Pew (2008) pointed out three challenges for researchers in this field, including the needs for (1) simplified model development, (2) better capabilities for articulating and visualizing how the models work, and (3) model validation.

Recently, several efforts have been made to address these challenges. G2A (Amant, Freed, & Ritter, 2005) and ACT-Simple (Salvucci & Lee, 2003) were developed to automatically translate GOMS (Goals, Operators, Methods, and Selection rules) style operators into ACT-R production rules. Incorporating ACT-Simple, CogTool (John, Prevas, Salvucci, & Koedinger, 2004) simplified the construction of human-computer interaction tasks, allowing the modeling of web browsing tasks by user demonstration with the mouse and the keyboard. Integrating ACT-Simple and an ACT-R driving model (Salvucci, 2006), Distract-R (Salvucci, 2009) simplified the construction of models for human interaction with in-vehicle devices in driving scenarios. Using Visual Basic Application in Excel, a click-and-select user interface has been developed in Queueing Network-Model Human Processor (QN-MHP, Wu & Liu, 2008). It allows users to build QN-MHP models without learning any simulation language. Usability tests showed that this click-and-select interface can save time and reduce errors in model development (Wu & Liu, 2009). In addition, easy-to-use user interfaces have also been developed in E-GOMS (Gil, 2010) and SANLab-CM (Patton & Gray, 2010).

The previous efforts have simplified model development, reducing or eliminating the need for learning a modeling language. However, each of them still has limitation in one or more of the following aspects.

- The simplification of modeling work comes at the cost of limiting the task displays and controls that can be model to a limited set of tasks.
- The flexibility to construct customized models with customized parameters is limited, for example, not being able to define the road curvature of each road segment as in a human driving experiment.
- Human information processing that can be modeled is limited to procedural and perceptual-motor processes, lacking the capabilities to model complex cognitive tasks such as learning, decision making, and sentence comprehension.

writing texts in tables and selecting options from menus. Users start from selecting the single or dual task scenario or loading demonstrations (Figure 3). Previous research models are included as demos (samples), such as ACT-R Tutorial models, driving models, and a transcription typing and reading dual-task model. New models can be made by modifying existing demos.

When a single task is selected with a template such as the Day-Block-Trial template for discrete-event experiments or the World3D template for driving, a task setup window will appear, asking users for the information needed in the experiment setup. When a dual-task is selected, two windows will appear each defining a task. For example, the Day-Block-Trial template in MSA asks for configuration settings such as whether a display stage in a trial will be terminated when all responses are detected (Figure 4) and setup details such as the number of trials in a block and the type of a display item (e.g., text, line, button, and tone) (Figure 5). The World3D template defining a driving task asks for road and car details such as lane width, road type, road length, and other cars' speed (Figure 6).

Figure 3. Screenshot of selecting a task using Model Setup Assistant.

Figure 4. Screenshot of selecting task configuration options using Model Setup Assistant.

Figure 5. Screenshot of defining a discrete task using Model Setup Assistant.

Figure 6. Screenshot of defining a driving task using Model Setup Assistant.

After defining the task, MSA can also assist users to define the mind and the parameter parts of a model. Since these syntaxes in QN-ACTR are the same as in ACT-R, if existing ACT-R codes are available, users can simply copy the ACT-R codes and paste into a QN-ACTR model. If no existing code available, users can define the mind, including chunks (Figure 7a) and production rules (Figure 7b), and set parameters (Figure 7c) with the assistance of MSA, by filling in tables and selecting from lists without the need to learn the knowledge description language used in ACT-R.

The model generated by MSA is also written in syntaxes. The resulted syntaxes can be saved, edited, or directly used to run the model. Simple modification of a model such as changing a few parameters can be easily achieved by directly editing the syntax file.

Visualization of 3D dynamic tasks

Previous work has developed the visualization of mental information processing, discrete experiment displays and controls, and the multi-dimensional mental workload (Cao & Liu, 2011a, 2011b). A new feature added to the visualization capabilities of QN-ACTR in this study is visualizing 3D dynamic tasks.

Using Animator3D in Micro Saint® Sharp, 3D dynamic tasks such as driving in single or dual task scenarios can be visualized in real time while the model is performing the task, which allows intuitive observation of model performance. The system refresh rate can be set by the user (10 ms by default). System dynamics such as speed, steering angle, and lateral deviation are visualized and recorded. Figure 8 illustrates that the model is driving a car while performing an arithmetic addition task. The model is following the car in the right lane and is visually focusing on the car. At the same time, the model is speaking "three" in response to the question of "1 + 2," which is displayed through the auditory channel and visualized on the right hand side of the figure.

Figure 7. Screenshot of defining the mind and the parameter parts of a model using Model Setup Assistant, including (a) chunks, (b) production rules, and (c) parameters.

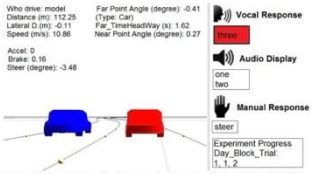


Figure 8. Visualization of a driving and arithmetic addition dual-task in QN-ACTR.

Integrated human experiment interface

The same task interface with which the model interacts can also serve as the interface for human participants to complete the same tasks. We have developed a human driving interface in QN-ACTR that supports simulated driving experiments with steering wheels and pedals. This feature allows the model and the human to perform and be compared in the same tasks with identical interfaces, with no need to replicate the real world experiment system in the modeling. Using the same experiment platform avoids any discrepancy between human and model tests due to the experiment setup.

FINDINGS

The usability development of QN-ACTR is evaluated using Nielsen's ten heuristics for user interface design (1994).

Visibility of system status. MSA always shows the stage of model development at the top-left corner. The visualization of the mind and the task keeps users informed about what is going on in the model during the simulation. Buttons in MSA are dimmed and disabled when their actions cannot be performed in some cases. Program responses and feedbacks are immediate with no delay.

Match between system and the real world. All the column headers in MSA tables and the items in menus use self-explanatory phrases without abbreviation. The steps of modeling in MSA follow the logical order shown in Figure 2. Full names and detailed descriptions are shown for each abbreviated ACT-R parameter name (Figure 7c).

User control and freedom. MSA supports undo (e.g., change the road name, delete a chunk, and reset a table) and redo (e.g., go back to the previous stage, and then go next again). A cancel button is provided at each stage to exit the setup at any time, and then users can restart MSA if needed.

Consistency and standards. Definitions and names are used consistently throughout all modeling steps. Tables and menus follow similar layouts and styles. Button position is the same between templates and stages.

Error prevention. The use of menu selection in MSA tables prevents the input of invalid items. Table cells automatically perform validation check, and users are notified when an input is of an invalid type or out of the valid range. Duplicated names assigned by users (e.g., chunk names) are automatically revised to prevent run-time errors. Syntax errors are also reported before the simulation starts.

Recognition rather than recall. MSA provides menus for users to select their options and tables to fill in. Model developing knowledge is provided to users in the interface. For example, users do not have to learn any modeling syntax. Instead, they can describe the model in natural language and fill in blanks or select items (Figure 7a, b). The default value, valid range, and description of model parameters are displayed for the users (Figure 7c).

Flexibility and efficiency of use. The syntax method and MSA cater to both inexperienced and experienced users. Experienced user can speed up the modeling work by directly copying and editing syntaxes. Syntaxes for the mind and the parameters can also be directly copied from ACT-R codes.

Aesthetic and minimalist design. MSA tables and menus are organized and aligned in groups. Introductions and explanations are concise.

Help users recognize, diagnose, and recover from errors. Error messages are expressed in plain language (no codes) and precisely indicate the problem. For example, "Error! Set General Parameters needs para_name: :lf to be a double rather than: nil."

Harper, 2007). This scenario used the same DISALT shooting testbed and provided behavioural data to model the shooting task. Production rules were defined to model the shooting-only task as the process of visual searching, manual aiming, and pulling the trigger.

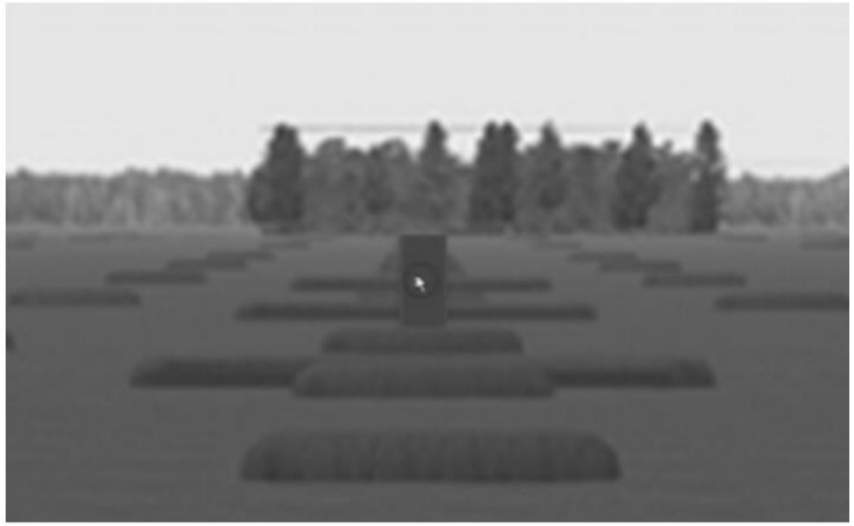


Figure 4 Visualization of the task environment with an enemy-target in QN-ACTR. The cursor represents the iron sight aiming. Background picture from (Scribner et al., 2007).

Relationship between QN and Task Network Modeling

CHAPTER 30

Queuing and Network Models

Yili Liu

Abstract

Comprehensive and computational models of human performance have both scientific and practical importance to human-machine system design and human-centered computing. This chapter describes human performance models that are based on queuing and network theories. Queuing-based models and network-based models were initially developed as two separate schools of models, as summarized in the first part of the chapter. Recent work based on queuing networks not only integrates the two schools of models into a unified framework but also allows integration of several other schools of approaches such as symbolic models, as described in the second part of this chapter.

Key Words: queuing models, network models, queuing network models, cognitive modeling, human-machine interaction models, systems modeling

Introduction

The increasing complexity of advanced human-machine systems makes it necessary for system designers to consider human capabilities and limitations as early as possible in system design. In order to reduce risks associated with poor task design with appropriate tools and methods for task analysis and function allocation, it is important to develop models of human performance and human-system interaction that are comprehensive, computational, science-driven, and application-relevant.

Models of human performance and human-system interaction should be comprehensive to capture the whole range of concurrent perceptual, cognitive, motor, and communication activities of human-system performance (also see Byrne, this handbook). These models should be computational and computerized to allow quantitative and rigorous simulation and analysis of design alternatives and scenarios. These models should be science-driven, with deep roots in and strong connections with cognitive science theories and principles. These models should also be application-relevant, striving to tackle and solve practical design problems, with an engineering philosophy that having an "imperfect" or approximate solution is better than no solution at all.

Human performance models for complex human-machine systems must also take into account the fact that operators in human-machine systems often need to perform a number of concurrent activities at once (see also Salvucci, this handbook). Examples of multitask situations abound and include an automobile driver who has to ensure the smooth operation of a vehicle while time-sharing between the instrument panel and the forward view of the roadway, and a traffic controller who has to divide attention between various visual and auditory sources of information while making time-critical decisions and performing intensive communications activities.

Many computational models have been proposed to model multitask performance and address the nature and the cause of task interference, in

Relationship between QN and Task Network Modeling

addition to various conceptual theories and qualitative models. This chapter focuses on computational models of multitask performance that are based on queuing and network theories, which represent some of the most prominent approaches in computational multitask modeling.

The specific contribution of this chapter to the challenges of engineered or technological systems is its emphasis and demonstration of the importance and value of queuing and network models in human-machine system design. The chapter first summarizes queuing-based models and network-based models, which were initially developed as two separate schools of models, and then describes recent work based on queuing networks that not only integrates the two schools of models into a unified framework but also allows integration of several other schools of approaches such as symbolic models.

Single-Server Queuing Models

Historically, computational modeling of multitask performance started with the school of computational modeling that we call single-server queuing models. This school of models encapsulates computationally a prominent conceptual theory of multitask performance called the single-channel theory of selective attention. Its roots can be traced to the single-channel theory of human information processing originally proposed by Craik (1947), which assumes that the human information processing system has bottlenecks that can process only one stimulus or piece of information at a time, and thus the system functions through a series of selections about which stimulus or piece of information to process (see also Broadbent, 1958; Deutsch & Deutsch, 1963; Welford, 1967).

The single-channel psychological theory of selective attention has been the fundamental basis of numerous engineering models of human performance (Carbonell, 1966; Rouse, 1980; Senders, 1964; Senders & Posner, 1976; Schmidt, 1978). These engineering models postulate that the human is a single-channel processor or a time-shared computer with a single central processing unit (CPU), which quickly switches and allocates its processing capacity among a variety of tasks in a sequential and all-or-none fashion. The models view human multitask performance as a single-server queuing problem or multitask sequencing problem in which multiple tasks or diverse sources of information are queued for service from the single-server human information processing system. For a comprehensive and detailed review of this school of models, see Liu (1997).

Task Network Models

Another school of engineering models of human performance that has had a long history and wide range of successful applications is the task network models. Starting with the systems analysis of integrated networks of tasks (SAINT) modeling methodology developed by Siegal and Wolf (1969), the task network approach models the human interaction with the environment as a sequence of tasks (also called paths) and acknowledges the existence of alternative paths to accomplish a goal or different goals in certain circumstances. These alternative paths form a task network. Parallel paths in a task network represent alternatives rather than concurrence of processing.

The modeling methodology of SAINT has been substantially and significantly extended into a family of network-based models, prominent among them includes Micro Saint Sharp, which is a general-purpose, discrete event simulation software tool that has been used successfully in many areas including human factors and the military, manufacturing, and service sectors (Laughery, 1989). Micro Saint was also used as the platform to develop the flagship task network modeling tool of the Army Research Lab (ARL) called IMPRINT (Improved Performance Research Integration Tool). IMPRINT is arguably the most powerful of the Army's Human System Integration (HIS) tools developed over the past two decades. It is a Windows-based, dynamic, stochastic, discrete event-modeling framework. When certain assumptions hold—that is, when the system of interest can be adequately described by task activities and networked sequencing, when dynamic processes and random variability are of interest, and when any continuous tasks can be fairly transformed into discrete tasks—then IMPRINT is an appropriate tool to use to represent and analyze soldier-system performance. As a system design and acquisition support tool, IMPRINT can be used to help set realistic system requirements, to identify soldier-driven constraints on system design, and to evaluate the capability of available manpower and personnel to effectively operate and maintain a system under environmental stressors. IMPRINT is also used to target human performance concerns in system acquisition, to estimate user-centered requirements early, and to make those estimates count in the decision-making process (Hawley, Lockett, & Allender, 2005).

Relationship between QN and Task Network Modeling

In spite of their differences, one of the common features of these models is their reliance on the fundamental assumption that humans can process only one piece of information at a time. Human multitask performance is modeled as a process of selecting tasks for sequential action according to some service discipline or cost function, which is usually based on the assumption that there is a mental cost to switching attention and/or there is a cost of being unable to attend to a critical instrument in a timely fashion (Rouse, 1980; Sheridan, 1972). Another common characteristic of these models is their focus on time as the underlying dimension and the metric of processing. Time is what is competed for by multiple tasks in a serial fashion, and completion time defines the difficulty or demand of each task or task component. The models are relatively silent as to the intensity aspects of task demand.

Recent work based on queuing networks not only integrates the two schools of models into a unified framework but also allows integration of several other schools of approaches, such as symbolic models (e.g., Anderson et al., 2004) and multiple resources theories (Wickens, 1984; Wickens & Liu, 1988), as described in the second part of this chapter.

Queuing Network Models

As described above, in order to support human-machine system analysis and design, it is important to develop models of human performance and human-system interaction that are comprehensive, computational, science-driven, and application-relevant. Along this line of research, a queuing network (QN)-based unified theory and computational architecture of human performance and human-system interaction has been developed that simultaneously meets the criteria listed above: comprehensive, computational, science-driven, and application-relevant. As reflected in the name "queuing networks," this approach explicitly considers both the queuing and the network aspects of human performance and modeling. Several major steps have been taken along this direction, each producing significant results and generating unique insights on human performance modeling in general and the role of queuing networks in human performance modeling in particular.

The following sections of this chapter first summarize accomplishments along this line of research and then discuss the next steps of research. The queuing network approach to human performance modeling was developed in several steps, starting from basic psychological functions and fundamental psychological issues and then moving to more complex tasks in human-machine systems. More specifically, since reaction time (RT) is arguably regarded as the most important and most widely used human performance measurement, the first step of the QN work was the successful development of a QN theory of reaction time (RT) that integrates the influential psychological architectural RT models as special cases, including the serial discrete-stages, the serial continuous-flow, and the discrete network models (such as the critical path network model). Further, the QN models cover a broader range of mental architectures and can be subjected to well-defined empirical tests. In the second step, the focus was on the relationship between RT and response accuracy, whose tradeoff is one of the fundamental characteristics of human performance. In this step of QN work, the architectural RT models and the sequential information sampling RT/accuracy models are unified through QN-RMD (Reflected Multidimensional Diffusions). Specifically, the "state" of a K-server QN of mental architecture is represented as a reflected diffusion space of K dimensions, in which "reflecting barriers" reveal architectural constraints, while "absorbing barriers" represent accuracy-related response criteria. QN-RMD moves beyond the current one-dimensional random walk/diffusion/accumulator models that have successfully accounted for but are limited to single-stage fast responses.

In the third step, QN-MHP (Model Human Processor) was developed to bridge the mathematical and the symbolic models of mental architecture and to support mathematical modeling and real-time generation of multitask performance and mental workload. QN-MHP expands the three discrete serial stages of perceptual, cognitive, and motor processing in MHP into three continuous-transmission subnetworks of servers, each performing distinct psychological functions specified with a procedural/symbolic language. Multitask performance and workload emerges as the network behavior of multiple streams of information flowing through a network. QN-MHP has been applied to generate and model a variety of tasks including the psychological refractory period, visual search, transcription typing, and driving a vehicle simulator.

QN Architecture of RT: Integrating Architectural and Information Transmission Models

Historically, the first groups of computational models that examine human performance and the

Relationship between QN and Task Network Modeling

Queueing Network Modeling of Human Performance of Concurrent Spatial and Verbal Tasks

Yili Liu, *Member, IEEE*

Abstract— **This article describes a three-node queueing network model of human multitask performance to account for interferences between concurrent spatial and verbal tasks. The model integrates considerations of single-channel queuing theoretic models of selective attention and parallel processing, multiple-resource models of divided attention, and provides a computational framework for modeling both the serial processing and the concurrent execution aspects of human multitask performance. The single-channel and the multiple-resource concepts and their applications in engineering models are reviewed. Experimental evidence in support of the queueing network model is summarized. The potential value of using queueing network methods to integrate currently isolated concepts of human multitask performance and in modeling human machine interaction in general is discussed.**

I. Introduction

ONE OF THE common characteristics of an operator's task in human–machine systems is the need to perform a number of concurrent activities at once. Examples of multitask situations abound and include an automobile driver who has to ensure the smooth operation of a vehicle while time-sharing between the instrument panel and the forward view of the roadway, and a traffic controller who has to divide attention between various visual and auditory sources of information, while making time-critical decisions and performing intensive communications activities. The requirements for processing multiple sources of information often push the operators in multitask human machine systems to the upper bound of their attention capabilities.

Fortunately, the increasing power and sophistication of hardware and software technology are providing additional options for human–machine system design that could take into account the characteristics of the human operator. In this regard, as indicated by a recent report of the Committee on Human Factors, National Research Council, comprehensive engineering-based predictive models of operator performance and workload in complex multitask systems become increasingly important [3]. These models can help assess the impact of the technological infusions and determine the most effective design before a system is configured, and will allow the human factors professionals to communicate their knowledge to the engineering community more effectively in a language that is compatible with the designers' existing terminology and conceptual base.

Many predictive models of multitask performance have been proposed to address the nature and the cause of task interference. Prominent among these models are the single-channel serial processing models and the multiple-resource parallel processing models. The two schools of models have fundamental differences in their views of the nature of multitask performance and in their research and modeling methodology. The single-channel serial processing models treat multitask performance as an issue of task selection and scheduling: human information processing systems can only attend to one task at a time, and multitask performance relies on the rapid switching of attention among the tasks competing for attention [9], [36], [45]. The multiple-resource parallel processing models, in contrast, treat multitask performance as an issue of parallel allocation and division of processing resources among simultaneous tasks: multiple tasks can be processed at the same time as long as the total demand does not exceed the limit of attentional capacity or processing resources [59].

Until recently, there has been a substantial gap between the two schools of models. As models of human behavior, both schools of models have received substantial support from a multitude of experimental studies. But at the same time, it has become increasingly evident that neither school of models alone is sufficient in providing fully satisfactory explanations to the empirical data. From the perspective of engineering modeling, the single-channel assumptions have thus far enjoyed a greater success, as indicated by the existence of a set of well-established models such as the queueing theoretic models and the network models reviewed in the following section. These models provide formal mechanisms for representing and codifying the single-channel assumptions of task selection in engineering terms. The multiple-resource models, in contrast, have only recently started to see some of their concerns being gradually accommodated in several simulation models of human performance, and there is still a lack of a set of computational methods to represent the assumptions of simultaneous execution and resource allocation in engineering terms. Furthermore, there does not exist a set of integrated engineering-based methods to model the concerns of both schools of models and to bridge the gap between the two. As indicated in the recent National Research Council report, there is a lack of methods to model the two most important features of a macromodel: task selection and simultaneous execution [3].

Manuscript received December 11, 1994; revised May 11, 1996. An earlier version of this paper was presented at the 1994 IEEE International Conference on Systems, Man, and Cybernetics, San Antonio, TX, October 1994, and published in the conference proceedings.

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

Publisher Item Identifier S 1083-4427(97)00138-0.

Relationship between QN and Task Network Modeling

Recently, Liu [29], [30] proposed that queueing network models and related methods employed widely in industrial engineering and system performance analysis may provide us an integrated computational framework for modeling the complex structural and temporal arrangements that multiple tasks might assume. The structural arrangements include both serial selection and parallel execution, and the temporal arrangements include both immediate activities and delayed processing. The purpose of this article is to examine the gap between the single-channel and the multiple-resource models of multitask performance, to describe a three-node queueing network model of interference between concurrent spatial and verbal tasks, and to illustrate the potential power of the queueing network approach for modeling multitask performance. As described below, the queueing network model provides a computational framework to integrate the concerns of the single-channel and the multiple-resource models, which, in essence, can be treated as special cases of the queueing network model.

The structure of the article is as follows. The first two sections review the single-channel and the multiple-resource models in detail, in terms of their theoretical assumptions and their applications in engineering modeling. Both the successes and the limitations of the two modeling approaches will be discussed. Then, a specific pattern of task interference between concurrent spatial and verbal tasks will be discussed as an illustration of the gap between the two classes of models. A three-node queueing network model is then described as a plausible and intuitive account of interference between spatial and verbal tasks and as an attempt to bridge the gap between the single-channel and the multiple-resource models. The value and the potential power of using queueing network methods to integrate some other currently isolated concepts of human performance and in modeling human-computer systems in general is discussed at the end of the article.

II. Single-Channel, Queueing Theoretic Models of Multitask Performance

As mentioned above, a prominent theory of multitask performance is the single-channel theory of selective attention. Its roots can be traced to the single-channel theory of human information processing originally proposed by Craik [12] to explain the psychological refractory period in human information processing discovered by Telford [51]. Telford discovered that when two reaction time tasks are presented close together in time, the reaction time to the second task stimulus is consistently delayed from a single task control condition. Various forms of single-channel theories have been subsequently proposed and elaborated [6], [14], [35], [56]. What is consistent about these theories is their common assumption that the human information processing system has bottlenecks that can only process one stimulus or piece of information at a time, and thus the system functions through a series of selections about which stimulus or piece of information to process. The focus of investigation is the identification of the bottlenecks, and the topics of debate among these theories are their different opinions regarding the locus of the bottlenecks and the factors that influence the selection processes.

The single-channel psychological theory of selective attention has been the fundamental basis of numerous engineering models of human performance [9], [42], [45]. These engineering models postulate that the human is a single-channel processor or a time-shared computer with a single central processing unit (CPU), which quickly switches and allocates its processing capacity among a variety of tasks in a sequential and all-or-none fashion. The models view human multitask performance as a single server queueing problem or multitask sequencing problem in which multiple tasks or diverse sources of information are queued for service from the single-server human information processing system.

Early models in this tradition have focused on modeling human visual sampling and monitoring behavior. Senders developed an instrument monitoring model, which integrated the single-channel concept and the sampling theorem of Shannon's information theory in making its predictions about the observer's fractional dwell time on each monitored instrument [45]. Carbonell proposed a single server priority queueing model of multi-instrument visual sampling [9]. The priority of each instrument at any instant is modeled as the combined effect of both the probability and the cost of exceeding a prescribed limit. The model integrates concepts from queueing theory, information theory, and decision theory. Carbonell used simulation to solve the model, and showed a close fit between the model's predictions and the subjects' actual performance in flying a simulator in terms of the fraction of attention devoted to each instrument. Senders and Posner further developed the queuing theoretic approach to instrument monitoring and provided analytical solutions to a model that they developed for display sampling [46]. Schmidt applied queueing theoretic method to the analysis of an air traffic control task [43].

A systematic and extensive effort in applying the queuing theoretic methods to the modeling of human machine systems can be found in a series of studies conducted by Rouse and his colleagues. Rouse described human–computer interaction as a queueing system with the human and the computer as two servers [42]. He formulated a queueing theoretic model of dynamic allocation of responsibility between the human and the computer in multitask situations, and illustrated the potential utility of this model with simulation experiments. Chu and Rouse later investigated the predictive power of the model with a behavioral experiment that simulated a multitask flight management situation [10]. A similar task scenario was also used in an earlier study by Walden and Rouse that investigated the suitability of a single-server queueing model of pilot decision-making [55]. Greenstein and Rouse integrated a pattern recognition technique called discriminant analysis with queueing theory methods in their two-stage model of human decision making in multiprocess monitoring situations [19]. Discriminant analysis was used in the first stage to generate estimates of event occurrence probability, and queueing theory was then applied in the second stage to incorporate these probabilities into the solution of the attention allocation problem.

Relationship between QN and Task Network Modeling

The single-channel assumptions can also be found in several other engineering models. For example, Sheridan assumed that there is a mental cost to switching attention which will determine how often different information sources in the environment are sampled, and thereby influence the sampling behavior [47]. Kleinman and Curry developed a model for human operator display monitoring, which assumed that the human is a single-channel time-shared processing channel [26]. Tulga modeled the multitask attention allocation problem as a dynamic single-machine sequencing problem [53]. The concepts and assumptions of single-channel processing and sequencing were also employed in the semi-Markov dynamic decision model of human task selection performance proposed by Pattipati, Kleinman, and Ephrath [40].

The single-channel concepts have been the fundamental basis of numerous simulation models of human performance. Notable examples include the task network models PROCRU [4], [28] and HOS [57]. Starting with the systems analysis of integrated networks of tasks (SAINT) modeling methodology developed by Siegal and Wolf [48], the task network approach models the human interaction with the environment as a sequence of tasks (also called paths), and acknowledges the existence of alternative paths to accomplish a goal or different goals in certain circumstances. These alternative paths form a task network. Parallel paths in a task network represent alternatives rather than concurrence of processing. Furthermore, a task in a network cannot be started until the preceding task on the same path of the network has been completed. Thus, at any instant only one task on a path can be executed.

Procedure-oriented crew model (PROCRU) is a control theory-based simulation model that has received widespread recognition [4]. The model is a closed-loop system model incorporating submodels for the aircraft, aircraft crew members, and the air traffic controller. The crew member submodel is a detailed human performance model and has a comprehensive coverage of human activities in monitoring and control of a large system. The model assumes that the crew members have a set of procedures or tasks to perform, and one task is chosen at a given instant of time, which is the one perceived to have the highest expected gain for execution at that time. The model contains a procedure selector, which is responsible for task selection and sequencing.

The human operator simulator (HOS) uses a library of human performance micromodels to simulate the operator's perceptual, cognitive and motor responses [21], [50], [57]. The original versions of the HOS approach assumes that humans are single-channel processors, and that human behavior is goal-oriented and can be defined as a sequence of discrete micro-tasks, which can be aggregated to predict task performance. Other simulation models of complex task performance that are based on single-channel assumptions include STALL [11], SIMWAM [17], [33], the model developed by Tulga and Sheridan [54], and its subsequent modified versions (e.g., [37]).

In spite of their differences, one of the common features of these models is their reliance on the fundamental assumption that humans can only process one piece of information at a time. Human multitask performance is modeled as a process of selecting tasks for sequential action according to some service discipline or cost function, which is usually based on the assumption that there is a mental cost to switching attention and/or there is a cost of being unable to attend to a critical instrument in a timely fashion [39], [42], [47]. Another common characteristic of these models is their focus on time as the underlying dimension and the metric of processing. Time is what is competed for by multiple tasks in a serial fashion and completion time defines the difficulty or demand of each task or task component. The models are relatively silent as to the intensity aspects of task demand.

These single-channel based models have demonstrated tremendous success in modeling two aspects of human performance: visual sampling in process monitoring and strategic task scheduling in high workload situations. The primary concern of visual sampling is to find the optimal tactics for a single-channel sampler to sample sources of information sequentially when the information sources cannot be attended to at once and thus compete for the operator's focal attention [36]. For example, in monitoring instrument panels, owing to the need for foveal vision when accurate reading is required, the eyes must be pointed in an appropriate direction to scan and sample a source of information. Since the distances between instruments are most often greater than the radius of foveal vision (2° to 3° of visual angle), accurate reading of spatially separated instruments can only be done sequentially. Similarly, in high workload situations when time pressure is the major source of workload and the operators are free to choose the order in which the tasks should be done to avoid overload [1], [37], these single-channel models have been quite successful in capturing the strategic scheduling aspect of task performance.

However, intuition and experimental evidence both support the view that humans do have the ability to perform multiple tasks in a truly concurrent fashion under many real-world circumstances. "We must recognize that people in fact can do more than one thing at a time and normally do" [1, p. 5]. In these task situations, single-channel assumptions and the analogy of a single-CPU time-shared computer or a single-server queueing system may only capture part of the nature of human performance and may not be adequate to portray the complex cognitive mechanisms for concurrent processing. Furthermore, each of the concurrent tasks may have attentional demands varying in intensity as well as in time. Some task pairs may be more similar with each other in their task structure than with others, and thus produce different patterns of task interference when they are performed concurrently with each other than when they are with other tasks. Thus, it may be necessary to address the parallel processing aspect of performance, to consider the intensity as well as the time characteristics of task demand, and to analyze the structural aspects of concurrent tasks. These issues have been the focus of investigation of models of divided attention.

III. Multiple Resource, Parallel Processing Models of Multitask Performance

In contrast to the single-channel assumption adopted by selective attention theorists that attention capacity can only

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

1. RT: Reaction Time (QN-RT) and Mental Structure
2. RT and Accuracy (QN-RMD) (Mental Structure vs State of Mind)
3. Procedural Tasks (QN-MHP or QN-MHP-BE)
4. Complex Cognition Tasks (QN-ACTR)
5. Visual attention tasks (QN-NSEEV)
6. Manual or Continuous control tasks (QN-Control)
7. Basic Body Motion tasks (QN-MTM)
8. Mind-Body Interaction (QN-MBS)
9. Neural level (QN-Neural)
10. Nervous and Endocrine Systems (QN-NES)
11. Multi-Person Multi-Machine QN (QN-HMN)
12. Applications in HMI, HRI, HAI

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

0033-295X/96/$3.00

Queueing Network Modeling of Elementary Mental Processes

Yili Liu
University of Michigan

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

Why is there a delay between stimulus presentation and response initiation? This has been one of the most enduring and fundamental puzzles for psychologists. The current belief of cognitive psychologists is that this delay is a reflection of the dynamic activities of an underlying mental structure that transforms stimulus into response. Most important, because the cognitive system is not amenable to open inspection, the characteristics of this delay—also called *reaction time* (RT)—may offer important clues to the possible configurations of the mental structure.

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

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

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

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

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

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

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

Note. PERT = program evaluation and review technique.

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

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

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

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

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

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

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

ON THE SPEED OF MENTAL PROCESSES

F. C. DONDERS [1]

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

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

DONDERS INSTITUTE

Search

Donders Institute for Brain, Cognition and Behaviour

Students • Staff • Alumni

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

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

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

Biography F.C. Donders

F.C. Donders, the father of mental chronometry

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

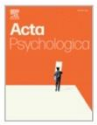
Donders was one of the pioneers of ophthalmology. His major contributions were in the areas of refraction and astigmatism. In 1858 Donders established the first eye hospital in the Netherlands. In 1864 his influential work "On the anomalies of accommodation and refraction of the eye with a preliminary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

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

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

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

Acta Psychologica
Volume 30, 1969, Pages 276-315



The discovery of processing stages: Extensions of Donders' method

Saul Sternberg [1]

Show more

+ Add to Mendeley Share Cite

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

Abstract

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

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

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

Richard Schweickert

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

Abstract

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

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

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

RICHARD SCHWEICKERT

The University of Michigan

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

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

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

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

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

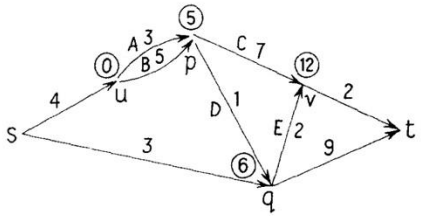


FIG. 2. A task network. Each arrow represents a process and associated with each arrow is a number giving the duration of the process. Circled numbers indicate the duration of the longest path from u to the point under the circled number.

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

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

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

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

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

Psychological Review

VOLUME 86 NUMBER 4 JULY 1979

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

James L. McClelland
University of California, San Diego

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

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

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

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

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

0033-295X/96/$3.00

Queueing Network Modeling of Elementary Mental Processes

Yili Liu
University of Michigan

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

Why is there a delay between stimulus presentation and response initiation? This has been one of the most enduring and fundamental puzzles for psychologists. The current belief of cognitive psychologists is that this delay is a reflection of the dynamic activities of an underlying mental structure that transforms stimulus into response. Most important, because the cognitive system is not amenable to open inspection, the characteristics of this delay—also called *reaction time* (RT)—may offer important clues to the possible configurations of the mental structure.

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

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

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

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

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

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

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

Note. PERT = program evaluation and review technique.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

RT as Network Sojourn Time

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

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

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

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

$$RT_d = T_1 + T_d. \quad (7)$$

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

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

$$RT_c = T_2 + T_c. \quad (8)$$

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

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

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

Series Queues as a Model for RT

Network Sojourn Time in Series Queues

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

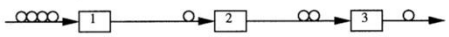


Figure 1. A series queueing network.

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

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

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

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

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

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

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

and the expected value has an upper bound expressed as

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

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

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

In the case of K identical servers, Equation 23 becomes

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

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

where

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

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

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

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

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

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

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

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

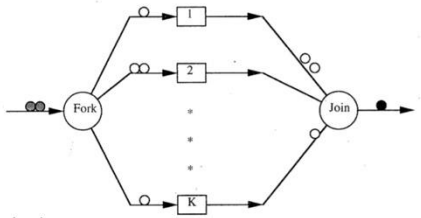


Figure 2. A parallel fork-join queueing network.

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

A Single-Server Feedback Queueing System

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

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

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

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

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

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

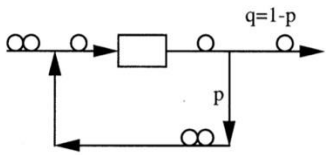


Figure 3. A queueing system with instantaneous Bernoulli feedback.

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

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

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

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

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

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

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

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

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

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

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

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

Simon–Foley Network and Overadditive Factor Interactions

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

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

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

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

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

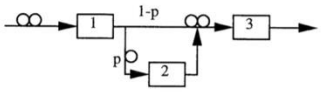


Figure 4. A Simon–Foley network in which noise components may overtake signal components.

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

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

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

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

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

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

Closed Queueing Networks and Underadditive Factor Interactions

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

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

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

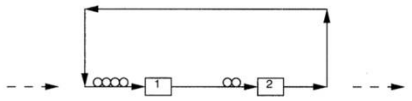


Figure 5. A cyclic queueing network that allows a fixed number of stimulus components to exist in the system.

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

1. RT: Reaction Time (QN-RT) and Mental Structure
2. RT and Accuracy (QN-RMD) (Mental Structure vs State of Mind)
3. Procedural Tasks (QN-MHP or QN-MHP-BE)
4. Complex Cognition Tasks (QN-ACTR)
5. Visual attention tasks (QN-NSEEV)
6. Manual or Continuous control tasks (QN-Control)
7. Basic Body Motion tasks (QN-MTM)
8. Mind-Body Interaction (QN-MBS)
9. Neural level (QN-Neural)
10. Nervous and Endocrine Systems (QN-NES)
11. Multi-Person Multi-Machine QN (QN-HMN)
12. Applications in HMI, HRI, HAI

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

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

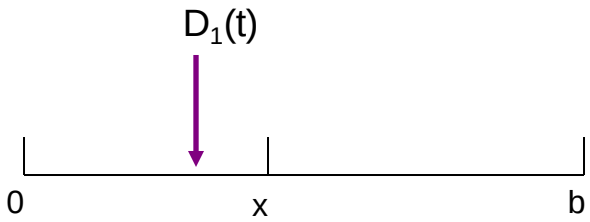
(from Liu, 1996, "Queueing network modeling of elementary mental processes," Psychological Review, 103(1), pp. 116-136).

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

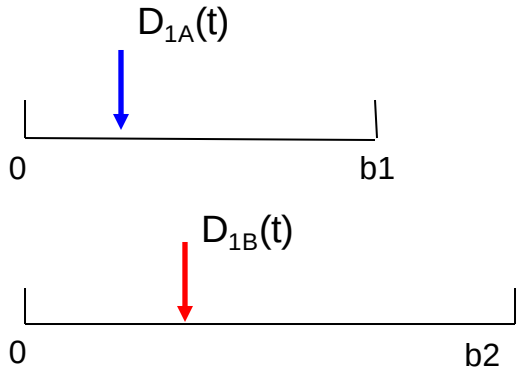
- Non-unidirectional flow
- Non-selective influence
- By-passing
- Fixed capacity

Counter/accumulator
Random-walk

Accumulator
Diffusion



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



2. Counter/Accumulator Model

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

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

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

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

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

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

Counter/accumulator
Random-walk

Accumulator
Diffusion

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

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

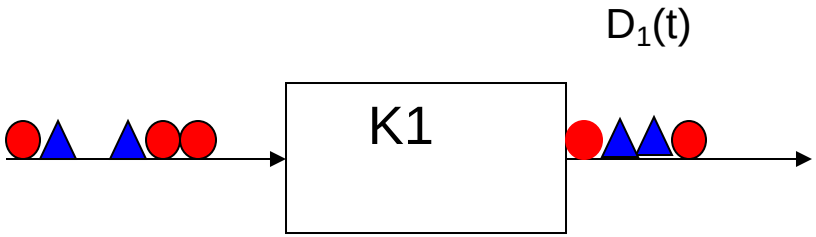
(from Liu, 1996, "Queueing network modeling of elementary mental processes," Psychological Review, 103(1), pp. 116-136).

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

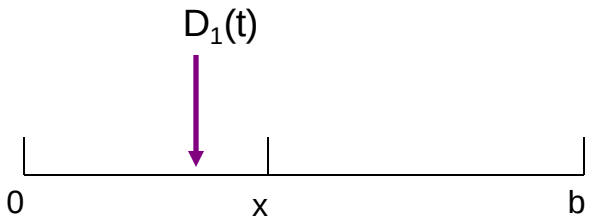
- **Non-unidirectional flow**
- **Non-selective influence**
- **By-passing**
- **Fixed capacity**

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

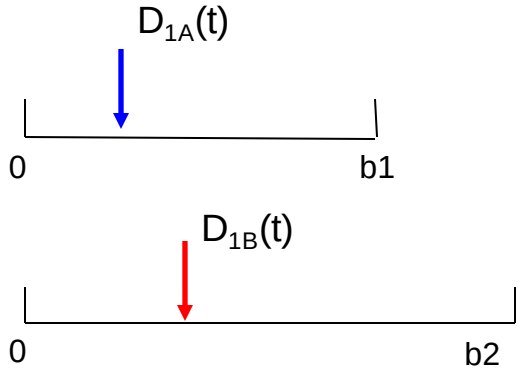
Reflected Multidimensional Diffusions



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

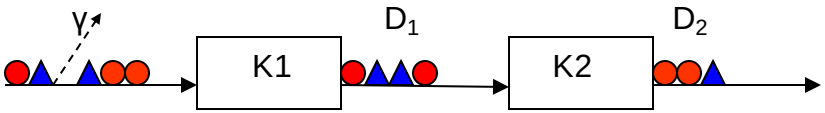


b). One 1-d diffusion with two absorbing barriers

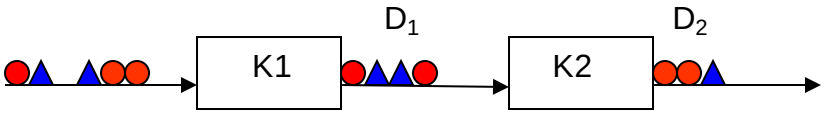


c). Two 1-d diffusions, each with one absorbing and one reflecting barrier

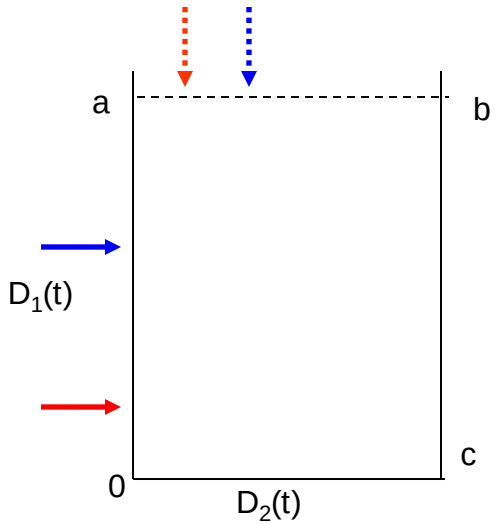
Figure 2: A single server queueing system and its alternative 1-d diffusion representations



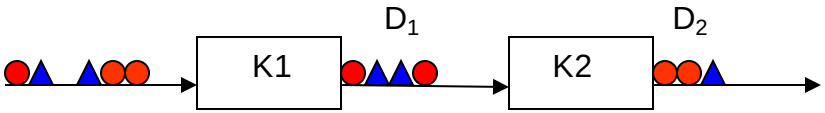
a). A tandem two-server system with two types of customers: type A (“triangles”) and type B (“circles”)



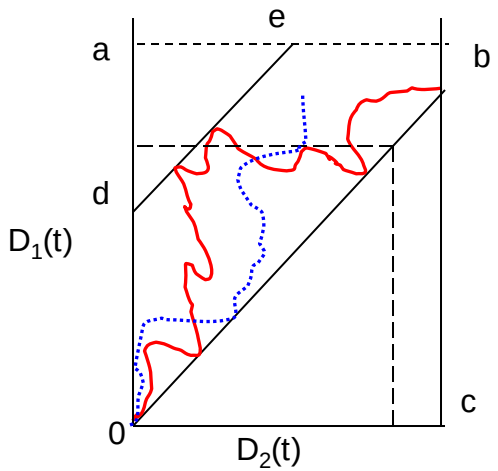
a). A tandem two-server system with two types of customers: type A ("triangles") and type B ("circles")



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



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

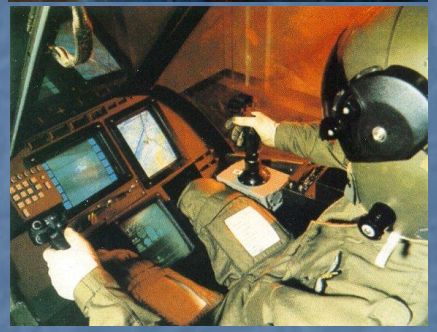


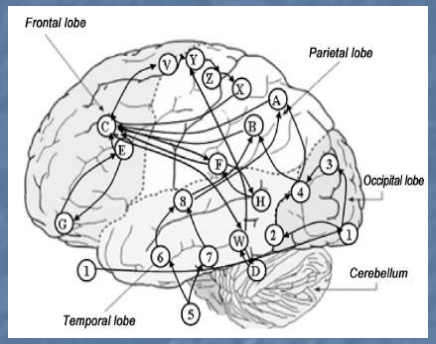
b). 2-D diffusion representation of {$D_1(t)$, $D_2(t)$} in a continuous tandem queue

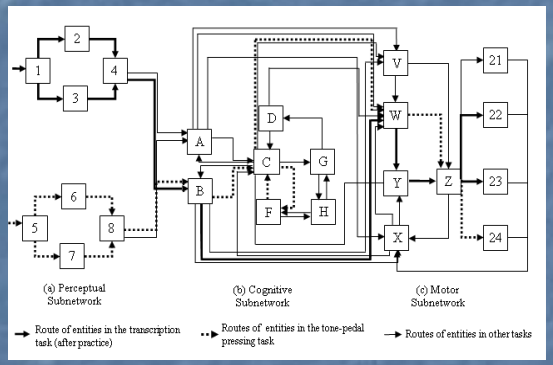
Queueing Network (QN)
Models of Human Behavior (HB)
QN-MHB

1. RT: Reaction Time (QN-RT)
2. RT and Accuracy (QN-RMD) (Mental Structure vs State of Mind)
3. Procedural Tasks (QN-MHP or QN-MHP-BE)
4. Complex Cognition Tasks (QN-ACTR)
5. Visual attention tasks (QN-NSEEV)
6. Manual or Continuous control tasks (QN-Control)
7. Basic Body Motion tasks (QN-MTM)
8. Mind-Body Interaction (QN-MBS)
9. Neural level (QN-Neural)
10. Nervous and Endocrine Systems (QN-NES)
11. Multi-Person Multi-Machine QN (QN-HMN)
12. Applications in HMI, HRI, HAI

Queueing Network Modeling of Cognitive Architecture and Human-Machine Systems







Queueing Network of Mental Architecture

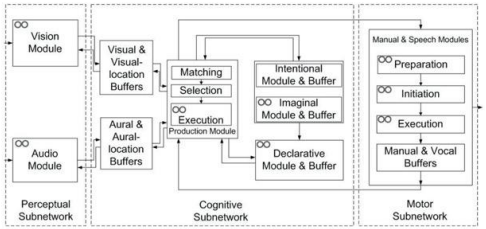


Figure 3. Server structure of QN-ACTR. Queue symbols (shown as two circles) mark the servers where queues are added from the QN's perspective. All the server processing logics in the QN-ACTR are identical to the corresponding algorithms in ACT-R (adapted from Cao & Liu, 2012c).

The Psychology of Human-Computer Interaction

Stuart K. Card
Thomas P. Moran
Xerox Palo Alto Research Center

Allen Newell
Carnegie-Mellon University

LEA
1983 LAWRENCE ERLBAUM ASSOCIATES, PUBLISHERS
Hillsdale, New Jersey London

2. The Human Information-Processor

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

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

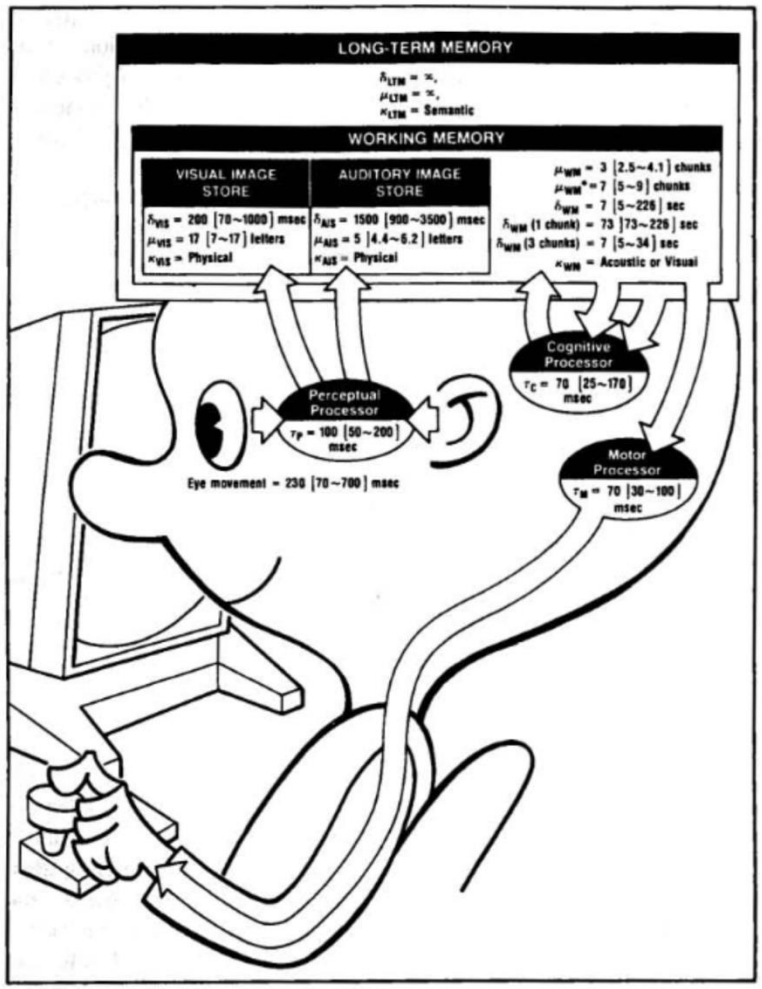


Figure 2.1. The Model Human Processor—memories and processors.

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

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

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

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

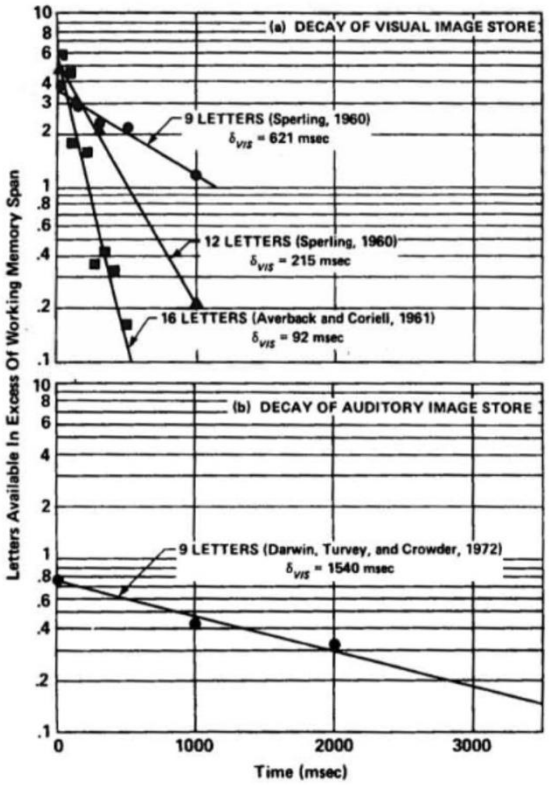
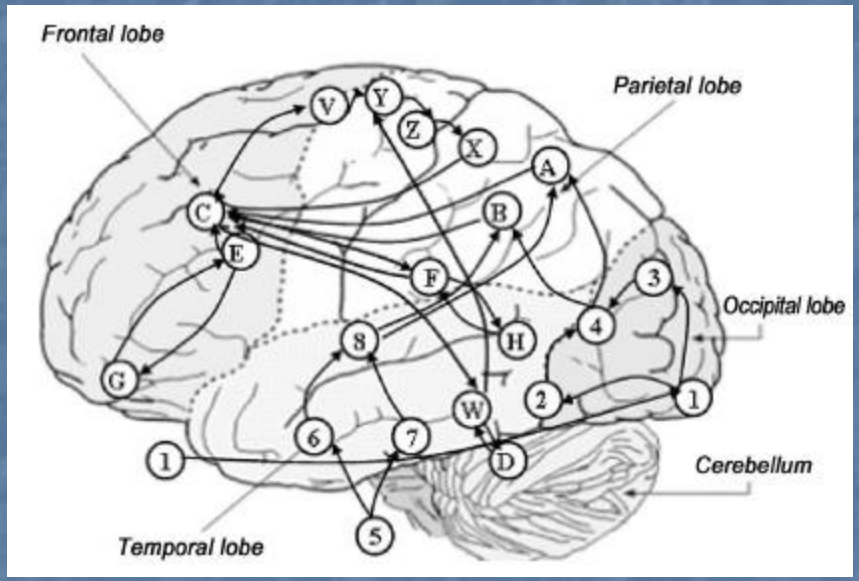
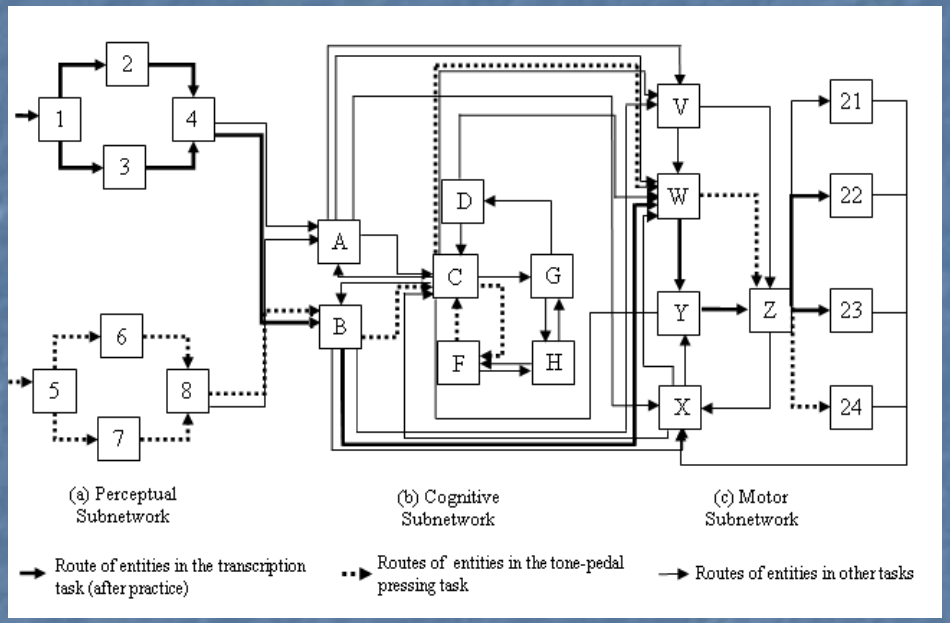


Figure 2.3. Time decay of Visual and Auditory Image Stores.
(a) Decay of the Visual Image Store. In each experiment, a matrix of letters was made observable tachistoscopically for 50 msec. In the case of the Sperling experiments, a tone sounded after the offset of the letters to indicate which row should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





Human Brain

Queueing Network of Mental Architecture

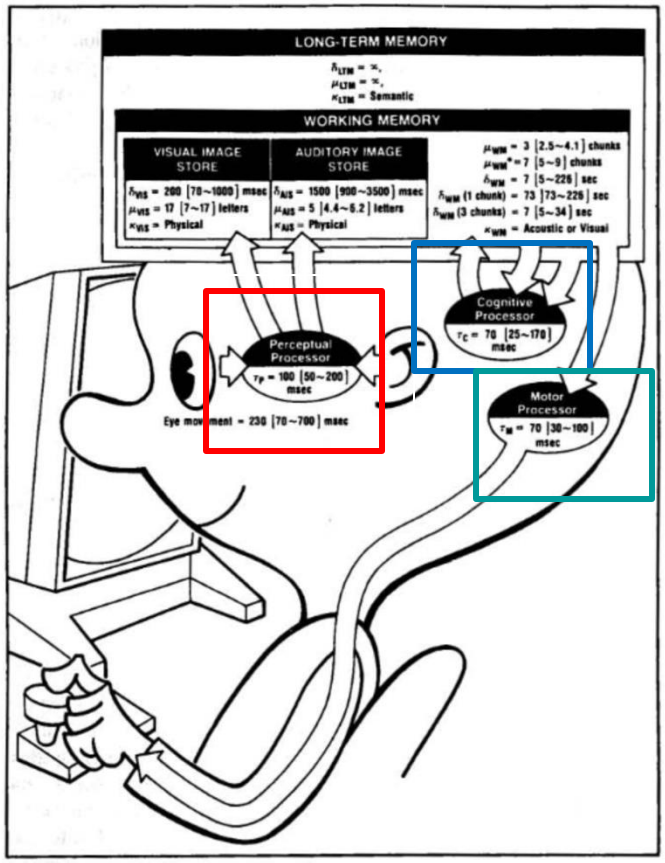
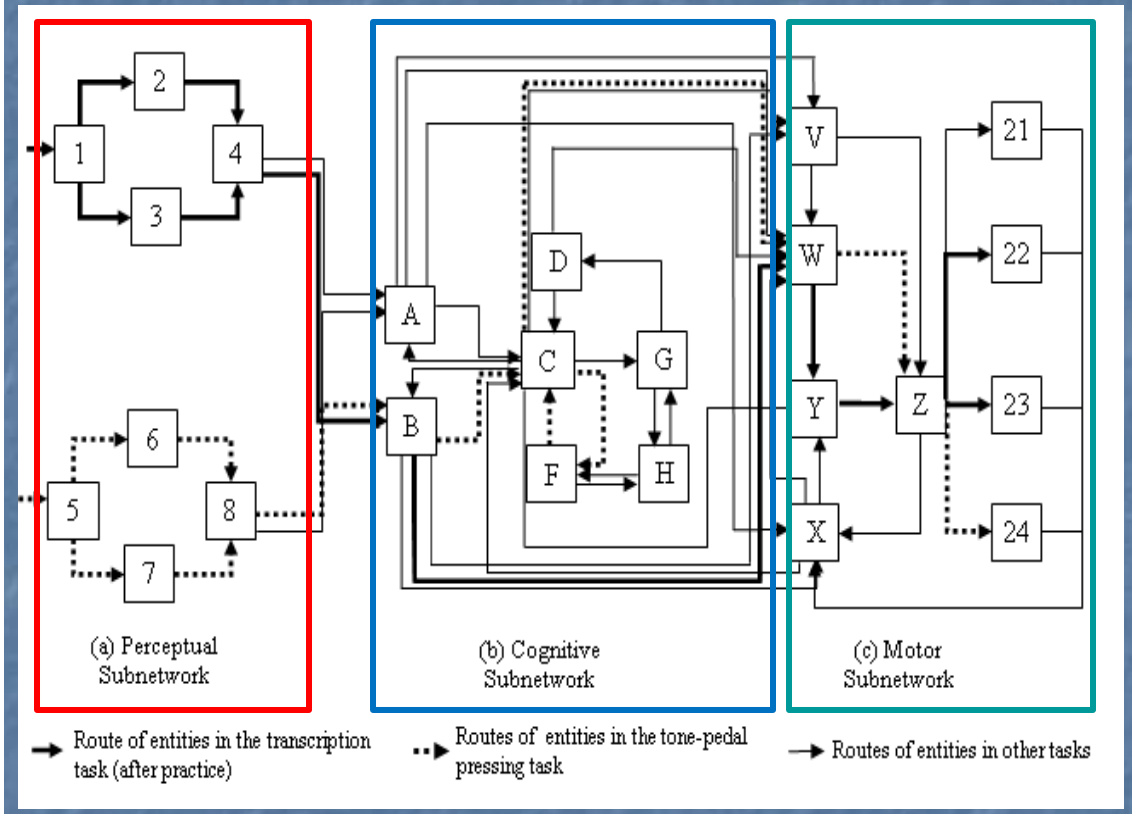


Figure 2.1. The Model Human Processor—memories and processors.

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



Queueing Network - MHP

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

In Chemistry, we have:

- Elements
- Compounds
- Mixtures

- Chemical structure (high-school)

In QN-MHP, we have:

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

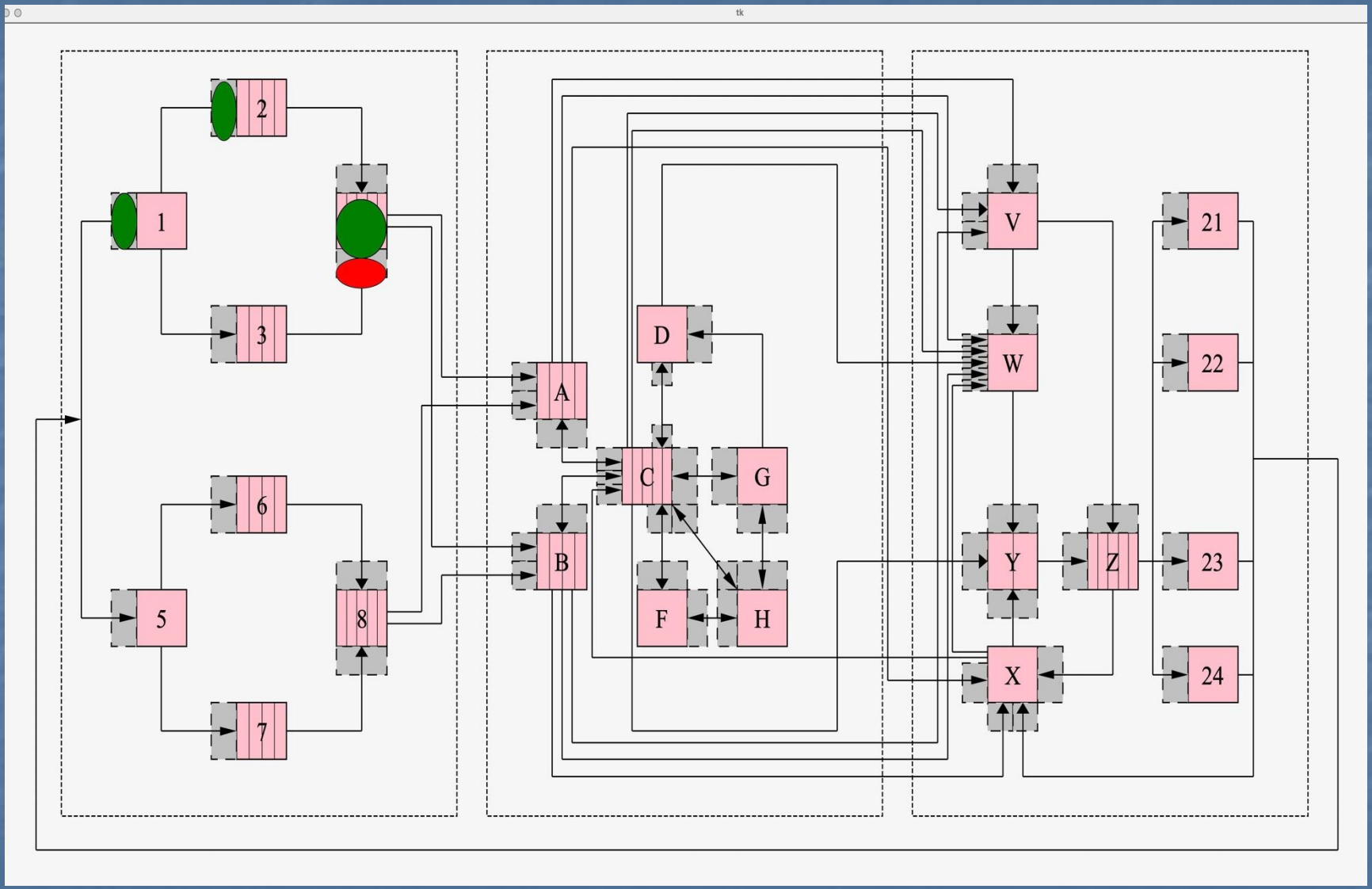
- Queueing Network Structure

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

Examples:

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

QN-MHP-BE or QN-MHP



Behavior Compounds: Compound_BE

Examples:

1. Look_for

A Compound of Basic BEs:

Look_at

See

Judge_identity

Store_to_WM (or update_WM)

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

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

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

Behavior Compounds: Compound_BE

Examples:

Scan_SEEV

A Compound of Basic BEs:

Look_at

Fixate

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

Or (more advanced/complex versions)

A Compound of Basic BEs:

Look_at

Fixate

[Retrieve from LTM

update_WM] periodically

Behavior Compounds: Compound_BE

Examples:

1. Tracking_1D
2. Tracking_2D or 3D
3. Tracing_1D
4. Tracing_2D or 3D

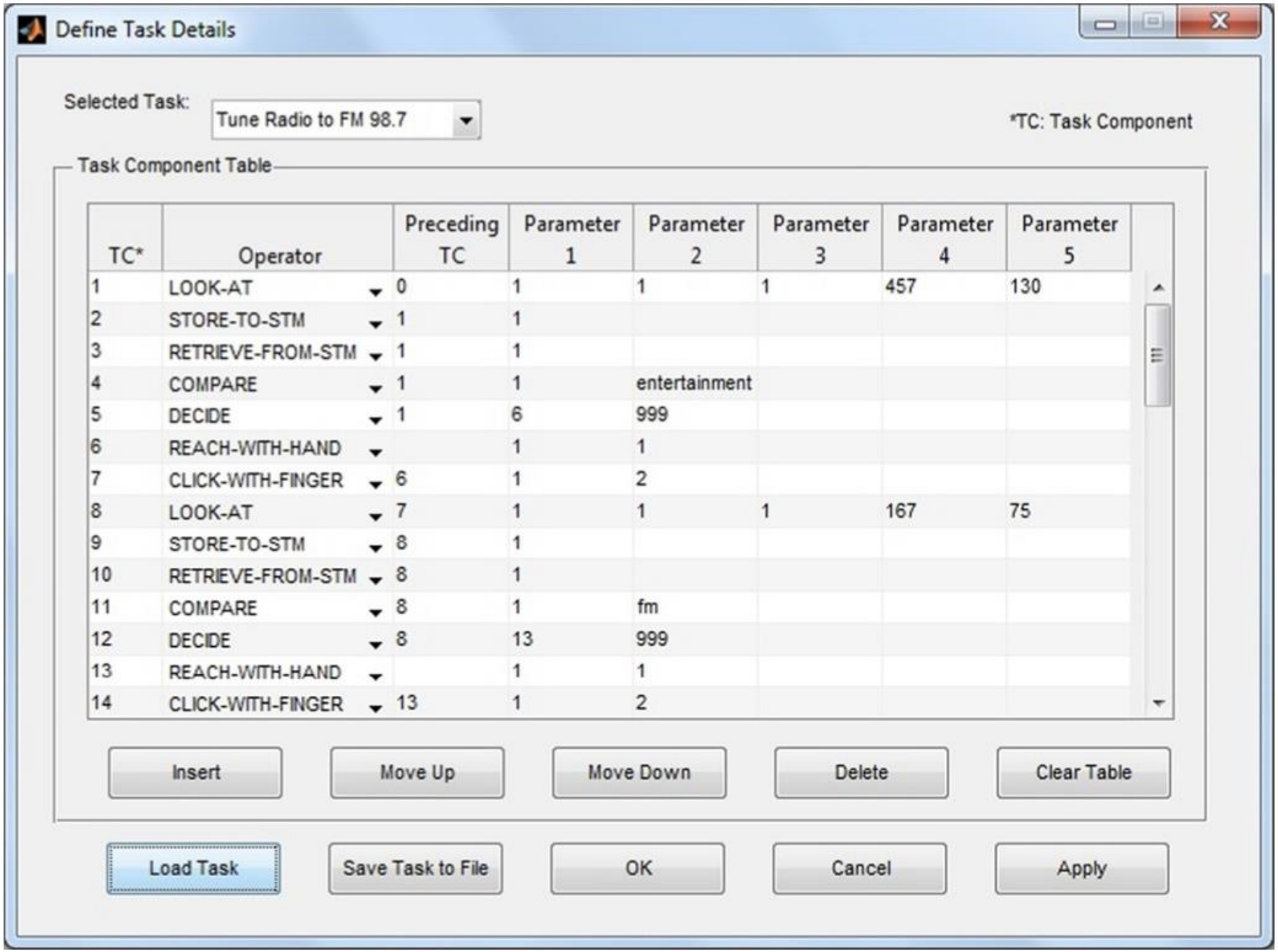
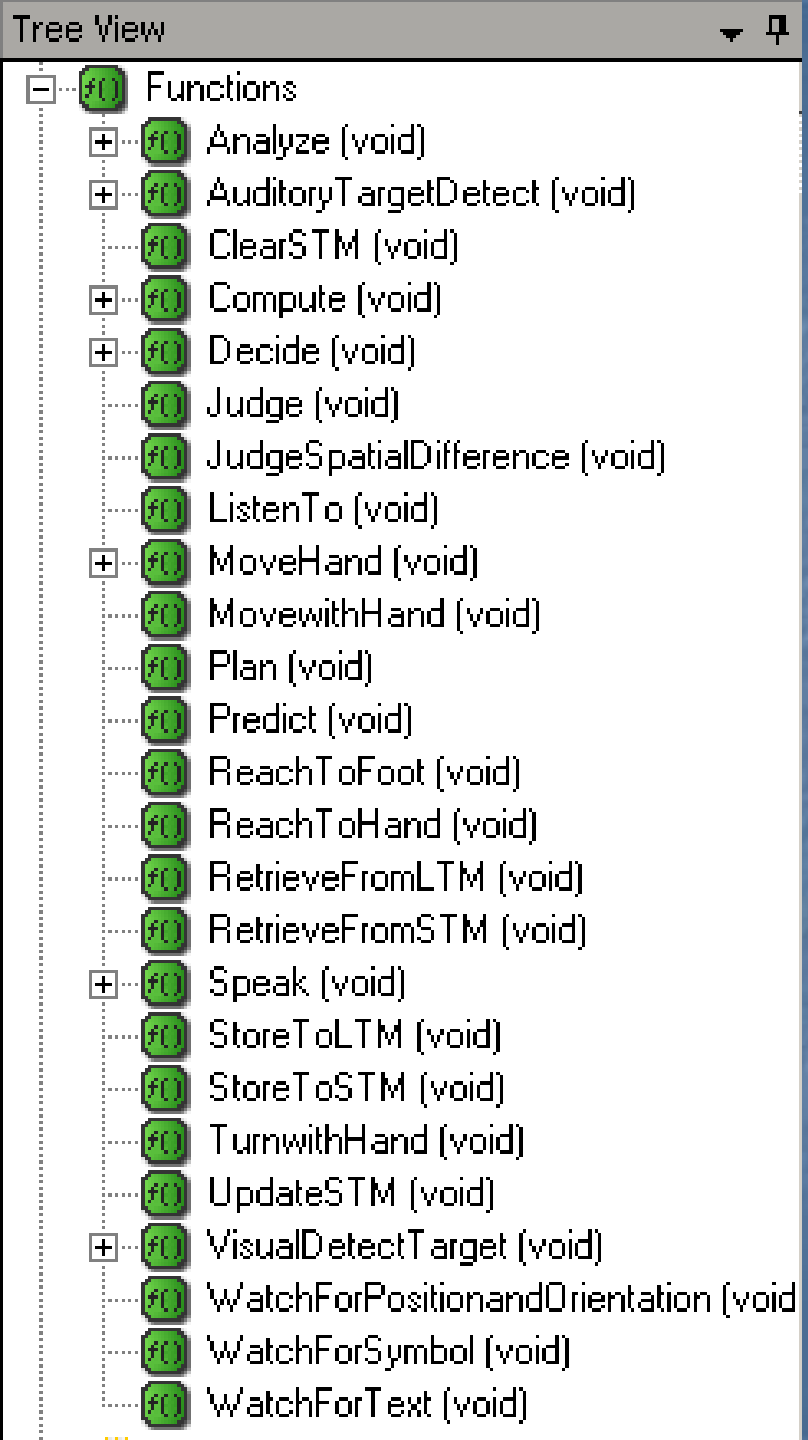


Fig. 6. Task specifications with QN-MHP task components.



Library of Task Elements

- Implemented as functions called from IMPRINT tasks
- Pass parameters about the task from which it is called, maximum completion time, desired path, and whether to cascade to the next task
- Wait for notice that the task is done and possibly additional information (e.g., what path to take)

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

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

Queuing Network Modeling of the Psychological Refractory Period (PRP)

Changxu Wu and Yili Liu
University of Michigan

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

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

Performing multiple tasks at the same time is common in daily life; for example, drivers can steer a car and at the same time talk with friends in the car, and telephone operators can answer customer phone calls and type textual information into a computer. Among the wide range of multiple task situations, the psychological refractory period (PRP) is one of the most basic and simplest forms of a dual-task situation. In a PRP experiment, two reaction-time (RT) tasks are presented close together in time, and participants are asked to perform the two tasks as quickly as possible. Typically, participants' response to the second of the two RT tasks is performed more slowly than to the first when the interval between the presentation times of these two tasks is short. PRP has been studied in laboratories over 100 years, from the behavioral (Creamer, 1963; Kantowitz, 1974; Oberauer & Kliegl, 2004; Pashler, 1984, 1994b; Schumacher et al., 1999; Solomons & Stein, 1896; Welch, 1898; Welford, 1952) to the neurological level (Jiang, Saxe, & Kanwisher, 2004; Sommer, Leuthold, & Schubert, 2001). It is also the subject of extensive theoretical work and the focal point of an important theoretical controversy between several computational models of cognition. There are several important cognitive models of PRP, including the response-selection bottleneck (RSB) or central bottleneck model proposed by Pashler (1984, 1990, 1994a, 1994b, 1994c), the executive-process interactive control (EPIC) model proposed by Meyer and Kieras (1997a, 1997b), and the model based on the ACT-R/perceptual-motor system (ACT-R/PM) proposed by Byrne and Anderson (2001). Each of these models is able to account for some of the important aspects of PRP; however, each appears to encounter at least one experimental counterexample to its predictions (Jiang et al., 2004; Meyer & Kieras, 1997a, 1997b; Oberauer & Kliegl, 2004; Ruthruff, Pashler, & Klaassen, 2001). Therefore, the questions remain about how to model these experimental results, provide a unified account of the discoveries in behavioral and neuroscience studies, and gain further insights into the mechanisms of dual-task performance.

This article takes further steps toward addressing these important questions with a queuing network-based computational cognitive architecture (Liu, 1996, 1997; Liu, Feyen, & Tsimhoni, 2006; Wu & Liu, 2007a, 2007b, 2008; Wu, Liu, & Walsh, in press; Wu, Tsimhoni, & Liu, in press). First, we introduce the major experimental results in PRP studies and the major PRP effects. Second, the major existing models of PRP are described, including their advantages and their counterexamples. Third, we describe the queuing network model, including its major assumptions and components. In the fourth section, we describe how the queuing network model mathematically accounts for and theoretically explains the PRP phenomena. Finally, we discuss the implications of the model, as well as its extension in future research.

EXPERIMENTAL STUDIES IN PRP

In the following section, we introduce the major findings in experimental studies of PRP, including the basic PRP experiment paradigm and the major PRP effects that have been the subject of theoretical controversy—the subadditive difficulty effect, the response grouping effect, the practice effect, and brain imaging patterns in PRP.

Basic PRP Experiment Paradigm

The basic PRP experiment paradigm requires the participants to perform two tasks called Task 1 (T1) and Task 2 (T2) concur-

Changxu Wu and Yili Liu, Department of Industrial and Operations Engineering, University of Michigan.

Changxu Wu is now at the Department of Industrial and Systems Engineering, State University of New York at Buffalo.

We appreciate the support from the National Science Foundation (NSF) for this work (NSF Grant 0308000).

Correspondence concerning this article should be addressed to Changxu Wu, State University of New York (SUNY)—Buffalo, 414 Bell Hall, Buffalo, NY 14260-2050, or to Yili Liu, Department of Industrial and Operations Engineering, University of Michigan, 1205 Beal Avenue, Ann Arbor, MI 48109. E-mail: changxu@buffalo.edu or yililiu@umich.edu

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

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

Queuing Network Modeling of the Psychological Refractory Period (PRP)

Changxu Wu and Yili Liu
University of Michigan

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

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

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

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

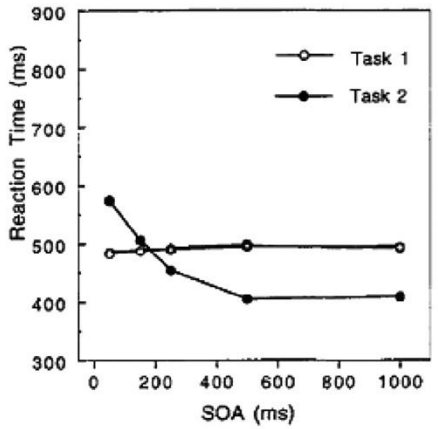


Figure 1. Typical experimental results in the basic psychological refractory period experiment paradigm. Adapted from "Concurrent Response-Selection Processes in Dual-Task Performance: Evidence for Adaptive Executive Control of Task Scheduling," by E. H. Schumacher et al., 1999, *Journal of Experimental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

Subadditive Difficulty Effect

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

Schumacher et al.'s (1999) Study

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

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

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

Karlin and Kestenbaum's (1968) Study

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

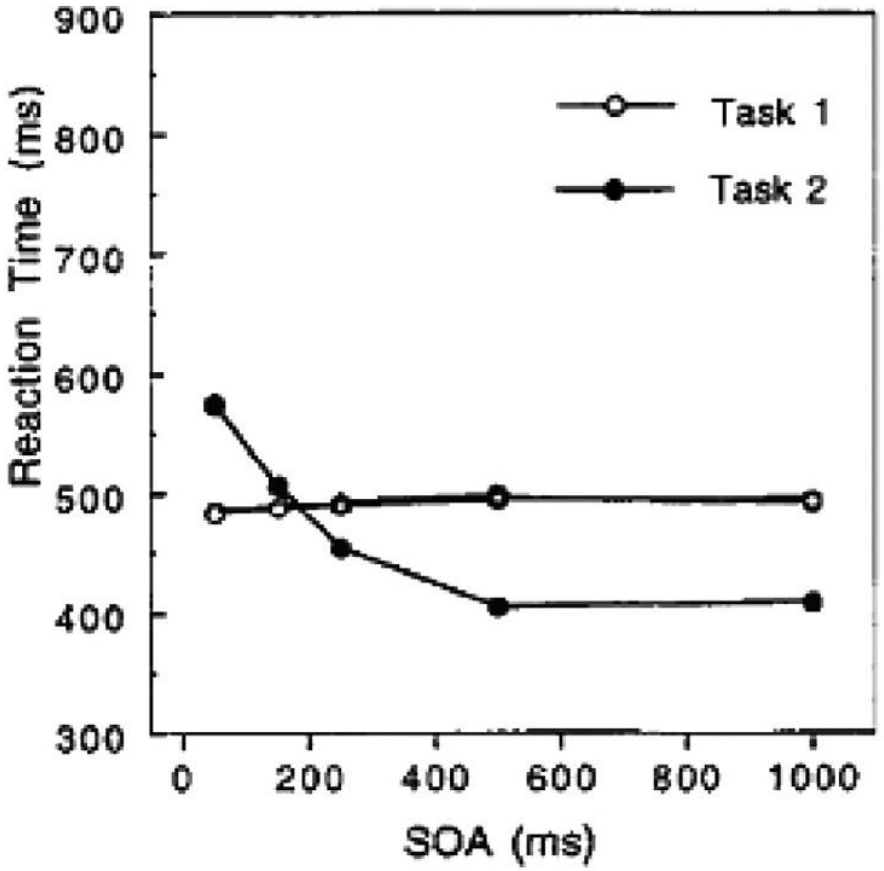
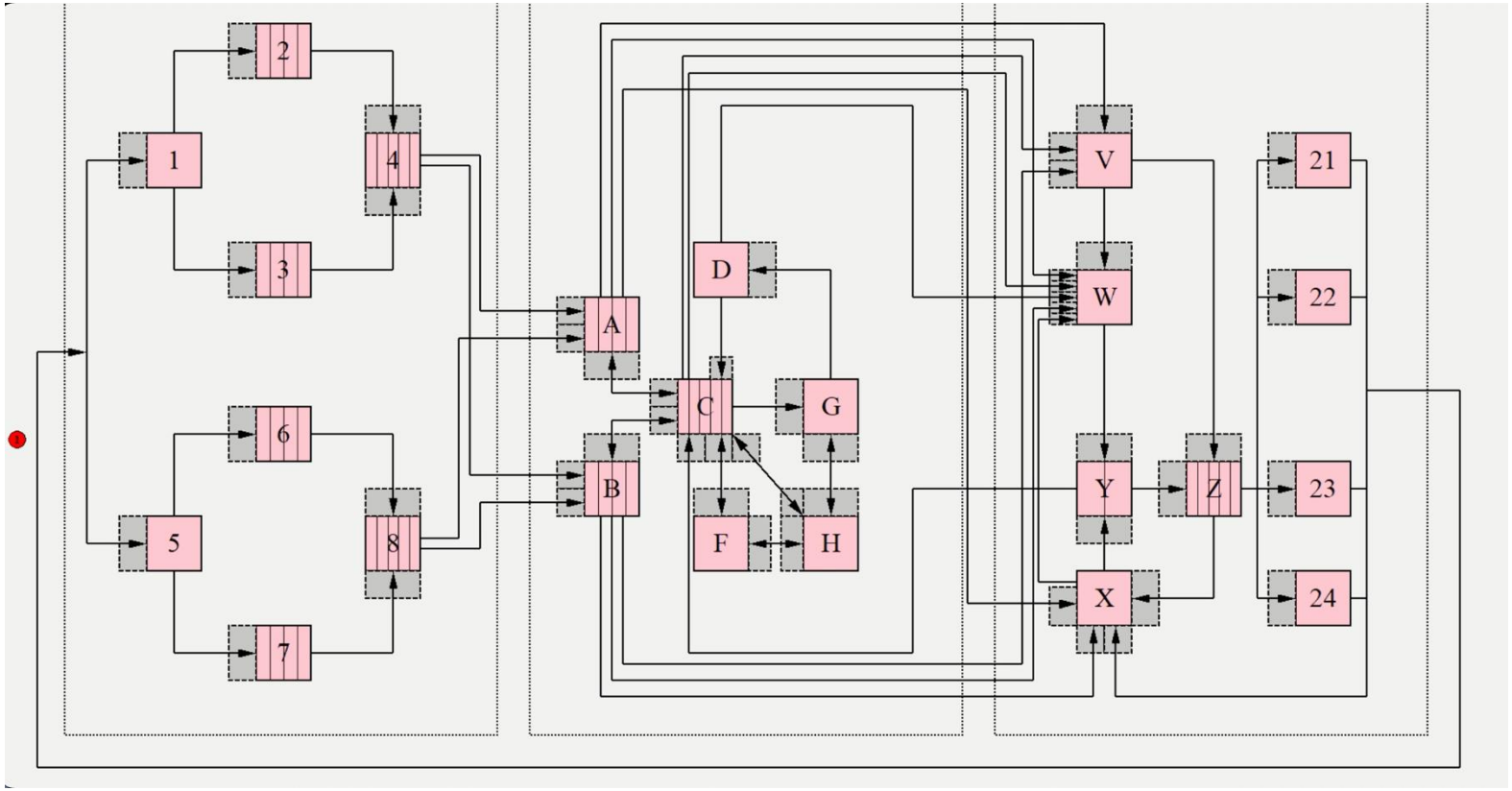
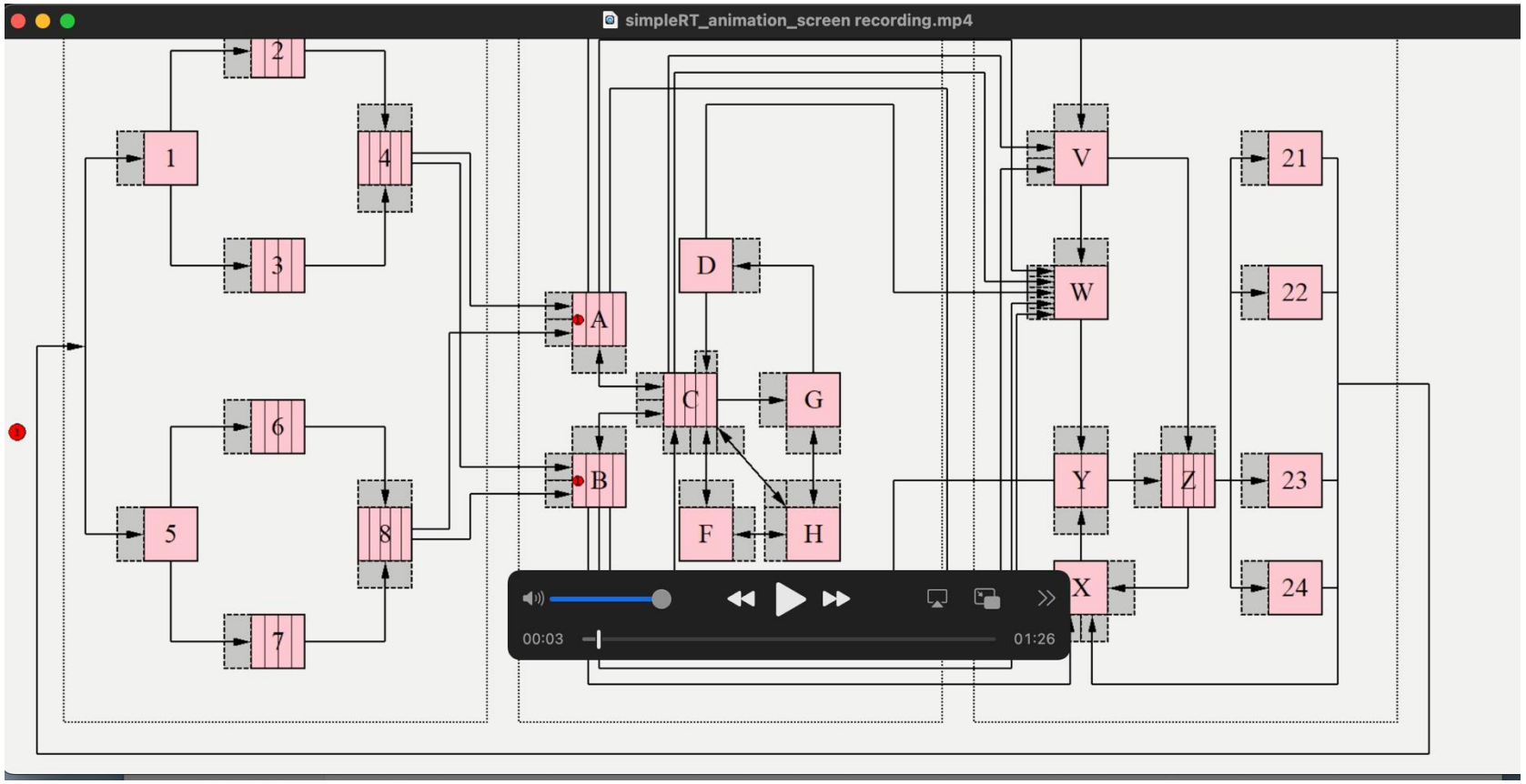


Figure 1. Typical experimental results in the basic psychological refractory period experiment paradigm. Adapted from "Concurrent Response-Selection Processes in Dual-Task Performance: Evidence for Adaptive Executive Control of Task Scheduling," by E. H. Schumacher et al., 1999, *Journal of Experimental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

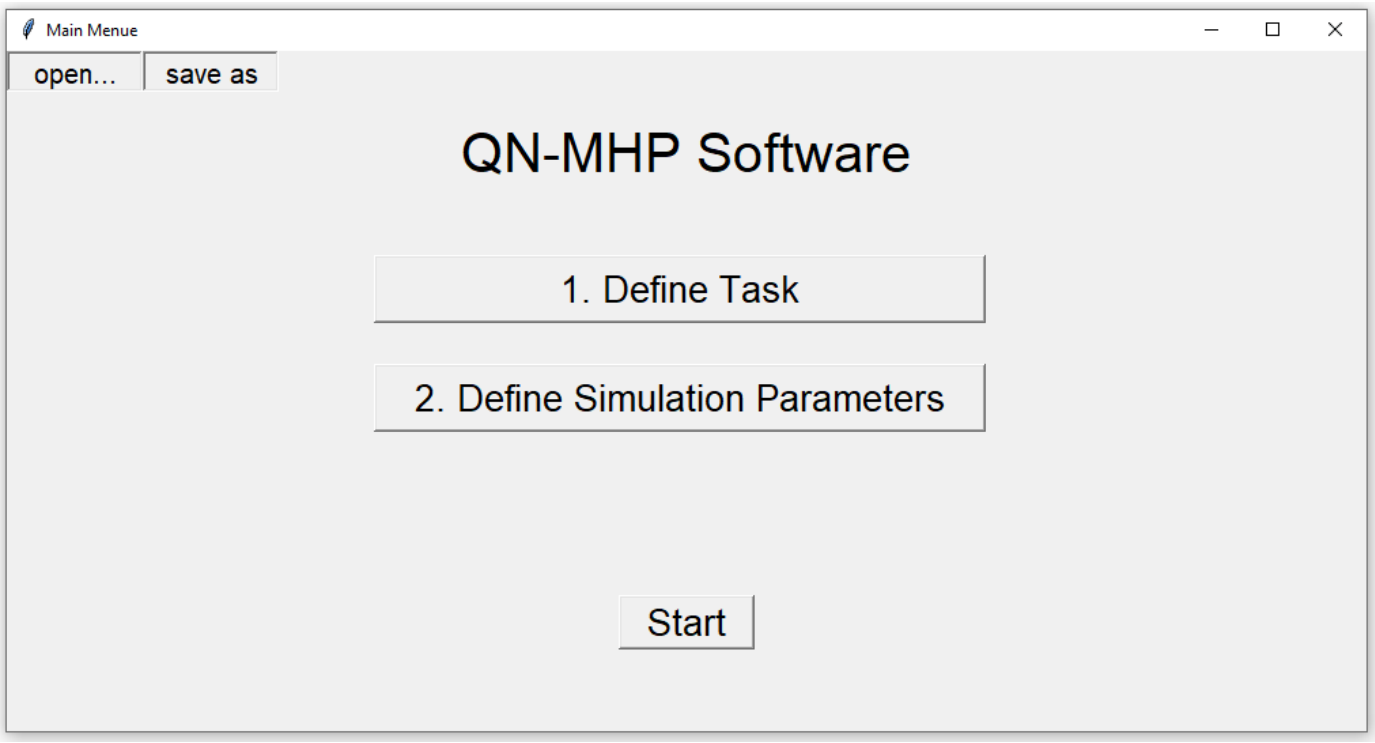
QN-MHP



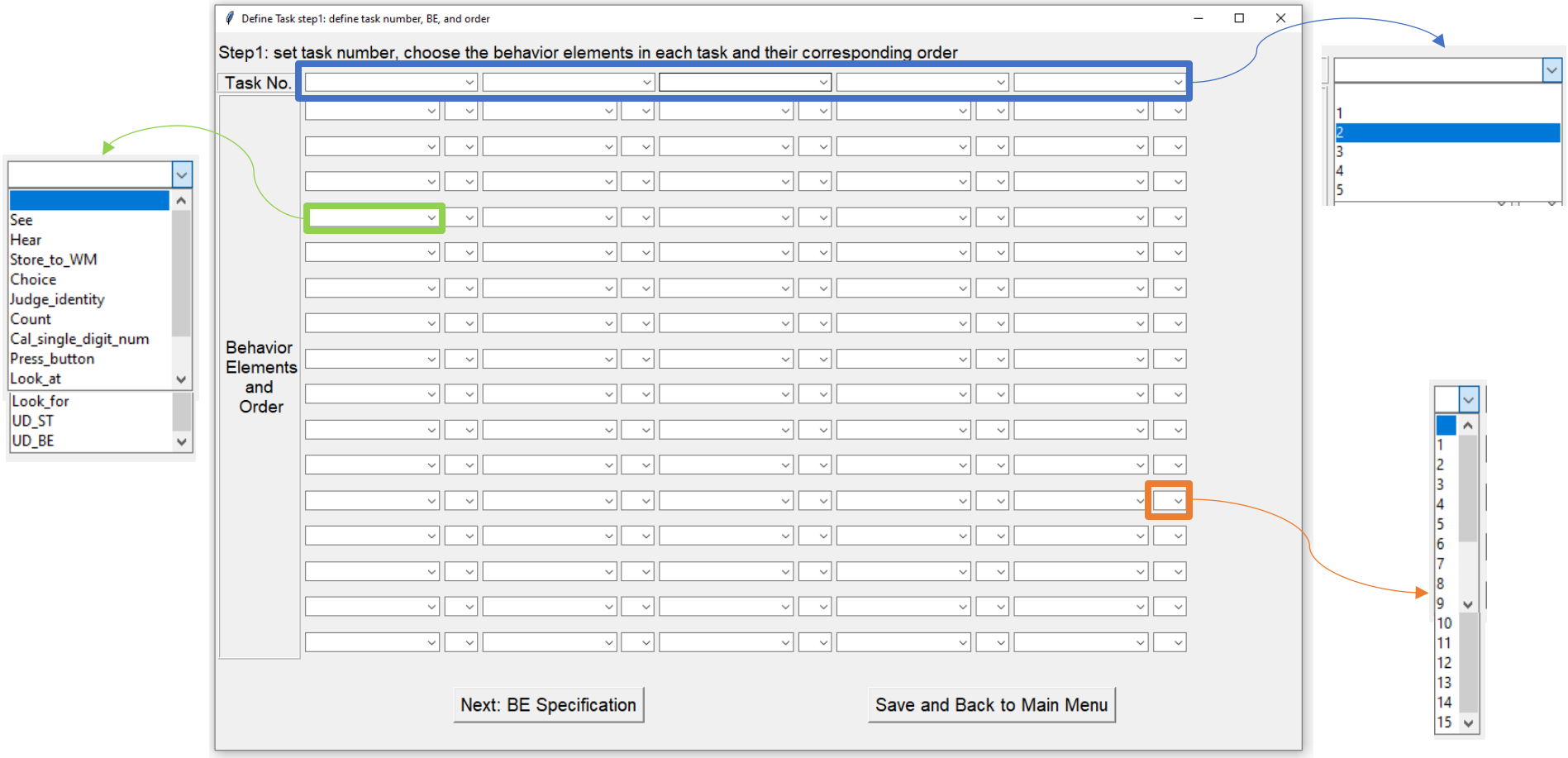
QN-MHP


simpleRT_animation_screen recording.mp4
1
2
3
4
5
6
7
8
A
B
C
D
F
G
H
V
W
X
Y
Z
21
22
23
24
00:03
01:26

Main Menu

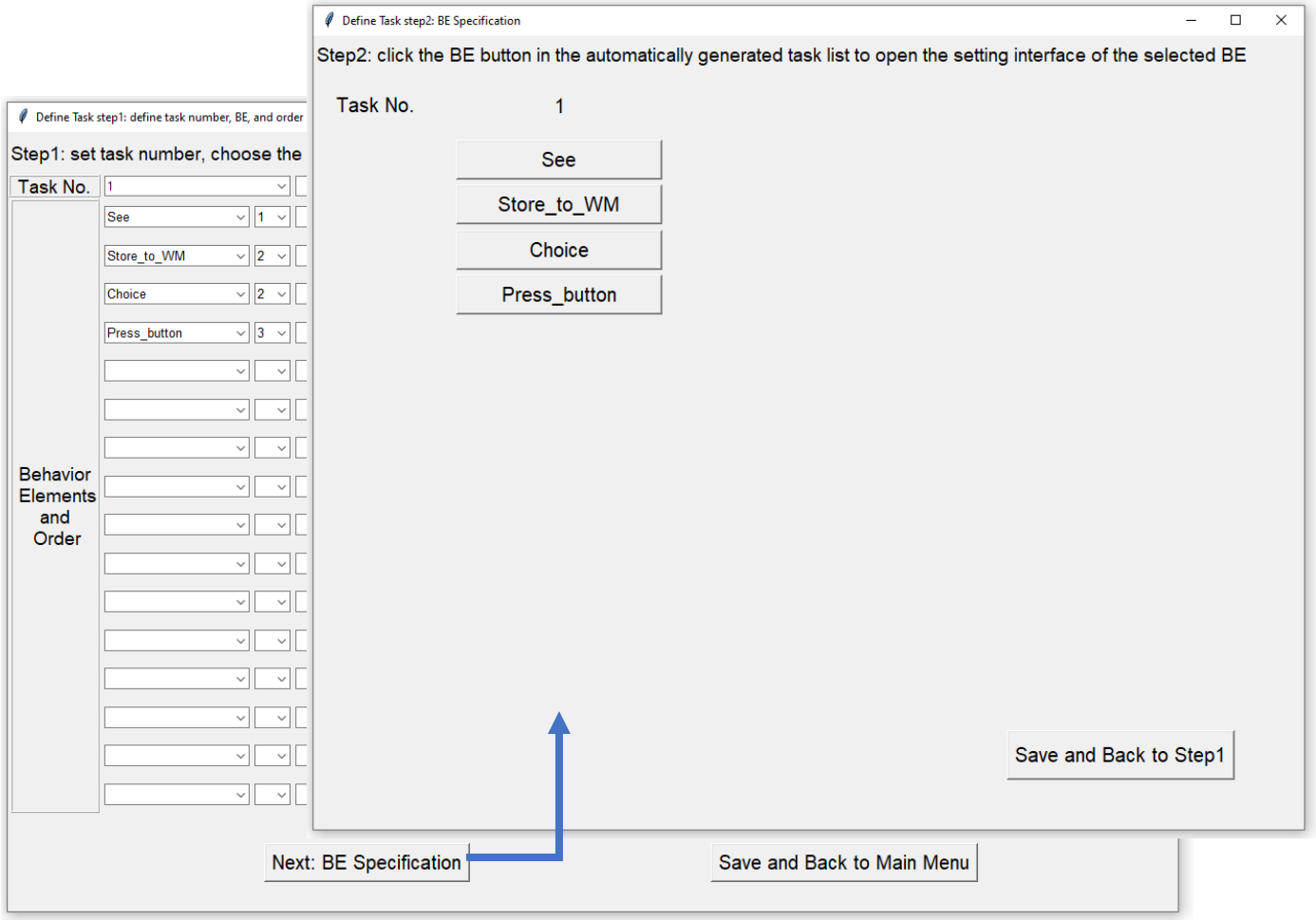

Main Menue
open...
save as
QN-MHP Software
1. Define Task
2. Define Simulation Parameters
Start

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

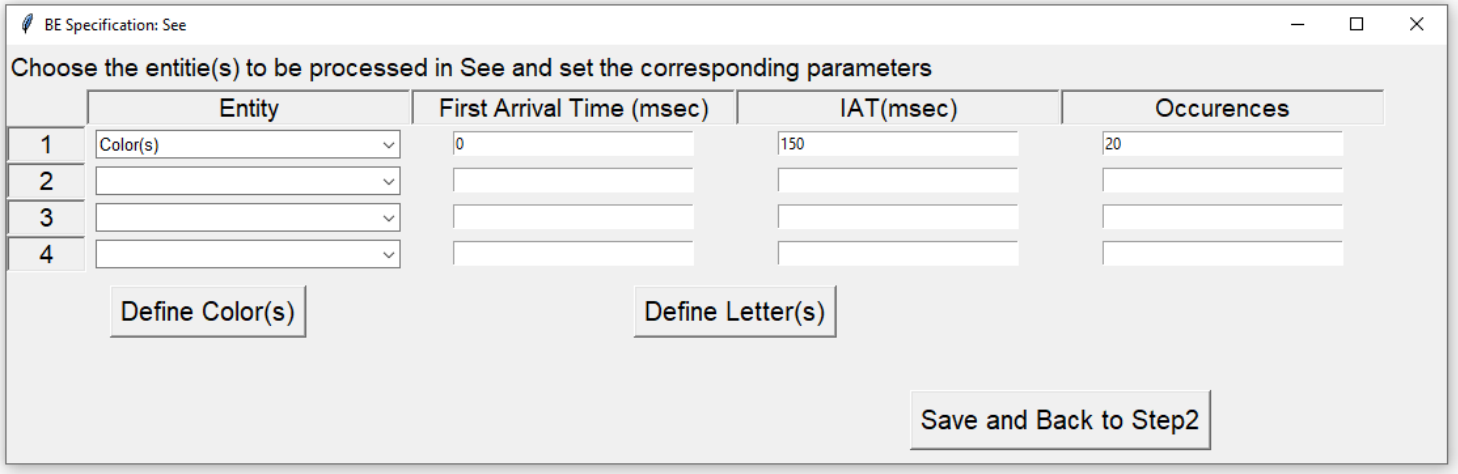


1. Define Task-step2: BE specification

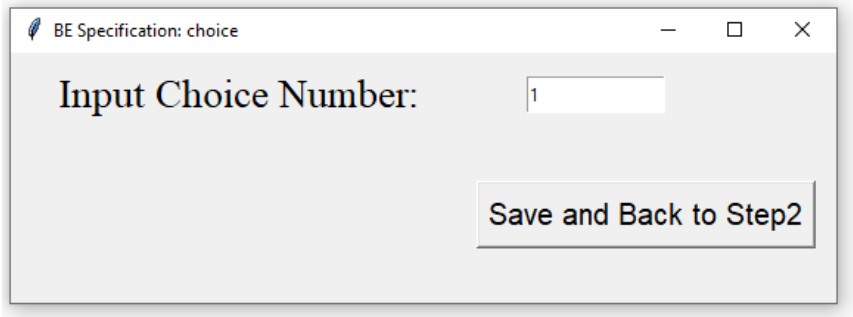
E.g. simple RT task



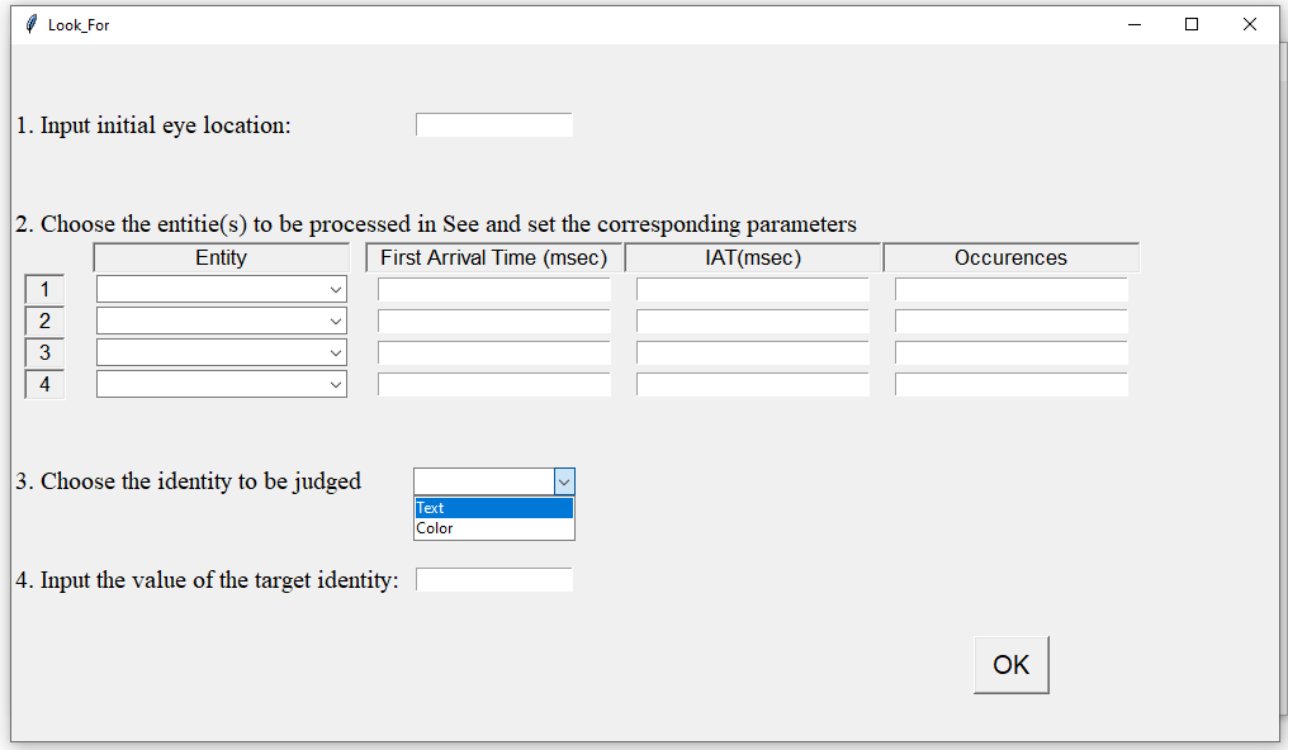
BE specification: see



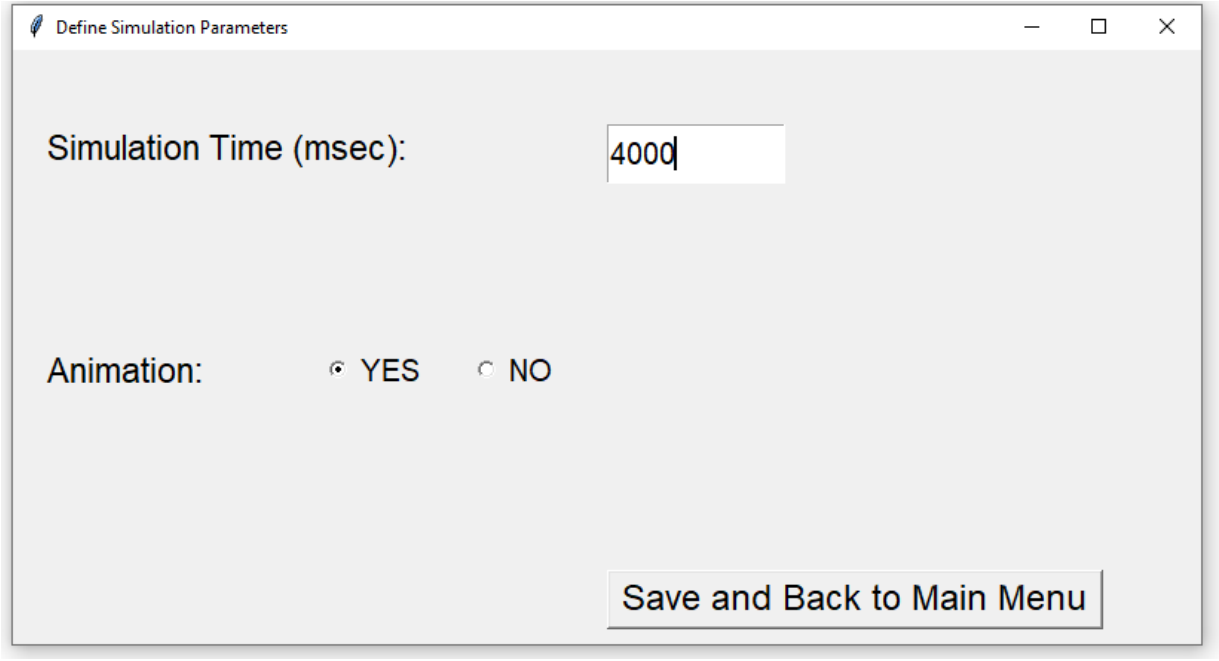
BE specification: choice



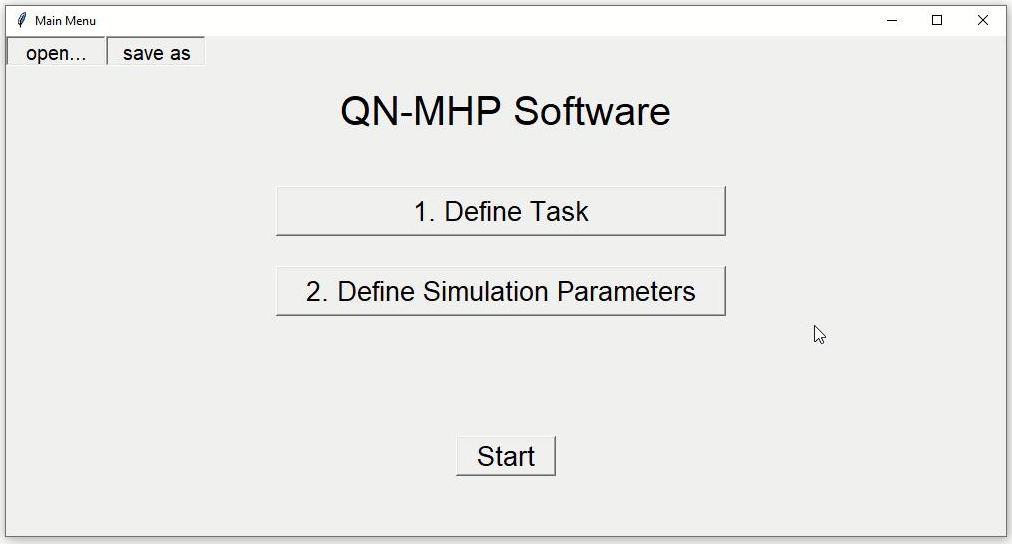
BE specification: Look for



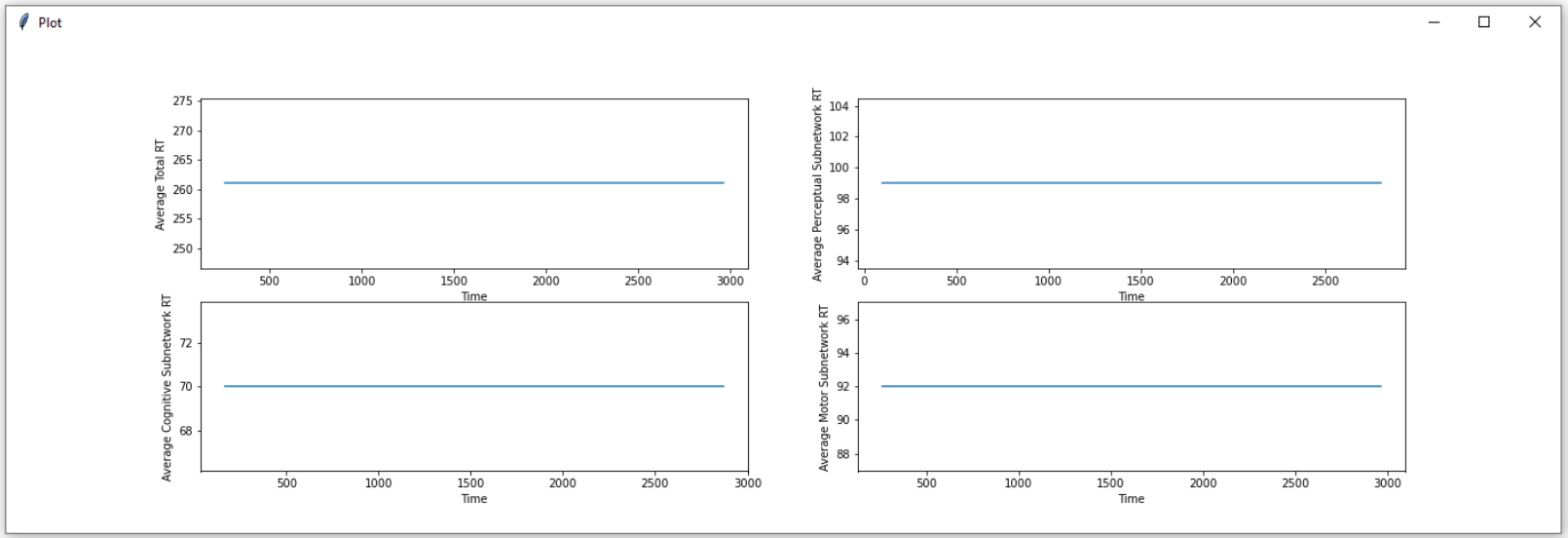
2. Define Simulation Parameters



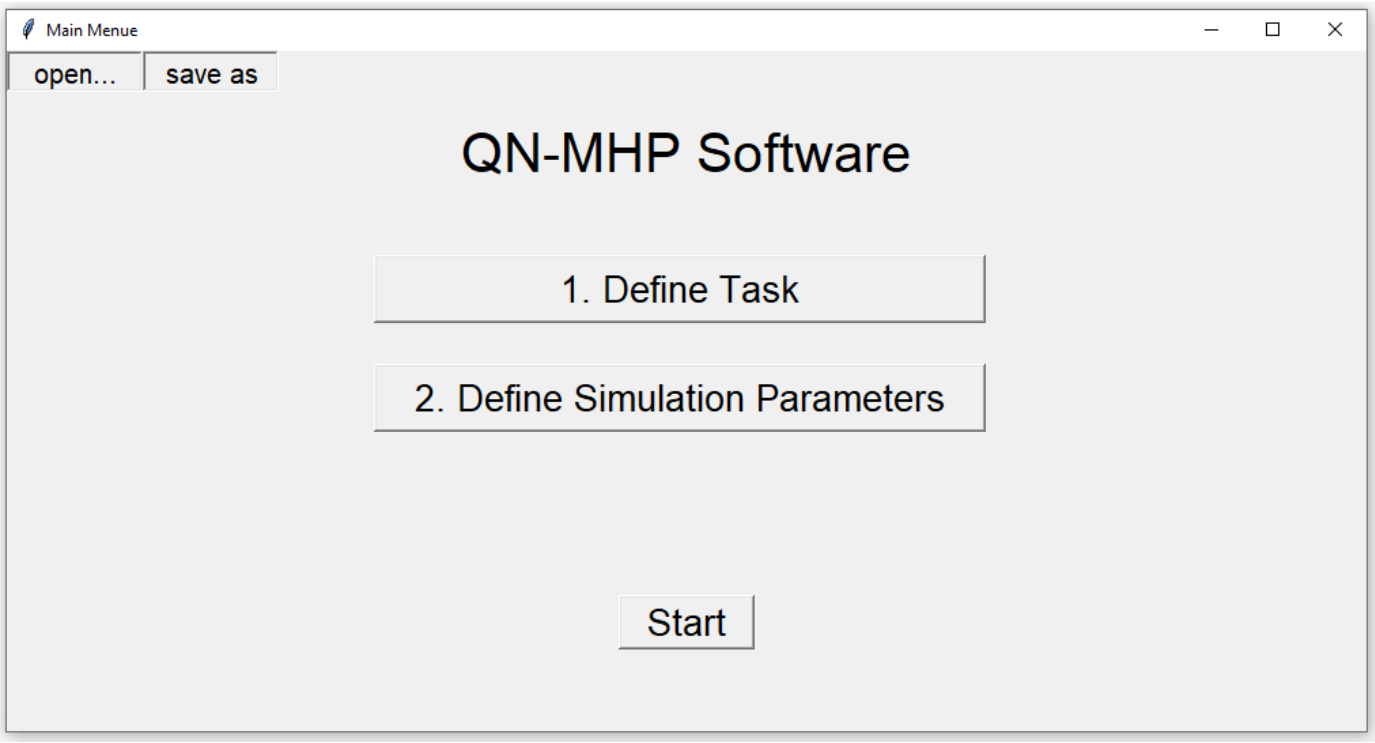
Animation for Choice RT task



Plot for simpleRT task



Main Menu


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

QN-MHP can perform

- Driving (steering and map reading)
- Transcription typing
- Visual search
- RT task
- Psychological Refractory Period
- and more “procedure” tasks

Can also be used to visualize “mental workload”

QN-MHP driving

Queueing Network Modeling of Human Performance in Complex Cognitive Multi-task Scenarios

by

Shi Cao

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Industrial and Operations Engineering)
in the University of Michigan
2013

Table of Contents

Modeling Dual-Task Concurrency and Effort in QN-ACTR and IMPRINT

by

Christopher Jason Best

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Industrial and Operations Engineering)
in The University of Michigan
2013

Doctoral Committee:

Professor Yili Liu, Chair
Assistant Professor Victoria Booth
John F. Locket III, US Army Research Laboratory
Professor Nadine B. Sarter

Modeling Dual-Task Concurrency and Effort in QN-ACTR and IMPRINT

by

Christopher Jason Best

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Industrial and Operations Engineering)
in The University of Michigan
2013

TABLE OF CONTENTS

Modeling of Army Research Lab (ARL) cognitive load shooting task

- Primary task: "friend" or "foe" shooting task
- Secondary task: "Arithmetic addition"

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