Development and Evaluation of an Ergonomic Software Package for Predicting Multiple-task Human Performance and Mental Workload in Human-Machine Interface Design and Evaluation

Changxu Wu
Department of Industrial and Systems Engineering
State University of New York (SUNY)-Buffalo

Yili Liu
Department of Industrial & Operations Engineering
University of Michigan

Abstract

Predicting human performance and mental workload in multiple task situations at an early stage of system design can save a significant amount of time and cost. However, existing modeling tools either can only predict human performance or require users of tools to learn a new programming language. Queueing Network-Model Human Processor (QN-MHP) is a new cognitive architecture for modeling both human performance and mental workload in multiple tasks. This paper describes the development of a Visual Basic Application in Excel (VBA) software package and an illustrative case study to evaluate its effectiveness. The software package has an easy-to-use user interface for QN-MHP that assists users of the modeling tool to simulate a dual task including definition of the tasks and interfaces by clicking buttons to select options and filling texts in a table, with no need to learn a simulation language. It allows the model user to intuitively observe the information processing state of the model during simulation, and conveniently compare the simulated human performance and mental workload for different designs. The illustrative case study showed that naïve users without prior simulation language programming experience can model human performance and mental workload in a complex multitask situation within 3 minutes; and this software package can save 71% of modeling time and reduce 30% of modeling errors. Further developments of the VBA software package of QN-MHP are also discussed on how to make it a comprehensive proactive ergonomic design and analysis tool.

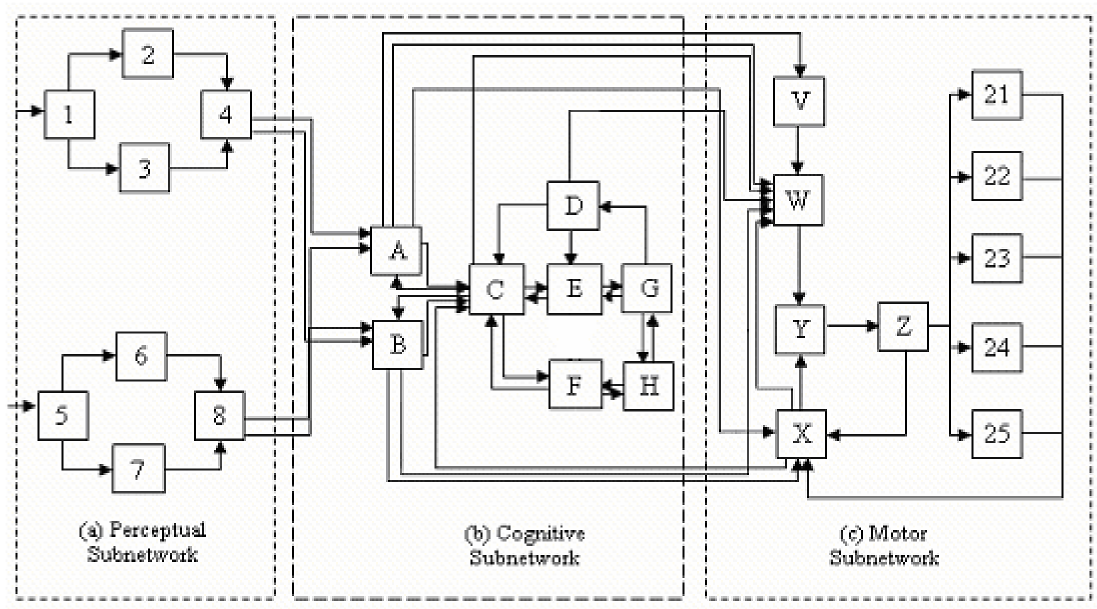
Keywords: Ergonomics, Software package, Queueing Network, Human performance, Mental workload

1. Introduction

Performing two tasks concurrently is one of the common activities in human-machine interaction. Operators in control rooms in manufacturing industry may operate a device and monitor several displays at the same time. Pilots need to control the airplane and at the same time communicate with the air traffic controllers. Drivers may operate an instrument panel or use a Global Positional System (GPS) while steering a car simultaneously. Predicting human performance and mental workload in dual task conditions at an early stage of system design can save system development teams (engineers, human-machine interface designers and even managers) a significant amount of time and cost in comparison to revising the systems at a later stage of system development (Gore, 2000, 2002). The need for models that can predict both performance and mental workload has often been mentioned in the literature on human modeling (Olsen & Olsen, 1990).

Besides several digital modeling tools to predict and assess human physical movement (Kuo & Chu, 2005; Resnick & Zanotti, 1997; Shidar, SAl-Araimi, & Omurtag, 2002; Yoon & Kim, 1996), several cognitive modeling techniques have been recently developed and they are mainly used to predict human performance, including CRITIQUE (Hudson & Stasko, 1995), Micro Saint (Laughery, 1989; Schunk, 2000), APEX (M. Freed, Matessa, Remington, & Vera, 2003; M. A. Freed & Remington, 2000), and QN-MHP (Wu & Liu, 2004a, 2004b, 2004c, 2006a, 2006b, 2006c, 2007, 2008, In Press). Hudson et al. (1995, 1999) developed a innovative modeling tool called CRITIQUE (the Convenient Rapid, Interactive Tool for Integration Quick Usability Evaluations), which was able to automatically produce KLM (Keystroke-Level Model) to predict single task performance time. CRITIQUE uses features of the subArctic input model to transparently record detailed logs of user interactions. Micro Saint was developed in 1985, and it is another valuable human performance modeling software package (Laughery, 1989; Schunk, 2000). It uses a task network modeling method to predict performance time: activities of target users of a device or system are represented in a diagram as nodes and arrows between the nodes, which represent the sequences in which the activities are performed. Researchers in the NASA Ames Research Center (e.g., Freed & Remington, 2000) developed a useful modeling tool called APEX—a GOMS (Goal-Operator Methods-Selection)-like framework that incorporates mechanisms and methodologies for predicting certain forms of human error.

Queueing network modeling approach has been established both as a psychological theory and a human performance modeling technique in human-computer interaction. In modeling human performance, computational models based on queueing networks have successfully integrated a large number of mathematical models in response time (Liu, 1996) and in multitask performance (Liu, 1997) as special cases of queueing networks. A simulation model of a queueing network mental architecture, called the Queueing Network-Model Human Processor (QN-MHP), has been developed to represent information processing in the mental system as a queueing network on the basis of neuroscience and psychological findings. Ample research evidence has shown that major brain areas with certain information processing functions are localized and connected with each other via neural pathways (Bear, Connors, & Paradiso, 2001; Faw, 2003; Roland, 1993; Smith & Jonides, 1998), which is highly similar to a queueing network of servers that can process entities traveling through the routes serially or/and in parallel depending on specific network arrangements. Therefore, brain regions with similar functions can be regarded as servers and neural pathways connecting them are treated as routes in the queueing network (see Figures 1 and 2). Information being processed in the network is represented by entities traveling network.



Perceptual Subnetwork

1. Common visual processing
2. Visual recognition
3. Visual location
4. Visual recognition and location integration
5. Common auditory processing
6. Auditory recognition
7. Auditory location
8. Auditory recognition and location integration

Cognitive Subnetwork

A. Visuospatial sketchpad
B. Phonological loop
C. Central executive
D. Long-term procedural memory
E. Performance monitor
F. Complex cognitive function
G. Goal initiation
H. Long-term declarative & spatial memory

Motor Subnetwork

V. Sensorimotor integration
W. Motor program retrieval
X. Feedback information collection
Y. Motor program assembling and error detecting
Z. Sending information to body parts
21-25: Body parts: eye, mouth, left hand, right hand, foot

Figure 1. The general structure of the queueing network model (Wu & Liu, 2007)

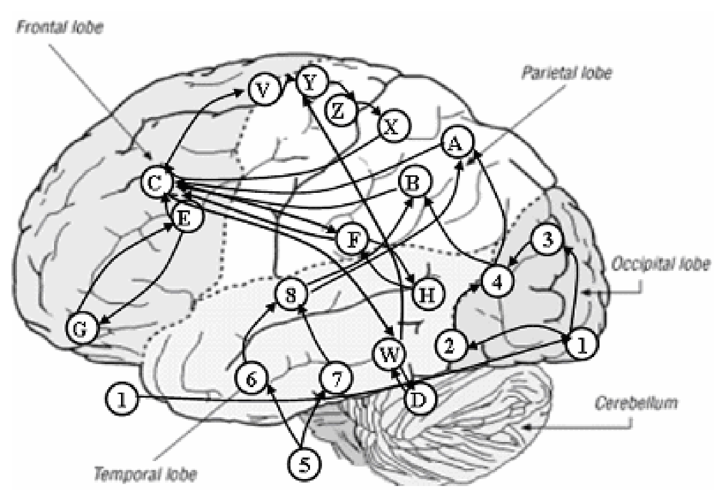


Figure 2. Approximate mapping of servers in the queueing network model onto human brain (Wu & Liu, 2007)

QN-MHP has been successfully used to generate human behavior in real time, including simple and choice reaction time (Feyen, 2002), transcription typing (Wu & Liu, 2004a, 2004b, 2008), psychological refractory period (Wu & Liu, 2004c, In Press), physiological index of mental workload in visual-manual tracking (Wu & Liu, 2006a, In Press), subjective index of mental workload in driving (Wu & Liu, 2006b, 2006c, 2007), and driver performance (Liu, et al., 2006). One of the most important advantages of QN-MHP in modeling multiple tasks is that: multitask performance emerges as the behavior of multiple streams of information (represented by entities in the queueing network) flowing through a network, with no need to devise complex, task specific procedures to either interleave production rules into a serial program or for an executive process to interactively control task processes (Liu et al., 2006). Similar to the other modeling tools, however, users of the current version of QN-MHP (called QN-MHP 2.0) need to learn a manufacturing simulation language—Promodel, with which QN-MHP is currently implemented.

Table 1. Summary of existing modeling tools

Existing Tools	**Functions**		**Ease of Use**	
	Modeling multitask performance	Modeling mental workload	Usage of widespread UI rapid designing tool	Need to learn a new programming language
CRITIQUE	No	No	subArctic	Yes (subArctic)
Micro Saint	Yes	Yes	-	Yes (Micro Saint)
APEX	Yes	No	-	Yes (PDL)
QN-MHP 2.0 without VBA package	Yes	Yes	No	Yes (Promodel)
QN-MHP 2.0 with VBA package	**Yes**	**Yes**	**VBA in Excel**	**No**

-: Not mentioned in their documents or under development

Table 1 summarizes the functions and ease of use of the existing modeling tools introduced above. This review of the major existing tools clearly indicates that there is a need to develop an easy-to-use software tool for modeling human performance and mental workload in multitask situations, especially for users (engineers, interface designers and even managers) who do not have prior programming experience. In the following sections of this article, we describe the VBA software package of QN-MHP 2.0 as a new modeling tool for users who do not have prior simulation programming language experience, including its detailed software architecture as well as information transmission between the tool and the original Promodel file. In addition, we also describe a case study to illustrate the effectiveness of the software package in reducing the amount of time and errors in task modeling and in enhancing users' subjective evaluation.

2. Development of VBA Software Package for QN-MHP 2.0

Original Structure of QN-MHP 2.0

Before we discuss the development of the VBA software package for QN-MHP 2.0, it is necessary to introduce the original structure of the QN-MHP 2.0 without the package.

The original model is composed of two components: an Excel file containing the task and interface information and a Promodel file including all of the servers and routes as a task-independent cognitive architecture (see Figure 3). Once the researchers know the critical modeling information, including detailed information of the task to be simulated and interfaces to be evaluated, they need to define the Excel sheets in the Excel file according to the modeling information by using a set of pre-defined numerical codes to represent the operators or actions—an element in task analysis in QN-MHP and a task can be decomposed in a combination of these operators. Then, they need code the three parts in the Promodel file: arrival (define the arrival of task information), macro (define the type of task), and array (define which sheets in the Excel file to use). After the researchers define the Excel file and Promodel file, they need to open and run the Promodel file to get the simulation results of human performance and mental workload. The Promodel file reads the Excel file during the simulation process.

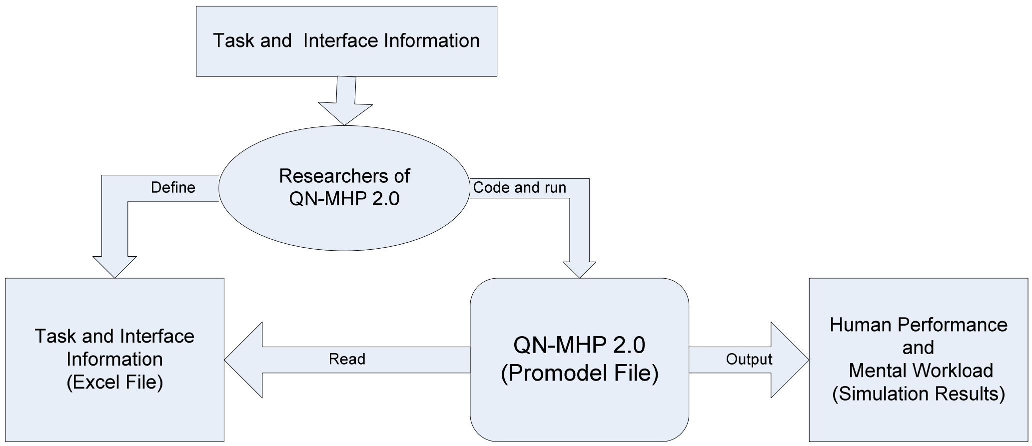


Figure 3. Original structure of the QN-MHP 2.0 without VBA software package

Overall Structure of QN-MHP 2.0 with VBA Software Package

Built on the original structure of QN-MHP 2.0, a Visual Basic Application in Excel (VBA) package was developed and this package was composed of two components (see Figure 4): an easy-to-use VBA user interface of the model and an ActiveX module (these two components together are called "VBA software package" or "modeling tool" of QN-MHP 2.0 in this paper). The role of the VBA user interface of the Model is to help users of the QN-MHP 2.0 define the task and interface information as well as automatically export the task and interface information to the Excel file. Then the information in the Excel file is exported to the ActiveX module by the software itself. The ActiveX module automatically codes the original QN-MHP 2.0 Promodel file according to the information from the Excel file and runs the Promodel file to generate the simulation results.

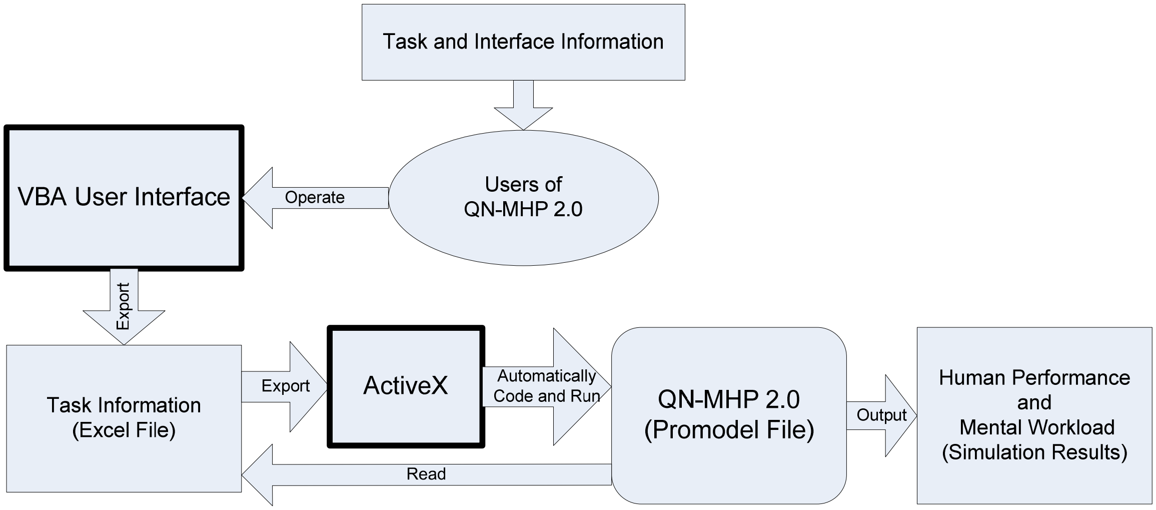


Figure 4. Overall Structure of QN-MHP 2.0 with VBA software package including a VBA user interface and an ActiveX modules (the two boxes highlighted with bold borders)

VBA User Interface

The user interface of QN-MHP 2.0 is developed using a widely used rapid UI design software: Visual Basic Application (VBA) in Excel. Figure 5 describes the flow chart in using this VBA user interface to define the task and interface information. To define a single or a dual task, users of the modeling tool need three steps to define the task and interface information using this VBA user interface. First, users have an option to choose whether the target task to be simulated is a single or dual task. After that, the users can define the single and dual tasks individually (see Figures 6 and 7).

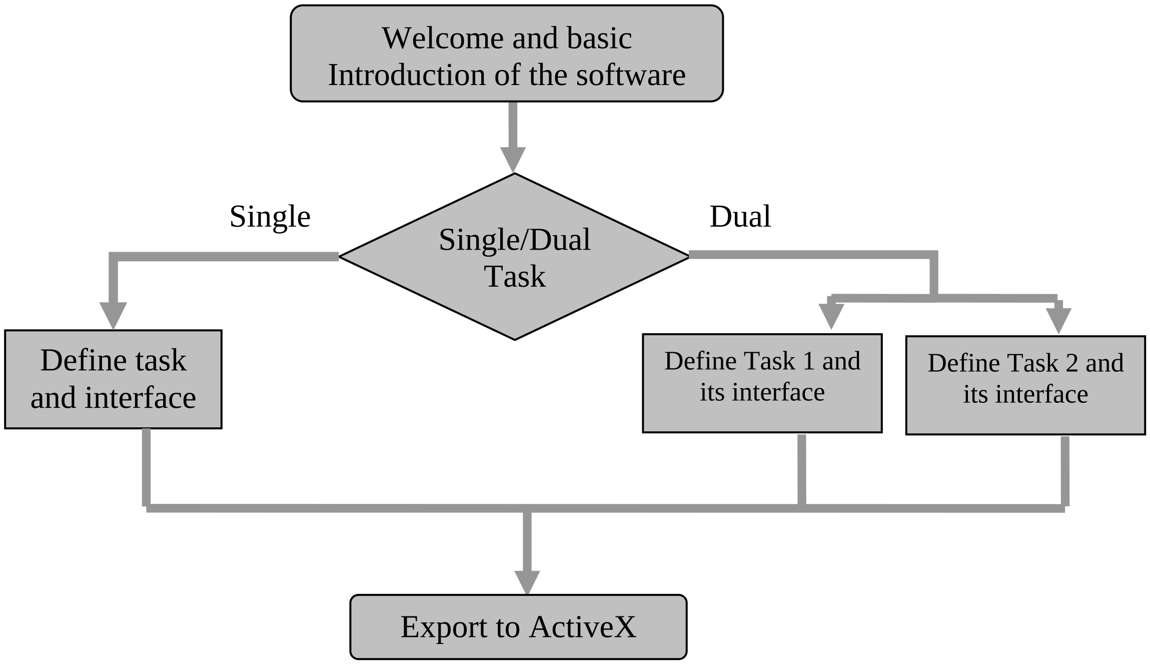


Figure 5. Flow Chart of the VBA user interface

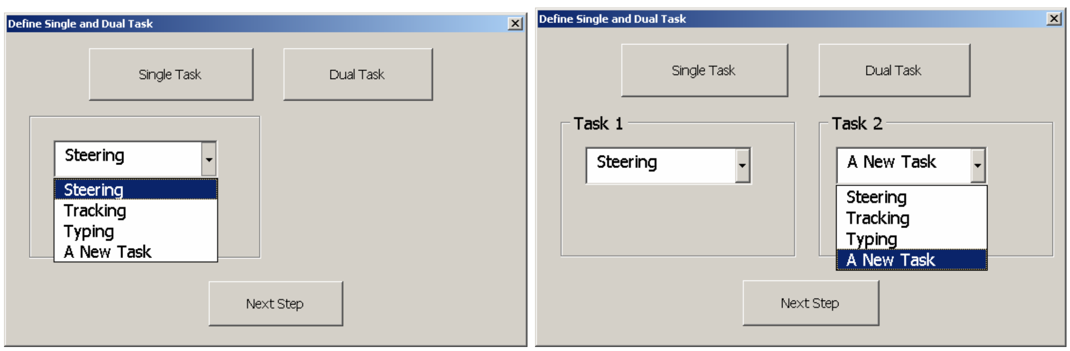


Figure 6. Screenshots of the user interface in selection of a single or dual task modeling

Second, in defining a single task, users have four options including three options to use the existing module: steering, typing and visual manual tracking and the fourth option to define the task as a new task. Similarly, in defining a dual task, users have these four options for both tasks (see Figures 6 and 7). If the "a new task" option is selected, a table automatically shows up so that the users can define the user interface mock-ups (the graphic images of the UIs and their path on the computer) and the name of objects in these user interface mock-ups (see Table 2).

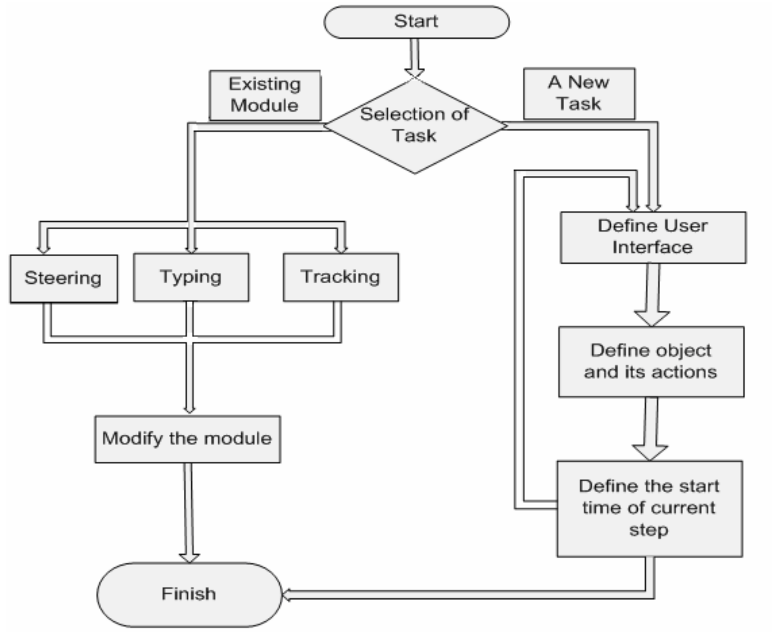


Figure 7. Flow chart of the VBA user interface in defining a task

(A single task or Task 1 or Task 2 in a dual task situation)

Table 2. A sample table for defining the user interfaces and objects in them

InterfaceID	Interface	Path	ObjectID	Object
1	UI_1	E:\UI_photos\GPS.jpg	1	Menu Button
			2	Direction Arrow
			3	Message
2	UI_2	E:\UI_photos\Plate.jpg	4	Address Button

Third, the interface automatically changes according to users' selection of the tasks (see Figures 8 and 9): Figure 8 shows the interface when users choose steering as Task 1 and a new task as Task 2 in a dual task condition. Users can define a change of UI in the image window at the left side of the dialogue box. Starting from the top on the right side of the dialogue box, users specify an object in the UI (e.g., the "menu" button) previously defined in the table (see Table 2), its sensory channel (visual, auditory, or tactile), and a series of actions or operators corresponding to that object (e.g., "look at," "reach to it (by hand)" etc.). Rather than typing a numerical code into an Excel sheet, when users demonstrate the tasks by choosing a certain action/operator, the VBA user interface automatically translates this action/operator and store its numerical code to the Excel file. The definition of a series of actions corresponding to an object is called a step, and users specify the start time of these actions by: 1) waiting until the start of the previous step (start at the same time as the previous step), 2) waiting until several actions of the previous step occur, 3) waiting until the end of the previous step, or 4) starting independently of the previous step (users are allowed to define the start time by an event or absolute start time). Users can then click on the next step to define a series of actions corresponding to another object. In the dual task condition, after users define both tasks, users can define the priority of each task by clicking on the "Task Priority" button.

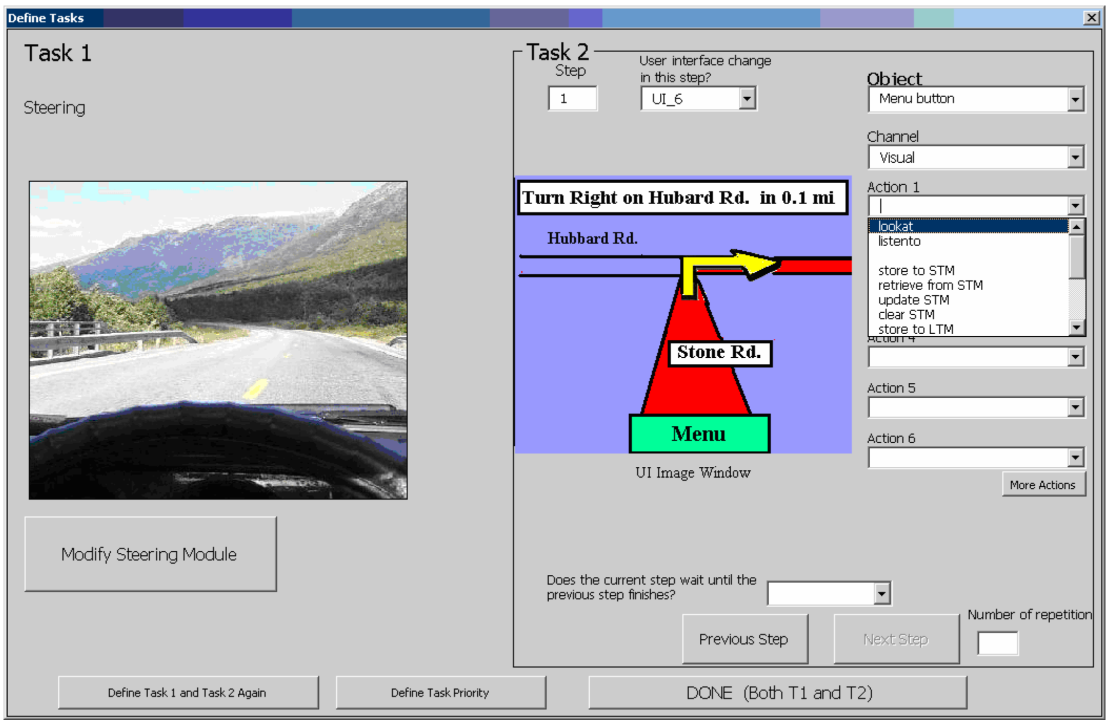


Figure 8. A screenshot of the VBA user interface in defining a dual task (Steering as Task 1 and a new task as Task 2 in a dual task condition)

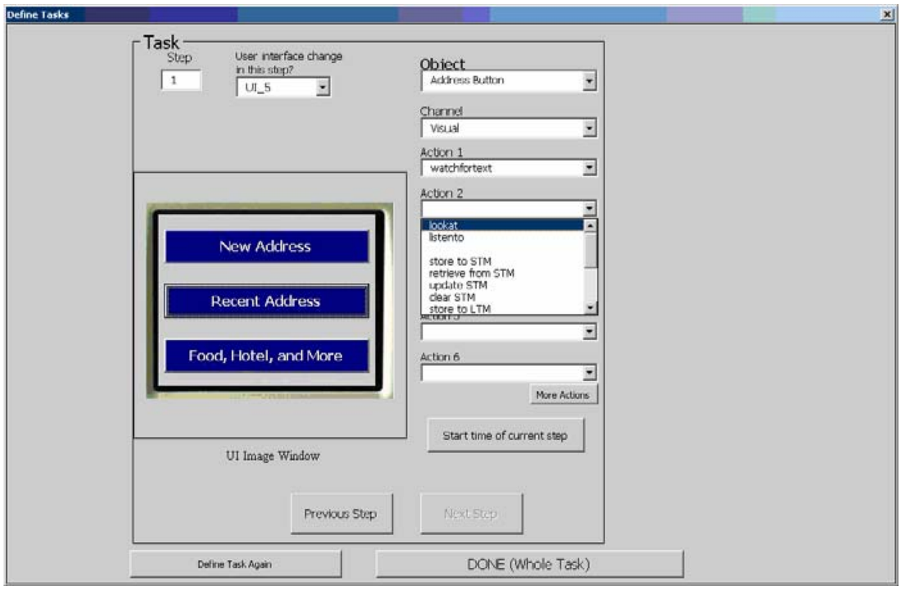


Figure 9. A screenshot of the VBA user interface of QN-MHP 2.0 in defining a single task (A new task in a single task condition)

All of the task and interface information is exported automatically to the Excel file when users press the "Done" button located at the bottom of the interface. At the same time, the information in the Excel file is also exported to the ActiveX module introduced in the following section.

ActiveX and Promodel Simulation

The ActiveX module in the software is developed based on the ActiveX module provided in the Promodel software package. The module is composed of: a) Several Excel macros and each of these macros is corresponding to a component in Promodel software (e.g., arrays, arrivals, and processing, etc.) and these macros are able to automatically communicate with the corresponding components in Promodel to update the Promodel codes (Promodel, 2003); b) A "Control" Excel sheet which manages all of these Excel macros and it contains subroutings (a set of VBA codes) controlling and activating all of the Excel macros. For example, a self-developed "Update Promodel File" subrouting activates all the macros to export codes to a Promodel file and this subrouting can be initiated by a "Update Promodel File" button on the "Control" sheet.

The function of this ActiveX module in the VBA software package is to automatically update three parts of the Promodel file (arrival, array and macro) corresponding to the information imported from the Excel file after users press the "Update Promodel File" button (see Figure 10). Once users click on the "Run Simulation" button on the "Control" sheet, the Promodel file is activated and run.

During the simulation, users can observe the dynamic activities of entities in the network and the changes of utilization of subnetworks on the dynamic plot (see Figure 11). The simulation results of human performance and index of mental workload (utilization of subnetwork) is reported after the Promodel file finishes running.

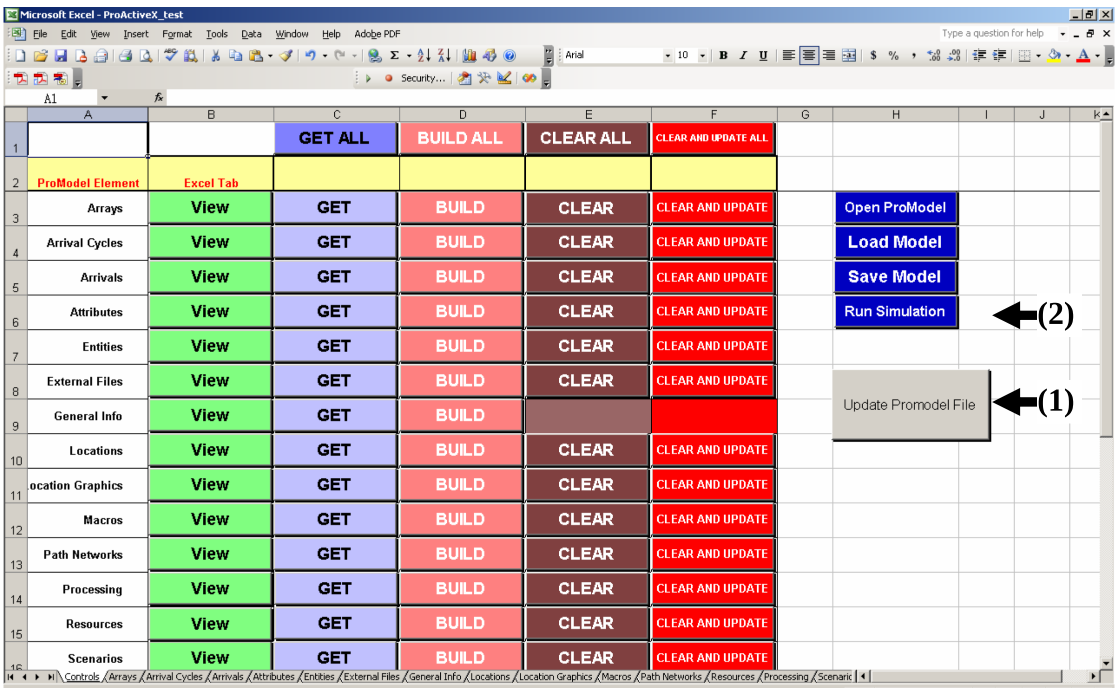


Figure 10. The ActiveX module in the new development of QN-MHP 2.0 (users of the modeling tool only need to click on the "Update Promodel File" (1) and "Run Simulation" button (2) on the "Control" sheet)

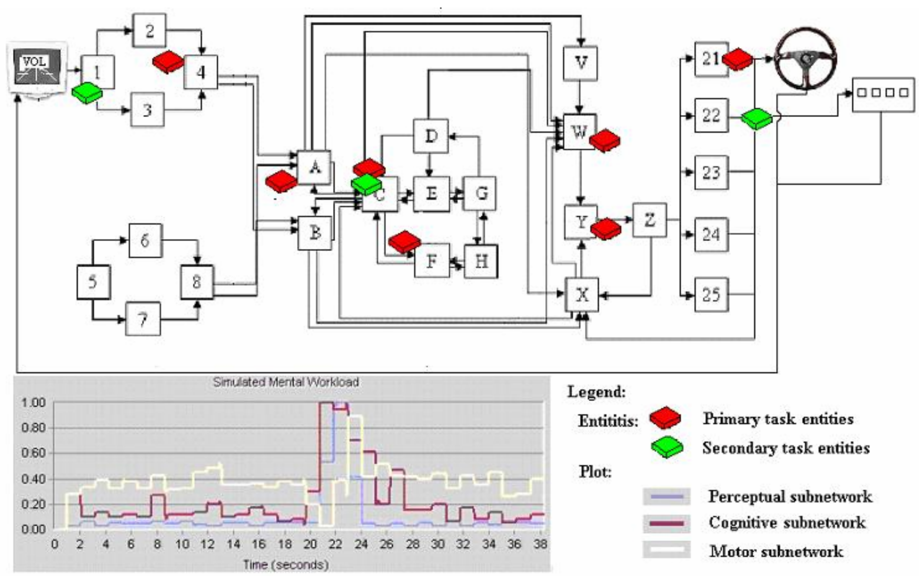


Figure 11. Dynamic change of subnetwork utilization as an index of mental workload during the Promodel simulation (see a short move clip on the website: http:www.acsu.buffalo.edu/changxu/) (Wu & Liu, 2007)

Figures 12-15 show the sample simulation results in Promodel and results in Excel imported from the Promodel's simulation results. Users of the modeling tool not only can compare the human performance and mental workload in different designs of the prototypes (see Figure 12), but also modify the design based on the simulation results of subnetworks and servers' utilization (Figures 13 and 14) and the state of entities during simulation (see Figure 15). For example, Figure 12 (a-d) suggests that Design 1 is the best compared to Designs 2 and 3 in terms of lane deviation in steering (Task 1), reaction time of Task 2, as well as mental workload. In addition, based on Figure 13, users of the modeling tool can also observe at which time point mental workload may exceed a certain "red line" so that they can modify the design or consider some intelligent adaptive user interface to prevent the occurrence of the extremely high workload. Users of the modeling tool can also evaluate designs to balance the workload of certain servers (e.g., right and left hand) based on the simulation results shown in Figure 14. Users can modify the arrival frequency of the task information by reducing the information presentation speed (e.g., a display presenting messages to a driver) to reduce the percentage of entities being blocked or waiting based on Figure 15.

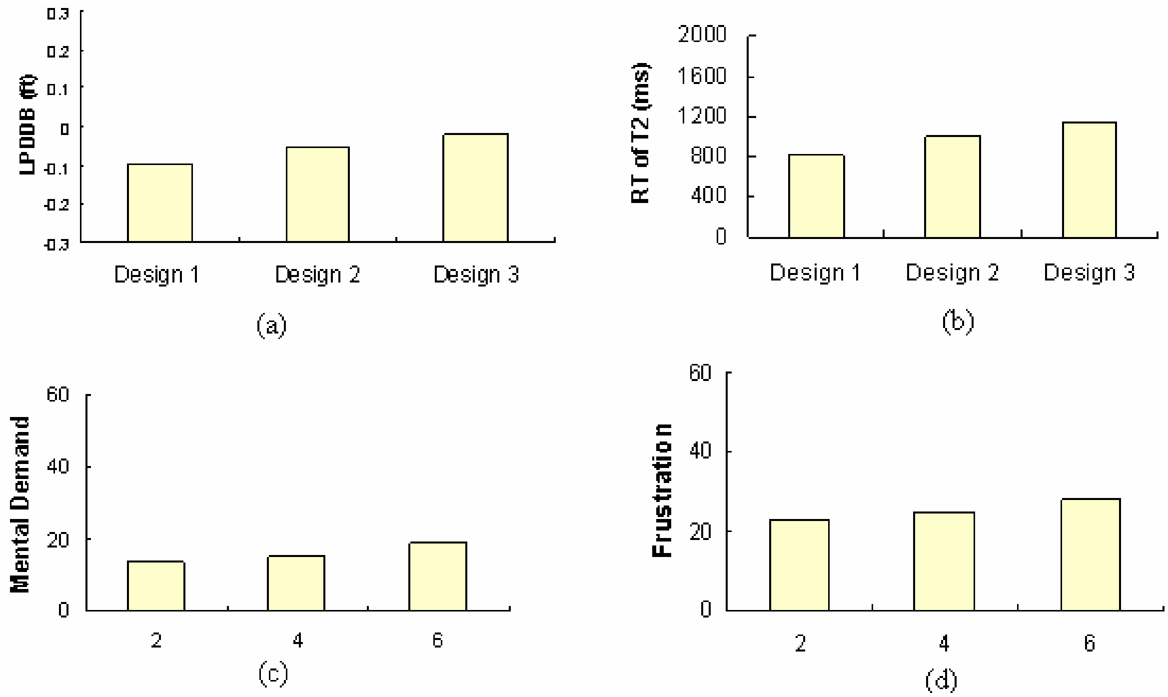


Figure 12. Sample simulation results of human performance in Excel imported from the Promodel's simulation results (a: LPDDB (lane position deviation difference from the baseline); b: RT of T2 (reaction time of the secondary task); c and d: two indexes of mental workload in NASA-TLX scale: mental demand and frustration) in comparing three different designs of the in-vehicle systems (see detailed simulation results of the other four index of mental workload in NASA-TLX scale and comparison between the simulation and experimental results in Wu & Liu, 2006b, 2006c)

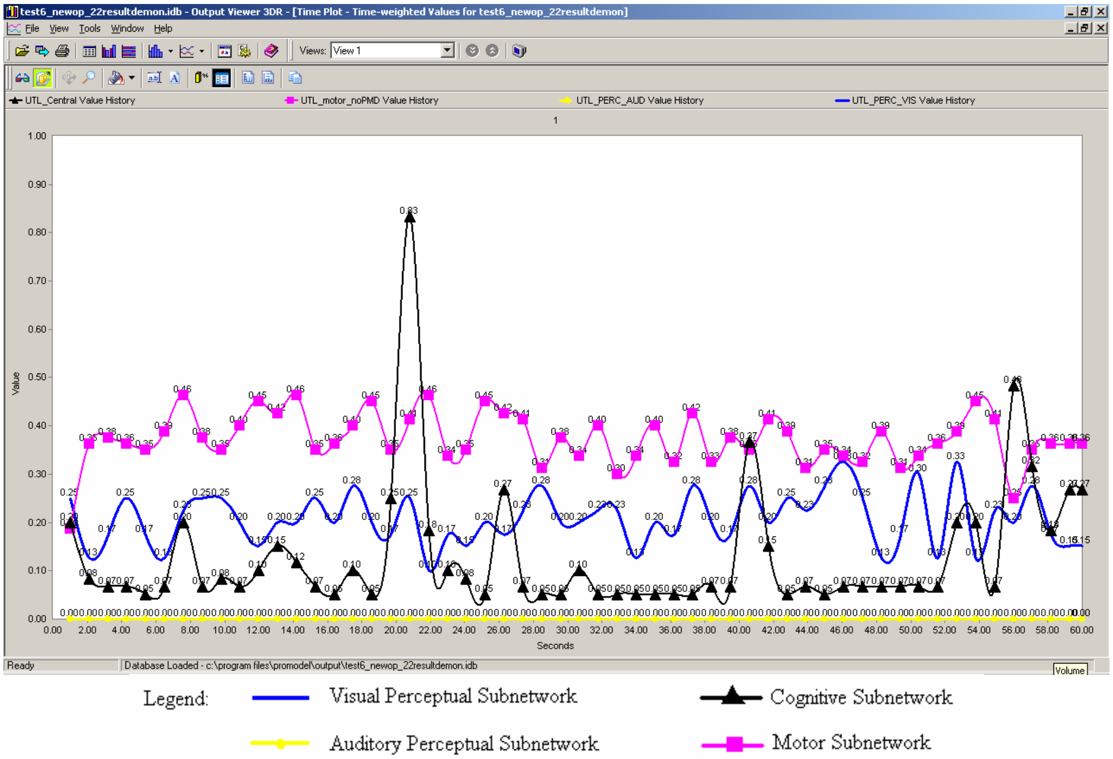


Figure 13. Sample simulation results in Promodel of subnetwork's utilization (X-axis is the simulation time and Y-axis is the utilization of the four subnetworks)

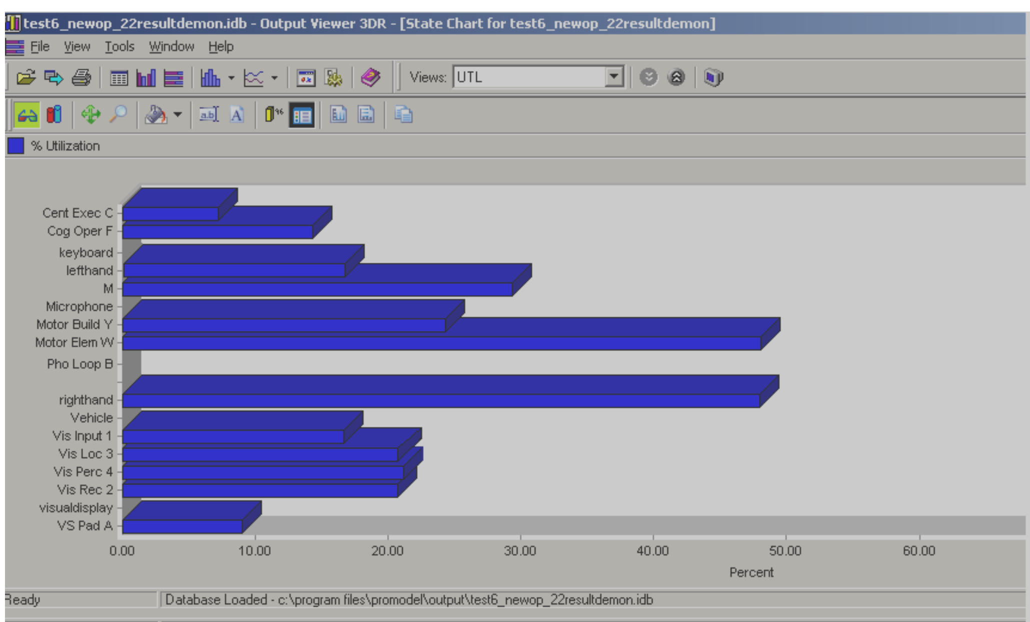


Figure 14. Sample simulation results of the averaged utilizations of servers in the queueing network (X-axis is the averaged utilization of servers and Y-axis is the name of servers described in Figure 1)

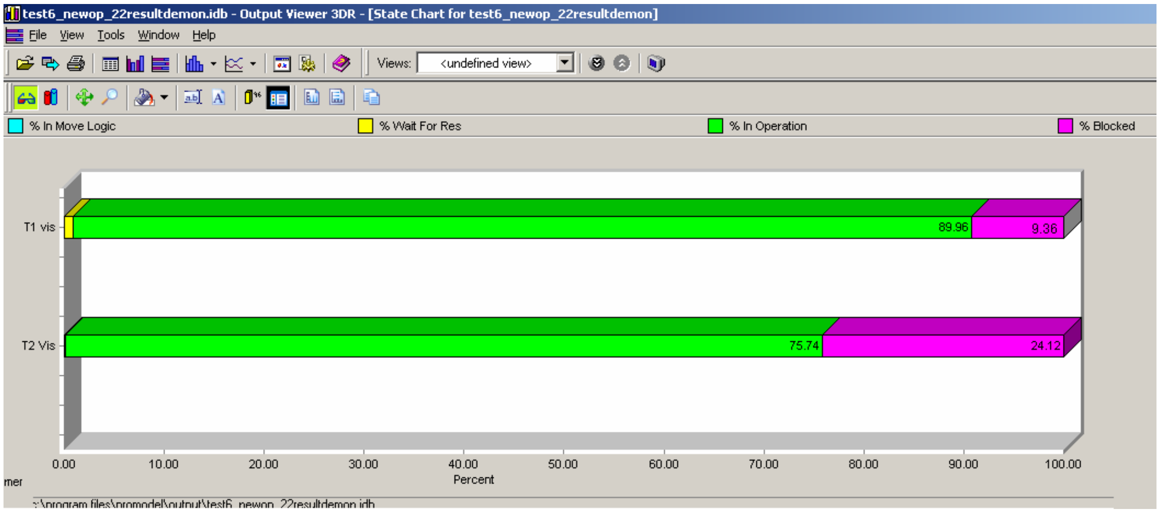


Figure 15. Sample simulation results of entities' activity in terms of percentage in operation, blocked and waiting for resource (T1_vis: entity of Task 1 perceived via visual perceptual subnetwork; T2_vis: entity of Task 2 perceived via visual perceptual subnetwork)

Overall, users of the modeling tool step through the VBA software package (a VBA user interface and an ActiveX module) by simply clicking buttons and selection options, or filling some text in the table, with no need to learn a new programming language. Users can intuitively observe the information processing state of the model during simulation, and easily compare the simulated human performance and mental workload for different designs. Moreover, the VBA software package was developed using one of the most popular rapid user interface prototype tools (VBA in Excel) and users of the modeling tool can easily connect this modeling tool with the prototype of the interface mock-ups.

3. An Illustrative Case Study

An experimental study, as an illustrative case study of the VBA software package of QN-MHP, has been conducted to evaluate its effectiveness in helping users of the modeling tool perform a modeling task.

3.1 Tasks to be Modeled

Two sample tasks that have been modeled by researchers of QN-MHP 2.0 were used in this case study: a dual task of visual-manual tracking and tone counting (Wu & Liu, 2006a) and a dual task of steering and visual stimuli-key pressing (Wu & Liu, 2007). The first dual task is the task employed in the experimental study of Wickens, et al., (1983), in which the primary task of the subjects was to manipulate a joystick and attempt to superimpose a cursor on a target which was moving in a series of discrete horizontal displacement on a visual display; in the secondary task, subjects in that experiment were instructed to count the number of occurrences of low-pitched tones in a series of tones of high or low pitch. For the second dual task, in the primary vehicle steering task, subjects in the experimental study (Feyen and Liu, 1998) were asked to keep the vehicle in control and maintain the speed and lane position; in the secondary button-pressing task, subjects were instructed to press one of the buttons on a panel mounted on the right side of the steering wheel when they saw a command presented on the display.

3.2 Subjects, Experimental Procedure and Variables

8 graduate students (4 male and 4 female, 20-30 years old，M=25.7 years) who did not have prior simulation language programming experience participated the current experimental study. These subjects never used the current modeling tool or the original QN-MHP before. Subjects were paid $10 per hour in the experiment as compensation for their participation.

There are two experimental groups in the case study defined according to which group uses the modeling tool first: Group 1 uses modeling tool first and then uses the original QN-MHP 2.0 without the VBA software package to model the dual tasks; in Group 2, this order is reversed. For Group 1, after a 10-minute brief introduction of the current modeling tool including how to step through the VBA user interface and how to define the task and interface information, subjects were asked to model the two dual tasks introduced above (the order of these two dual tasks are balanced across different subjects). After that, an experimenter gave a 10-minute brief introduction to the subjects in Group 1 on how to perform multiple task modeling with the original QN-MHP 2.0 without using the VBA software package and subjects were asked to model the same tasks again only using the Excel files and Promodel file. Finally, subjects were asked to evaluate the two ways of modeling (modeling with or without the VBA software package) in terms of its ease of learning and stepping through the software in the modeling process on a 7-point Likert scale (7: extremely easy; 4: neural; 1: extremely difficult).

The independent variable in this case study is the two ways of modeling: using the new developed VBA software package of QN-MHP 2.0 (with the VBA software package) vs. using the original QN-MHP 2.0 (without the VBA software package). The dependent variable is the average time to model a dual task, percentage of correct steps (measured by the number of correct steps in using the software package divided by the total number of steps), and average rating scores in the subjective evaluation. The correct steps were obtained from a pilot study that as long as users of the tool follow these steps the model will generate the simulated human performance and mental workload which were consistent with the experimental data (Wu & Liu, 2006a, 2007).

3.3 Experimental Results

Table 3 summarizes the experimental results in comparing the subjects' performance and their subjective evaluations in modeling tasks with or without using the VBA software package.

Table 3. Summery of Experimental Results

Measurements	Without VBA	With VBA
Modeling Time of a Dual Task (sec)	167 (48)*	436 (93)
Percentage of Correct Steps in Modeling	92%	62%
Ease of Learning (Subjective Evaluation)	2.2 (.98)	6.2 (.75)
Ease of Stepping Through the Software (Subjective Evaluation)	1.8 (.75)	6.4(.49)

* SD is shown in the parenthesis

Figure 16 shows the average time of modeling the dual tasks with or without using the VBA software package. Subjects who used the software package saved about 71% of modeling time: the average time to model a dual task was 2 minutes 47 seconds (SD=48 sec) when using the VBA software package, but was 7 minute 16 seconds (SD=93 sec) when using the original QN-MHP. Mann-Whitney U test, a nonparametric test equivalent to the t test and suitable for small sample size (n<30), found that there was a significant difference in the modeling time between using or not using the software package (Z=-2.884, asymmetric significance (2-tailed) =.004<.05).

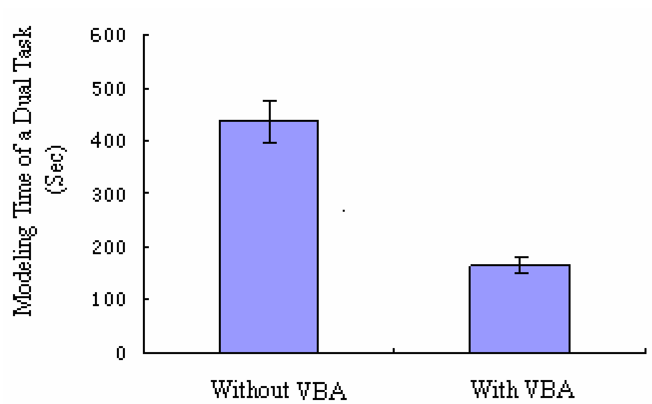


Figure 16. Average time of modeling a dual task in the experiment (not using vs. using the VBA software package of QN-MHP 2.0). Vertical bars show ± 1 standard error

The average percentage of correct steps of the subjects in modeling the dual tasks is shown in Figure 17. The use of the VBA software package in performing the modeling task significantly increased the percentage of correct steps from 62% to 92% on average (Mann-Whitney U test, Z=-2.912, asymmetric significance (2-tailed) =.004<.05).

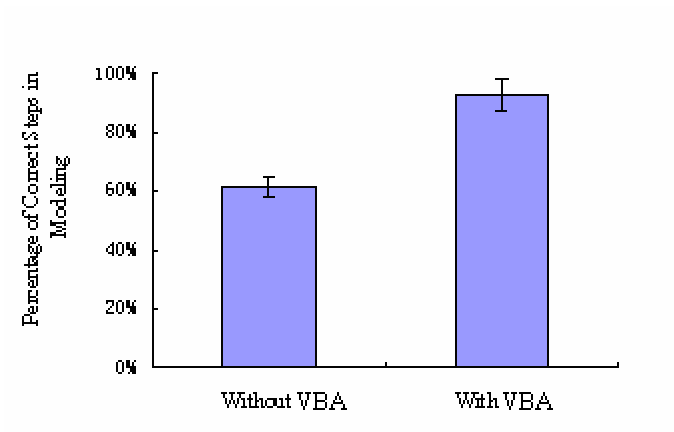


Figure 17. Average percentage of correct steps of modeling a dual task in the experiment (not using vs. using the VBA software package of QN-MHP 2.0). Vertical bars show ± 1 standard error

Figure 18 shows the subjective evaluations of the two ways of modeling in terms of ease of learning and stepping through the software in modeling. In each index, the average score in using the VBA software package was significantly higher than that without using the package (ease of learning: Mann-Whitney U test, Z=-2.916, asymmetric significance (2-tailed) =.004<.05; ease of stepping through the software: Mann-Whitney U test, Z=-2.917, asymmetric significance (2-tailed) =.004<.05)

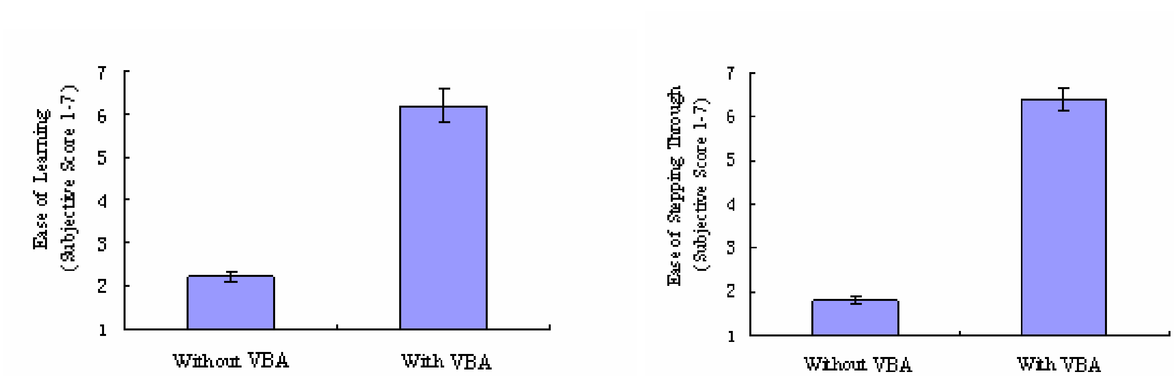


Figure 18. Average scores of subjective evaluations in terms of ease of learning and stepping through the software (not using vs. using the VBA software package of QN-MHP 2.0). Vertical bars show ± 1 standard error

4. Discussion

This paper describes the development of VBA software package with an easy-to-use user interface for QN-MHP 2.0 which is able to help users of the modeling tool define the tasks and interfaces in multitask systems easily by clicking and typing rather than learning a new programming language to code this information. The experimental results in the case study indicate that using this software package significantly reduced the modeling time and errors of the users in simulating human performance and mental workload in multitask situations.

Users of the new software package do not need to learn any new programming language to model multiple tasks. They only need to design the UI mock-ups in a graphic format, store them as images in the computer, and then choose the corresponding actions/operators to demonstrate the task in the VBA software package by clicking and typing, decreasing their working memory load in translating the operators/actions into numerical codes and reducing their errors in manually inputting the codes in Excel file as well as in coding the Promodel file. The current experimental study found that after a 10-minute brief introduction of the software, the naïve users who have never used this modeling tool before can model human performance and mental workload in a complex multitask situation within 3 minutes. The efficiency and ease of use in modeling the task via this new software package is also confirmed by high subjective evaluation scores after usage (We also checked the order effect—subjects who used the VBA package first vs. subjects who used the original package without the VBA package first, and found no significant difference in performance and subjective evaluation between these two groups). In addition, the new software package of QN-MHP is developed using VBA—a widespread UI rapid design tool which makes further development of the interface easier and more compatible with other commonly used software development tools.

The current work demonstrates the value of the easy-to-use modeling tool of QN-MHP, and we are exploring several important further developments for the current modeling tool. It is promising to connect the Visual Basic Application with other Microsoft Office software and web design tools. Most of the commonly used software developed by Microsoft® Cooperation is able to use Visual Basic as well as its applications including Microsoft Visual Studio Family (Visual Basic.net, Visual C++.net, Visual J#.net, Visual C#.net), Microsoft Word, PowerPoint, Excel, Access, Outlook and more importantly FrontPage. For example, once users of the modeling tool create their web pages in FrontPage, a VBA program can automatically transfer the users' behavior and web pages into VBA user interface of QN-MHP so that users of the modeling tool can directly demonstrate the task in the FrontPage. Another important aspect of the software currently being developed is the state-transition diagram, where the state of objects on the user interface is a node and the actions defined by users are the transitions between the nodes. In addition, we are adding additional features on the VBA software package so that users of the tool are able to conveniently define the properties of target users of the UI mock-ups (e.g., age, perceptual motor speed, mental operation speed, handedness) as well as the conditions of the target environment where the UI mock-ups are used (e.g., lighting condition, noise level, vibration level etc.). Finally, a related research work is being conducted to connect the QN-MHP with the human motion models including JACK and 3DSSPP/AutoCAD (Feyen, Liu, Chaffin, Jimmerson, & Joseph, 2000; Reed & Tsimhoni, 2006). These developments will further enhance the capabilities of the combined models to make comprehensive predictions of human performance, human movement, mental and physical workload.

We are developing the modeling tool to make the human performance and mental workload modeling easier for engineers, user interface designers and even managers, so that they can efficiently compare and improve system prototypes at an early stage of system development. Our comprehensive computational model of human performance—QN-MHP, offers not only theoretical insights to human performance and mental workload, but is a step toward a comprehensive proactive ergonomic design and analysis tool for user interface design.

Acknowledgments

This article is based upon work supported by the National Science Foundation. However, any opinions, findings and conclusions or recommendations expressed in this article are those of the authors and do not necessarily reflect the views of the National Science Foundation (NSF).

Reference

Bear, M. F., Connors, B. W., & Paradiso, M. A. (2001). *Neuroscience: exploring the brain* (8th ed.). Baltimore, MD: Lippincott Williams & Wilkins,.

Card, S. K., Moran, T. P., & Newell, A. (1980). The keystroke-level model for user performance time with interactive systems. *Communications of the ACM, 23*(7), 396-410.

Faw, B. (2003). Pre-frontal executive committee for perception, working memory, attention, long-term memory, motor control, and thinking: A tutorial review. *Consciousness and Cognition, 12*(1), 83-139.

Feyen, R. (2002). *Modeling Human Performance using the Queuing Network-Model Human Processor (QN-MHP).* University of Michigan, Ann Arbor.

Feyen, R., Liu, Y., Chaffin, D., Jimmerson, G., & Joseh, B. (1999). New software tools improve workplace design. *Ergonomics in Design,* 24-30.

Feyen, R., Liu, Y., Chaffin, D., Jimmerson, G., & Joseph, B. (2000). Computer aided ergonomics: A case study of incorporating ergonomics analysis into workplace design. *Applied Ergonomics, 31*, 291-300.

Freed, M., Matessa, M., Remington, R., & Vera, A. (2003). *How Apex Automates CPM-GOMS.* Paper presented at the International Conference on Cognitive Modeling 2003.

Freed, M. A., & Remington, R. (2000). *Making human-machine system simulation a practical engineering tool: An APEX overview.* Paper presented at the Proceedings of the 3rd International Conference on Cognitive Modeling, Veenendaal (NL).

Gore, B. F. (2000). *The study of distributed cognition in free flight: a human performance modeling tool structural comparison.* Paper presented at the SAE Digital Human Modeling for Design and Engineering International Conference and Exposition, Dearborn, MI.

Gore, B. F. (2002). Human performance cognitive-behavioral modeling: A benefit for occupational safety. *International Journal of Occupational Safety and Ergonomics 8*(3), 339-351.

Hudson, S. E., & Stasko, J. (1995). *Animation Support in User Interface Toolkits: Flexible, Robust, and Reusable Abstractions.* Paper presented at the Proceedings of the ACM Symposium on User Interface Software and Technology.

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

Kuo, C., & Chu, C. (2005). An online ergonmic evaluator for 3D product design. *Computers in Industry, 56*, 479-492.

Laughery, K. (1989). Micro SAINT: A tool for modeling human performance in systems. In G. R. McMillan, D. Beevis & E. Salas (Eds.), *Applications of Human Performance Models to System Design* (pp. 219-230). New York: Plenum.

Liu, Y. (1996). Queuing network modeling of elementary mental processes. *Psychological Review, 103*(1), 116-136.

Liu, Y. (1997). Queuing network modeling of human performance of concurrent spatial and verbal tasks. *IEEE Transactions on Systems Man and Cybernetics Part a-Systems and Humans, 27*(2), 195-207.

Liu, Y., Feyen, R., & Tsimhoni, O. (2006). Queueing Network-Model Human Processor (QN-MHP): A Computational Architecture for Multitask Performance. *ACM Transaction on Human Computer Interaction, 13*(1).

Magrab, E. B., & E.B., M. (1997). *Integrated Product and Process Design and Development*. Boca Raton, FL, USA: Taylor & Francis CRC Press LLC.

Newell, A. (1990). *Unified theories of cognition*. Cambridge, MA: Harvard University Press.

Olsen, J. R., & Olsen, G. M. (1990). The Growth of Cognitive Modeling in Human-Computer Interaction Since GOMS. *Human-Computer Interaction, 5*(2&3), 221-265

Promodel. (2003). Promodel User Guide (Version 6): Promodel Cooperation.

Reed, M., & Tsimhoni, O. (2006, May 23-24). *The Virtual Driver: Integrating Cognitive and Physical Models of Driver Performance.* . Paper presented at the 12th Annual Army Research Center Conference, Ann Arbor, MI.

Resnick, M. L., & Zanotti, A. (1997). Using ergonomics to target productivity improvements. *Computers & Industrial Engineering, 33*, 185-188.

Schunk, D. (2000). *Modeling with the Micro Saint Simulation Package.* Paper presented at the Proceeding of the 2000 Winter Simulation Conference.

Shidar, A., SAl-Araimi, S., & Omurtag, B. (2002). Development of a software package for ergonomics assessment of manufacturing industry. *Computers & Industrial Engineering, 43*, 485-493.

Smith, E. E., & Jonides, J. (1998). Neuroimaging analyses of human working memory. *Proc. Natl. Acad. Sci. USA, 95*, 12061-12068.

Williams, K. E., Hultman, E., & Graesser, A. C. (1998). CAT: A Tool for Eliciting Knowledge on How to Perform Procedures *Behavior Research Methods, Instruments, & Computers 30*(4), 565 - 575.

Wu, C., & Liu, Y. (2004a). *Modeling Behavioral and Brain Imaging Phenomena in Transcription Typing with Queuing Networks and Reinforcement Learning Algorithms.* Paper presented at the Proceedings of the 6th International Conference on Cognitive Modeling (ICCM-2004), Pittsburgh, PA, USA. 314-319.

Wu, C., & Liu, Y. (2004b). *Modeling human transcription typing with queuing network-model human processor.* Paper presented at the Proceedings of the 48th Annual Meeting of Human Factors and Ergonomics Society, New Orleans, Louisiana, USA. 381-385.

Wu, C., & Liu, Y. (2004c). *Modeling Psychological Refractory Period (PRP) and Practice Effect on PRP with Queueing Networks and Reinforcement Learning Algorithms.* Paper presented at the Proceedings of the 6th International Conference on Cognitive Modeling (ICCM-2004), Pittsburgh, PA, USA. 320-325.

Wu, C., & Liu, Y. (2006a). *Queuing Network Modeling of a Real-time Psychophysiological Index of Mental Workload—P300 Amplitude in Event-Related Potential (ERP).* Paper presented at the 50th Annual Conference of the Human Factors and Ergonomics Society, San Francisco, CA, USA.

Wu, C., & Liu, Y. (2006b). *Queuing Network Modeling of Age Differences in Driver Mental Workload and Performance.* Paper presented at the 50th Annual Conference of the Human Factors and Ergonomics Society, San Francisco, CA, USA.

Wu, C., & Liu, Y. (2006c). *Queuing Network Modeling of Driver Workload and Performance.* Paper presented at the 50th Annual Conference of the Human Factors and Ergonomics Society, San Francisco, CA, USA.

Wu, C., & Liu, Y. (2007). Queuing Network Modeling of Driver Workload and Performance. *IEEE Transactions on Intelligent Transportation System*. vol 8 (3), 528-537

Wu, C., & Liu, Y. (2008). Queueing Network Modeling of Transcription Typing. *ACM Transactions on Computer-Human Interaction (Accepted with minor revision), 15*(1), In Press.

Wu, C., & Liu, Y. (In Press). Queuing Network Modeling of Psychological Refractory Peroid. *Psychological Review*.

Wu, C., Liu, Y., & Walsh, C. (2007). Queuing Network Modeling of a Real-time Psychophysiological Index of Mental Workload--P300 in Event-Related Potential (ERP). *IEEE Transactions on Systems, Man, and Cybernetics (Part A)*, In Press.

Wu, C., Tsimhoni, O., & Liu, Y. (In Press). Development of an Adaptive Workload Management System using Queueing Network-Model of Human Processor. *IEEE Transactions on Intelligent Transportation System*

Yoon, W. C., & Kim, Y. S. (1996). Aiding he analysis of human actions in large-scale systems: an intelligent interface approach. *Computers & Industrial Engineering, 30*, 569-577.