Proceedings of the 2002 Winter Simulation Conference
E. Yücesan, C.-H. Chen, J. L. Snowdon, and J. M. Charnes, eds.

MICRO SAINT MODELING AND THE HUMAN ELEMENT

Daniel W. Schunk
Wendy K. Bloechle
Ron Laughery

Micro Analysis and Design, Inc.
4949 Pearl East Circle
Boulder, CO 80301, U.S.A.

ABSTRACT

With new technologies emerging daily, one of the largest problems companies currently face is the challenge of upgrading slow, outdated systems. While it is common to consider things like cost efficiency and automation in new systems, companies often overlook the human element in their system redesign. By factoring in the human element, companies can avoid having to make costly adjustments to their system because of unexpected human error. Task network modeling is one approach to modeling human performance in complex systems. Micro Saint is a modeling environment that supports task network modeling and human performance modeling. This paper will discuss how to model the human element using Micro Saint along with a brief case study.

1 INTRODUCTION

Human performance is often a high-risk element in the overall operational effectiveness of many types of systems. For example, approximately two thirds of aircraft accidents are now attributed to pilot error. Unfortunately, the traditional design process tends to put a disproportionate focus on the technical performance of equipment, with little regard for the human component.

In the past, it has been difficult to integrate human performance models within system performance models, because of the complexity of human behavior and the lack of computational power to address the variability in human performance. The techniques that have traditionally been used to examine the human performance issues have largely been manual and laborious in nature. However, modern tools and methods facilitate the transfer of this information in a format compatible with other system computer models and simulations. This provides a golden opportunity to ensure that problems associated with human performance are identified early in the design process to prevent costly changes and procurement delays.

Over the past few decades, tools and techniques for the modeling of human performance in systems have evolved and matured. These tools are now at a state of maturity where they could be effectively integrated into the systems engineering process. Outlining how human performance modeling fits into the world of computer simulation is the purpose of this paper.

2 A DISCRETE-EVENT SIMULATION APPROACH FOR MODELING HUMAN PERFORMANCE

Task network modeling is a discrete-event simulation approach to modeling human performance in complex systems that has evolved for several reasons. First, it is a reasonable means for extending the human factors engineering staple – the task analysis. Task analyses organized by sequence are the basis for the task network model. Second, in addition to complex operator models, task network models can include sophisticated submodels of the equipment hardware and software to create a closed-loop representation of the human/machine system. Third, task network modeling is relatively easy to use and understand. Finally, task network modeling can provide reasonable input to many types of issues. With a task network model, the human factors engineer can examine a design (e.g., control panel redesign) and address questions such as "How much longer will it take to perform this procedure?" and "Will there be an increase in the error rate?" Generally, task network models can be developed in less time and with substantially less effort than would be required if a prototype were developed and human subjects used.

2.1 Task Network Modeling Fundamentals

Task network modeling involves the extension of a task analysis into a predictive model based around a network representation of the human's activity. The basic ingredient of a Micro Saint task network model is the task analysis as represented by a network or series of networks. The performance of the tasks can be interrelated through shared variables. The relationships among different components of the system (which are represented by different segments of the network) can then communicate through changes in these shared variables. For example, when an operator enters a command on a keyboard, this may initiate change in computer state or the information that is presented on an operator display. This task network is built in Micro Saint via a point and click drawing palette. Through this environment, the user creates a network as shown in Figure 1.

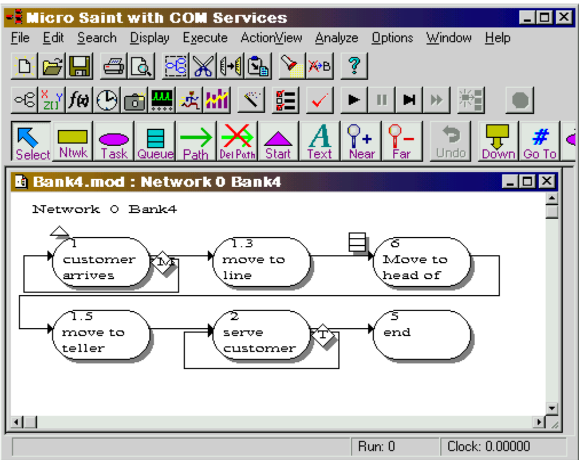

Figure 1: Micro Saint Task Network

To reflect complex task behavior and interrelationships, more detailed characteristics of the tasks can be defined. By pointing and double clicking on a task, the user opens up the Task Description Window as shown in Figure 2 whereby information describing task behavior and linkage with other system elements can be defined.

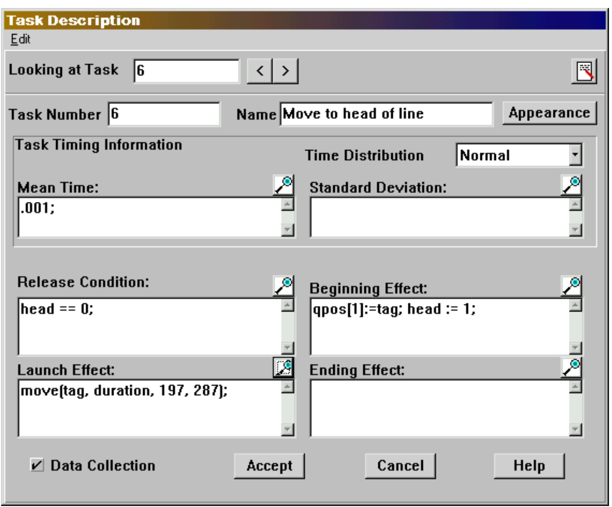

Figure 2: Task Description Box

Another notable aspect of the Task Network Diagram Window shown in Figure 1 is the diamond-shaped icons that follow some tasks. These are present every time more than one path out of a task is defined. In Micro Saint, these decision diamonds can be one of three types: tactical, probabilistic, and multiple. Multiple decisions mean that every path leaving the task will be taken. Tactical decision mean that based on logic only one path will be taken. Probabilistic decisions mean that only one path will be taken based on probabilities. Implicitly, this means that a decision must be made by the human to select which of the following potential courses of action should be followed. By opening a window into the decision, decision logic and algorithms can be developed to any level of complexity. From these basic building blocks, a task network model can be built to describe human and system behaviors of any size or level of complexity.

3 A CASE STUDY THAT EVALUATES CREW SIZE ON A HELICOPTER

3.1 Problem Explanation

Below is an example of a simulation model for a helicopter crew. It was developed to determine the impact of a reduced crew on the performance of a limited attack mission.

3.2 Model Explanation

The model being used for this analysis is a discrete event simulation model developed using Micro Saint simulation software. The model is based upon an attack helicopter in a limited attack mission. In this mission, the helicopter is on patrol and is attacked by a number of threats. Originally, the helicopter was designed to have both a pilot and gunner so that if a threat is present, the gunner can perform all the necessary engagement duties while the pilot controls the helicopter. The question this model is trying to answer: can having only one crewmember present be sufficient to perform all the necessary tasks in an engagement situation?

The model was developed so that if a gunner is available then the gunner could perform many of the engagement and communication tasks. If there is no gunner available, the pilot has to perform all the tasks. An assumption of the model is that the crewmembers can only perform one task at a time.

3.3 Micro Saint Model Development

3.3.1 Develop Task Flow Diagram

In this model, there are two general types of tasks that can occur. Many of these tasks in the model will occur probabilistically. The first type of task is called a “control” task. These are the tasks the pilot would perform in order to control the helicopter. These tasks are composed of very small actions that will occur over and over again in very small increments in an unprescribed order. These tasks include: controlling altitude, controlling airspeed, controlling heading, maintaining obstacle clearance, monitoring time, checking weapons and controlling drift. These tasks can be grouped into a network. (see Figure 3) During model execution, this network could spin around itself in order to accurately simulate the pilot exercising judgment in a scenario. This might cause the results of each individual scenario to be different.

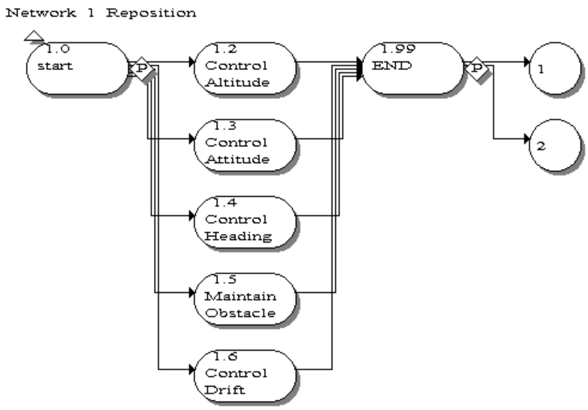

Figure 3: Reposition Network

Another set of controls is the responsibility of the gunner (if present). If the gunner is not present in the scenario, then these responsibilities are the pilot's as well. These gunner controls are broken out into 4 different networks: receive target information, select/prepare weapon, acquire target, and engage target. The receive target information network (see Figure 4) consists of tasks the gunner/pilot goes through in receiving target information. The select/prepare weapon networks (see Figure 5) are the tasks required for selecting and preparing the appropriate and available weapons for the target present.

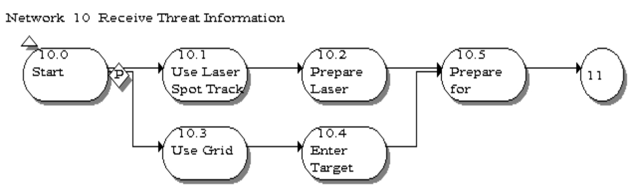

Figure 4: Receive Target Information Network

Other tasks and networks inside the model include tasks that define the steps required for acquiring the target using the helicopter imaging equipment, the tasks needed for the helicopter to engage and destroy the target and tasks that include whether an enemy threat will attack the helicopter as well as what happens after the helicopter is attacked (helicopter destroyed, damaged, threat attack misses).

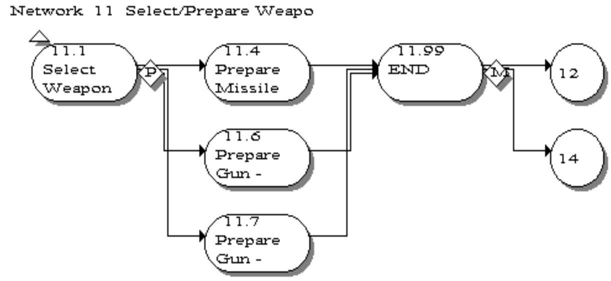

Figure 5: Select/Prepare Weapon Network

3.3.2 Define Decision Nodes

In the model, all decisions nodes in dealing with the helicopter are probabilistic decisions. This is to simulate the randomness of controlling a helicopter. Some other nodes are also probabilistic, this includes the actions of the enemy as well as how the helicopter will react when hit (helicopter destroyed or still functional). This is to help randomize the environment the pilot is functioning in, in order to show when and when not another crewmember would be required.

The tactical decision node is used when an attack is calculated to hit as well as when an enemy might attack again. Finally the multiple decision node is utilized when more than one task is needed to be performed in the helicopter at same time. If the helicopter has a pilot and gunner, these tasks will be completed at the same time. If there is only a pilot then the tasks will be separated out causing time to be lost and possibly causing the helicopter to be destroyed by the enemy.

3.3.3 Variables

The model analyzes five areas of this scenario: pilot utilization, gunner utilization, number of hits the aircraft takes, number of crashes, number of available targets per scenario, and the simulation time for each scenario run. These data variables are defined as:

a) Crash – flag for if the helicopter crashed in this scenario.
b) Tgtavail – number of targets available for this scenario
c) Clock – system variable for simulation time
d) Hit – number of hits helicopter takes per scenario.

Variables should be created also for the pilot and the gunner. There should also be seven flag variables created (flag1, flag2, etc.) to mark that various tasks have been completed as well as four path flag variables (path1, path2, path3, path4) to flag which path to take during tactical decisions.

Finally the following variables need to be created for various record keeping activities:

a) Cutofftime – time at which the probability of a hit increases
b) Operator – indicates which crewmember is busy
c) probhit – probability of enemy hitting helicopter
d) xpos – X position of helicopter
e) ypos – Y position of helicopter

3.3.4 Model Logic

In order to get meaningful outputs, the next area to develop is the model logic. The underlying rule of this model is that the crewmembers of the helicopter can only perform one job at a time. This means for each human, each task that is being performed, in order for that task to start a qualified crewmember must be available.

If the task is one of controlling the helicopter (altitude, drift, etc.), then the pilot is the only crewmember qualified to perform this task. The appropriate coding techniques are shown in Figure 6. If the task deals with the helicopter engagement, then it is the gunner's task, unless there is no gunner present, then it is also the pilots task. Logic must be placed in the task to check to see if there is a gunner and to use the gunner whenever appropriate. This is shown in Figure 7.

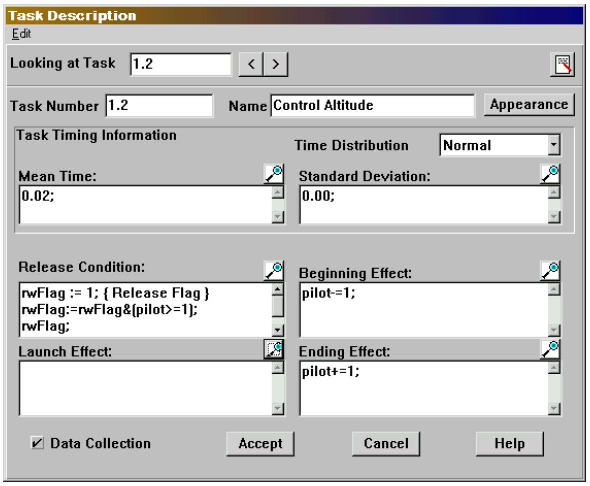

Figure 6: Pilot Only Task

The rest of the model inputs consist of simple timing data as well as logic to dictate which paths to take or to have the simulation model not continue until all previous tasks have been completed.

3.3.5 Data Inputs

There are two events that occur at the start of the model. The first one is used to identify whether a gunner is

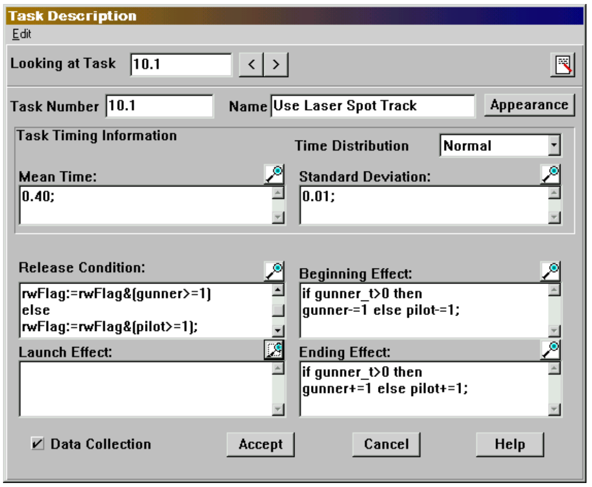

Figure 7: Gunner or Pilot Task

available. The variable "gunner_t" sets the number of gunners available, so if the user sets gunner_t := 0, the pilot will do all the tasks in the network. The second event establishes the value of the "cuttofftime" variable. If the engagement has not begun by this time, then the probability of the threat destroying the helicopter is increased.

3.3.6 Data Outputs

The model collects two types of outputs: crewmember utilization and system performance. The utilization outputs are in the gunner.rsc and pilot.rsc files. When gunner_t is set to 0, gunner.rsc shows zero utilization and pilot.rsc shows 100% utilization. This indicates that the resource allocation portion of the model is running as designed. The system performance output can be found in the ah64.res file. This file indicates the total length of the run, the number of times the aircraft was hit, whether it was destroyed, and the number of threats we engaged. In cases where the aircraft was destroyed, the length of the run is truncated. This is because that run of the model is halted by the "halt()" command in the "A/C Not Operational" task. It is interesting to note that our aircraft will be destroyed more often if we cannot share tasks between the gunner and the pilot. That is because the pilot cannot always finish all the necessary tasks before the cut off time and the probability of the threat destroying him is increased.

With this finished example, the user can analyze the effects of decreased personnel in an attack helicopter as well as assess if the costs outweigh the benefits. This is a way for users to execute "what if" analyses without having to commit resources.

4 SUMMARY

Obviously, manpower is a crucial aspect of any task and operation. Accurately identifying when there is a need and when there is excess is a sure path to success. Task network modeling provides one way to analyze crewmember performance in the operating environment. Together with usability analysis to ensure model accuracy and systems analysis to define how the system will be used, a better assessment of the value of human performance can be gained.

AUTHOR BIOGRAPHIES

DANIEL W. SCHUNK is an industrial engineer for Micro Analysis and Design, Inc. He has a Bachelor of Science in Industrial Engineering from Purdue University. His email address is <dschunk@maad.com>.

WENDY K. BLOECHLE is the Director of Sales and Marketing for Micro Analysis and Design, Inc. She has a Bachelor's of Science in Industrial Engineering from the University of Illinois and a Master of Business Administration from the University of Colorado. Her email address is <wbloechle@maad.com>.

RON LAUGHERY is the President and founder of Micro Analysis and Design, Inc., the developer of Micro Saint and other human performance modeling and simulation tools. He has a Ph.D. in Industrial Engineering at the State University of New York at Buffalo and worked in the aerospace industry until the establishment of Micro Analysis and Design. His email address is <rlaughery@maad.com>.