V.C
Computer Simulation

Handbook of Industrial Engineering: Technology and Operations Management, Third Edition.
Edited by Gavriel Salvendy

CHAPTER 93

Modeling Human Performance in Complex Systems

K. Ronald Laughery, Jr.
Susan Archer
Micro Analysis and Design, Inc.

Kevin Corker
San José State University

Handbook of Industrial Engineering: Technology and Operations Management, Third Edition.

1. INTRODUCTION

Over the past few decades, human factors and ergonomics practitioners have increasingly been called upon early in the system design and development process. Early input from all disciplines results in better and more integrated designs as well as lower costs than if one or more disciplines finds that changes are required later. Our goal as human factors and ergonomics practitioners should be to provide substantive and well-supported input regarding the human(s), his or her interaction(s) with the system, and the resulting total performance. Furthermore, we should be prepared to provide this input from the earliest stages of system concept development and then throughout the entire system or product life cycle.

To meet this challenge, many human factors and ergonomics tools and technologies have evolved over the years to support early analysis and design. Two specific types of technologies are design guidance (e.g., O'Hara et al. 1995; Boff et al. 1986) and high-fidelity rapid prototyping of user interfaces (e.g., Dahl et al. 1995). Design guidance technologies, in the form of either handbooks or computerized decision support systems, put selected portions of the human factors and ergonomics knowledge base at the fingertips of the designer, often in a form tailored to a particular problem such as nuclear power plant design or UNIX computer interface design. However, design guides have the shortcoming that they do not often provide methods for making quantitative trade-offs in *system* performance as a function of design. For example, design guides may tell us that a high-resolution color display will be better than a black-and-white display, and they may even tell us the value in terms of increased response time and reduced error rates. However, this type of guidance will rarely provide good insight into the value of this improved element of the human's performance to the *overall system's* performance. As such, design guidance has limited value for providing concrete input to system-level performance prediction.

Rapid prototyping, on the other hand, supports analysis of how a specific design and task allocation will affect human and system-level performance. The disadvantage of prototyping, as with all human-subjects experimentation, is that it can be costly. In particular, prototypes of hardware-based systems, such as aircraft and machinery, are very expensive to develop, particularly at early design stages when there are many widely divergent design concepts. In spite of the expense, hardware and software prototyping are important tools for the human factors practitioner, and their use is growing in virtually every application area.

While these technologies are valuable to the human factors practitioner, what is often needed is an integrating methodology that can extrapolate from the base of human factors and ergonomics data, as reflected in design guides and the literature, in order to support system-level performance predictions as a function of design alternatives. This methodology should also bind with rapid prototyping and experimentation in a mutually supportive and iterative way. As has become the case in many engineering disciplines, a prime candidate for this integrating methodology is computer modeling and simulation.

Computer modeling of human behavior and performance is not a new endeavor. Computer models of complex cognitive behavior have been around for over 20 years (e.g., Newell and Simon 1972) and tools for computer modeling of task-level performance have been available since the 1970s (e.g., Wortman et al. 1978). However, two things have changed appreciably in the past decade that promote the use of computer modeling and simulation of human performance as a standard tool for the practitioner. First is the rapid increase in computer power and the associated development of easier-to-use modeling tools. Individuals with an interest in predicting human performance through simulation can select from a variety of computer-based tools (for a comprehensive list of these tools, see McMillan et al. 1989). Second is the increased focus by the research community on the development of *predictive* models of human performance rather than simply descriptive models. For example, the GOMS model (Gray et al. 1993) represents the integration of research into a model for making predictions of how humans will perform in a realistic task environment. Another example is the research in cognitive workload that has been represented as computer algorithms (e.g., McCracken and Aldrich 1984; Farmer et al. 1995). Given a description of the tasks and equipment with which humans are engaged, these algorithms support assessment of when workload-related performance problems are likely to occur and often include identification of the quantitative impact of those problems on overall system performance (Hahler et al. 1991). These algorithms are particularly useful when embedded as key components in computer simulation models of the tasks and the environment.

Perhaps the most powerful aspect of computer modeling and simulation is that it provides a method through which the human factors and ergonomics team can step up to the table with the other engineering disciplines that also rely increasingly on quantitative computer models. What we will discuss in this chapter are the methods through which the human factors and ergonomics community can contribute early to system design tradeoff decisions.

2. OBJECTIVES OF THIS CHAPTER

This chapter will discuss some existing computer tools for modeling and simulating human/system performance. It is intended to provide the reader with:

1. An understanding of the types of human factors and ergonomics issues that can be addressed with modeling and simulation
2. An understanding of some of the tools that are now available to assist the human factors and ergonomics specialist in conducting model-based analyses
3. An appreciation of the level of expertise and effort that will be required to use these technologies

We begin this chapter with two caveats. First, we are not yet at a point where computer modeling of human behavior allows sufficiently accurate predictions that no other analysis method (e.g., prototyping) is likely to be needed. In early stages of system concept development, high-level modeling of human–system interaction may be all that is possible. As the system moves through the design process, human factors and ergonomics designers will often want to augment modeling and simulation predictions with prototyping and experimentation. In addition to providing high-fidelity system performance data, these data can be used to enhance and refine the models. This concept of human performance modeling supporting and being supported by experimentation with human subjects is represented in Figure 1. In essence, simulation provides the human factors and ergonomics practitioner with a means of extending the knowledge base of human factors and of amplifying the effectiveness of limited experimentation.

Second, the technologies discussed here are evolving rapidly. We can be certain that (1) every tool discussed is undergoing constant change and (2) new modeling tools are being developed. We are discussing computer-based tools, and we expect the pace of change in these tools to mirror the pace in other computer tools such as word processors and spreadsheets. These detailed discussions of several of the modeling tools are included to facilitate a better understanding of human performance modeling tools. We encourage the reader to contact citations in this chapter to assess the state of any tool at that time.

3. THE TYPES OF QUESTIONS THAT ARE BEING ADDRESSED BY HUMAN PERFORMANCE MODELS

Below are a few classes of problems to which human/system modeling has been applied:

- How long will it take a human or team of humans to perform a set of tasks as a function of system design, task allocation, and individual capabilities?
- What are the trade-offs in performance for different combinations of design, task allocation, and individual capability selections?
- What are the workload demands on the human as a function of system design and automation? How will human performance and resulting system performance change as the demands of the environment change? How many individuals are required on a team to ensure safe, successful performance? How should tasks be allocated to optimize performance?
- How will environmental stressors such as heat, cold, and the use of drugs affect human–system performance?

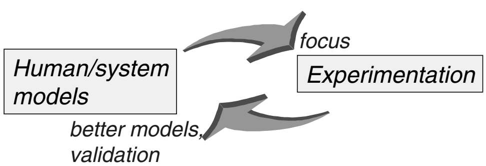

Figure 1 The Synergy between Modeling and Experimentation.

The above is a sampling rather than an exhaustive list. The tools we discuss in this chapter are inherently flexible, and we consistently discover that these tools can be used to solve problems that the tool developers never conceived.

To assess the potential of simulation to answer questions, every potential human performance modeling project should first determine the specific questions that the project is trying to answer. Then conduct a critical assessment of what is important in the human–machine system being modeled. This will define the required content and fidelity of the model. The questions that should be considered about the system include:

- *Human performance representation:* What time or duration of performance is important? How is human performance initiated and what resolution of behavior is required? What aspects of human performance, including task management, load management, and goal management, are expected?
- *Equipment representation:* What equipment is used to accomplish the tasks? To what level of functional and physical description can and should the equipment be represented? Is it operable by more than one human or system component?
- *Interface requirements:* What information needs to be conveyed to the humans and when? Is transformation of information required?
- *Control requirements:* What processes need to be controlled by the human and to what level of resolution?
- *Logical and physical constraints:* How is performance supported through equipment operability and procedural sequences? What alarms and alerts should be represented?
- *Simulation driver:* What makes the system function—the occurrence of well-defined events (e.g., a procedure), the movement of time (e.g., the control of a vehicle), or a hybrid of both?

By defining the purpose of the model and then answering the above questions, the human factors practitioner will get a sense of what is important in the system and therefore what may need to be represented in a model. In using human performance models, perhaps the most significant task of the human factors practitioner will be to determine what aspects of the human/machine system to include in the model and what to leave out. Many modeling studies have failed because of the inclusion of too many factors that, while a part of human–system performance, were not system performance drivers. Consequently, the models become overly complex and expensive to develop. In our experience, it is better to begin with a model with too few aspects of the system represented and then add to it than to begin a modeling project by trying to model everything. The first approach may succeed, while the second is often doomed.

Second, the human factors practitioner should consider the measures of effectiveness of the system that the model should be designed to predict. In building the model, it is important to remember that the goal of the model will be to predict measures of human performance that will impact system performance. Therefore, a clear definition of what is important to performance is necessary. The following aspects of performance measures should be considered:

- *Success criteria:* What operational success measures are important to the system? Can these be stated in relative terms, or must they be measured in absolute terms?
- *Range of performance to be studied:* What are the experimental variables that are to be explored by the model? How important is it to establish a range of performance for each experimental condition as a function of the stochastic (i.e., random) behavior of the system?

By asking the above questions prior to beginning a modeling project, the human factors practitioner can develop a better sense of what is important in the system in terms of both aspects that drive system performance as well as the measures of effectiveness that are truly of interest. Then and only then can a human performance modeling project begin with a reasonable hope of success.

The remainder of this chapter will discuss two classes of modeling tools for human performance simulation. After discussing each class of modeling tool, we will provide specific examples of a modeling tool and then provide case studies about how these tools have been used in answering real human performance questions.

4. THE TWO CLASSES OF SIMULATION MODELS OF HUMAN–SYSTEM PERFORMANCE

Human performance can be highly complex and involve many types of processes and behavior. Over the years, many models have been developed that predict sensory processes (e.g., Gawron et al. 1983), aspects of human cognition (e.g., Newell 1990), and human motor response (e.g., Fitts's law).

The current literature in the areas of cognitive engineering, error analysis, and human–computer interaction contains many models, descriptions, methodologies, metaphors, and functional analogies. However, in this chapter we are not focusing on the models of these individual elements of human behavior but rather on models that can be used to describe human performance in systems. These human/system performance models typically include some of these elemental behavioral models as components but provide a structural framework that allows them to be put in the context of human performance of tasks in systems.

We separate the world of human–system performance models into two general categories: *reductionist* models and *first principle* models. Reductionist models use human/system task sequences as the primary organizing structure, as shown in Figure 2. The individual models of human behavior for each task or task element are connected to this task-sequencing structure. We refer to it as reductionist because the process of modeling human behavior involves taking the larger aspects of human/system behavior (e.g., ‘‘perform the mission’’) and then successively reducing them to smaller elements of behavior (e.g., ‘‘perform the function,’’ ‘‘perform the tasks’’). This continues until a level of decomposition is reached at which reasonable estimates of human performance for the task elements can be made. One can also think of this as a top-down approach to modeling human/system performance. The example of this type of modeling that we will use in this chapter will be task network modeling, where the basis of the human–system model is a task analysis.

First-principle models of human behavior are structured around an organizing framework that represents the underlying goals and principles of human performance. Tools that support first-principle modeling of human behavior have structures embedded in them that represent elemental aspects of the human. For example, these models might directly represent processes such as goal-seeking behavior, task scheduling, sensation and perception, cognition, and motor output. To use tools that support first-principle modeling, one must describe how the system and environment interact with the modeled human processes. An example of a very simple structure that supports this type of modeling environment is presented in Figure 3. The example we will use of a tool designed to support this type of modeling is the Man–Machine Integrated Design and Analysis System (MIDAS).

It is worth noting that these two modeling strategies are not mutually exclusive and, in fact, can be mutually supportive in any given modeling project. Often, when one is modeling using a reductionist approach, one needs models of basic human behavior to represent behavioral phenomena accurately and therefore must draw on elements of first-principle models. Alternatively, when one is modeling human–system performance using a first-principled approach, some aspects of human–system performance and interrelationships between tasks may be more easily defined using a reductionist approach. Either class of model has been used to model individual and team performance. It is also worth noting that recent advances in human performance modeling tool development are blurring the distinctions between these two classes (e.g., Hoagland et al. 2001; LaVine 2000). Increased emphasis on interoperability between models has caused researchers and developers to focus on integrating reductionist and first-principle models.

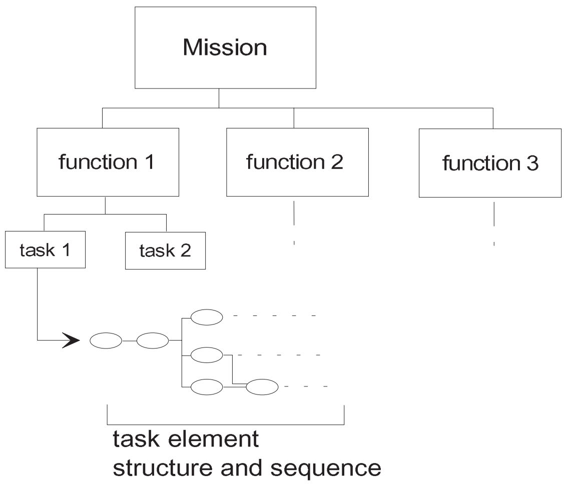

Figure 2 The Concept of Reductionist Models of Human Performance.

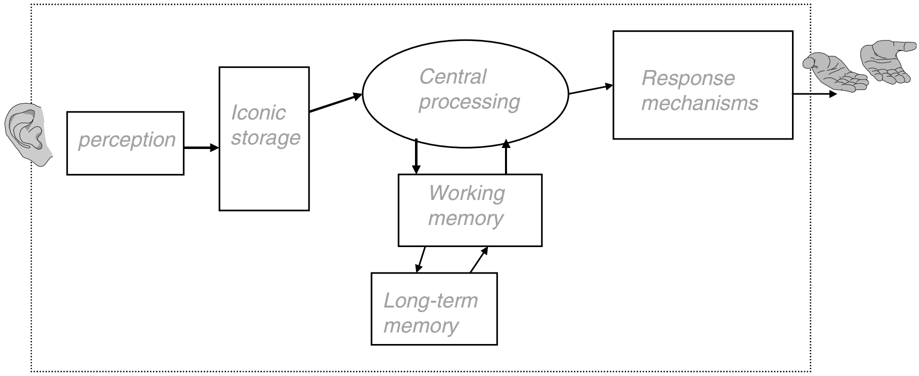

Figure 3 An Example of the Concept of First-Principle Models of Human Performance.

5. AN EXAMPLE OF A REDUCTIONIST APPROACH: TASK NETWORK MODELING

One technology that has proven useful for predicting human-system performance is task network modeling. In a task network model, human performance is decomposed into tasks. The fidelity of this decomposition can be selective, with some functions being decomposed several levels and others just one or two. This is, in human factors engineering terms, the task analysis. The sequence of tasks is defined by constructing a task network. This concept is illustrated in Figure 4, which presents a sample task network for dialing a telephone.

Task network modeling is an 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 staple—the task analysis. Task analyses organized by task sequence are the basis for the task network model. Second, task network models can include sophisticated submodels of the system hardware and software to create a closed-loop representation of relevant aspects of the human/machine system. Third, task network modeling is relatively easy to use and understand. Recent advancements in task network modeling technology have made this technology more accessible to human factors practitioners. Finally, task network modeling can provide efficient, valid, and useful 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. However, as stated before, for revolutionary designs, modeling may not alleviate the need for empirical data collection.

Task network models of human performance have been subjected to validation studies with favorable results (e.g., Lawless et al. 1995; However, as with any modeling approach, the real level at which validation must be considered is with respect to a particular model, not with respect to the general approach.

5.1. What Goes into a Task Network Model?

To represent complex, dynamic human/system behavior, many aspects of the system may need to be modeled in addition to simply task lists and sequence. In this subsection, we will use the task network modeling tool Micro Saint as an example.

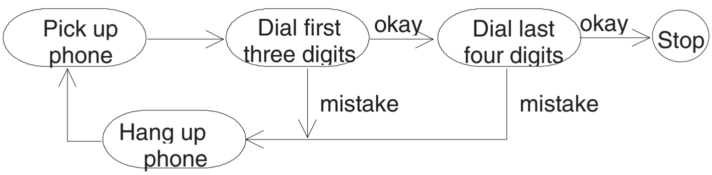

Figure 4 An Example of a Task Network Model Representing a Human Dialing a Telephone.

The basic ingredient of a Micro Saint task network model is the task analysis as represented by a network or series of networks. The level of system decomposition (i.e., how finely we decompose the tasks) and the amount of the system that is simulated depends on the particular problem. For example, in a power plant model, one can create separate networks for each of the operators and one for the power plant itself. While the networks may be independent, performance of the tasks can be interrelated through shared variables. The relationships among different components of the system, represented by different segments of the network, can then communicate through changes in these shared variables. For example, when an operator manipulates a control, this may initiate an open valve task in a network representing the plant. This could ripple through to a network representing other operators and subsystems and their response to the open valve.

This basic 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 5. Networks can be embedded within networks, allowing for hierarchical construction. In addition, the shape of the nodes on the diagram can be chosen in order to represent specific types of activity.

To reflect complex task behavior and interrelationships, more detailed characteristics of the tasks need to be defined. By double clicking on a task, the user opens up the task description window, as shown in Figure 6. Below are descriptions of each of the items on this window.

Task number: An arbitrary number for task referencing.

Task name: any name used to identify the task.

Time distribution: Micro Saint will conduct Monte Carlo simulations with task performance times sampled from a distribution as defined by this option (e.g., normal, beta, exponential).

Mean time: This parameter defines average task performance time for this task. This can be a number, equation, or algorithm, as can all values in the fields described below.

Standard deviation: Standard deviation of task performance time.

Release condition: Data in this field will determine when a task begins executing. For example, a condition stating that this task will not start before an operator is available might be represented by a release condition such as the following:

$$\text{operator} >= 1$$

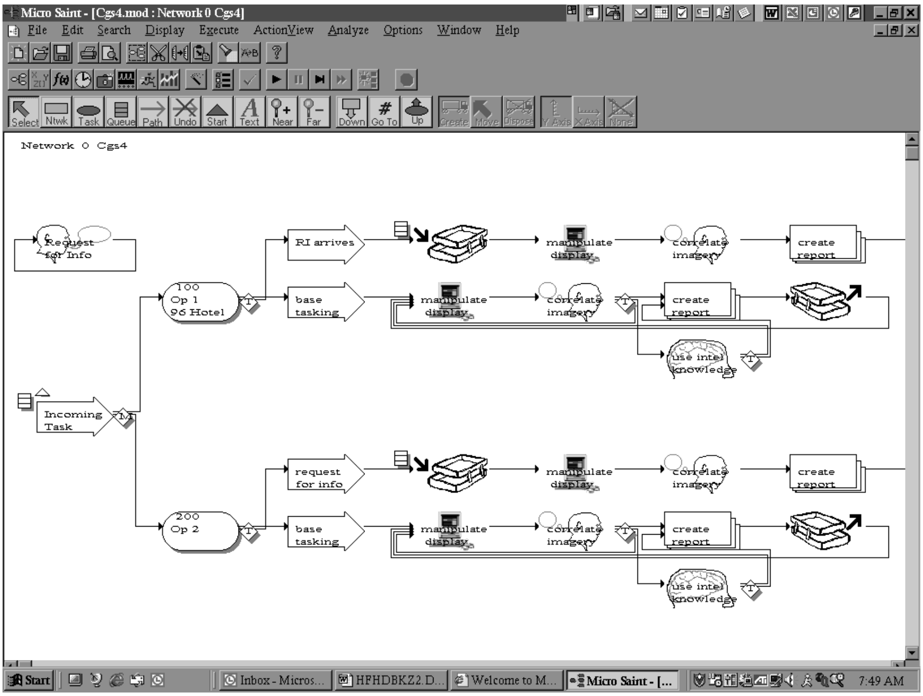

Figure 5 The Main Window in Micro Saint for Task Network Construction and Viewing.

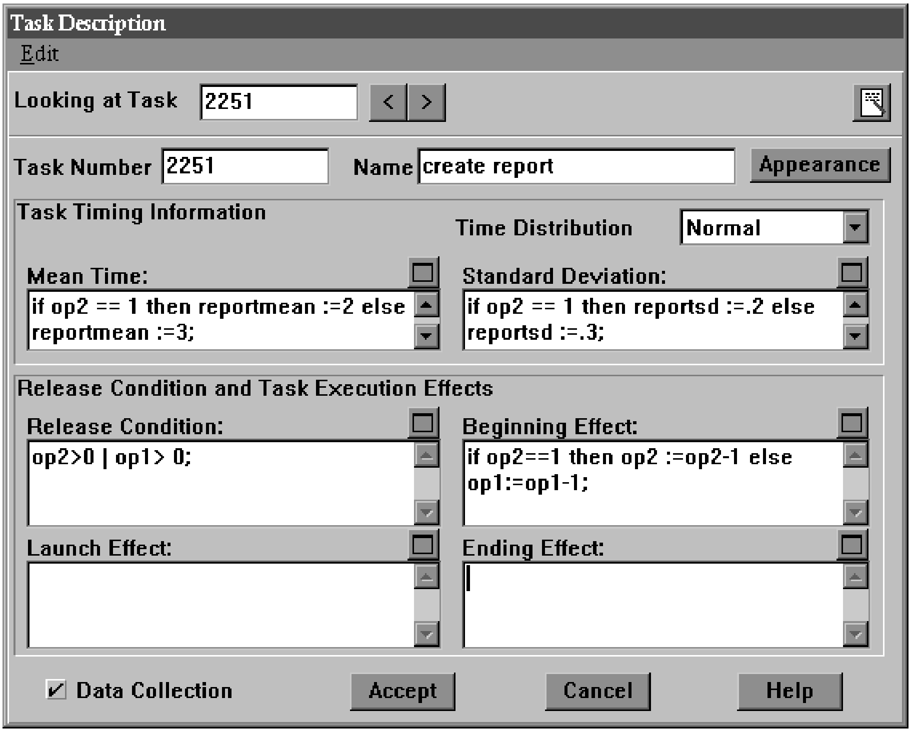

Figure 6 The User Interface in Micro Saint for Providing Input on a Task.

In other words, there must be at least one operator available for the task to commence. If all operators were busy, the value of the variable operator would equal zero until a task is completed, at which time an operator becomes available. This task would wait until the condition was true before beginning execution, which would probably occur as a result of the operator completing the task he or she is currently performing.

Beginning effect: This field permits the user to define how the system will change as a result of the commencement of this task. For example, if this task used an operator that other tasks might need, we could set the following condition to show that the operator is unavailable while he performed this task:

$$\text{operator} := \text{operator} - 1$$

Assignment and modification of variables in beginning effects are one key way in which tasks are interrelated.

Launch effect: Similar to a task beginning effect but used to launch high-resolution animation of the task.

Ending effect: This field permits the definition of how the system will change as a result of the completion of this task. From the previous example, when this task was complete and the operator became available, we could set the ending effect as follows

$$\text{operator} := \text{operator} + 1$$

at which point another task waiting for an operator to become available could begin. Ending effects are another key way in which tasks can be interrelated through the assignment and modification of variables.

Another notable aspect of the task network diagram window shown in Figure 5 is the diamond-shaped icons that follow some tasks. These are present every time more than one path out of a task is drawn. In a task network model, this means that several tasks might commence at the completion of this task. Often this represents a human decision-making process. In that case, the branches align to potential courses of action that the modeled human could select. To define the decision logic, the user of Micro Saint would double-click on the diamond to open up a window, as shown in Figure 7.

There are only three general types of decisions to model:

- *Probabilistic:* In probabilistic decisions, the human will begin one of several tasks based on a random draw weighted by the probabilistic branch value. These weightings can be dynamically calculated to represent the current context of the decision. For example, this decision type might be used to represent human error likelihoods and would be connected to the subsequent tasks that would be performed.
- *Tactical:* In tactical decisions, the human will begin one of several tasks based on the branch with the highest "value." This could be used to model the many types of rule-based decisions that humans make, as illustrated in Figure 7.
- *Multiple:* This would be used to begin several tasks at the completion of this task, such as when one human issues a command that begins other crew members' activities.

The fields in Figure 7 labeled "Routing Condition" represent the values associated with each branch. The values can be numbers, expressions, or complicated algorithms defining the probability (for probabilistic branches) or the desirability (for tactical and multiple branches) of taking a particular branch in the network. Again, any value on this screen can not simply be numbers but also include variables, algebraic expressions, logical expressions, or groups of algebraic and logical expressions that would essentially form a subroutine. As the model executes, Micro Saint includes a parser that evaluates the expressions included in the branching logic when it is encountered in the task network flow. This results in a dynamic network in which the flow through the tasks can be controlled with variables that represent equipment state, scenario context, or the task loading of the humans in the

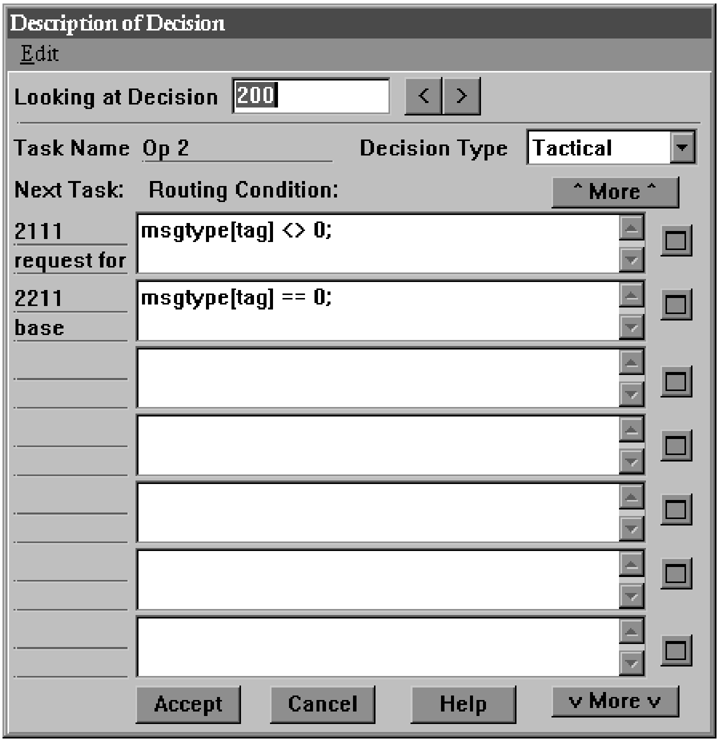

Figure 7 The User Interface in Micro Saint for Defining Task Branching Decision Logic.

system, to name a few examples. It is the power of this parser that provides many task network models the ability to address complex problems.

There are other aspects of task network model development. Some items are defining a simulation scenario, defining continuous processes within the model, and defining queues in front of tasks. Further details of these features can be obtained from the Micro Saint User's Guide (Micro Analysis and Design 1999).

As a model is being developed and debugged, the user can execute the model to test it and collect data. There are several display modes, reflecting differing levels of information provided to the user during execution. In the most detailed mode, the simulation pauses after every simulated task. Another mode shows the user nothing about the simulation except when it is completed. There is also a model animation mode in which the task network is drawn on the screen and tasks that are currently executing are highlighted. In the model animation mode, the analyst can get a very clear picture of what events are occurring in what sequence in the model. Figure 8 presents a sample display during model animation.

Once a model is executed and data are collected, the analyst has a number of alternatives for data analysis. The data created during a model execution can be reviewed in the model analysis environment or exported to statistical and graphics packages.

The above discussion should indicate that task network modeling is a relatively straightforward concept that is a logical extension of task and systems analysis. Task network modeling is an evolution, not a revolution, to the human factors practitioner. As stated before, the basis for task network models of human performance is the mainstay of human engineering analysis—the task analysis. Much of the information discussed is generally included in the task analysis. Task network modeling, however, greatly increases the power of task analysis since the ability to simulate a task network with a computer permits prediction of human performance, rather than simply the description of human performance that a task analysis provides. What may not be as apparent, however, is the power of task network modeling as a means for modeling human performance in systems. Simply describing the systems activities in this step-by-step manner allows complex models of the system to be developed where the human's interaction with the system can be represented in a closed-loop manner.

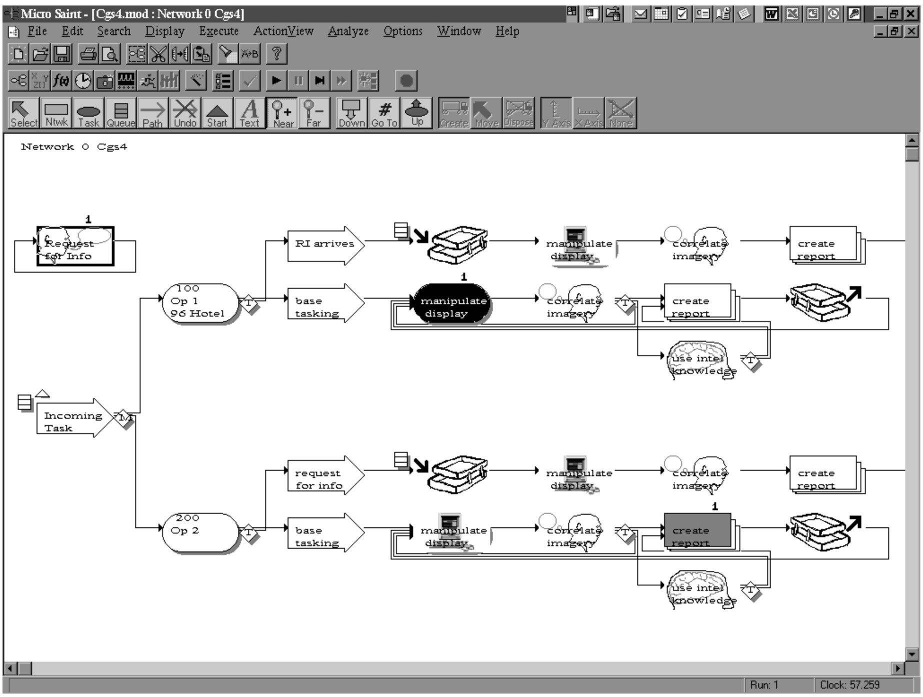

Figure 8 An Example of a Task Network Animation during Model Execution in Micro Saint.

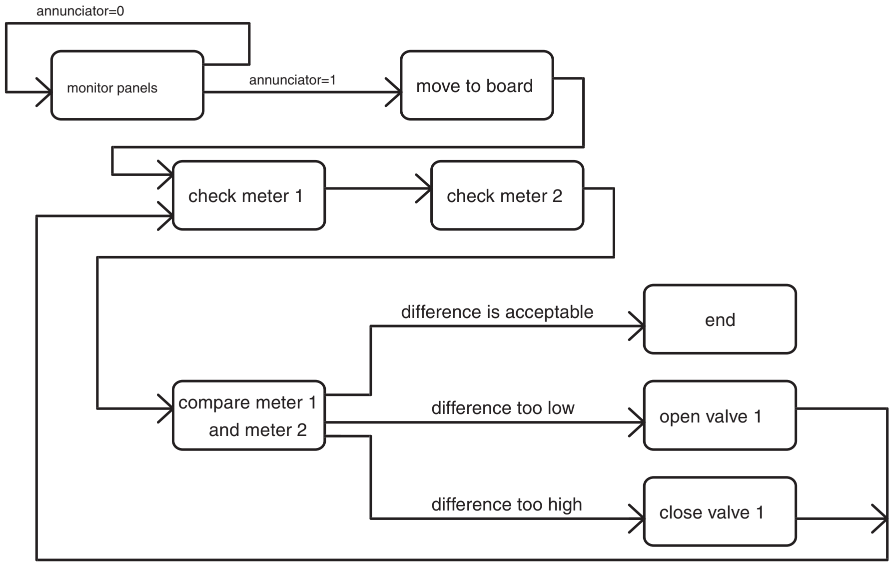

Figure 9 Sample Task Network Model of a Process Control Operator Responding to an Annunciator.

The above discussion, in addition to being an introduction to the concepts, is also intended to support the argument that task network modeling is a mature technology ready for application in a wide range of problem domains.

5.2. An Example of a Task Network Model of a Process Control Operator

This simple hypothetical example illustrates how many of the basic concepts of task network modeling can be applied to studying human performance in a process control environment. It is intended to illustrate many of the concepts described above.

The simple human task that we want to model is of an operator responding to an annunciator. The procedure requires that the operator compare readings on two meters. Based on the relative values of these readings, the operator must either open or close a valve until the values on the two meters are nearly the same. The task network in Figure 9 represents the operator activities for this model. Also, to allow the study of the effects of different plant dynamics (e.g., control lags), a simple one-node model of the line in which the valve is being opened is included in Figure 10.

The operator portion of the model will run the monitor panels task until the values of the variables meter1 and meter2 are different. The simulation could begin with these values being equal and then precipitate a change in values based on what is referred to as a scenario event (e.g., an event representing the effects of a line break on plant state). This event could be as simple as:

$$\text{meter1} := \text{meter1} + 2.0$$

or as complex as an expression defining the change in the meter as a function of line break size,

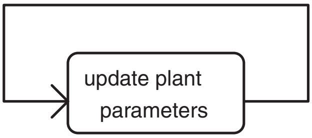

Figure 10 A Simple One-Node Model of the Plant That Is Integrated with the Detailed Operator Model.

flow rates, and so on. An issue that consistently arises in model construction is how complex the plant system model should be. If the problem under study is purely operator performance, simple models will usually suffice. However, if overall plant behavior is of interest, then the models of plant dynamics, such as meter values, are more important. Again, we recommend the ''start simple'' approach whenever possible.

When the transient occurs and the values of meter1 and meter2 start to diverge, the annunciator signal will trigger. This annunciator would be triggered in the plant portion of the model by a task-ending effect such as:

if meter1 <> meter2 then annunciator := 1

Once the plant model sets the value of the variable annunciator to 1, the operator will begin to move to the appropriate board. Then the operator will continue through a loop to check the values for meter1 and meter2 and either open valve1, close valve1, or make no change. The determination of whether to make a control input is determined by the difference in values between the two meters. If the value is less than the acceptable threshold, then the operator would open the valve further. If the value is greater than the threshold, then the operator would close the valve. This opening and closing of the valve would be represented by changes in the value of the variable valve1 as a task-ending effect of the tasks open valve1 and close valve1. In this simple model, operators do not consider rates of change in values for meter1 and therefore would get into an operator-induced oscillation if there were any response lag. A more sophisticated operator model could use rates of change in the value for meter1 in deciding whether to open or close valves.

Again, this is a very small model reflecting simple operator activity on one control via a review of two displays. However, it illustrates how large models of operator teams looking at numerous controls and manipulating many displays could be built via the same building blocks used in this model. The central concepts of a task network and shared variable reflecting human–system dynamics remain the same.

Given a task network model of a process control operator in a current control room, how might the model be modified to address human-centered design questions? Some examples are:

1. Modifying task times based on changes in the time required to access a new display
2. Modifying task times and accuracies based upon changes in the content and format of displays
3. Changing task sequence, eliminating tasks, and/or adding tasks based upon changes in plant procedures
4. Changing allocation of tasks and ensuing task sequence based upon reallocation of tasks among operators
5. Changing task times and accuracies based upon stressors such as sleep loss or the effects of circadian rhythm

The above is not intended as a definitive list of all the ways that these models may be used to study design or operations concepts, but it should illustrate how these models can be used to address design and operational issues.

5.3. Case Studies in the Use of Task Network Modeling to Address Specific Design Issues

In this section, we will examine two case studies in the use of task network simulation for studying human performance issues. The first case study explores how task network modeling can be used to assess task allocation issues in a cognitively demanding environment. The second example explores how task network modeling has been used to extend laboratory and field research on human performance under stress to new task environments.

We should state clearly that these examples are intended to be representative of the types of issues that task network modeling can address, as well as approaches to modeling human performance with respect to these issues. *They are not intended to be comprehensive with respect to either the issues that might be addressed or the possible techniques that the human factors practitioner might apply.* Simulation modeling is a technology whose application leaves much room for creativity on the part of the human factors practitioner with respect to application areas and methods. These two case studies are representative.

5.4. Using Task Network Modeling to Evaluate Crew Workload

Perhaps the greatest contributor to human error in many systems is the extensive workload placed upon the human operator. The inability of the operator to cope effectively with all of his or her information and responsibilities contributes to many accidents and inefficiencies. In recognition of this problem, new automation technologies have been introduced to reduce workload during periods of high stress. Some of these technologies are in the form of enhanced controls and displays, some are in the form of tools that push information to the operator and alert the operator in order to focus attention, and still others consist of adaptive tools that take over tasks when they sense that the operator is overloaded. Unfortunately, these technical solutions often introduce new tasks to be performed that affect the visual, auditory, and/or psychomotor workload of the operators.

Recently, new concepts in crew coordination have focused on better management of human workload. This area shows tremendous promise and is benefiting from efforts of human factors researchers. However, their efforts are hindered because there are limited opportunities to examine empirically the performance of different combinations of equipment and crew composition in a realistic scenario or context. Additionally, high workload is not typically caused by a single task but by situations in which multiple tasks must be performed or managed simultaneously. It is not simply the quantity of tasks that can lead to overload, but also the composition of those tasks. For example, two cognitive tasks being performed in parallel are much more effortful than a simple motor task and an oral communication task being performed together. The occurrence of these situations will not typically be discovered through normal human engineering task analysis or subjective workload analysis until there is a system to be tested. That is often too late to influence design.

To rectify this problem, there has been a significant amount of recent research and development aimed at human workload *prediction* models. Predictive models allow the designers of a system to estimate operator workload *without human subjects experimentation*. From this and other research, a solid theoretical basis for human workload prediction has evolved, as is described in Wickens (1989).

This section discusses a study using task network modeling to predict the impact of task allocation on human workload. While these examples are posed in the context of the design of a military system, the same techniques have been used in nonmilitary applications such as process control and user–computer interface design.

5.4.1. Modeling the Workload of a Future Command and Control Process

The Army command and control (C2) community is concerned with how new information, technology, and organizational changes projected for tomorrow's battlefield will impact soldier tasks and workload. To address this concern, an initiative was taken to model soldier performance under current and future operational conditions. In this way, the impact of performance differences could be quantitatively assessed so that equipment and doctrine design could be influenced in a timely and effective manner.

In one C2 project, the primary concern was to determine how tasks should be allocated and automated such that a C2 team could evaluate all the relevant data and make decisions within an environment with particularly high time pressure. Specifically, the effort was to address the following key questions:

- How many crew members do you need?
- How do you divide tasks among jobs?
- How does decision authority flow?
- Can the crew meet decision time line requirements?
- Is needed information usable and accessible?

We used task network modeling to study crew member, task and scenario combinations in order to examine these questions.

Figure 11 shows the top level diagram of the task network. Essentially, the crew members receive and monitor information about the system and the environment until an event occurs that pushes them out of the 10000 and 20000 networks into either a series of planning tasks, and/or a series of evaluation, decision, direction, and execution tasks. The purpose of the planning task is to update tactical battle plans based on new information received from the system or the environment. Receipt of new intelligence data about the enemy's intention or capability is an example of an event that would cause the crew members to undertake planning tasks. Similarly, receipt of information from the system about resource limitations might trigger the crew members to proceed down the alternative path (through evaluate to execute). Specifically, limited resources might cause the crew members to evaluate whether the engagement is proceeding appropriately (30000), decide how to adjust system parameters (40000), direct the appropriate response to the correct level of command (50000), and then execute the order (60000). Upon completion, the crew members would return to monitoring the system and situation.

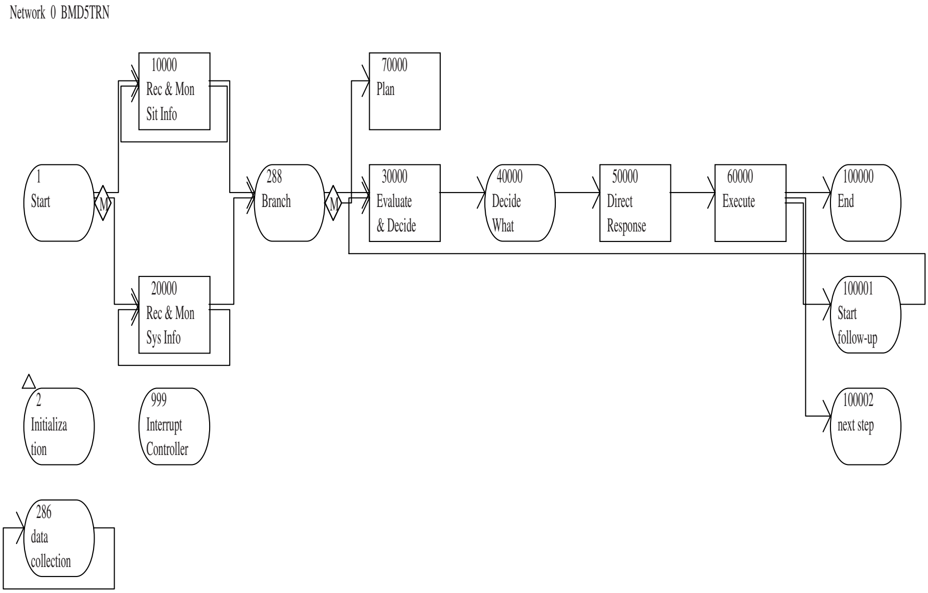

Figure 11 Upper-Level Task Network.

Each of the rectangles in the task network shown in Figure 11 actually consists of a network of tasks. An example of the tasks that belong to Network 10000, receive and monitor situation information, is shown in Figure 12. As described under Figure 7 of this chapter, the tasks in Network 10000 are linked by probabilistic and tactical decisions.

Each of the tasks in the C2 task network is associated with several items of human performance data. These include:

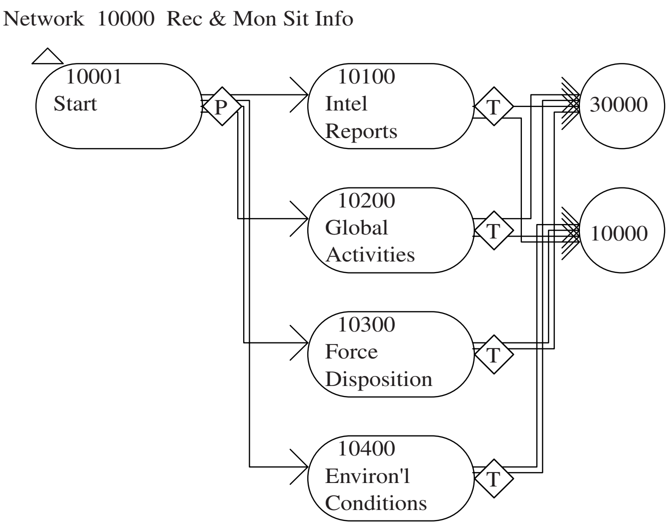

Figure 12 Second- Level Task Network.

- *Task performance time:* These data consist of a mean, standard deviation, and distribution. The data were collected from a combination of three sources: (1) human factors literature (e.g., Fitts's Law); (2) empirical studies during operator-in-the-loop simulator exercises; and (3) subject matter experts.
- *Branching logic:* While the task network indicates a general process flow, this particular model was designed to respond to scenario events. Because of that design decision, each task includes logic to determine the following task. For example, if the scenario is very intense and multiple target tracks are available, the crew members will follow a different task flow than if they were performing routine system checks.
- *Release rules:* Logic controlling the number and types of parallel tasks each crew member can perform are contained in each task's release condition.

Since one purpose of the model was to examine various task-allocation strategies, the model was designed to incorporate several measures of crew member workload. The basis of this technique is an assumption that excessive human workload is not usually caused by one particular task required of the operator. Rather, the human having to perform several tasks simultaneously leads to overload. Since the factors that cause this type of workload are intricately linked to these dynamic aspects of the human's task requirements, task network modeling provides a good basis for studying how task allocation and sequencing can affect operator workload.

However, task network modeling is not inherently a model of human workload. The only relevant output common to all task network models is the time required to perform a set of tasks and the sequence in which the tasks are performed. Time information alone would suffice for some workload-evaluation techniques, such as Siegel and Wolf (1969), whereby workload is estimated by comparing the time available to perform a group of tasks to the time required to perform the tasks. Time available is driven by system performance needs, and time required can be computed with a task network model. However, it has long been recognized that this simplistic analysis misses many aspects of the human's tasks that influence both perceived workload as well as ensuing performance. At the very least, this approach misses the fact that some pairs of tasks can be performed in combination better than other pairs of tasks.

One of the most promising theories of operator workload, which is consistent with task network modeling, is the multiple resource theory proposed by Wickens (e.g., Wickens et al. 1983). Simply stated, the multiple-resource theory suggests that humans have several different resources that can be tapped simultaneously. Depending upon the nature of the information-processing tasks required of a human, these resources would have to process information sequentially (if different tasks require the same types of resources) or possibly in parallel (if different tasks required different types of resources). There are many versions of this multiple-resource theory in workload literature (e.g., McCracken and Aldrich: Bierbaum et al. 1989; North and Riley 1989; Little et al. 1993; Knapp et al. 1999). In this chapter, we will provide a discussion of the underlying methodology of the basic theory.

Multiple-resource workload theory is implemented in a task model in a fairly straightforward manner. First, each task in the task network is characterized by the workload demand required in each human resource, often referred to as a workload channel. Examples of commonly used channels include auditory, visual, cognitive, and psychomotor. Particular implementations of the theory vary in the channels that are included and the fidelity with which each channel is measured (high, medium, low vs. seven-point scale). In fact, Bierbaum et al. 1989 present reliable benchmark scales for determining demand for each channel. As an example, the scale for visual demand is presented in Figure 13.

Similar scales have been developed for the auditory, cognitive, and psychomotor channels. With this approach, each operator task can be characterized as requiring some amount of each of the four kinds of resources, as represented by a value between one and seven. All operator tasks can be analyzed with respect to these demands and values assigned accordingly.

In performing a set of tasks pursuant to a common goal (e.g., engage an enemy target), crew members frequently must perform several tasks simultaneously, or at least nearly so. For example, they may be required to monitor a communication network while visually searching a display for target track. Given this, the workload literature indicates that the crew member may either accept the increased workload (with some risk of performance degrading) or begin dumping tasks perceived as less important. To factor these two issues into task network simulations, two approaches can be incorporated: (1) evaluate combined operator workload demands for tasks that are being performed concurrently, and/or (2) determine when the operator would begin dumping tasks due to overload.

During a task network simulation, the model of the crew may indicate they are required to perform several tasks simultaneously. The task network model evaluates total attentional demands for each human resource (e.g., visual, auditory, psychomotor, and cognitive) by combining the attentional demands across all tasks that are being performed simultaneously. This combination leads to an overall workload demand score for each crewmember.

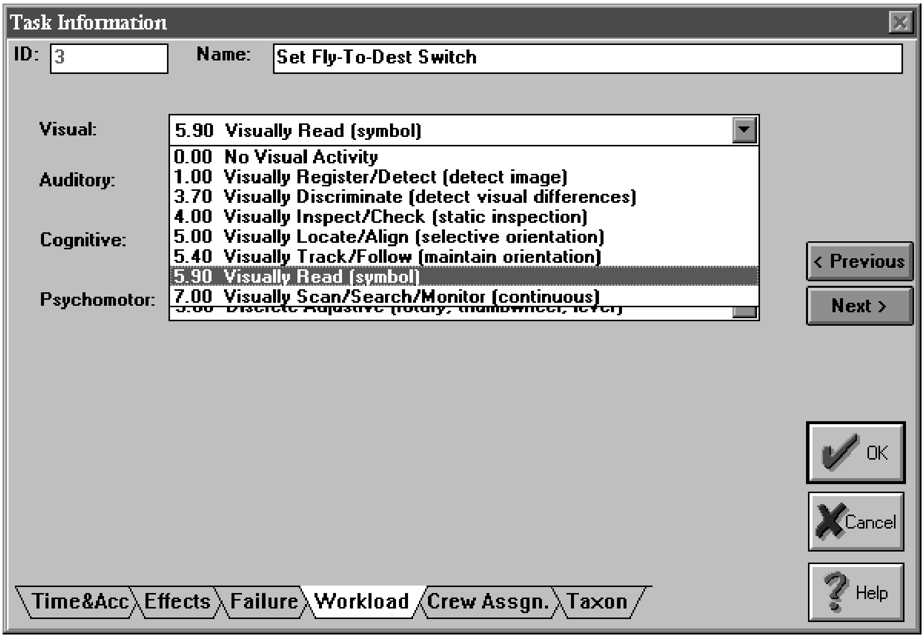

Figure 13 An Example of a Visual Workload Scale.

To implement this approach in Micro Saint, the task beginning effect can be used to increment variables that represent the current workload score in each resource. Then, while the tasks are being performed, these variables track attentional demands. When the tasks are completed, the task ending effects can decrement the values of these variables accordingly. Therefore, if these workload variables were recorded and then plotted as the model runs, the output would look something like what is shown in Figure 14. This result can be used to identify points of high workload throughout the scenario being modeled. The human factors practitioner can then review the tasks that led to the points of high workload and determine whether they should be reallocated or redesigned in order to alleviate the peak. This is a common approach to modeling workload.

Once the task networks were verified with knowledgeable crewmembers, they became part of the human factors team's analytical test bed. Figure 15 shows the overall method that we used to examine aspects of crew member performance across a wide variety of operational scenarios and crew configuration concepts.

The center of this diagram, labeled task network, represents the tasks that the crew performs as we just described them. The network itself, representing the flow of the tasks, does not change between model runs. Rather, the model has been parameterized so that an event scenario stimulates the network. The left side of the diagram illustrates the types of data that are used to drive the task network model. In this case, those data include crew configurations, or allocations of tasks to different crew members and automation devices, as well as scenario events. The scenario events represent an externally generated time-ordered list of the events that trigger the crew members to perform tasks in the task network. The right side of Figure 15 represents the types of outputs that can be produced from this task network model. One of the primary outputs is a crew member workload graph, such as that shown in Figure 14.

5.4.2. Extensions of This Approach to Simulating Crew Mental Workload to Other Environments

The workload-analysis methodology described above has recently been developed into a stand-alone task network modeling tool by the Army Research Laboratory (ARL) Human Research and Engineering Directorate (HRED), as part of the Improved Performance Research Integration Tool (IMPRINT) (Allender et al. 1994; Archer and Adkins 1999). IMPRINT integrates task network modeling software with features to support specifically the multiple-resource theory of workload discussed

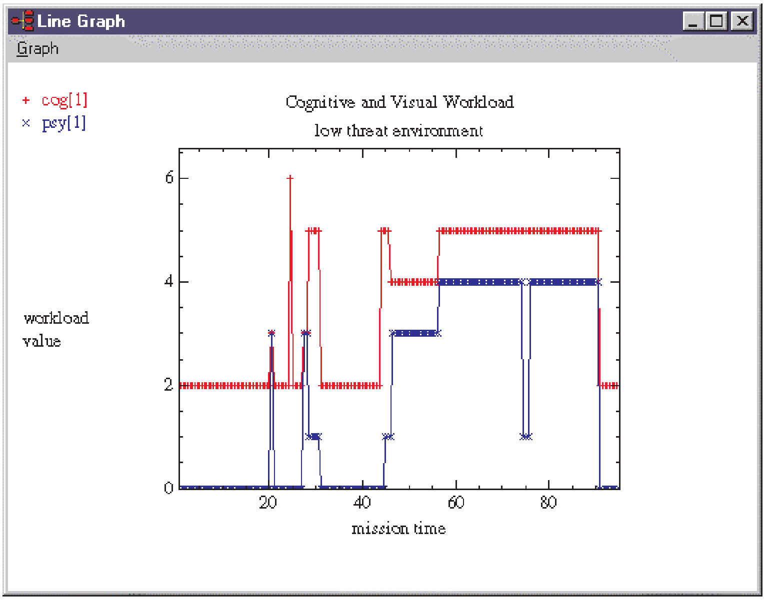

Figure 14 An Example Workload Output from a Task Network Model.

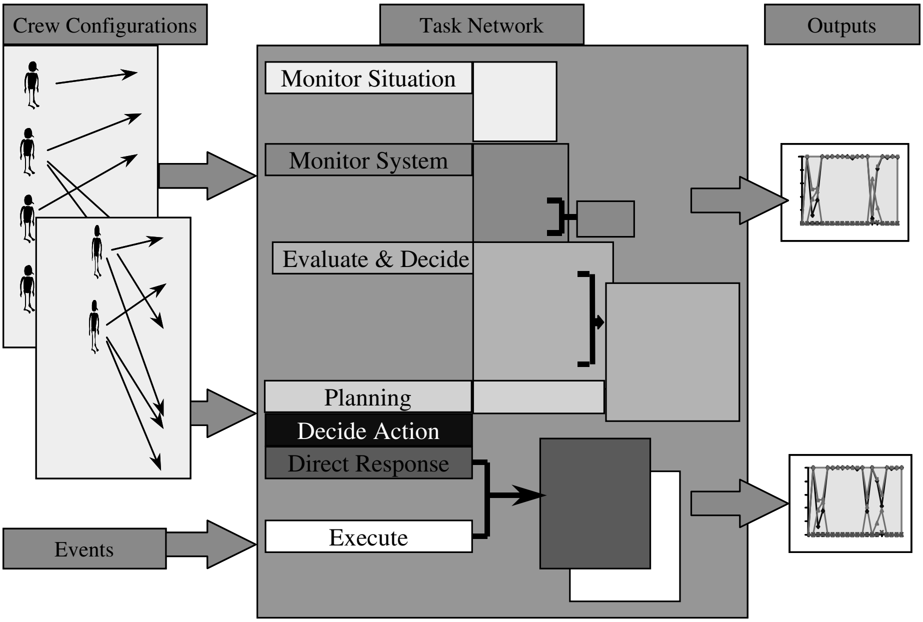

Figure 15 Overall Method for Examining Workload in a Complex System.

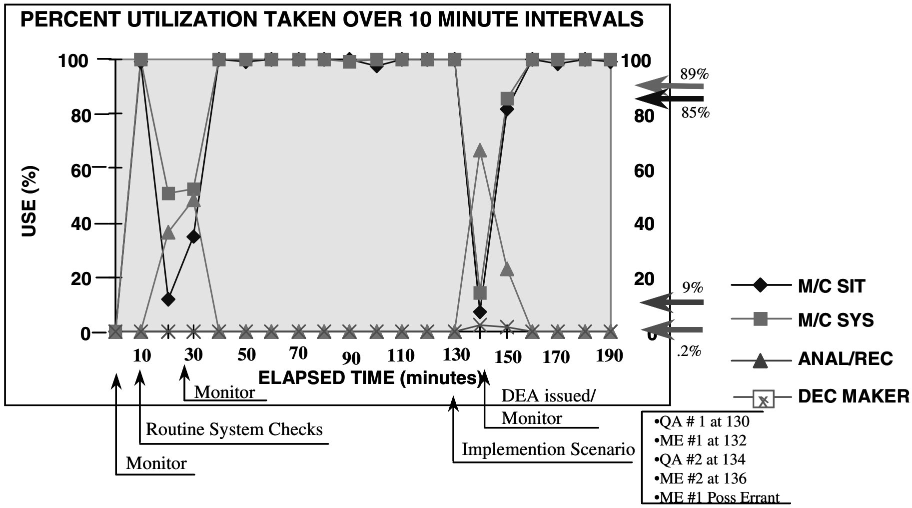

Figure 16 Example of Model Predictions of Operator Utilization over Time.

above. It provides the human factors practitioner with an environment that supports the analysis of task assignment to crewmembers based on four factors:

1. *Workload of crew members:* Tasks should be assigned to minimize the amount of time crew members will spend in situations of excessive workload.
2. *Time performance requirements:* Tasks must be assigned and/or sequenced so that they are completed within the available time. This consideration is essential because time constraints often will drive the need to perform several tasks simultaneously.
3. *Likelihood of successful performance and consequences of failure:* Tasks must be assigned and/or sequenced so that they can be completed within a specified accuracy measure.
4. *Access to controls and displays:* Tasks cannot be assigned to crew members who do not have access to the necessary controls and displays.

Of course, there are numerous theoretical questions regarding this simplistic approach to assessing workload in an operational environment. However, even the use of this simple approach has been shown to provide useful insight during design. For example, in a study conducted by the Army (Allender 1995), a three-man crew design was evaluated using a task network model. The three-man model was constructed using data from a prototype four-man system. From this model-based analysis, the three-man design was found to be unworkable . Later, human subjects experimentation verified that the model's workload predictions were sufficiently accurate to point the design team in a valid direction.

Finally, ARL HRED has developed another custom software package named WinCrew (Plott 1995). It includes capabilities, also available in an advanced version of IMPRINT, to implement more refined methods of predicting workload. WinCrew overlays the W/INDEX manifestation of the multiple resource theory of workload (Boettcher et al. 1989) into a task network-based environment. In addition to a better estimate of workload, WinCrew is unique in that it has built-in constructs for simulating workload management strategies that operators would employ to accommodate points of high operator workload. The ultimate result of simulating the workload management strategies is that the operator task network being modeled is dynamic. In other words, the task sequence, assignments to operators, and individual task performance may change in response to excessive operator workload as the task network model executes. These changes may be as simple as one operator handing tasks off to another operator to reduce workload to an acceptable level or as complex as the operator beginning to time share tasks in order to complete all the assigned tasks, potentially task time and/ or accuracy. Ultimately, the tool provides an estimate of system-level performance as a result of these realistic workload management strategies. This innovation in modeling provides greater fidelity in efforts that model human behavior in the context of system performance, particularly in high-workload environments such as complex system control and management.

5.5. Using Task Network Modeling as a Means of Extending Research Findings on Human Performance under Stress to New Task Environments

Task network modeling was used by LaVine et al. (1995) to extend laboratory data and field data collected on one set of human tasks to predicting performance on similar tasks. The problem of extending laboratory or field human performance data to other tasks has plagued the human engineering community for years. We intuitively know that human performance data can be used to predict performance for similar tasks. However, often the task whose performance we want to predict is similar in some ways but different in others. The approach described below uses a skill taxonomy to quantify task similarity and therefore provides a means for determining how other tasks will be affected when exposed to a common stressor on human performance. Once functional relationships are defined between a skill type and a stressor, task network modeling is used to determine the effect of the stressor on performance of a complex task that simultaneously uses many of these skills.

The specific approach below is being used by the U.S. Army to predict crew performance degradation as a function of a variety of stressors. It not intended to represent a universally acceptable taxonomy for simulating human response to stress. The selection of the best taxonomy would depend upon the particular tasks and stressors being studied. What this example is intended to illustrate is another way that task network modeling can be used to predict human performance by making a series of reasonable assumptions that can be played together in a model for the purpose of making predictions that would be impossible to make otherwise.

The methodology for predicting human performance degradation as a function of stressors consists of three parts: (1) a *taxonomy* for classifying tasks according to basic human skills, (2) *degradation functions* for each skill type for each stressor, and (3) *task network models* for the human-based system whose performance is being predicted. Conceptually, either laboratory or field data can be used to develop links between a human performance stressor (e.g., heat, fatigue) and basic human skills. By selecting a skills taxonomy that is sufficiently discriminating to make this assumption reasonable, one can assume that the effects of the stressor on all tasks involving the skill will be approximately the same. The links between the level of a stressor (e.g., fatigue) and resulting skill performance (e.g., the expected task time increase from fatigue) are defined mathematically as the degradation function. The task network model is the means for linking these back to complex human/system performance.

5.5.1. The Taxonomy

The basic premise behind the taxonomy is that the tasks that humans perform can be broken down into basic human skills or atomic tasks (Roth 1992). The taxonomy used by Roth consists of five skill types: attention, perception, psychomotor, physical, and cognitive skills. These taxonomic skills are described by Roth as follows:

1. *Attention:* The ability to attend actively to a stimulus complex for extended periods of time in order to detect specified changes or classes of changes that indicate the occurrence of some phenomenon that is critical to task performance.
2. *Perception:* The ability to detect and categorize specific stimulus patterns embedded in a stimulus complex.
3. *Psychomotor:* The ability to maintain one or more characteristics of a situation within a set of defined conditions over a period of time, either by direct manipulation or by manipulation of controls that cause changes in the characteristics.
4. *Physical:* The ability to accomplish sustained, effortful muscular work.
5. *Cognitive:* The ability to apply concepts and rules to information from the environment and from memory in order to select or generate a course of action or a plan. This includes communicating the course of action or plan to others.

These five skills covered most of the tasks that were of interest to the Army for this study and still provided a manageable number of categories for an analyst to use.

5.5.2. Degradation Functions

The degradation functions quantitatively link skill performance to the level of a stressor. The degradation functions can be developed from any data source, including standard test batteries or actual human tasks. Through statistical analysis, one can build skill-degradation functions for each taxon. These functions map the performance decrement expected on a skill based on the parameters of the

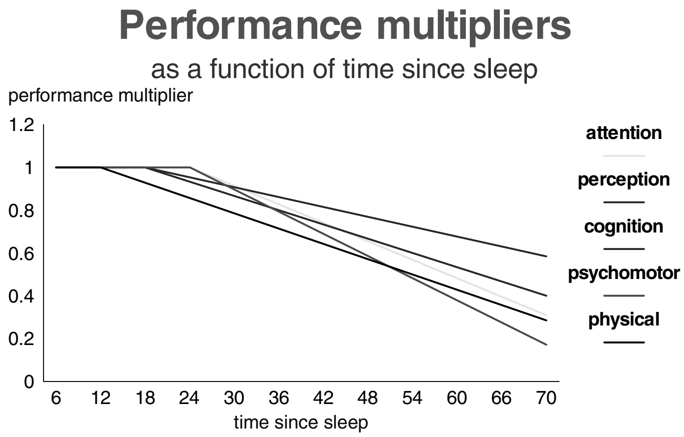

Figure 17 An Example of the Performance Degradation Functions Associated with each of the Human Skills from the Taxonomy.

performance-shaping factor (e.g., time since sleep). An example of these functions is presented in Figure 17.

5.5.2.1. Incorporating the Degradation Functions into Task Network Models to Predict Overall Human/System Performance Degradation The key to making this approach useful to predicting complex human performance is the task network model of the new task. In the task network model of the human's activities, all tasks are defined with respect to the percentage of each skill required from the taxonomy. For example, the following are ratings for tasks faced by a console operator responding to telephone contacts:

- Detect ring—50% attention, 50% perception
- Select menu item using a mouse—40% attention, 60% psychomotor
- Interpret customer's request for information—100% cognitive

In building the task network model, one can build functions to degrade a specific task's performance through an arithmetic weighting of skill-degradation multipliers that are derived from the degradation functions. For example, if the fatigue parameter was time since sleep and the value of that parameter was 36 hours since sleep, the task time performance multipliers would be as follows in the example above:

- Attention performance multiplier = 0.82
- Perception performance multiplier = 0.808
- Cognition performance multiplier = 0.856
- Psychomotor performance multiplier = 0.784
- Physical performance multiplier = 0.727

Based upon these multipliers and the above task weightings, the specific task effects would be:

- Detect ring—50% attention, 50% perception

$$\text{Task multiplier} = 0.5 \times 0.82 + 0.5 \times 0.808 = 0.814$$

- Select menu item using a mouse—40% attention, 60% psychomotor

$$\text{Task multiplier} = 0.4 \times 0.82 + 0.6 \times 0.784 = 0.7984$$

- Interpret customer's request for information—100% cognitive

$$\text{Task multiplier} = 0.856$$

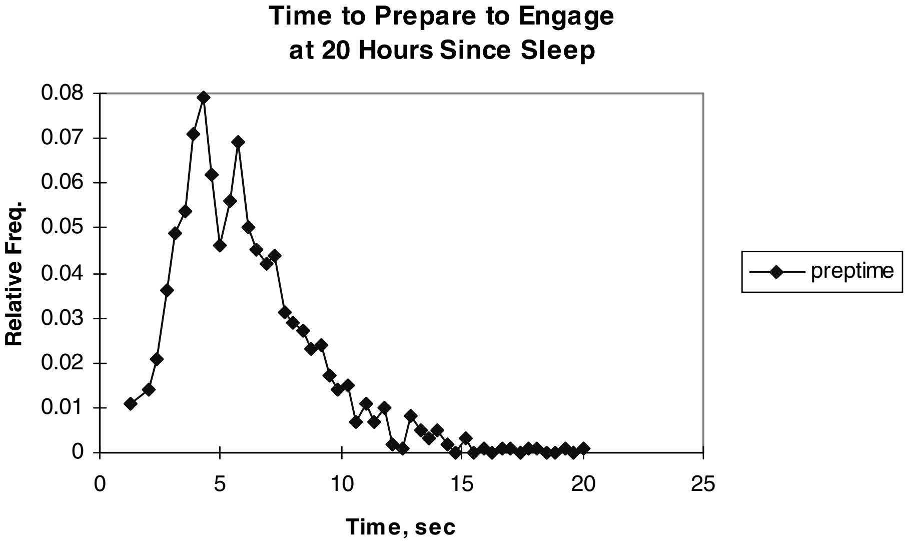

Figure 18 Frequency Distribution of Expected Human Performance as a Function of Time Since Sleep That Was Derived Using Task Network Modeling.

In a model of the complex tasks examined by LaVine et al. (1995), the task networks consisted of several dozen or even several hundred tasks. Through the approach described above, each task in a model exhibited a unique response to a stressor depending upon the particular skills that it required. The task network model then provided the means for relating the individual task performance to overall human/system performance as a function of stressor level (e.g., the time to perform a complex series of tasks involving decision making and error correction). Through this type of analysis, LaVine et al. were able to develop curves such as that shown in Figure 18 relating human performance to a stressor. These relationships would have been virtually impossible to develop experimentally.

Again, there were a number of simplifying assumptions that were made in this research. However, by being willing to accept these assumptions, LaVine et al. were able to characterize how complex human/system performance would be affected by a variety of stressors over a wide range in a relatively short time. They were thus able to estimate the effects of stressors that would have otherwise been pure guesswork.

5.6. Summary of Examples of Task Network Modeling of Human–System Performance

Once again, the above are intended to serve as examples, not as a catalogue of problems or approaches that are appropriate for task network modeling. Task network modeling is an approach to extend task and systems analysis to make predictions of human system performance. The creative human factors and ergonomics practitioner will find many other useful applications and approaches.

6. AN EXAMPLE OF A FIRST-PRINCIPLED APPROACH TO HUMAN/SYSTEM PERFORMANCE MODELING: THE MAN MACHINE INTEGRATED DESIGN SYSTEM (MIDAS)

The other fundamental approach to modeling human performance is based upon the mechanisms that underlie and cause human behavior. Since this approach is based on fundamental principles of the human and his or her interaction with the system and environment, we have designated them first-principle models. By integrating these models with models of the system and environment, the human factors specialist can predict the full behavior of large-scale interactive human–machine systems. The Man–Machine Integrated Design and Analysis System (MIDAS) follows in the tradition of integrated, first-principled models of human performance such as PROCRU (Baron et al. (1980) in that the modeling framework provides models of emergent human behavior based on elementary models of human behaviors such as perception, attention, working memory, and decision making. In the operation of these elementary models, MIDAS shares some of the characteristics of the task network approach. However, MIDAS is focused around an integrated architecture where micromodels of

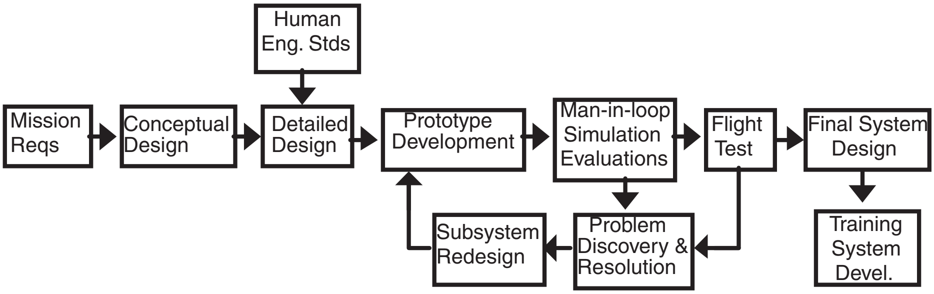

Figure 19 Current Crew-Station and Human–Computer Interaction Development Process.

human performance feed forward and feedback to the other constituent models in the human system rather than being linked primarily to the human's activities as in task network models.

6.1. Background

The A^3I Program was initiated in 1985 to support exploration of computational representations of human–machine performance to aid designers of crew systems. The major product of this effort was a human factors computer-aided engineering system called MIDAS (Man–Machine Integration Design and Analysis System). MIDAS is intended to revise the system design process in order to place more accurate information into the hands of the designers early in the process of human engineering design so that the impact and cost of changes are minimal. It is also intended to identify and model human–automation interactions with flexible representations of human–machine function. The crew station development process, as it is currently undertaken, is illustrated in Figure 19. The design proceeds from requirements and capabilities in conceptual design through increasing specification to hardware and software prototypes and simulation tests. Human performance evaluation occurs after prototype design and development. Results from testing the prototype are then used to guide prototype redesign.

MIDAS integrates the design process by using human performance models in the conceptual design phases of system development. Human–system integration and development enabled by the computational human performance models methodology are illustrated in Figure 20.

In this revised process, human performance considerations are accounted for early in the designs and played out for evaluation in the simulation mode. Iteration in this mode is flexible and timely. The flow then proceeds with a refined design into the standard design and prototype development phases. MIDAS provides a prototyping test bench, based on human performance models. Designers can work with computational representations of the crew station and human operators rather than relying solely on hardware simulators and man-in-the-loop studies, to discover problems and ask ''what-if'' questions regarding the projected mission, equipment, and environment.

In addition to its use in development and design, MIDAS offers a structure or framework in which to test and implement models of human cognition. The MIDAS framework systematizes and unifies the interaction of human performance representations in a common structure and with a common

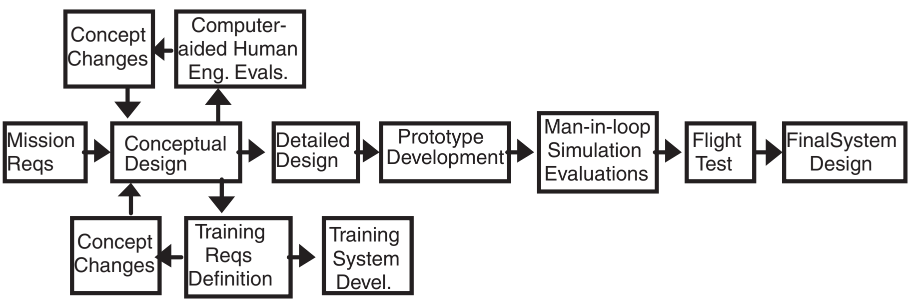

Figure 20 Design Methodology Made Possible through Modeling and MIDAS.

language for interaction. The representation is a tightly linked set of computational descriptions of the elemental aspects of human performance. Models of human performance from perception through cognition and action are implemented within this framework. The interplay of the models produces simulations of behavior.

In developing MIDAS, three challenges were considered. First, the level of representation of human behavior has to be sufficiently detailed to predict individual performance and guide design for individual aiding and support systems. At the same time, that behavioral representation must be able to provide input into large-scale analyses to predict global consequences of system modification. Second, the models of human–system performance have to be sufficiently computational to support design specification in control theoretic terms but also sufficiently flexible and robust to account for a range of human behavior influenced by cultural (corporate, professional, and national) and environmental context. Third, the human performance model has to represent individuals and teams of human operators. This requires control of not only single cognitive behavior but also the cognitive activities associated with group and organizational behavior. Such applications require representation of many intelligent agents sharing world models, and coordinating action/intention with cooperative scheduling of goals and actions in a potentially unpredictable world of operations. The simulated operators' activity structures must provide for anticipation (knowledge of the intention and action of remote operators) and respond to failures of the system and other operators in the system in context-sensitive processes.

MIDAS is intended to meet these challenges by including the following characteristics:

- *Modifiability and manipulability:* The basic mode of operation for MIDAS users is to explore the impact of changes to the baseline design. Thus, the capability for systematic change is critical. Of equal importance is system extensibility. To be generally useful, the modeling environment should be applicable to many types of design changes, and to many operational domains. The MIDAS architecture is designed to allow extensions of this type with minimal disruption to the existing core MIDAS system.
- *Transparency:* The analysis system must provide designers with explicit and transparent reference to the rules, decision-making strategies, heuristics, and assumptions under which the human–machine system is assumed to be operating, as well as predicted performance. For example, at any point in the simulation, a designer should be able to examine the cognitive state of the human operators, the rules that are being used to guide their behavior, and their nominal workload. The designer should also be able to perform sensitivity analyses on critical parameters of the human–machine system. Similarly, the state of equipment or mission progress should be able to be probed in order to relate the system state to the operator's performance.
- *Dynamic analysis capability:* The simulation system must produce a stream of behavior in the form of dynamic timelines describing not only its state and structure, but also sequences of action over time and contingent responses of the human/system behavior. The system must support testable hypotheses. Designers must be able to analyze the events occurring in a simulation scenario and relate this performance to man-in-the-loop simulation data. In MIDAS, each action taken, decision made, and communication event is logged by the analysis system.

6.2. System Architecture

There are two perspectives on the MIDAS system architecture that describe the system to support these modes of analysis: the functional architecture and structural bases of the system. Each is discussed below.

6.2.1. MIDAS Functional Architecture

The MIDAS system has evolved over a period of 15 years of development (Corker and Smith 1992). The basic structure of the core system presented here is based on the work of Tyler et al. (1998). This architectural version of MIDAS has throughout its development been used to evaluate helicopter crew stations, short-haul civil tiltrotor emergency handling operations, and the impact of MOPP flight gear on crew performance (Atencio et al. 1996, 1998; Shively et al. 1995). The specific development for analysis of air traffic management systems will be provided below.

The user enters the system through the graphical user interface (GUI), which provides the main interaction between the designer and the MIDAS system. The user selects among four functions in the system:

1. Create and/or edit a domain model that includes establishment and selection of the parameters of performance for the human operator model(s) in the simulation
2. Select the graphical animation or view to support that simulation or a set of simulations

3. Specify in the simulation module the parameters of execution and display for a given simulation set
4. Specify in the results analysis system the data to be collected and analyzed as a result of running the simulation.

The results-analysis system also provides for archival processes for various simulation sessions. The overall functional architecture is provided in Figure 21.

The user would typically use all of the top-level features to support a new simulation. If a user were exploring, for instance, the assignment of function between a human operator and an automated assistant, the user could maintain the majority of the extant domain, graphical, and analytic models and make modification through the domain model to the human operator model, the equipment model, and the simulation scenario.

6.2.1.1. Domain Model The domain model consists of descriptors and libraries supporting the creation of:

- *Vehicle characteristics:* Location space, aerodynamic models of arbitrarily detailed fidelity, and guidance models for vehicle (automatic) control.
- *Environment characteristics:* This provides the external interactions, including terrain from selected databases at varied levels of resolution, weather features insofar as they affect vehicle performance or operator sensory performance, and cultural features (towns, towers, wires, etc.). In short, the analyst here specifies the world of action of the experiment/simulation.
- *Crew Station/equipment characteristics:* The crew station design module and library is a critical component in the MIDAS operation. Descriptions of discrete and continuous control operation of the equipment simulations are provided at several levels of functional detail. The system can provide discrete equipment operation in a stimulus–response (black box) format, a time-scripted/event driven format, or a full discrete-space model of the transition among equipment states. Similarly, the simulated operator's knowledge of the system can be at the same varied levels of representation or can be systematically modified to simulate various states of misunderstanding the equipment function.
- *The human operator model* (*HO*): The human performance model in MIDAS allows for the production of behavior and response for single and multiple operators in the scenarios. The

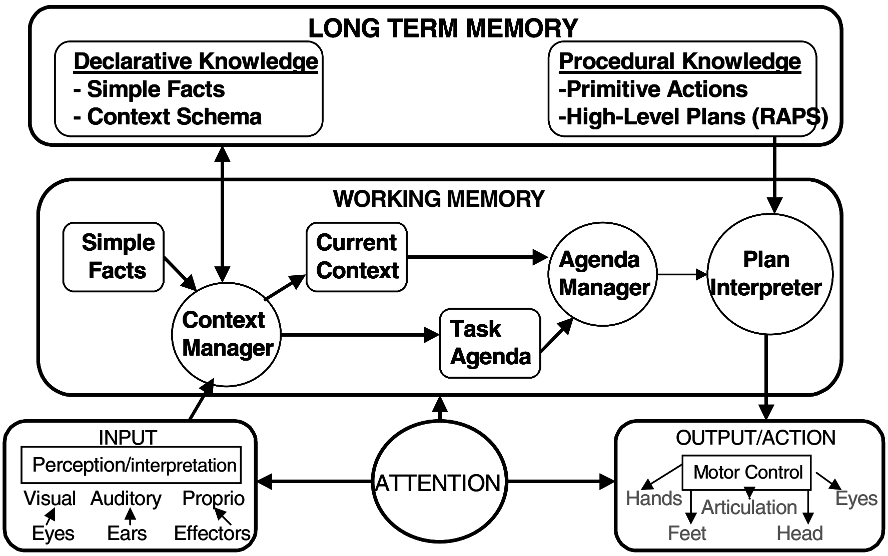

Figure 21 Illustration of the Overall Functional Architecture of the Human Operator Model in MIDAS. The functional architecture is repeated because multiple operators need to be represented in larger team simulations.

human operator model is the key to the MIDAS function as a predictive design aid. The human operator performance model is a combination of a series of functionally integrated micromodels of specific cognitive capabilities within a human operator. The human operator model functions as a closed-loop control model with inputs coming from the world and action being taken in the world. The model provides psychological plausibility through the explicit representation of cognitive constructs (detailed in Section 6.3).

- *Mission and activity models:* describe in a hierarchic structure the goals and the available recovery activities from missions-not-as-planned that make up the human operators' high-level behavioral repertoire in the mission. The next level of decomposition of the action of the mission is a set of high-level procedures (that can be stored as a fairly generic set of routines, e.g., look at or fixate). Finally, there are the specific activities in "active action packets" RAPS, which are the process by which the human operator affects the simulation.

6.2.1.2. Graphical Simulation Model MIDAS provides the user with a set of CAD development tools for both the design and modification of the simulation domain elements. The system also provides for the selection of a set of views or graphical simulation windows. These windows can provide various perspectives into the ongoing simulation, from that of a God's-eye view of the vehicles, environment and operators to views from the eyes of a particular human agent in the simulation.

In addition to the standard physical views, MIDAS supports a number of analytic views. The user can select views into the activity performance over time or views into the cognitive activities of any of the operators of the simulation. In addition, these views can be linked to the overall simulation view so that the user can view the cognitive activities and their effect in the real world at the same time. The time synching also allows for retrospective reply after a particular simulation has been performed. Figure 22 illustrates a standard set of views that might be selected during a simulation.

6.2.1.3. Analytic Tools After the simulation trials have been run, MIDAS provides the user with an editor to aid in the specification of analyses to be performed on the data generated. A set of analyses can be undertaken in the MIDAS model environment, which will allow the analyst views

Figure 22 Illustration of a Run-Time View of the MIDAS System with Cockpit, Pilot, and Copilot, and an Activity and Load Time as Well as the Procedures and Goals That Are Currently Active in the System.

TABLE 1 Representational Models Within MIDAS.

Micromodels	Empirical research
Visual processing (field of view)	Arditi and Azueta 1992; Lubin and Bergen 1992
Visual perception	Remington et al. 1992
Auditory processing	Card et al. 1983
Central processing and memory	Baddeley and Hitch 1974
Effectors/output behavior (35 primitive tasks)	Hamilton et al. 1990
Attention—multiple-resource theory	Wickens 1984
Anthropometric models	Badler et al. 1993 (28)

Source: Gore and Corker 1999.

into the task timeline and load levels for the operators simulated in the scenario. MIDAS supports packaging and exportation of specific data associated with the simulation entities to external statistical packages as well.

6.3. MIDAS Structural Architecture

The structural architecture of MIDAS is that of a federated set of models organized into groupings that represent the various agents in the simulation. We will concentrate here on the structural integration of the models that compose the human operator(s) in the MIDAS modeling system. These models have been developed in a structure that represents an empirically based human information-processing model. This structural integration has been termed a first-principles model, based on its intentional integration of cognitive models that represent separable elements of the cognitive process.

The first-principles models of human performance are based on the mechanisms that underlie and cause human behavior. First-principles models integrate human perceptual and cognitive systems and human motor systems, thus incorporating the higher-level behaviors that are characteristic of human performance. This incorporation supports emergent human behavior based on elementary model function. The representational models and the research upon which these models have been based can be seen in Table 1.

Figure 23 presents a schematic of the way that the models can be structured for many human performance-modeling applications.

The cognitive submodels function as follows.

6.3.1. Working and Long-Term Memory Stores

Working memory is the store that is susceptible to interference and loss in the ongoing task context. Long-term memory loss would represent, for instance, a loss of skills or deep procedural memory

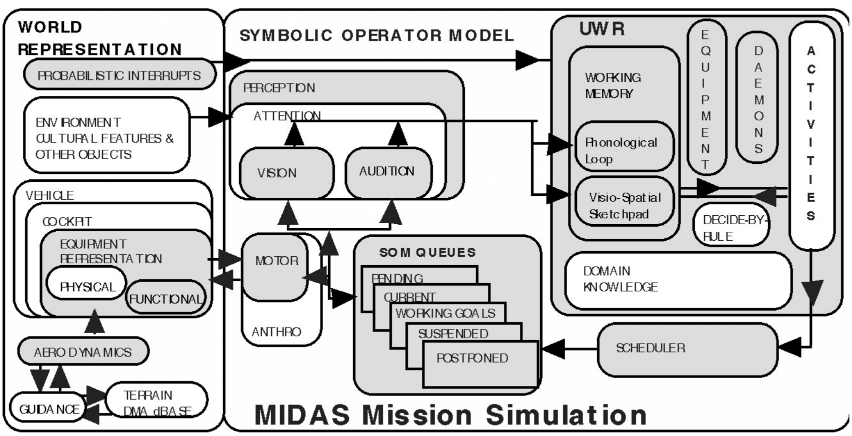

Figure 23 Structure of Cogntive Models.

of how to perform tasks. Neither the process of loss nor development of long-term memory, learning, is explicitly represented as a MIDAS function. We have modeled human memory structures as divided into long-term (knowledge) and working memory (short-term store). We have implemented working memory, described by Baddeley and Hitch (1974) as composed of a central control processor (of some limited capacity), an ''articulatory loop'' (temporary storage of speech-based information), and a ''visuo-spatial scratch pad'' (temporary storage of spatial information). The point of transference of information from the flight deck and ATC displays to the operators' working memory is the critical juncture in the subsequent use of the information that is exchanged.

Long-term memory structure is provided via a semantic net. The interaction of procedure with memory is provided by a goal decomposition method implemented as a form of cognitive schema. In order to capture the central role of schema and internal representation, we have an elaborate representation of both declarative and procedural information in the MIDAS model. In MIDAS, the internal updatable world representation (UWR) provides a structure whereby simulated operators access their own tailored or personalized information about the operational world. The structure and use of the UWR are akin to human long-term memory and are one of the aspects of MIDAS unique from most human–system modeling tools. UWR contents are defined by presimulation loading of required mission, procedural, and equipment information. Data are then updated in each operator's UWR as a function of the mediating perceptual and attentional mechanisms previously described. These mechanisms function as activation filters, allowing more or less of the stimuli in the modeled environment to enter the simulated operator's memory. Knowledge of what is on each operator's mind is a key modeling feature that allows MIDAS to examine decision making and the information exchange that is critical to decision making.

6.3.2. *Attentional Control*

Representation of human–automation integration requires functions of attentional control and concurrent task performance. Distributed attention and attention switching refer to an operator's ability to perform multiple tasks simultaneously. In many cases, a second task can be added to the performance of a primary task with little or no impact to the performance of the first task. In other cases, the performance of two tasks simultaneously has a disastrous interaction. Such context-and order-sensitive effects are determined in the scheduling and agenda management function provided in the MIDAS model. Attention capture functions are represented through a preattentive filter mechanism that responds to physical characteristics of environmental stimuli (e.g., color, blinking, auditory characteristics).

6.3.3. *Activity Representation*

Tasks or activities available to an operator are contained in that operator's UWR and generate a majority of the simulation behavior. Within MIDAS, a hierarchical representation is used (similar to, but more flexible than, the mission–phase–segment–function–task decomposition employed by many task-analysis systems). Each activity contains slots for attribute values, describing, for example, preconditions, temporal or logical execution constraints, satisfaction conditions, estimated duration, priority, and resource requirements. A continuum of contingent or decision-making behavior is also represented in MIDAS, following the skill, rule, knowledge-based distinction reported by Rasmussen (1983). The activity structures in MIDAS are currently being implemented as sketchy plans in the reactive action packets paradigm (RAPS) of Firby (1989). This structure of activities will interact with resource and context mangers to structure an agenda.

6.3.4. *Task Agenda*

The agenda structure stores instantiated RAPS as goals with subnetworks and logical control flags, object bindings, and history of state and completion. This network represents the current set of tasks to be performed by the operators of the simulation given the current goals and context. The network can complete successfully, be interrupted by other task networks, or be aborted. The relationship among the actions in terms of logic of performance (e.g. sequential or concurrent tasks) is also specified in the agenda structure. Whether, in fact, tasks can be performed concurrently is a function of resource relations in the cognitive model (sensation/reception, central/attentional/effectors). Work is currently underway to unify the representation of action and resources in the various version of the MIDAS system.

6.3.5. *Decision Making*

Quick, skill-based, low-effort responses to changes in values of information held in the UWR are captured by ''daemons'' when a triggering state or threshold value, sensed by perception, is reached. Daemons represent well-trained behaviors such as picking up a ringing phone or extinguishing a caution light. Classic production rule-based behavior is also available and is used when conditions in the simulation world match user-defined rule antecedent clauses active for the scenario modeled. Finally, more complex or optimization-oriented decision making is represented via a set of six prescriptive algorithms (e.g., weighted additive, elimination by aspect, etc.), as reported by Payne et al. (1988). Each of these algorithms uses a different combination of attribute values, weights, and cut-off values for calculating the "goodness" of the options.

6.3.6. Higher-Level Functions

The cognitive submodel architecture allows for the development of higher-order functions of cognition. For example, Shiveley et al. 1995 has developed and demonstrated a "situation awareness" function that combines characteristics of working memory, long-term memory search, and perceptual models to develop an abstraction termed situation, which can then be used to guide behavior or to serve as a measure of adequacy of information in the environment and in the crew's knowledge store to meet task demands.

Context is a combination of declarative memory structures and incoming world information that is mapped to the agenda manager who is taking the plan (overall mission). This, combined with the plan interpreter, provides a series of actions to be performed in order to meet mission goals and handle contingent activities (such as interruption or plan repair). Verma (2000) has explored the extension of situation to "contextual control" (Hollnagel 1993) by using the elements of rule-based behavior, number of goals in working memory, and decision horizon in the MIDAS planning module.

6.4. Case Studies in MIDAS Applications to Aviation

The world community of aviation operations is engaged in a vast, system-wide evolution in human–system integration. The nature of this change is to relax restrictions in air transport operations wherever it is feasible. The relaxation includes schedule control, route control, and, potentially, separation authority in some phases of flight, such as aircraft self-separation in en route and oceanic operations. The consistent result of the relaxation of system constraints is to change and challenge human performance in that system in two dimensions. First, the decision-making process becomes distributed. This distributed decision differs from current operation and has a direct impact on crew and team resource management processes. Second, the dynamic concept of operations provides a new challenge to the human operators of that system. The human operators (pilots, air traffic controllers, and airline operations personnel) must monitor and predict any change in the distribution of authority and control that might result as a function of the airspace configuration, aircraft state or equipage, and other operational constraints. The operators are making decisions and sharing information not only about the management of the airspace, but also about the operating state of that airspace. In order to support collaborative and distributed control between air and ground for separation assurance, modifications to the roles and decision authority must be explored. The evolution of the air transportation system, therefore, profoundly challenges human performance prediction and the cognitive sciences. Previous models of human performance linked to machine performance have had distinct boundaries in the human–machine elements to be modeled. Current system design requires models of human, machine/automation, aircraft, airline operations, air traffic management, and National Airspace (NAS) management to be tightly coupled in order to guide design, evaluate the effectiveness, and ensure the safe operation of the system.

6.4.1. MIDAS Case Study 1: Predicting Flight Crew Performance in the Advanced Air Traffic Management System

We have focused our early investigation on critical issues in air ground coordination in relation to aircraft self-separation. The interaction among aircraft and controllers is proposed to occur at points in space around each aircraft called *alert* and *protected zones*.

These zones are to be used by an alerting system to monitor and advise the flight crew on conflicting traffic flying within these areas. In a cockpit-based system, the alerting system would warn the flight crew of any aircraft entering the alert zone. The crew could evaluate the situation and choose or negotiate a preferred deviation. If the intruding aircraft continued into the smaller warning zone, the crew would be advised to take evasive action.

Much discussion and debate has gone into the further definition of the warning and alert zones, including their description as complex surfaces that take into account the speed, performance, and turning radius of the aircraft. Up to this time, the process of definition lacked data for inclusion of human performance in the size and shape of these areas. Figure 6 proposes a redefinition of these zones based on a human–machine system performance. Built upon the well-defined physical aerodynamic response of the aircraft are the more varying machine (sensing, communication, computation) and human (perception, communication, decision, action) responses to any alert. These zones might also differ, depending on the speed of the aircraft, configuration of the conflict, and procedures used to process the conflict.

6.4.1.1. Study 1: Model Analysis The goal of this study was to develop a better understanding of the impact of joint and distributed decision making among flight crews and air traffic control on

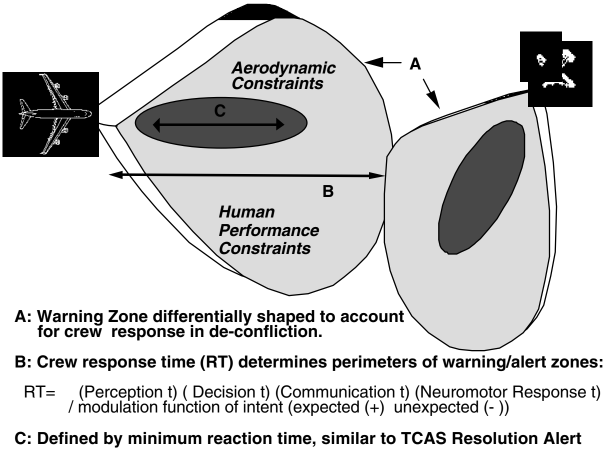

Figure 24 Alert and Protected Zones Calibrated to Human Performance Parameters, Aircraft Performance Parameters, and Communication Systems Parameters.

the size and shape of the alert zones. This was accomplished by first analyzing and modeling the cognitive and procedural requirements of several candidate encounter scenarios. These models were then populated with performance data derived from human-in-the-loop experiments. The specified scenarios were then represented within the MIDAS computational modeling and simulation system.

With Monte Carlo simulation techniques, each scenario could be exercised many times, eventually establishing a statistical distribution for the human–machine performance of that configuration. Combining this with the aerodynamic performance of the system (in this case the closing speed of conflicting aircraft at differing encounter angles) meant that the differences in warning requirements between the different scenarios should have emerged. All encounters were assumed to be two-ship interactions.

- *Scenario 1:* In this scenario, both aircraft are equipped with some type of CDTI detection equipment. Here a single aircraft can detect and avoid the conflicting aircraft by acting on its own (no communications are necessary). This might describe a situation where one aircraft is slowly closing on another from behind.
- *Scenario 2:* Both aircraft are again equipped. However, because of the geometry of the encounter and conflicting goals, both aircraft must be involved and negotiate to resolve the problem. The solution would be arrived at though communication and negotiation between the two flight crews.
- *Scenario 3:* This scenario describes an encounter where communications with ATC are required to resolve the problem. This is necessary because one aircraft is equipped with the required suite of equipment while the other is not. Such encounters might be common early in the implementation of free flight or when encountering older, nonupgraded aircraft.

6.4.1.2. High-Level Activity Definition in the MIDAS Model An initial cognitive and physical task analysis was performed for each of the three scenario cases. The result was a sequential model identifying the high-level processes (or activities) performed by the operators. In scenarios 2 and 3, the activities that were to be performed in parallel by the other flight crew and ATC were also defined. Falling out of this analysis was a recognizable cycle of alert, recognition, communication, decision, then communication and action by the crews. This process was replicated throughout the scenarios for each flight crew interaction.

6.4.1.3. Lower-Level Activity Specification Using these sequences as a guide, the lower, or leaf-level, activities (corresponding to the physical or cognitive tasks actually performed by the operators) were defined for each high level task. Columns 2 and 3 of Figure 7 show the high- and lower (leaf)-level activities defined for scenario 1. The remaining columns show the interrupt recovery, duration, and VACM (visual, auditory, cognitive, and motor channel capacity requirements) specifications assigned to those activities. Where possible, the activities were chosen to correspond to research that had been performed in previous studies (Corker and Pisanich 1995). This provided access to fully defined activity specifications. New activities, along with their specifications, were developed by interpolating prior results.

The MIDAS model can contain activities that may interrupt the flight crew from the normal activities (for example, a question in the cockpit may interrupt a flight crew member from a CDU entry task). The interrupt resumption specifications define how an activity is resumed after being suspended. Resumption methods are individually defined based on the characteristics of the activity and the sequence in which it operates. The resumption methods used on this simulation include not-interruptible (cannot be interrupted); resume (resume activity where interrupted); and restart (restart the activity from its beginning). Interruptions and the way that an activity is resumed directly affect the duration of the activity sequence.

6.4.1.4. Experimental Runs After specification and testing, each scenario was loaded into Air-MIDAS and 50 Monte Carlo runs were gathered for that simulation. The data recorded for each run included the activity sequence along with the individual activities and their duration for that sequence (including any interrupt activities). These data were written to a file for analysis in Microsoft Excel format. This data was postprocessed using the rules described earlier to extract a proper time for the parallel activity sets. This allowed the establishment of a total duration (time required for all operators to complete their tasks) for each scenario run. This was the dependent variable in this study.

6.4.1.5. Results A standard set of descriptive statistics was generated for each scenario based on the set of 50 Monte Carlo runs. The temporal performance data were also plotted as a histogram using a bin size of 10 seconds, illustrated in Figure 25.

The performance observed in each scenario above was applied to a 90° crossing conflict. In this geometry, the initial traffic alert was proposed to be signaled at 40 miles from the crossing point and assumed a typical commercial aircraft cruise speed (Mach 0.82). The measure in this case was the closing distance (straight-line distance between the aircraft). The minimum, maximum, and average human–machine performance times are illustrated in Figure 25. This calculation also allowed the determination of a closing distance for each performance time (potentially from 56 down to 0 miles). Using this geometry, an idea of the initial warning distance could be inferred using the worst-case performance criteria. Although the average clearing distances in scenarios 1 and 3 differ significantly, given a warning at 40 miles, the worst-case time in both scenarios would allow an avoidance maneuver to begin well before the aircraft were 20 miles from each other. In scenario 3, however, the

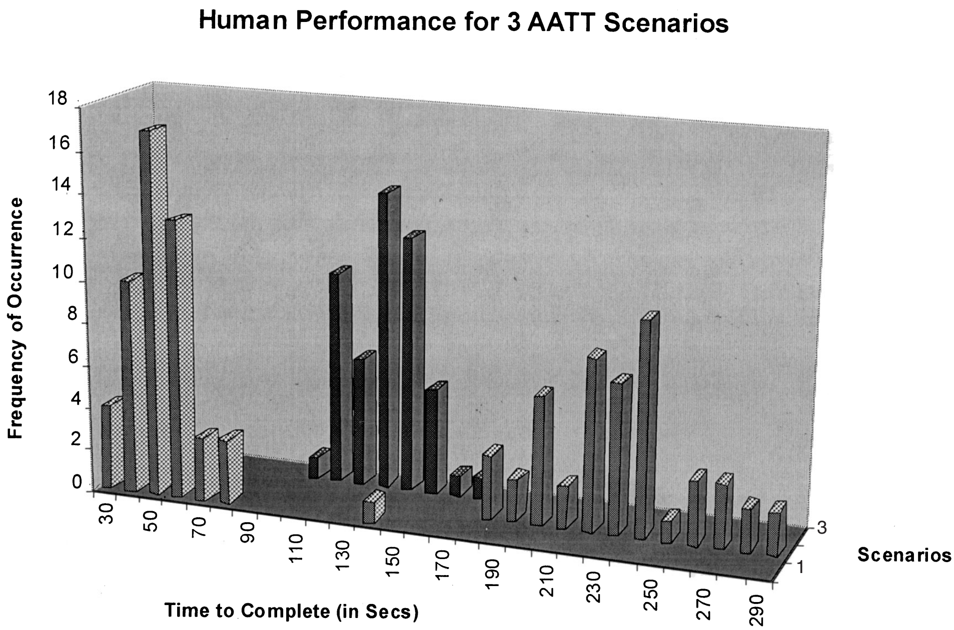

Figure 25 Response Times for an Air-to-Air Encounter at 90° Intercept.

worst, and even average, clearing distances observed would indicate that the alert point for that type of interaction should be initiated well beyond 40 miles.

To investigate this idea further, a second application of the human performance data was performed. Calculations were again made with both aircraft maintaining a speed of Mach 0.82. For each 15° angle around the aircraft, the resulting closing speed was calculated. Combining that speed and the performance distribution of each scenario resulted in a distance traveled for that angle. In this case, two standard deviations above and below the average were used as the minimum and maximum points respectively. Five miles were added to these distances to account for the warning zone. When plotted, these points create the heart-shaped rosettes shown in Figure 26.

In addition to showing the difference in warning distance needed to maintain the same performance at differing closing angles, these plots are interesting because they also illustrate a difference in performance area (size of the area between minimum and maximum performance) between the three scenarios. Given the performance observed, the higher closing speeds actually exacerbate the differences between the scenarios. Although scenarios 1 and 3 looked comparable in the 90° closure shown earlier, at shallower angles scenario 3 actually requires a significantly earlier warning point to maintain the five mile alert zone.

6.4.2. Extension of Model to Air Traffic Control

The second example of MIDAS use concentrates on the air traffic controller operations in response to free-flight self-maneuvering aircraft. In this study, the human operator's performance measures in the distributed air/ground air traffic management (ATM) that characterizes the multiple-controller, multiple-aircraft system include visual monitoring, perception, spatial reasoning, planning, decision making, communication, procedure selection, and execution. Two scenarios will be created in the current modeling effort. The first scenario will be operated consistent with the air traffic control rules of flight. The second scenario will be operated consistent with free flight rules of operation established by RTCA. Each scenario involves a response to a scripted conflict situation in a number of conditions.

The MIDAS model of the controller was developed using the rules of current and free-flight operation. Estimates and measured times for controllers performance on these tasks was input into the task descriptions. Here the interest was in how stable the system was to emergency conditions under current and free-flight conditions. The control scenario was focused on the handoff between

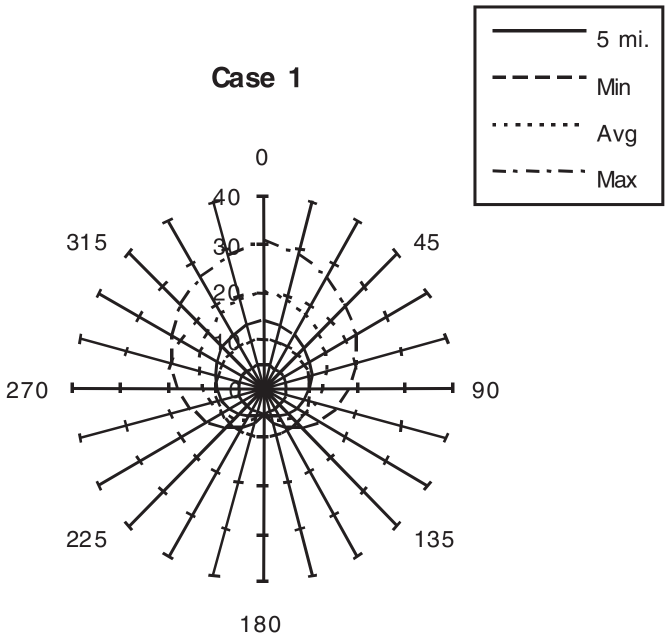

Figure 26 Cartoid Shape of Minimum Safe Distance to Alert Calculated as a Function of Model-Generated Crew Response Times.

sectors in an enroute segment of flight. This scenario then required the development and interaction of several MIDAS human performance models. There were two controllers modeled that interacted with a single aircraft flight crew. The scenario is illustrated in Figure 27.

Results were compiled for various conditions of weather and emergency management under the two flight regimes and controllers' performance profiles for workload were developed. In addition, operational measures, such as aircraft maneuvers and points of closest approach, were also calculated. The details of these analyses can be found in Gore (1999).

6.5. Other Modeling Strategies That Have Demonstrated Utility in Modeling Human Performance in Systems

There has been a flurry of interest over the past decade in the use of object-oriented tools that focus on the representation of human knowledge and how humans use that knowledge in real situations to make decisions and act. An example of this approach is the Distributed Operator Model Architecture (DOMAR), developed by Dr. Michael Young of the Wright Patterson Air Force Research Laboratories and Mr. Stephen Deutsch of BBN Inc. (Young and Deutsch 1999). The system is composed of a framework of software languages and model-development tools. As with the other models discussed in this section, the intention is to provide the analyst with the necessary components to develop models that simulate both the human operators and the systems with which they interact.

DOMAR is unique in its intention to allow these models to interact not only to allow with other entities in the model environment, that is, with simulations, but to allow the human performance models developed in OMAR to interact with live human operators and real systems in a hybrid human–simulation modeling capability.

The component software tools that are part of DOMAR allow the developers to construct their own unique cognitive architecture for the operators simulated in DOMAR.

DOMAR is made up of tools to support the development of appropriate cognitive models of human operators interacting with systems. The following description is provided by Young and Deutsch (1999):

- The Simple Frame Language (SFL) is used to represent knowledge about objects, their attributes, and the relationships among objects. It provides an object-oriented substrate for the entire simulation environment. One of the most important classes of objects is an agent—the special type of object that can execute procedures and thereby run in the DOMAR simulator. The Rule Definition Language (RDL) provides computational primitives for representing rules (i.e., condition action pairs of the form if-then). Rules are developed as rule packets, which specify when and how the rules are to be applied to a specific situation. The Simulation Core Language

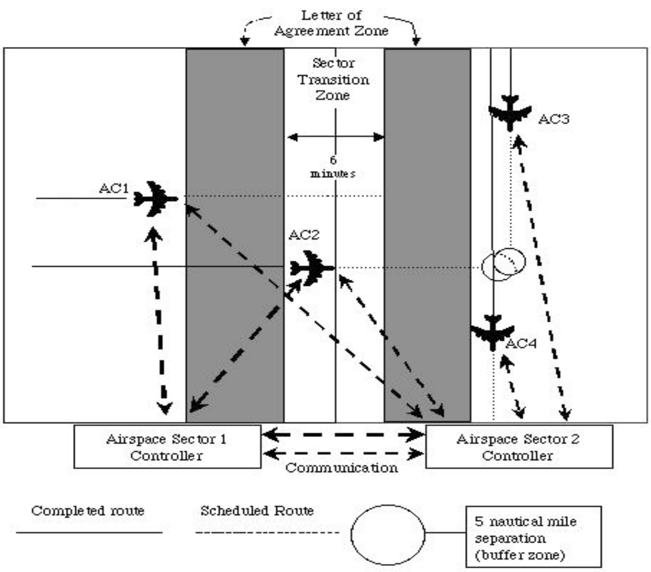

Figure 27 Cross-Sector Handoff Scenario

(SCORE) is a procedure definition language that provides a set of computational primitives for modeling the priority-based serial or parallel execution of goals, plans, and tasks.

- The DOMAR publish–subscribe protocol, implemented in a set of SCORE forms, plays a central role in several important aspects of model building. Within the human performance models themselves, it is used to coordinate the concurrent operation of related procedures. It is frequently employed in the modeling of person-to-person communication, and it is the basis for the distributed execution, in real time or fast time, that DOMAR supports. Very large agent-based simulations may be developed in a distributed computing environment.
- Knowledge engineering: DOMAR provides graphical tools to support knowledge acquisition, the development and management of large bodies of code, and data-analysis tools to support model development and experiment evaluation. SFL objects and agents are defined using the graphical concept editor, which provides an interactive graphical display of the hierarchical relationship among object definitions as a directed network of connected nodes. It provides a table window for displaying and editing slots associated with objects. SCORE procedures are created with a text editor (e.g., EMACS). They can then be viewed with the procedure browser, which provides several different graphical views of the programmatic interrelationships among the procedures.
- Analysis capabilities: DOMAR provides tools for monitoring events as reported by the simulator during a model run. Events that can be monitored include goal events, procedure events, stimulus-and-response events, and user-defined events such as communication events. Using a menu, the DOMAR user chooses the events to be displayed before the start of a simulation or at any point during a run. Finally, DOMAR has two post-run analysis displays. The event timeline window displays a listing of time-tagged events for one or more simulation agents. Individual agents have their own timelines within the display. The task timeline window displays the set of concurrently running tasks for one or more agents over a specific period of time. It provides the start and end time for a task and the task's status and priority. All of the displays are written in the Java language and can be connected to any DOMAR image that may be running on a local or remote computer.

6.5.1. Sample Applications

DOMAR was created to support the development of human performance models. DOMAR's sophisticated agent framework has recently been used by other researchers to develop a variety of agent-based systems. DOMAR is being used by AFRL in its Human Interaction with Software Agents (HISA) project to create human–computer interfaces incorporating intelligent agents; by the Defense Advanced Research Project Agency (DARPA) in its Collaborating Agent Based Systems (CoABS) program to demonstrate a collaborating agent communication framework and in its Joint Forces Air Component Commander (JFACC) program to create a business enterprise model of the JFACC process; and by the Navy in its Conning Officer Virtual Environment (COVE) program to create an intelligent training system.

7. SUMMARY

This chapter has provided the need for simulating performance of complex human-based systems as an integral part of system design, development, testing, and life-cycle support. It has also defined two fundamentally different approaches to modeling human performance: a reductionist approach and a first-principled approach. Additionally, we have provided detailed examples of two modeling environments that typify these two approaches, along with representative case studies.

As we have stated and demonstrated repeatedly throughout this chapter, the technology for modeling human performance in systems is evolving rapidly. Furthermore, the breadth of questions being addressed by models is constantly expanding. We encourage the human factors practitioner with a little creativity and computer savvy to consider how computer simulation can provide a better and more cost-effective basis for human factors analysis.

REFERENCES

Allender, L. (1995), personal communication, December.

Allender, L., Lockett, J., Headley, D., Promisel, D., Kelley, T., Salvi, L., Richer, C., Mitchell, D., and Feng, T. (1994), ''HARDMAN III and IMPRINT Verification, Validation, and Accreditation Report,'' Prepared for the US Army Research Laboratory, Human Research and Engineering Directorate, December.

Archer, S. G. ,and Adkins, R. (1999), ''Improved Performance Research Integration Tool (IMPRINT) Version 5 User's Manual,'' Prepared for the the US Army Research Laboratory, Human Research and Engineering Directorate, December.

Arditi, A., and Azueta, S. (1992), "Visualization of 2-D and 3-D Aspects of Human Binocular Vision," Presented at the Society for Information Display International Symposium (available from the Lighthouse, Inc.).

Atencio, A., Shively, R. J., and Shankar, R. (1996), "Evaluation of Air Warrior Baselines in a Longbow Apache Helicopter Crewstation in a MIDAS Simulation," American Helicopter Society 52nd Annual Forum, Washington, DC.

Atencio, A., Banda, C., and Tamais, G. (1998), "Evaluation of a Short Haul Civil Tiltrotor in Emergency Go-Around—A MIDAS Simulation," American Helicopter Society 54th Annual Forum, Washington, DC.

Baddeley, A. D., and Hitch, G. J. (1974), "Working Memory," in *Advances in Research and Theory,* Vol. 8 of The Psychology of Learning and Motivation, G. H. Bower, Ed., Academic Press, New York, pp. 47–90.

Badler, N., Phillips, C., and Weber, B. (1993), *Simulating Humans: Computer Graphics, Animation and Control,* Oxford University Press, New York.

Baron, S., Zacharias, G., Muralidharan, R., and Lancraft, R. (1980), "PROCRU: A Model for Analyzing Flight Crew Procedures in Approach to Landing," in *Proceedings of the Eighth IFAC World Congress* (Tokyo).

Bierbaum, C., Szabo, S., and Aldrich, T. (1989), *Task Analysis of the UH-60 Mission and Decision Rules for Developing a UH-60 Workload Prediction Model.* US Army Research Institute Aviation R&D Activity, Fort Rucker, AL.

Boettcher, K., Riley, V., and Collins, C. (1989), "A Modeling Approach for Analyzing Human-Machine Interaction Dynamics in Computer Systems," in *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics* (Cambridge, MA).

Boff, K. R., Kaufman, L., and Thomas, J.P. (1986), *Handbook of Perception and Cognition,* John Wiley & Sons, New York.

Card, S. K., Moran, T. P., and Newell, A. (1983), *The Psychology of Human–Computer Interaction,* Erlbaum, Hillsdale, NJ.

Corker, K. M., and Pisanich, G. M. (1995), "Analysis and Modeling of Flight Crew Performance in Automated Air Traffic Management Systems," Presented at: 6th IFAC/IFIP/IFORS/IEA Symposium: Analysis, Design, and Evaluation of Man–Machine Systems, Boston.

Corker, K. M., and Smith, B. (1992), "An Architecture and Model for Cognitive Engineering Simulation Analysis: Application to Advanced Aviation Analysis," Presented at AIAA Conference on Computing in Aerospace, San Diego.

Dahl, S. G., Allender, L., Kelley, T., and Adkins, R. (1995), Transitioning Software to the Windows Environment—Challenges and Innovations," in *Proceedings of the 1995 Human Factors and Ergonomics Society Meeting, Human Factors and Ergonomics Society* (Santa Monica, CA, October).

Farmer, E. W., Belyavin, A. J., Jordan, C. S., Bunting, A. J., Tattershall, A. J., and Jones, D. M. (1995), "Predictive Workload Assessment: Final Report," Report No. DRA/AS/MMI/CR95100/, Defence Research Agency, Farnborough, UK, March.

Firby, R. J. (1989), "Adaptive Execution in Complex Dynamic Worlds," Ph.D. Thesis, Yale University, Technical Report YALEU/CSD/RR #672.

Gawron, V. J., Laughery, K. R., Jorgensen, C. C., and Polito, J. (1983), "A Computer Simulation of Visual Detection Performance Derived from Published Data," in *Proceedings of the Ohio State University Aviation Psychology Symposium* (Columbus, April).

Gore, B., F., and Corker, K. M. (1999), "System Interaction in 'Free Flight': A Modeling Tool Cross Comparison," in *Proceedings of the Digital Human Modeling Conference and Exposition,* SAE International Paper # 199-01-1987, Warrendale, PA, June.

Gray, W. D., John, B. E., and Atwood, M. E. (1993), "Project Ernestine: Validating a GOMS Analysis for Predicting and Explaining Real-World Task Performance," *Human–Computer Interaction,* Vol. 8, pp. 237–309.

Hahler, B., Dahl, S., Laughery, R., Lockett, J., and Thein, B. (1991), "CREWCUT—A Tool for Modeling the Effects of High Workload on Human Performance," in *Preceedings of the 35th Annual Human Factors Society Conference,* (San Francisco).

Hamilton, D. B., Bierbaum, C. R., and Fulford, L. A. (1990), "Task Analysis/Workload (TAWL) User's Guide, Version 4," Research Project 91-11, U.S. Army Research Institute for the Behavioral and Social Sciences (AD A241 861), Alexandria, VA.

Hoagland, D., Martin, E., Anesgart, M , Brett, B., LaVine, N., and Archer, S. (2001), "Representing Goal-Oriented Human Performance in Constructive Simulations: Validation of a Model Performing Complex Time-Critical-Target Missions,'' in *Proceedings of the Simulation Interoperability Workshop* (Orlando, FL, April).

Hollnagel, E. (1993), *Human Reliability Analysis: Context and Control,* Academic Press, London.

Knapp, B., Archer, S. G., Archer, R. D., and Walters, B. (1999), ''Innovative Approaches to Modeling – An Application for National Missile Defense,'' in *Proceedings of the Society for Computer Simulation Conference* (Chicago).

LaVine, N. D., Peters, S. D., and Laughery, K. R. (1995), ''A Methodology for Predicting and Applying Human Response to Environmental Stressors,'' Micro Analysis & Design, Inc., Boulder, CO, December.

Lawless, M. L., Laughery, K. R., and Persensky, J. J. (1995), ''Micro Saint to Predict Performance in a Nuclear Power Plant Control Room: A Test of Validity and Feasibility,'' NUREG/CR-6159, Nuclear Regulatory Commission, Washington, DC, August.

Little, R., Dahl, S. G., Plott, B., Wickens, C., Powers, J., Tillman, B., Davilla, D., and Hutchins, C. (1993), ''Crew Reduction in Armored Vehicles Ergonomic Study (CRAVES),'' Report No. ARL-CR-80, prepared for the Army Research Laboratory, July.

Lubin, J., and Bergen, J. (1992), ''NASA Cockpit Display Visibility Modeling Project,'' Final Report NAS2-12852, SRI/David Sarnoff Research Center, Moffett Field, CA.

McCracken, J. H., and Aldrich, T. B. (1984), ''Analysis of Selected LHX Mission Functions: Implications for Operator Workload and System Automation Goals,'' Technical Note ASI 479-024-84(B), prepared by Anacapa Sciences, Inc., June.

McMillan, G. R., Beevis, D., Salas, E., Strub, M. H., Sutton, R., and Van Breda, L. (1989), *Applications of Human Performance Models to System Design,* Plenum Press, New York.

Micro Analysis and Design (1999), *Micro Saint, Version 3.0: User's Guide, Micro Analysis and Design,* Boulder, CO, May.

Newell, A. (1990), *Unified Theories of Cognition,* Harvard University Press, 1990.

Newell, A., and Simon, H. A. (1972), *Human Problem Solving,* Prentice Hall, Englewood Cliffs, NJ.

North, R. A., and Riley, V. (1989), ''W/INDEX: A Predictive Model of Operator Workload,'' in *Applications of Human Performance Models to System Design,* G. McMillan et al., Eds., Plenum Press, New York.

O'Hara, J. M., Brown, W. S., Stubler, W. F., Wachtel, J. A., and Persensky, J. J. (1995), ''Human–System Interface Design Review Guideline: Draft Report for Comment,'' NUREG-0700 Rev.1, U.S. Nuclear Regulatory Commission, Washington, DC.

Payne, W., Bettman, J. R., and Johnson, E. J. (1988), ''Adaptive Strategy in Decision Making,'' *Journal of Experimental Psychology: Learning, Memory and Cognition,* Vol. 14, No. 3, pp. 534–552.

Plott, B. (1995), ''Software User's Manual for WinCrew, the Windows-Based Workload and Task Analysis Tool,'' U.S. Army Research Laboratory, Aberdeen Proving Ground, MD.

Rasmussen, J. (1983), ''Skills, Rules, and Knowledge; Signals, Signs and Symbols, and Other Distinctions in Human Performance Models,'' *IEEE Transactions on Systems, Man, and Cybernetics,* Vol. SMC-13, No. 3, pp. 257–266.

Remington, R. W., Johnston, J. C., and Yantis, S. (1992), ''Involuntary Attentional Capture by Abrupt Onsets,'' *Perception and Psychophysics,* Vol. 51, No. 3, pp. 279–290.

Roth, J. T. (1992), ''Reliability and Validity Assessment of a Taxonomy for Predicting Relative Stressor Effects on Human Task Performance,'' Technical Report 5060-1, prepared under contract DNA001-90-C-0139, Micro Analysis and Design, Inc., Boulder, CO, July.

Shively, R. J. et al. (1995), ''MIDAS Evaluation of AH-64D Longbow Crew Procedures in a Air-Ground Flight Segment: MOPP versus Unencumbered,'' Sterling Federal Systems, NASA Ames Research Center.

Siegel, A. I., and Wolf, J. A. (1969), *Man–Machine Simulation Models,* Wiley-Interscience, New York.

Tyler, S., Neukom, C., Logan, M., and Shively, J. (1998), ''The MIDAS Human Performance Model,'' in *Proceedings of the Human Factors and Ergonomics Society 42nd Annual Meeting* (Chicago), pp. 320–325.

Verma, S. (2000), ''Introduction of Context in Human Performance Models as Applied to Dynamic Resectorization,'' Master's thesis, San José State University, Human Factors and Ergonomics Program, San José, CA.

Wickens, C. D. (1984), *Engineering Psychology and Human Performance,* Merrill, Columbus, OH.

Wickens, C. D. (1989), ''Models of Multitask Situations,'' in *Applications of Models to System Design*, G. McMillan, Ed., Plenum Press, New York, pp. 259-273.

Wickens, C. D., Sandry, D. L., and Vidulich, M. (1983), "Compatibility and Resource Competition Between Modalities of Input, Central Processing, and Output," *Human Factors,* Vol. 25, pp. 227–248.

Wortman, D. B., Duket, S. D., Seifert, D. J., Hann, R. l., and Chubb, A. P. (1978), *Simulation Using SAINT: A User-Oriented Instruction Manual,* Aerospace Medical Research Laboratory, AMRL-TR-77-61. Wright-Patterson Air Force Base, OH, July.

Young, M. and Deutsch, S. (1999), www.he.afrl.af.mil/hes/hess/programs/omar/2omarnext.html.

ADDITIONAL READING

Agha, G., "Actors: A Model of Concurrent Computation in Distributed Systems," Technical Report 844, MIT Artificial Intelligence Laboratory, Cambridge, MA, 1985.

Allender, L., Kelley, T., Salvi, L., Headley, D. B., Promisel, D., Mitchell, D., Richer, C., and Feng, T., "Verification, Validation, and Accreditation of a Soldier-System Modeling Tool," *Proceedings of the 39th Human Factors and Ergonomics Society Meeting* (San Diego, October 9–13, 1995) (available from the Human Factors and Ergonomics Society, Santa Monica, CA).

Archer, R., Drews, C. W., Laughery, K. R., and Dahl, S. G., "Data on the Usability of Micro Saint," in *Proceedings of NAECON Meeting* (Dayton, OH, May 1986).

Baron, S., and Corker, K., "Engineering-Based Approaches to Human Performance Modeling," in *Applications of Human Performance Models to System Design,* Grant McMillan, Ed., Plenum Press, New York, 1989.

Corker, K. M., and Pisanich, G. M., "When Reasonable Expert Systems Disagree," Presented at the Topical Meeting of the American Nuclear Society, Computer-Based Human Support Systems: Technology, Methods, and Future, Philadelphia, 1995.

Corker, K. M., Lozito, S., and Pisanich, G., "Flight Crew Performance in Automated Air Traffic Management," in *Human Factors in Aviation Operations,* R. Fuller, N. Johnston, and N. McDonald, Eds., Vol. 3, Avebury Aviation, Hants, UK.

Corker, K. M., and Smith, B., "An Architecture and Model for Cognitive Engineering Simulation Analysis: Application to Advanced Aviation Analysis," Presented at AIAA Conference on Computing in Aerospace, San Diego, 1993.

Craik, K. J. W., "Theory of the Human Operator in Control Systems I. The Operator as an Engineering System," *British Journal of Psychology,* Vol. 38, 1947, pp. 56–61.

Drews, C., and Laughery, K. R., "A Modeling System Designed Around the User Interface," Presented at the Summer Computer Simulation Conference, Chicago, July 1985.

Drews, C., Laughery, R. R., Kramme, K., and Archer, R., "LHX Cockpits: Micro SAINT Simulation Study and Results," Report prepared for Texas Instruments—Equipment Group, Dallas, June 1985.

Gore, B., "A Comparison of Human Performance Models Applied to Advanced Air Traffic Management Operations," Master's thesis, San José State University, San José, CA.

Reason, J. T., "The Chernobyl Errors," *Bulletin of the British Psychological Society,* Vol. 40, 1987, pp. 201–206.

Reason, J. T., *Human Error,* Cambridge University Press, Cambridge, 1990.

Remington, R. W., Johnston, J. C., and Yantis, S., "Involuntary Attentional Capture by Abrupt Onsets," *Perception and Psychophysics,* Vol. 51, No. 3, pp. 279–290.

Shankar, R., "Z-Scheduler: Integrating Theories of Scheduling Behavior into a Computational Model," in *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics,* pp. 1219–1223.

Tustin, A., "The Nature of the Operator's Response in Manual Control and Its Implication for Controller Design," *Journal of the IEE,* Vol. 94 (Part IIA, No. 2).

Tversky, A., and D. Kahneman, "Judgment under Uncertainty: Heuristics and Biases," *Science,* Vol. 185, 1974, pp. 1124–1131.