International Journal of Systems Science, 2000, volume 31, number 6, pages 713–729

A rule-based pilot performance model

MICHAEL S. MCCOY† and REUVEN R. LEVARY‡

A new method of modelling task execution time and its relationship to resource utilization has been developed in this study in order to simulate accurately pilot performance in a single-seat fighter aircraft. By developing the model in a rule-based expert system program, it is suitable for incorporation in an intelligent decision support system to help manage pilot workload and improve total man–machine system performance. This model can also be employed in the design of new aircraft crewstations to analyse new man–machine system interfaces. Validation of the model was accomplished through comparison of critical predicted measures of merit against observed pilot performance in a manned flight simulator.

1. Introduction

Human performance models are computer-based mathematical simulation models used to predict operator decision making, task and procedure execution, and mental workload. These models have been applied in the design of aircraft cockpits, nuclear power plant control stations, and air traffic control centres to anticipate problems with operability of the system. More recently proposed applications include supervisory control systems and decision aiding systems (Sheridan 1992). These new applications require a high level of precision in modelling the human decision making process and behaviour.

Early human performance models were developed with the sensory, organism and response construct (Siegel and Wolf 1969, Asiala *et al.* 1982). These models did not account for conflict between resource usage in the performance of tasks in parallel. Citing this deficiency, Wickens (1985) developed a multiple resource theory that used expert subjective opinion to estimate the percentages of resources required to perform tasks. Then, when the sum of demanded resources exceeded the availability of resources, tasks were shed or delayed until resources became available. However, this method depended on a highly subjective estimate of the percentage of resources required by the operator in the performance of the tasks. In addition, Wickens did not account for the extra time required when operators performed tasks in parallel, rather than singularly, i.e. in sequential order. Therefore, using measured task performance times, a new method of modelling the relationship between task execution time and resource utilization was developed in this study to account for these deficiencies, and to simulate more accurately human behaviour in a real-world, complex operating environment.

A human–machine system consists of a computer control system designed to provide an interface between a human operator and a task environment to achieve specific goals, such as navigating through the sea in a ship, generating electricity in a nuclear power plant, managing air traffic in an airport, or flying an aircraft from an origin to a destination.

Human performance models have been classified into four major categories (*a*) information processing, (*b*) control theory, (*c*) task network, and (*d*) knowledge based. Each type of model has advantages and disadvantages that must be weighed when selecting a technique for a specific application.

Information processing models were best typified by Senders' (1977) model and the Human Operator Model (Wherry 1976, Lane *et al.* 1981). The strengths of these models were generally detailed enough to produce data much like that collected in human-in-the-loop simulation experiments. Sub-models were aggregated to produce total task times. The models often were sensitive to equipment layout. However, the major weaknesses of information processing models involved their

Received 11 February 1998. Accepted 16 April 1999.

† The Boeing Corporation, Strategic Analysis and Planning, M.S. 064 2233, P.O. Box 516, St. Louis, MO 63166-0516, U.S.A.

‡Department of Decision Sciences and MIS, School of Business and Administration, St. Louis University, 3674 Lindell Blvd. St. Louis, MO 63108, U.S.A.

International Journal of Systems Science ISSN 0020–7721 print/ISSN 1464–5319 online © 2000 Taylor & Francis Ltd
http://www.tandf.co.uk/journals

predominantly micro-task orientation. This made them unsuitable for high-level analysis. Also, since information processing models were comprised of many sub-models, they were hard to validate.

Three control theory models best represented this category of human performance model. First, a frequency domain model, employing classical control theory, was developed (McRuer and Krendel 1974). This model was a quasi-linear describing function model that simulated manual control. Next, Baron and Levison (1980) developed the Optimal Control Model that utilized modern control theory to model manual control in the time domain. Finally, a combination of procedures, time-domain control theory, and rule-based modelling was employed in the Procedure Oriented Crew Model (PROCRU) developed by Baron and associates (Baron *et al.* 1980).

An advantage of control theory models was that their modular structure allowed sub-models to be incorporated into them. PROCRU permitted information processing to be incorporated in the task selection routines along with priority setting for task selection and execution. Disadvantages of these models included: lack of validation of the overall integrated models, apparent ability to work well only in highly structured procedures, and the high level of mathematical sophistication required limits their understanding and use.

The next category of human performance model was the task network model. This technique was first explored by Siegel and Wolf (1969). They built simple models that simulated task completions and accuracy. The US Air Force then commissioned development of a simulation language called 'Systems Analysis of Integrated Networks of Tasks' (SAINT), which provided an environment for defining networks, task times, network branching, and probabilities (Pritsker *et al.* 1974). Another network model developed was the Technique for Human Error Rate Prediction (THERP) that explained human reliability through the use of fault trees (Swain and Guttman 1980). Finally, queueing models were developed that simulate one task at a time using a computed priority algorithm for selection of the next task to be executed (Chu and Rouse 1979). This model employed resource utilization, much like the SAINT models.

One advantage of task network models was that operator procedures could be modelled to any level of detail. These models could include a knowledge base for branching logic from one task to another and could mix the level of detail between different sub-networks. Another advantage of this technique was that it fostered a top-down, hierarchical approach to task decomposition and modelling (Baron *et al.* 1990). The disadvantages of the task network modelling technique included the fact that the complexity created by integrating many sub-models together, made this type of model very difficult to validate. Because of the flexibility in modelling, tasks often were not developed adequately for good analysis. Assumptions often embedded in a particular model that only may have been valid during specific circumstances.

The last category of human performance models considered were knowledge-based models. Little has been done to adapt knowledge-based technology to human performance models (Elkind *et al.* 1989). Hunt and Rouse (1984) demonstrated the application of fuzzy sets and rule-based modelling in human problem solving. The concept required the definition of appropriate data structures, such as frames, schemas and scripts (Minsky 1968, Schank and Abelson 1977). These data structures developed into the latest technology of object oriented programming that defined rules to relate patterns of objects or frames (data) and initiated changes in the database. Finally, a conflict resolution mechanism was required to determine which rules to use at any given instant. Expert system programs are now commercially available to provide the mechanization for this concept.

Rule-based models simulate cognitive processes, such as decision making and problem solving. Rule-based models foster a better understanding of how people solve problems in a specific domain. This concept was deemed good for modelling cognitive activities in supervisory control (Baron *et al.* 1990).

One disadvantage of rule-based models was that the process of extracting knowledge from experts (knowledge acquisition) did not readily apply across groups. Often there was conflict between experts on problem solving methodology. Because of the computational intensity of rule-based models, they typically were not applied to real-time processes (Pilot's Associate 1991). The extraction of knowledge has been a difficult process to construct (Nisbitt and Wilson 1977). Also, rule-based models were difficult to evaluate in terms of their fidelity to the acquired expertise they attempted to reproduce (Baron *et al.* 1990).

Many models have been developed over the years to simulate human performance, starting with Siegel and Wolf (1969). These models were developed for different purposes and, thus, differed in assumptions, structure, and implementation (Schuppe 1988). Because of the large number of models and the various techniques used to simulate human behaviour, several good summaries have been written over the years. Rather than attempt to present another summary of existing models, a summary of the critiques is presented.

Knoop (1978) was very critical of the ability of models to represent characteristics of operator behaviour. She suggested that research had to be conducted to investigate the appropriate sub-models to be built for each aspect of the operator's characteristics.

Rouse (1980) presented a good outline of models and their applications. He gave a very fundamental explanation of models in the categories (*a*) estimation theory, (*b*) optimal control theory, (*c*) queueing theory, (*d*) fuzzy sets, and (*e*) production systems. With each tutorial on the theory, Rouse presented the applications to which that technique had been applied.

In the area of supervisory control, Sheridan and Hennessy (1984) presented a summary of the models that seemed applicable. They recognized that the design team could benefit from the application of modelling to assess the design. No model was found appropriate for all aspects of supervisory control, but several models were found applicable for analysis and experimentation. Different models were applicable to different stages of supervisory control system development.

McMillan *et al.* (1989) presented the proceedings of a NATO research study group workshop on applications of human performance models to system design which summarizes all of the major models. A good critique and review was included in this volume.

Baron *et al.* (1990) summarized models and recommended research for improving human performance models. They recommended research in merging various sub-models and modelling techniques (like knowledge-based models, networks, and queueing models) realizing that a super-model would not be all encompassing.

Finally, Schuppe (1988) cited Wickens' (1984) review as germane to his research of investigating or contrasting a multiple resource model and a queueing model. Schuppe found that there was no significant advantage to incorporating the complex multiple resource model in the queueing model for predicting pilot workload.

Several network models were developed to account for parallel processing of tasks. Asiala *et al.* (1982) developed a model that broke tasks into micro levels and associated a resource requirement to each subtask. No two subtasks could be executed simultaneously if they shared the same resource. The resources available consisted of vision, audition, cognition, right hand and left hand. With this model, the operator (an aircraft pilot) could employ the right hand on the stick (controlling the attitude of aircraft), while looking at a display and thinking about a tactical situation simultaneously, since these tasks did not require the same resources.

The Human Operator Simulator (Harris *et al.* 1987) is currently being modified to account for parallel processing, in much the same way that the Pilot Simulation Model of Asiala and colleagues did so. However, as Wickens (1989) noted, there is no method incorporated in either of these models to account for interaction between parallel tasks.

In order to account for the demand for tasks, Laughery *et al.* (1986) incorporated the task demand table developed by McCracken and Aldrich (1984) and by Aldrich *et al.* (1988). Each type of task (scan, read, decode, etc) was assigned a demand level from 0 to 7. Then, when the sum of task demands exceeded a threshold, one or more tasks were queued until they could be executed. There was no relationship established between channels (vision, cognition, motor resources, etc), thus, visual activity has no impact on cognitive activity being executed simultaneously.

Wickens (1989) examined several limitations to this approach. Most notably, there was experimental evidence that visual tasks did conflict with cognitive ones (Wickens 1984). He indicated that his multiple resource theory accounted for conflict between channels (Wickens 1987, Wickens and Liu 1988).

Wickens' (Wickens 1987, Wickens and Liu 1988) multiple resource theory proposed three dichotomous dimensions, each of which defined two resources. There were processing codes (spatial-analogues versus visual-manual), processing modality (auditory-speech versus visual-manual), and processing stages (perceptual-cognitive versus response). North (North 1985, North and Riley 1988) developed the WINDEX system, which uses many of the assumptions of the multiple resource theory by incorporating a conflict matrix that penalized tasks based on the conflict between resource requirements. Each task required some percentage of perceptual, cognitive and motor resource. Two tasks could be executed simultaneously, as long as the sum of the resources did not exceed a predetermined threshold. The conflict matrix was used to evaluate the percentage of resource requirements competing for attention during each task. Task time was not penalized except by the task execution time delay created by resource saturation.

The major difficulty with the multiple resource model approach was that experts were required to estimate the percentage of resource requirements for each task. Many expert subjects provided a wide range of estimates of the required percentage of cognition needed for one task. Therefore, this approach appeared too subjective. Schuppe (1988) demonstrated that a queueing model could provide equivalent results in a fighter mission to the WINDEX concept. This eliminated the necessity of having experts estimate the task cognition percentages. Schuppe's method did not consider conflicts between tasks. Therefore, a model was needed that combined the network, queueing, modelling technique with the rule-based modelling technique that accounted for task conflict without relying on the pilot's subjective estimates of the percentage of resource-attentional demand for tasks.

The prediction of pilot performance and workload goes beyond simply defining and evaluating operator interaction and contributions to overall system behaviour. The definition of operator performance is quite intuitive yet measurable in terms of time and accuracy. The concept of workload appears to be nebulous. Although there is not a common definition, workload measurement efforts have been pursued for a number of years (Schuppe 1988). Several measurement techniques have shown promise in discriminating between different levels of workload. Given an existing system, subjective measures of workload are the most common method of measuring workload. However, physiological measures are being explored as substitute discriminators or, more likely, as augmentation to subjective measures. However, those methods assume an existing system is available for experimentation. When the system design is in the early stages, the physical system will not be available for workload or performance evaluation.

An alternative approach, during the early phases of system design, is to develop a performance and workload prediction system. Several methods of workload prediction exist. They rely on time line analysis or psychological factors, such as stress. However, these models do not lend themselves to being incorporated in decision support systems since they are stand-alone simulation models.

A human performance model that could adequately predict performance and workload, as well as fit into an 'expert system' type of decision support system would be quite a beneficial design tool. If pilot workload and performance could be predicted accurately enough to provide reasonable indications of possible design deficiencies early in the aircraft design process, certainly the effort put forth developing such a model would be justified. A computer simulation model with the capabilities described would enable aeronautical design engineers to identify human–machine interaction problems and evaluate alternative designs with relative ease while the aircraft was still 'on the drawing board'. Thus, the time and expense incurred in building a physical prototype of an inadequate aircraft could be avoided (Sage 1991, Sheridan 1992).

If the human performance model, developed early in the design process can be validated, this model can be used for two distinct purposes: evaluation of avionics systems and embedded decision support. The model can be used during system design and deployment, a significant cost saving can be realized in system development. The model can also be incorporated into the decision support system to aid operator performance and manage workload during system deployment.

As technology continually progresses, humans are designing more complex systems requiring human-in-the-loop interactions. These systems are designed so that the humans operate as controllers. As a controller, the human must be capable of perceiving inputs from the environment, processing those inputs cognitively, and responding to those inputs in a way that ensures the human–machine system efficiently achieves its purpose. One of the most complicated components of a technological system is the human–machine interaction.

Systems design is becoming more sophisticated in an attempt to streamline the human–machine interface. The application of artificial intelligence, and, specifically, expert systems technology, is being heavily investigated to allow machines to assume more responsibility for performing mundane tasks, thus permitting the humans to perform more complex tasks suited to their abilities. While developing intricate systems, the issues of coordination and areas of responsibility between the human and the system become significant factors in the design and operation of the system (Rouse *et al.* 1988).

In order for complex human–machine systems to operate effectively and efficiently with humans, these systems must possess knowledge of both the humans' performance, and their workload. Performance information allows the system to anticipate the human's actions, and workload knowledge helps the system judge when the human may need assistance (Sage 1991). When good models have been incorporated in supervisory control systems, or other forms of decision support systems, they have enhanced the overall human–machine system performance (Morris *et al.* 1985). Therefore, a good predictive model of human performance and workload can be critical for the efficient operation of a human–machine system (McCoy and Boys 1987).

A concern in the development of human–computer interaction is the workload imposed on the human by the system design. Workload refers to the mental or cognitive load on the operator. There are two methods of managing workload during the design process. First, the designer may predict operator workload as the system design progresses and choose the design that imposes minimum workload. Second, the operator's workload can be reduced by incorporating a model of operator workload into the human–machine system, and by allowing the system to adapt to the task environment. Therefore, a method of assessing operator workload early in the system design is critical to managing operator workload during system operation. As the workload model evolves, it can be embedded in a decision support system that provides systemic reasoning and adjusts to specific situations and operator states. Since operator performance predictions can be used to adjust system performance, the model should predict the timing and activity of the operator.

Because of these concerns the objective of this study was threefold. First, an improved human performance model was deemed necessary to predict operator task performance time and workload. By establishing a relationship between operator task execution time and human resource utilization and conflict, a more accurate prediction could be established and used to provide decision support. The second objective was to extend the concepts that Curry *et al.* (1985) developed with a knowledge-based simulation model and that McCoy and Levary (1988) investigated using an activity scanning model. Profiting from lessons learned in both of these studies, the current human performance model was written in a rule-based language which better modelled the decision making process. Third, by adding the network concept developed by Siegel and Wolf (1969) and subsequently used by Asiala *et al.* (1982), rapid prototyping of task knowledge could be incorporated into the model.

In order to investigate performance and workload modelling, this study concentrated on assessing and predicting a pilot's performance and workload while flying a single-seat fighter aircraft. A human performance model was developed and validated in this study to predict pilot performance and workload for later use in decision support in modern cockpits. This study determined how well model predictions correlated with measures of actual performance in a dynamic pilot-in-the-loop simulation.

The model developed in this study differed significantly from other models because it incorporated a mechanism to account for task resource conflict within the context of a knowledge-based model. By incorporating the conflict matrix into the baseline rule-based model, resource conflict was related to task performance and workload, thus improving the quality of the model predictions. By using a rule-based framework, the model was better suited for incorporation in an expert system decision support environment.

The model developed in this study combined rule-based simulation with task network discrete event simulation (Keller and Stanley 1991) in the form of an activity scanning model as illustrated in figure 1. Comparisons of critical predicted measures against observed pilot performance in a McDonnell Douglas manned flight simulator were used to validate the performance aspect of the model.

The human performance model is described in section 2. Section 3 deals with the model verification and validation. A summary and conclusions are provided in the final section.

2. The pilot performance model

2.1. *Model development process*

The first step in the model development process consisted of defining the mission scenarios. These scenarios provided an objective of performing procedures in order to accomplish specific objectives. When the procedures were defined, they were analysed to determine the functions and corresponding tasks that had to be performed to accomplish the mission. Two types of data requirements were defined, model input data and measures of merit for model validation. Each type of data required specific procedures to be executed in a realistic environment. Both part task and part mission simulation experiments were conducted to gather this data. A block diagram of the model development process is illustrated in figure 2.

Part task simulation was the process of having pilots perform all tasks required for a procedure, in a McDonnell Douglas simulator cockpit, without the burden of having to perform or attend to other tasks as distractions. In this way, task execution times and pilot workload could be estimated on each procedure. The model then used these results as inputs to help profile the tasks and procedures being simulated by the pilots.

Part mission simulation was the process of having pilots perform the entire mission scenario. This required the executions of multiple procedures simultaneously. In performing these procedures in parallel, conflict could arise between tasks competing for human resources (per-

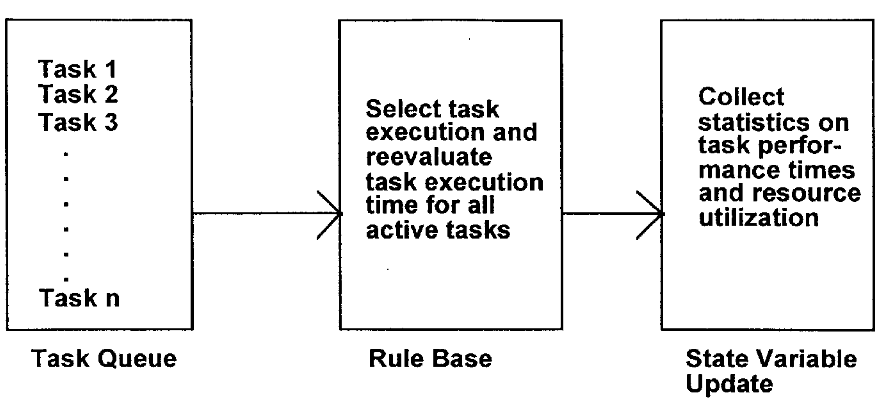

Figure 1. Rule-based model execution.

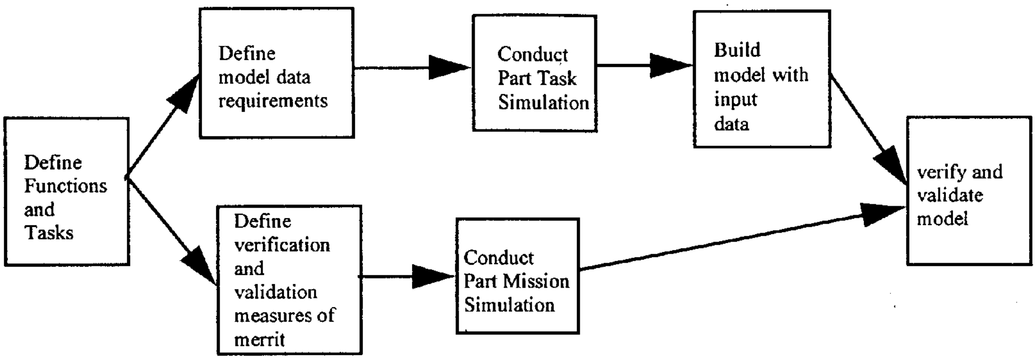

Figure 2. Model development process.

ceptual, cognitive or motor). If the model properly predicted pilot activity, it would also predict delays due to conflict between tasks. Therefore, measures of merit for validation were task execution times as well as delays due to the conflict of tasks competing for human resources.

Using the data collected through part task simulation experiments, the model was built. Each task was defined as part of a network representing the different procedures. Task inputs consisted of execution times, predessor and successor relationships, task information requirements, and task criticality. Task criticality consisted of classifing tasks as contributing to survival (highly critical), effectiveness (medium critical) and efficiency (low critical). Part task simulation (execution of tasks without interference from demands for other conflicting activity) was performed on seven procedures:

- Manually inputting a radio frequency
- Engaging the auto pilot
- Selecting the tactical navigation system
- Selecting a preset radio channel
- Inputting a navigational waypoint
- Selecting the Identify Friend or Foe frequency
- Engaging the instrument landing system.

These data, reported in McCoy and Seavers (1993), were then used to build the model for each procedure. In addition, these data were used to verify that the model was operating properly by comparing the model prediction of pilot procedure execution with actual pilot performance.

The next step was to validate the simulation model. Validation consisted of running the model to predict total mission performance. This constituted executing selected procedures while flying the aircraft. In this way, tasks would compete with each other for resources and cause potential delays. By comparing model prediction of model performance with actual pilot performance in the McDonnell Douglas simulator, a determination of model validation was made. The following section describes these steps in detail.

2.2. *Mission scenarios*

The purpose of developing mission scenarios was to define the goals of the mission and the procedures necessary to meet those goals. In the case of this exercise, the goals were to establish a route that required the pilot to execute representative procedures that could be used to measure pilot performance, and to compare these measures with a computer-generated prediction of pilot performance. To accomplish these goals, two mission scenarios were adapted, requiring pilots to perform flight control functions (see figures 3 and 4). In addition, the pilot had to maintain communications, navigation, and identification functions while performing the flight control functions.

The flight profile consisted of a set of waypoints placed strategically on a terrain board. Each waypoint was represented by a church steeple that was highly visible both on the map and in flight. Figure 3 shows the route that was flown by the aircrews. The first leg of the route was 15.8 miles long on a heading of 20°. Each pilot was instructed to fly the entire route 500 feet above

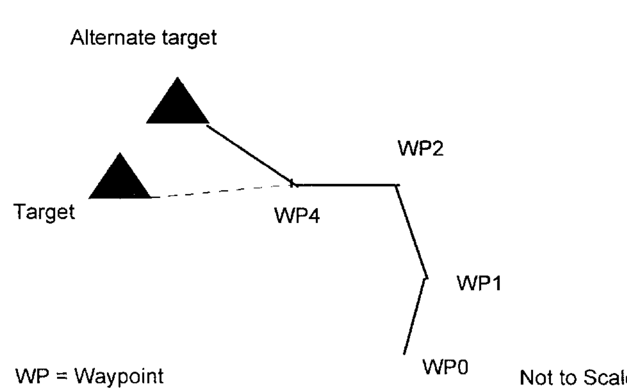

Figure 3. Mission scenario.

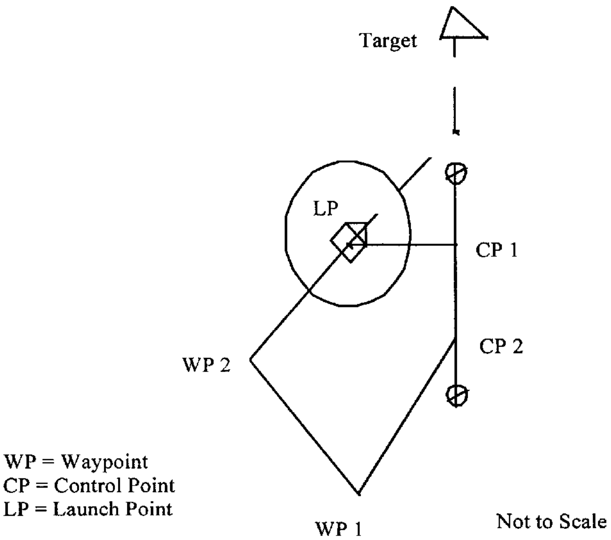

Figure 4. SLAM mission scenario.

ground level at a speed of 480 knots. At that rate, the first waypoint would be reached in 1 minute, 50 seconds. Once the pilot overflew the church, he executed a standard turn to a new heading of 310° and flew 19.7 miles. This leg lasted 2 minutes, 30 seconds. Again, upon overflying the next church steeple, the pilot executed a standard turn to a new heading of 270° for 16.3 miles, or approximately 2 minutes.

When the pilot flew the route illustrated in figure 3, data were collected on his accuracy in maintaining the flight path. In addition to flight control, the pilot was required to execute several event procedures consisting of (*a*) changing the frequency on a manual radio channel, (*b*) changing the frequency on the Identify Friend or Foe (IFF) system, and (*c*) inputting a new target location to include latitude, longitude and elevation. These events were presented to the pilot in what appeared to be a random order through a radio transmission instructing the pilot to execute each procedure with 1 minute, 30 seconds remaining in the flight segment.

The second mission scenario required the pilot to fly a predefined route and deliver a stand-off land attack missile (SLAM). The pilot was required to fly the route depicted in figure 4. During the first leg of the route (from WP_1 to WP_2), the pilot was required to perform SLAM weapons management while maintaining the flight plan. Weapons management consisted of (*a*) selecting the stores display, (*b*) selecting the SLAM missile, (*c*) selecting the data link for future communications with the missile in flight, (*d*) down-loading a predefined missile mission and, finally, (*e*) selecting the SLAM video display for future use.

At the waypoint (WP_2), the pilot had to turn to a new heading to fly to the launch point (LP). During this leg, the pilot maintained a specific heading and altitude while slowing down to a particular ground speed, all of which conformed with the pre-programmed mission loaded into the SLAM missile. By reaching the launch point using the correct flight parameters, the missile guidance system was more accurate, since the SLAM was launched under optimal initial conditions. At the launch point, the pilot pressed the appropriate button to launch the SLAM missile.

After launching the missile, the pilot changed headings and flew to the first control point (CP_1). Upon arrival at that location, the pilot assumed a new heading on a course parallel with the SLAM missile, and flew to the second control point (CP_2). While on this heading, the pilot could monitor the SLAM video as the missile approached the target.

2.3. *Task decomposition and analysis*

In order to develop a simulation model of pilot activity, a task analysis had to be performed, generating data for input to the model and data to be used for validating the model. Initial task analyses for the F/A-18 aircraft were reported by Wise and Asiala (1977), Wise *et al.* (1977). The data presented in these reports provided an initial base for developing the model. Additional data were collected in conjunction with experiments conducted at McDonnell Douglas Aerospace Company to satisfy the requirements of the Human Engineering Dynamic Simulation Plan (McCoy *et al.* 1993). The data collected during these exercises were reported in a series of White Papers that will be included in latter revisions of the Human Engineering Systems Analysis Report (McCoy and Seavers 1993).

Given the mission scenarios described, the pilot had to perform four major tasks, or functions: (*a*) flight control, (*b*) communications, (*c*) identification, and (*d*) navigation. In order to determine which variables to measure and what tasks to model in the simulation, it was necessary to decompose the functional tasks. For flight control, the pilot primarily concentrated on manipulation of the stick. This required viewing the Head Up Display (HUD), comparing the heading to the desired flight path, and seizing the stick with the right hand to adjust the aircraft's flight path. The declarative knowledge for this task was the actual and desired heading. The procedural knowledge for this function became:

If (actual heading – desired heading) is too large,
Then adjust heading (by manipulating the stick).

The type of data collected in the simulator were the mean time to manipulate the stick (adjust heading and altitude) and the mean time between adjustments.

The next function that was decomposed was communications. The pilot was instructed, via a radio transmission, to select a new radio frequency on the manual channel. In order to accomplish this procedure, several individual steps had to be performed. Using the Up-Front Control (UFC), the pilot selected the appropriate rotary 'comm' knob, and then viewed the frequency set on the knob. If the frequency was not 'manual' the pilot was required to rotate the knob and select 'manual'. The next step involved pulling the knob that changed the UFC to data entry mode to display the current frequency. Then, the pilot compared the current frequency to the desired one and chose to input the new frequency by using the mechanical keypad to input up to six digits. Finally, the pilot pressed 'enter' to signify that the old frequency had been overridden.

Another function the pilot performed was inputting the IFF code. Again, the pilot received a radio transmission instructing them to input this code. Upon receiving the command, the pilot pressed the appropriate button on the UFC to select the IFF input mode. The pilot compared the current with the desired four-digit code and chose to update it. Then, the pilot pressed 'enter' to complete the operation.

The final, and by far the most complicated, procedure executed by the aircrew was changing the waypoint, or target location. After receiving a radio transmission with instructions to input the new target location the aircrew selected the data entry panel from the Horizontal Situation Indicator (HSI) display. Once the waypoint input mode was selected, the pilot entered the hemisphere 'N' followed by a six-digit latitude. Next, the pilot entered 'W' and a seven-digit longitude. Finally, the pilot selected a three-digit elevation and pressed 'enter'. The data defined and collected for each task in the described procedures consisted of: task execution time (mean and standard deviation); information requirements; preconditions for execution; task goals or anticipated outcome; resource requirements; and, shed conditions.)

Task times were measured by video taping the execution of procedures and recording the time needed to complete each component of the task. Since data input was being recorded by the computer, the time each button as pushed was also recorded. By using these data along with guidelines extracted from Card *et al.* (1983), each task was measured for its perceptual, cognitive, and motor components. Then, this information was averaged across all pilots who participated in the data-gathering procedure. This produced an average execution time and a standard deviation for each task. Empirical distribution functions for each task were defined based on this data and used in the pilot performance simulation model (McCoy 1995).

The remaining data defined and collected for the model were gathered through pilot interviews. Pilots defined the information required to perform each task, the goals which prompted them to initiate the task and preconditions for task execution. In addition, they defined reasons for abandoning a task (shed conditions).

2.4. *Simulation model development*

The human performance model developed for this study was comprised of a combination of conventional discrete event simulation modelling techniques and an activity scanning, or knowledge-based, representation of the pilot decision making process. These two techniques were highly integrated to ensure that they accurately represented the environment exerting the demands on the aircrew, and the decision making and task execution processes that represented pilot activity. A description of the data generated from the model will be presented in section 3 to illustrate which variables compared favourably to observed pilot performance.

2.5. *Model description*

The human performance model (HPM) used in this study was a special form of activity scanning model. The model was built using the 'C' Language Interactive Production System (CLIPS) program. CLIPS is an expert systems tool developed by the artificial intelligence section of the NASA/Johnson Space Center (Giarratano 1993). Although CLIPS is a language designed for writing expert systems, it could be used to develop and execute simulation models. The basic elements of CLIPS are (*a*) a fact base, (*b*) a knowledge base, and (*c*) an inference engine. A program written in CLIPS consists of facts and rules. The facts represent the major component of the declarative knowledge of the model, while the rules constitute the procedural knowledge of the model. The inference engine decides which rules should be executed.

Declarative knowledge

CLIPs language has three basic components, declarative data, procedural data, and the inference engine that relates the two. Declarative data in CLIPs consist of facts. Facts can be simple structures, such as goals, or they can be complex structures called templates. A template defines a complex fact with associated fields or attributes. Two major templates were defined for this model, a resource template and a task template. Figure 5 illustrates the resource template.

The first line of figure 5 defines the template for resources. A resource represents a class of objects that share the same fields. Therefore, this template was used to define those common fields for the class of objects assigned the name 'resource'.

Line Number — **Code for CLIPs Resource Fact Template**

```
1  (deftemplate resource "define the resource object"
2      (slot name (type SYMBOL) (default ?NONE)
3      (allowed-words vision audition cognition
4              left-hand right-hand speech none))
5     (slot status (type SYMBOL) (default idle)
6     (allowed-words idle busy))
7     (slot sum-used (type FLOAT)(default 0.0))
8     (slot sum-used squared (type NUMBER)
9             (default 0.0))
10    (slot mean (type FLOAT) (default 0.0))
11    (slot stdev (type FLOAT) (default 0.0)))
```

Figure 5. Resource template.

The second line defines a name for each resource that was created. This name was distinct and was used for matching resource status in the rules. Lines 3 and 4 indicate that a limited number of resource names were allowed in the model (*a*) vision, (*b*) audition, (*c*) cognition, (*d*) left-hand, (*e*) right-hand and (*f*) speech. This limitation enabled the model to check that only allowable names were used, otherwise no match could occur on the resource name.

The status of the resource is addressed in lines 5 and 6. Either the resource status of 'idle' or 'busy' was allowed. Depending on the status, the system either implemented or delayed tasks requiring that resource. The execution time of the task as well as delay times were affected by the resource utilization state. The remaining five lines of code define statistical variables to be controlled. In order to collect statistics on the utilization of the resources, data had to be gathered on the amount of time the resource was busy. Therefore, two more fields were defined for this template, the sum-used and the sum-used-squared. These were used to compute the mean utilization rate and standard deviation by recording these times each time the resource changed status.

The second template in the model defined the tasks that the aircrew performed. Figure 6 presents the CLIPS coded task template. Again, templates are the mechanism for defining task attributes and are used by the rules to scan the tasks, or activities, for patterns of data that will allow task execution. The first attribute of the task is the name—again, an identifying keyword (see figure 6, line 2).

The field illustrated on the third line of figure 6 is the task status. When tasks were first defined, they had a status of 'null'. However, rules in the model dictated when a task status would change. The task status and conditions for those allowable status values consisted of (Sacerdoti 1977):

(1) Planned = goal for task has been asserted.
(2) Enabled = predecessor task has been completed.
(3) Activated = all preconditions have been met.
(4) Executing = resource seized.
(5) Completed = task conditions met.

In order for a task to be considered 'planned', a goal must have been asserted. The goal field, described on line 6 of the task template, was used as part of the pattern-matching rules governing task planning. Defining the system in this way, made it a truly goal-oriented model of the pilot. Although, in this study project, one goal triggered a whole procedure, the model was defined with the ability to propagate goals throughout the procedure while integrating a variety of tasks. The major goals employed in this model consisted of (*a*) fly aircraft, (*b*) establish comm, (*c*) select IFF, and (*d*) select new waypoint. These goals were used to trigger (*a*) flying the aircraft, (*b*) inputting a manual frequency change, (*c*) inputting an IFF code, and (*d*) entering new waypoint coordinates, respectively.

The next set of fields (see figure 6, lines 13 and 14) consists of predecessor and successor fields that allowed the model to traverse through a network of tasks that constituted a procedure. Rules were developed to examine the status of the predecessor and to ensure that it was completed prior to activating the next task. Several tasks could be executed simultaneously, as long as they were members of different sub-networks. If a task was the first task of a procedure, its predecessor

```
Line
Number   CLIPs Code for Task Template

1   (deftemplate task "define the network task template"
2       (slot name (type WORD) (default ?NONE)
3       (slot status (type WORD)(default null)
4       (allowed-words null planned enabled
5        activated executing completed))
6       (slot goal (type WORD) (default ?NONE))
7       (slot resource (type WORD) (default none)
8       (allowed-words vision audition
9        cognition left-hand right-hand speech))
10      (slot child-of (type WORD) (default none))
11      (slot parent-of (type WORD)(default none))
12      (slot predecessor (type WORD) (default none))
13      (slot successor (type WORD) (default none))
14      (slot start-time (type NUMBER) (default 0))
15      (slot end-time (type NUMBER) (default 0))
16      (slot priority (type WORD)(default ?NONE)
17          (allowed-words survival effective
18            efficiency))
19      (slot mean (type NUMBER) (default 0))
20      (slot stdev (type NUMBER) (default 0))
21      (slot condition (type WORD) (default no)
22          (allowed-words yes no))
23
24      (slot demand (type NUMBER) (default 0))
25      (slot repeat (type WORD) ( default no)
26          (allowed-words yes no)))
```

Figure 6. Task object definition.

was assigned the status of 'nil'. Also, if a task was the last task in a procedure, its successor was set to 'nil'.

Task beginning and ending times were recorded when the task began execution and when it completed execution (see figure 6, lines 15 and 16). These data could be used to calculate average task execution times, if those data were deemed necessary. This proved particularly useful when tasks were repeated throughout a model execution. The repetitive tasks in this study were flight tasks.

Although resource availability and preconditions were used to dictate which tasks should be executed, there was also a priority scheme incorporated in the model. The priority could be (*a*) survival, (*b*) effectiveness, and (*c*) efficiency (see figure 6, lines 17–19). In the rules, survival tasks have the highest priority, followed by effectiveness and, finally, efficiency. Therefore, as the activities were scanned, prioritization was employed. Another priority scheme, which may be implemented in the future involves defining a deadline for a task. Then, a criticality based on the time available to perform the task could be computed.

Once a task has met all conditions, including resource availability, the task completion must be scheduled. The next two fields, mean and standard deviation, were derived from the task analysis described in the earlier data gathering section. Those data were put into the task profile and employed using an appropriate distribution function to determine the random task execution times. Often, an empirical distribution was used.

Procedural knowledge

The facts described in the previous section defined the state of the system at any time. The rules for the human performance model provided the mechanism for scanning all of these facts, recognizing a specific pattern of facts, and choosing which rule to execute. The types of rules available in this model consisted of (*a*) changing the status of a task or activity, (*b*) setting a new goal, and (*c*) advancing time in the simulated clock. When a rule was executed, new facts could be asserted and old ones could be either retracted or modified. New or modified facts could trigger execution of another rule by providing a pattern that matched the conditional part of the rule. This discussion of procedural knowledge describes the structure of rules employed in CLIPS as well as each of the rules developed for this pilot performance model.

The standard rule employed for CLIPS is illustrated in figure 7. The conditions on the conditional side of this rule structure typically represented a set of fact values that the model recognized. An inference engine, such as that imbedded in CLIPS, provided the most efficient algorithms for pattern recognition. Therefore, construction of the rules remained the most challenging activity for the modeller.

As mentioned before, the major rule types for the human performance model consisted of (*a*) changing task status, (*b*) setting new goals, and (*c*) advancing time. The potential task status consisted of (*a*) null, (*b*) planned, (*c*) enabled, (*d*) activated, (*e*) executing, and (*f*) completed. The first task status modification rule was:

If
status = null
goal of task is posted
Then
modify task status to planned

The next rule developed for this model was the enabling rule that advanced the task to enabled status when a predecessor task was completed. If this was the beginning task, it became enabled automatically. The rule read as follows:

If
the task status = planned
task predecessor = completed or nil
Then
change status = enabled.

In order for a task to proceed to the next status and be activated, one of two conditions had to be true. Either a task had no preconditions to meet, so it was enabled, or the preconditions were met, and it was currently enabled. Although none of the current tasks had preconditions, this stage was incorporated into the model for future growth. When a task had assumed the activated status, it could become 'executing'. When a task became 'activated', it was essentially waiting in a queue for the resources required to execute it.

When a task was in an activated state, and it required a resource to execute, the rule for changing the status of the task to 'executing' checked the state of the demanded resource. If the demanded resource was idle, then the resource was 'seized' (i.e. the state of the resource was modified to 'busy' and the task status was modified to 'executing'). Also, a random sample was used to schedule the task completion time. When the task began executing, the time of execution was recorded by the model for use in calculating total execution time and resource utilization time.

If
(conditions to be met for matching patterns)
Then
(execute statements)

Figure 7. Standard rules for CLIPS.

When the task completion time exceeded the clock time 'tnow', then the resource was modified to the idle state and the task status was modified to the complete state. In addition, the running total of resource usage time was updated with the total time that the task was executed. This time factor was used at the end of the simulation execution to compute the statistics on resource utilization.

Another form of procedural knowledge investigated was the rule for advancing the clock. Every time a task was scheduled for execution, the event–time fact was asserted with the time of the completion of the task. A rule was developed to match on the latest event–time fact and update 'tnow' when all activities had been completed for the current time. In this way the simulation was a 'next event' simulation, since it did not require the model to time step through the total modelling horizon. The rule was written as follows:

If
event–time> tnow
event–time $\leqslant$ other-event–time
Then
assert new tnow = event–time

In addition, when 'tnow' exceeded the target simulation cycle, the model terminated execution and report statistics of model execution and resource utilization.

A mechanism for predicting pilot workload was required. Wicken's (1985) Multiple Resource Theory was deemed inappropriate due to the fact that it assumed several tasks using the same visual, cognitive or motor resource could be performed simultaneously (McCoy 1995). This same study found conventional single channel operator models to be inadequate because they did not address conflict between tasks being performed simultaneously using different resources. Therefore, a technique was developed that combined these two methods to predict pilot workload, estimate the workload to performance relationship, and adjust task performance based on this relationship.

By using a single channel operator mechanism, the problem of multiple tasks using the same resource was eliminated. However, the conflict matrix was incorporated from the Pilots Associate program model (Pilots Associate 1990) and the workload calculation developed by North (1985) was used to estimate workload each time a task began or completed execution. Next, a performance operating characteristic, as reported by Boff and Lincoln (1988) was introduced into the model. This characteristic provided a threshold for comparison of pilot workload. When workload exceeded the threshold, all active task scheduled completion times were extended. When workload decreased below the threshold, all active task completion times were re-adjusted to reflect better performance (McCoy 1995).

Once the model was developed, verification and validation had to be performed. The next section describes this process along with the application of the additional part mission data collected in the McDonnell Douglas flight simulation facility.

3. Verification and validation

Verification is the process of ensuring that the simulation model functions as intended when executed on the computer (Velayas and Levary 1987, Pritsker 1984). An example of verification involves determining that all activities are executed in the proper sequence. Verification also ensures that the probability distributions used in the model properly represent the data. A series of examinations and tests was conducted on the model to verify its operational ability.

Validation is the process of ensuring that the simulation model represents the environment it is designed to emulate (Velayas and Levary 1987). By comparing observed data collected in the pilot-in-the-loop McDonnell Douglas simulator with comparable data generated by the simulation model, an assessment can be made as to how well the model predicts the pilot activity. The efforts employed both to verify and validate the simulation model of the pilot are described below.

3.1. *Verification*

The first verification exercise tested that all tasks were properly executed in a procedure network. Based on the planning paradigm incorporated in the model, each task advanced through the following sequence of states (*a*) planned, (*b*) enabled, (*c*) activated, (*d*) executing and, (*e*) completed. Figure 8 illustrates the sequential progression of these stages. A print statement was embedded in the rules that matched on the condition for changing the task state and modified that state. Therefore, a trace, or time line, of events changing the status of task was generated and inspected. This trace demonstrated that the order of task planning and execution was functioning correctly.

The next phase of model verification ensured that each task had the correct probability distribution which generated appropriate task execution times. Each task was simulated 63 times with pilots in the McDonnell Douglas manned flight simulator. The law of large numbers indicated that this number of trials was sufficient for comparison purposes. The tasks were compared with the original sequence to demonstrate that the distributions generated appropriate means. A series of tables report the comparisons between model-generated samples of the distributions and the data observed in the manned simulator.

Table 1 presents the task comparison that constitutes the communications procedure. In this table, 'pilot mean' represents the average time pilots required to perform the subtask. The 'model mean' represents the average time required by the model to execute the subtask. The 'F-test' reports the signifcance of the test of differences between means, and the 'P-value' reports the probability of error. In all cases, the probability of error was sufficiently large to indicate that no statistically significant difference existed between pilot performance and model-generated task execution times at the 5% level of significance. The results reported in this tabular summary proved that the model was functioning properly with respect to the communications procedure.

Once the task distributions obtained from part task simulation experiments were simulated individually and compared with the input data to confirm the distribution accuracies, the procedures had to be simulated and compared with those observed in the simulator. Table 2 reports the comparison of procedures being executed by the pilot in the simulator with the procedures being predicted by the model.

Again, the first procedure investigated in the model was communications. In the previous exercise each subtask was executed repeatedly and averaged for comparison with the distributions derived from observed pilot performance. The next step was to run the entire procedure repeatedly and compare each subtask, as well as the total procedure execution time, with that performed by the pilots. Table 2 compares the means for both the pilot and the model for each subtask (even though subtasks were executed within the entire procedure). As table 2 illustrates, all the subtasks generated by the model were statistically close to those observed while pilots were performing the procedures in the part task simulation exercise.

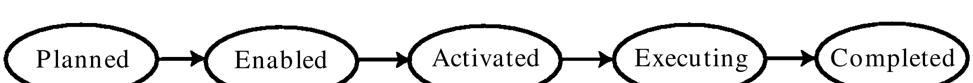

Figure 8. Task planning stages.

Table 1. Task verification for communications subtasks

Task name	Pilot mean	Model mean	F-test	P-value
Sel-Man-P	1.180	1.330	0.940	0.330
Sel-Man-c	1.700	1.700	0.000	0.989
Sel-Man-m	1.540	1.350	0.956	0.330
Ent-freq-p	0.900	0.850	0.415	0.520
ent-freq-c	2.100	1.900	2.100	0.150
ent-freq-m	0.890	0.950	0.370	0.540
ent-man-p	0.140	0.136	0.070	0.790
ent-man-c	0.360	0.400	2.600	0.110
ent-man-m	0.180	0.200	0.310	0.580
Total	9.010	8.840	0.195	0.660

*Task names are defined in McCoy (1995)

Table 2. Communication procedure verification

Task name	Pilot mean	Model mean	F-test	P-value
Sel-man-p	1.183	1.301	0.317	0.575
Sel-man-c	1.727	1.759	0.031	0.862
Sel-man-m	1.537	1.249	1.833	0.179
inp-freq-p	0.904	0.804	0.933	0.337
inp-freq-c	2.091	2.18	0.312	0.578
inp-freq-m	0.891	0.812	0.527	0.470
ent-freq-p	0.140	0.174	3.151	0.079
ent-freq-c	0.360	0.414	3.272	0.074
ent-freq-m	0.180	0.143	2.194	0.142
Total	9.012	8.769	0.232	0.631

*Task names are defined in McCoy (1995)

3.2. *Validation*

The previous section reported the exercises performed to verify that the model was functioning properly on the computer. This section reports the exercises performed to validate the model for the mission segments flown in the manned flight simulation facility. Only the procedure addressed in verification will be discussed. The remainder of the procedures executed by the pilot, used for validation are described in McCoy (1995).

First, pilots flew the navigational route and performed the communications, identify friend or foe, and navigational waypoint update procedures while maintaining the route. This imposed conflict between the flight control procedures, maintaining the route and airspeed, and the communications, navigation and identification (CNI) procedures mentioned. Then, the model was exercised to predict the pilot's performance of the same route and procedure execution. The purpose of this exercise was to compare the procedure execution time between the pilot and the model.

For each procedure, several statistics were reported. First was the average procedure time predicted by the model, based on the inputs. Next, was the average stick input time. This represented the average amount of time the pilots inputted stick manipulation to adjust course or altitude. Finally, the average time between stick inputs was presented. If the procedure was truly time consuming, then the time between stick input could be large since resources would be consumed when pilots manipulated the displays or entered data.

Two of the procedures showed large discrepancies between the execution times predicted by the model and the actual performance observed while a pilot flew the aircraft. The COMM procedure, entering a new radio frequency, and the Identify Friend or Foe (IFF) procedure both required the pilot to devote more than twice as much time to these tasks than the computer model had predicted.

The model was run in segments with one of the procedures being executed (the goal posted during that mission segment). Resource statistics were collected for the human during each mission segment and averaged for all repetitions of the model for that segment. Each segment was executed 35 times, a sufficient sample size to perform a statistical comparison. Table 3 reports the estimated resource usage that the model predicted regarding pilot workload.

Table 4 shows that two of the procedures, predicted by the model, were underestimated, compared with the observed pilot performance, when the pilot was simultaneously flying the airplane and performing the procedures. The procedures were entry of a new radio frequency and entry of a new IFF code. As with every procedure, the pilots were given approximately 90 seconds to perform the tasks. Upon arrival, the pilot was instructed to make a course correction. This was the time needed to fly to the next waypoint. Both procedures required less than 10 seconds to perform during the part task exercise. The model predicted pilot performance would be very similar when flying the aircraft. However, this was not the case. It is possible that the pilots actually performed the radio frequency and IFF code changes considerably slower than the model predicted because they experienced no sense of urgency. When the model was constructed, assumptions were

Table 3. Resource usage by mission segment

Segment	Vision	Audition	Cognition	Right hand	Left hand
COMM	0.347	0.040	0.225	0.510	0.083
WPT	0.420		0.365	0.425	0.129
IFF	0.373		0.159	0.536	0.037
SLAM	0.447		0.350	0.331	0.128

Table 4. Procedure validation

Procedure name	Model mean	Pilot mean	F-test	P-value
COMM	9.100	17.200	13.400	0.001
Stick Input	1.070	1.020	1.550	0.220
Bet. Stick	0.956	1.220	2.810	0.099
WYPT	16.410	20.600	0.657	0.420
Stick Input	1.080	1.150	3.630	0.062
Bet. Stick	1.290	1.070	21.200	0.000
IFF	4.840	16.950	48.500	0.000
Stick Input	1.090	1.166	6.940	0.011
Bet. Stick	0.830	1.000	11.100	0.002
SLAM	21.790	22.820	0.460	0.500
Stick Input	1.070	0.944	1.840	0.180
Bet. Stick	2.300	0.861	151.080	0.000

made that the pilots would devote immediate attention to these procedures, since no other complex functions, with sufficient demand to delay task execution, were competing for resources. Within those parameters, the model adequately predicted pilot performance. The next two procedures investigated consisted of inputting a new waypoint coordinate and pre-programming a SLAM missile for launch. Both of these procedures required much more time to perform (approximately 20 seconds). Apparently, the pilots perceived a sense of urgency and performed these tasks more expediently. The model, which accounted for urgency only in the priority system, predicted pilot performance times within 2 seconds of the times observed when pilots actually performed these procedures. In both cases, the model credibly predicted pilot activity.

Based on these results, the model did predict pilot performance, yet consistently underestimated pilot execution time. Incorporating the ability to account for interference between pilot resources (vision, audition, cognition, right hand, left hand, and speech) should enable the model to predict pilot activity with even greater accuracy.

4. Summary and conclusions

Systems engineering is a management technology that controls the interaction of science, the organization, and the systems environment (Sage 1992). The systems engineering process employs operations research techniques to improve the design of the system, at minimal cost. One critical aspect of systems engineering is the human–machine interaction. By combining mathematical modelling with traditional task analysis and other human factors procedures, human performance models have been developed to evaluate candidate system designs (Elkind *et al.* 1989). This has become increasingly important because development of full-scale, human-in-the-loop simulation studies have become expensive, in terms of personnel and time (Baron *et al.* 1990). Another motive for developing human performance models is that the rapid advance of modern information technology has redirected the way systems are being designed. With continuing improvements in computer technology, task automation has moved the human from immediate control of system operation to higher-level, supervisory tasks and long-term planning functions (Sheridan 1992). Without the thorough understanding of the implications of system design on the human–computer interface that a human performance model can provide, the overall system design can suffer. In such cases, inferior design can jeopardize the success of the entire system.

The fidelity of human performance models is key to the success of their application in both system design and in decision support implementation. If the model poorly predicts the impact of the system design on the human–computer interaction, the operator can become overloaded. Various methods of developing human performance models have been employed and each one has both advantages and disadvantages. Selecting any one technique can cause criticism, based on that method's acknowledged disadvantages. However, by minimizing these disadvantages and building a model suitable for a specific application, design criteria can be better met and the model will prove more suitable for embedding in real-time systems, such as decision support.

Three major modelling techniques were considered in this study because of their popularity, or ease of use. They were (*a*) the task network approach, (*b*) the rule-based approach, and (*c*) the multiple resource theory approach (Baron *et al.* 1990). Each of these approaches had both advantages and disadvantages. Advantages of the task network approach were (*a*) it encouraged a top-down structured design process, (*b*) it incorporated a natural hierarchy of tasks and procedures, and (*c*) the procedures could be decomposed to any level of detail, depending on the requirements of the model application.

The rule-based modelling approach (*a*) described how people solved difficult problems, (*b*) developed an understanding of how humans are transitioning to the role of supervisory controllers, (*c*) understood how the model could be embedded in knowledge-based decision support systems, and (*d*) expanded and became capable of learning as the environment changed.

The multiple resource model (Wicken 1984) was developed to account for the fact that humans appear to share resources while performing multiple tasks, simultaneously. Many previous modelling techniques ignored this phenomenon. Although difficult to quantify, the WINDEX model was developed to incorporate the multiple resource model (North 1985). It provided a mechanism to quantify channel demand and workload calculation that could be used to defer task execution based on a predicted excessive workload situation.

Each of these modelling techniques also had distinct disadvantage. With the network modelling paradigm, the interaction between tasks generally has not been modelled. Subprograms were not necessarily unique and care was taken in selecting which ones to model. Also, validation could be difficult because different levels of decomposition were used in the model compared with the data collected for validation.

Rule-based models have typically required extensive study of the individuals being modelled to determine the knowledge they employed while performing tasks within the system. Rule-based systems were generally used in slow, non-real-time environments where response times varied in magnitude from seconds to minutes. Aggregating submodels of perception, cognition, and motor response were required to model interactions between submodels. Finally, rule-based models were generally expensive to construct since they required developing an extensive knowledge base.

The multiple resource model, as implemented in the WINDEX environment (North 1985), quantified the channel demand and conflict between combinations of channel usage. The model generally allowed multiple tasks to be performed, simultaneously, as long as the predetermined workload limitation was not exceeded. Each task demanded a fraction of the resource to attend to each channel demanding attention. The major problem with this approach was determining what fraction of a resource was necessary for each channel within the task. Schuppe (1988) showed that the effort may not have been necessary since a queueing model did as well in predicting human performance. However, his queueing model, like others, still ignored the interaction of competing channels.

In order to combine the hierarchical advantages of the task network model with the flexibility of rule-based modelling, a joint modelling approach was implemented in this study. By developing the network environment within a rule-based, expert system program, the pattern recognition and other inherent knowledge-based features were available for enhancing the model environment. An initial demonstration of this technique was presented in the Task Network Tool that was developed for the Pilot's Associate program (Keller and Stanley 1991). However, this program also incorporated WINDEX for multiple resource theory workload calculations. In order to relieve the model of the problems found in the WINDEX implementation, the workload calculation was modified to allow only one task to seize a channel at a time. Two tasks could not share the same resource simultaneously. However, the model did retain the conflict penalty imposed from two channels being utilized simultaneously with different resources. In this way, multiple tasks could still be active simultaneously.

To develop a human performance model that was task-oriented, employed rule-based procedure generation, and incorporated accountability for interference between tasks performed simultaneously, data from various sources and modelling techniques were integrated. The data integration was accomplished throughout the following tasks: (*a*) task analysis data were selected from existing databases developed from a human-in-the-loop simulation plan (McCoy *et al.* 1993); (*b*) the task analysis was conducted using established techniques (Ausburn *et al.* 1980); (*c*) task demand for channel attention, based on McCracken and Aldrich (1984), was incorporated in the tasks reported by McCoy *et al.* (1993); (*d*) a task conflict matrix, as reported in the Pilot's Associate program (1990) for a crew station similar to that used for data collection, was performed; and, (*e*) a rule based on a performance operating characteristic derived from data reported in Boff and Lincoln (1988) was incorporated.

Developing the model also entailed integrating a series of techniques and modified methods to incorporate the advantages of each technique. These integration steps included: (*a*) adapting the model, based on task profiles developed for the human performance model reported by Asiala *et al.* (1982); (*b*) incorporating a combined task network and rule-based modelling approach similar to the Pilots Associate Task Network Tool (Keller and Stanley 1991); (*c*) building the model in a flexible, rule-based CLIPS environment (Giarratino 1993); (*d*) employing the planning paradigm proposed by Pilot's Associate (1991) and based on the method defined by Sacerdoti (1977); and (*e*) modifying the multiple resource theory (Wickens 1984) as implemented by WINDEX (North 1985) and applied to pilot modelling (Pilot's Associate 1991).

Because of the nature of integrating modelling techniques and data from multiple sources, there were a number of contributions made by this study. They consisted of (*a*) combining network and rule-based modeling approaches in CLIPS, (*b*) illustrating the use of an expert system program for modelling pilot performance, (*c*) demonstrating an environment that combines rule-based knowledge of task selection and execution, (*d*) validating the combined task network/rule-based model with data collected in a high fidelity, complex, human-in-the-loop simulation facility, (*e*) demonstrating a simplified workload calculation based on a single channel operator within a validated model, and (*f*) demonstrating the use of a performance operating characteristic, relating performance times to workload level, and adjusting task performance times.

The first contribution this study made toward improving human performance modelling techniques involved adapting the rule-based task network concept that Keller and Stanley (1991) developed (using an extensive mainframe computer network) for the Pilot's Associate program into a CLIPS expert system program (Giarratano and Riley 1994) running on a single personal computer. This provided an expedient medium for conducting experiments with the model using a convenient, low-cost facility. This also provided an environment for exploring the definition of declarative knowledge (in the form of templates and facts) as well as procedural knowledge, represented by rules.

Because declarative knowledge, facts about the operator state, system state, and environment, are represented, the model was useful for model-based reasoning. By imbedding this model in a decision support system, the model could produce operator activity predictions that could be compared with observed performance. Then, deviations from predicted activity and observed performance could be used to initiate more rules to aid the pilot.

The model developed in this study was validated using data collected from the human-in-the-loop flight simulator. Data were collected on pilot performance, while executing the procedures modelled. Those data were compared with the data generated from the Monte-Carlo simulation of the model. The data compared favourably.

The final demonstrated benefit of this study emanated from the model architecture. By using the rule-based modelling approach, a true activity scanning approach was developed. As events occurred, the model adjusted the fact database to reflect (*a*) the new state of the systems environment, (*b*) the system, (*c*) the operator's goals, and (*d*) the operator's state. The rules were developed to recognize patterns of states and to execute procedures based on those recognized patterns. This process represented procedural knowledge. One major advantage this methodology provided was an ease in adjusting task execution times. One rule was capable of adjusting all executing tasks based on the goal set by existing conditions representing changes in operator workload. This feature had been extremely difficult to implement using procedural languages, such as FORTRAN (McCoy and Levary 1988).

Acknowledgment

The authors would like to acknowledge MDC for providing the resources and facilities to conduct this study.

References

ALDRICH, T. B., SZABO, S. M., and BIERBAUM, C. R., 1988, The development and application of models to predict operator workload during system design. Unpublished manuscript, Anacapa Sciences, Inc., Fort Rucker, AL.

ASIALA, C. F., MILLER, J. T., WILPER, B. L., and MCCOY, M. S., 1982, *Israeli Ministry of Defence Pilot Workload Assessment Users' Reference Guide* (St. Louis, MO: McDonald Douglas Corp).

AUSBURN, F. B., AUSBURN, L. J., and RAGAN, T. J., 1980, Task analysis schema based on cognitive style and supplantational instructional design with applications to an air force training course (Report No. AFHRL-TR-79-59). Brooks Air Force Base, TX: Air Force Systems Command.

BARON, S., and LEVISON, W. H., 1980, The optimal control model: status and future directions. *Proceedings of the IEEE Conference of Cybernetics and Society* (Cambridge, MA: IEEE).

BARON, S., MURALIDHARAN, R., LANCRAFT, R., and ZACHARIAS, G. L., 1980, *PROCRU: A model for analyzing crew procedures in approach to landing* (BBN Report No. 4374) (Cambridge, MA: BBN Laboratories).

BARON, S., KRUSER, D. S., and HUEY, B. M., (Eds), 1990, *Quantitative Modeling of Human Performance in Complex Dynamic Systems* (Washington, DC: National Academy Press).

BOFF, K., and LINCOLN, J., (Eds), 1988, *Engineering Data Compendium: Human Perception and Performance* (Wright-Patterson Air Force Base, OH: Harry G. Armstrong Aerospace Medical Research Laboratory).

CARD, S. K., MORAN, T. P., and NEWEL, A., 1983, *The Psychology of Human–computer Interaction* (Hillsdale, NJ: Lawrence Erlbaum Associate).

CHU, Y., and ROUSE, W. B., 1979, Adaptive allocation of decision making responsibility between human and computer in multitask situations. *IEEE Transactions on Systems, Man and Cybernetics,* **9**(12).

CURRY, R. E., DICK, R., and PARKER, E., 1985, The use of knowledge-based simulation models to verify operability in new combat systems–An initial application to the an/sqqq-89 asw combat system (Report No. NPRDC-TR-86-3) (San Diego, CA: Navy Personnel Research Development Center).

ELKIND, J. I., CARD, S. K., HOCHBERG, J., and HUEY, B. M., (Eds), 1989, *Human Performance Models for Computer-aided Engineering* (Washington, DC: National Academy Press).

GIARRATANO, J. C., 1993, *CLIPS User's Guide* (Houston, TX: NASA Lyndon B. Johnson Space Center, Information Systems Directorate, Software Technology Branch).

GIARRATANO, J. C., and RILEY, G., 1994, *Expert Systems: Principals and Programming* (Boston, MA: PWS Publishing).

HARRIS, R. M., IAVECCHIA, H. P., ROSS, L. V., and SHAFFER, S. C., 1987, Microcomputer human operator simulator (HOS-IV). *Proceedings of the 32nd Annnual Meeting of the Human Factors Society* (Santa Monica, CA: Human Factors).

HUNT, R. M., and ROUSE, W. B., 1984, A fuzzy rule-based model of human problem solving. *IEEE Transactions on Systems, Man and Cybernetics,* **14**, 112–120.

KELLER, K. J., and STANLEY, K. C., 1991, Pilots associate: An integration architecture for decision making. *AIAA Conference on Guidance, Navigation and Control,* AIAA91-2759.

KNOOP, P. A., 1978, Survey of human operator modeling techniques for measurement applications. Report Number AFHRL-TR-78-35. Wright–Patterson Air Force Base, Ohio: Advanced Systems Division.

LAUGHERY, R., DREWS, C., and ARCHER, R., 1986, A micro SAINT simulation analyzing operator workload in a future attack helicopter. *Proceedings NAECON* (New York: IEEE).

LANE, N. E., STRIEB, M. I., GLENN, F. A., and WHERRY, R. J., 1981, The human operator simulator: an overview. In J. Moraal and K. F. Kraiss (Eds.) *Manned Systems Design: Methods, Equipment, and Applications* (New York: Plenum Press).

MCCOY, M. S., 1995, A rule-based pilot performance model. Ph.D. Dissertation, St. Louis, Mo.: St. Louis University.

McCoy, M. S., and Boys, R., 1987, Human performance models applied to intelligent decision support systems. *National Aerospace and Electronics Conference (NAECON).*

McCoy, M. S., and Levary, R. R., 1988, Augmenting knowledge acquisition processes of expert systems with human performance modeling techniques. *IEEE Transactions on Systems, Man, and Cybernetics,* **18,** 467–472.

McCoy, M. S., and Seavers, C. R., 1993, Human engineering systems analysis report (HESAR). (Report No. MDC 93B0081, Rev. A) (St. Louis, MO: McDonnell Douglas Corp).

McCoy, M. S., Hunter, M., Seavers, C., and McClure, A., 1993, Human engineering dynamic simulation plan. (Report No. MDC 93B0082, Rev. A) (St. Louis, MO: McDonnell Douglas Corp).

McCracken, J., and Aldrich, T. B., 1984, Analysis of selected LHX mission functions (Technical Note ASI 479-024-84(B)) (Fort Rucker, AL: Anacpa Sciences).

McRuer, D. T., and Krendel, E. S., 1974, *Mathematical Models of Human Pilot Behavior*. AGARD No. 188 (North Atlantic Treaty Organization: Advisory Group for Aerospace Research and Development).

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* (New York: Plenum Press).

Minskey, M. L., 1968, Monitoring behavior and supervisory control. In K. Boff, L. Kaufmann and J. Beatty (Eds.), *Handbook of Human Perception and Performance* (New York: Wiley).

Morris, N. M., Rouse, W. B., and Ward, S. L., 1985, Experimental evaluation of adaptive task allocation in the aerial search environment. *IFAC Man-Machine Systems*, Varese, Italy.

Nisbett, R. E., and Wilson, T. D., 1977, Telling more than we can know: verbal reports on mental processes., *Psychological Review*, **84,** 231–259.

North, R. A., 1985, WINDEX: A workload index for interactive crew station evaluation. *Proceedings NAECON* (New York: IEEE).

North, R. A., and Riley, V., 1988, WINDEX: a predictive model of operator workload. In G. MaxMillan, (Ed.), *Human Performance Models.* NATO AGARD Symposium, Orlando, FL.

Pilots Associate Program, 1990, Technical Operating Report (TOR) System Design Document. (Report No. MDC B2517) (St. Louis, MO: McDonnell Douglas Corp).

Pilots Associate Program, 1991, phase 1–Final report (Report No. MDC 91B0012) (St. Louis, MO: McDonnell Douglas Corp).

Pritsker, A. B., 1984, *Introduction to Simulation and SLAM II* (New York: Halstead Press).

Pritsker, A. B., Wortman, D. B., Seum, C., Chubb, G., and Seifert, D. J., 1974, *SAINT: Vol. I: Systems Analysis of an Integrated Network of Tasks* (Report No. AMRL-TR-73-126) (Wright–Patterson Air Force Base, OH: Aerospace Medical Reserach Laboratory).

Rouse, W. B., 1980, *Systems Engineering Models of Human–Machine Interaction* (New York: North Holland).

Rouse, W. B., Geddies, N. E., and Currey, R. E., 1988, An architecture for intelligent interfaces: Outline of an approach to supporting operators of complex systems. In *Human–Computer Interaction* (New York: Lawrence Erlbaum Associates) pp. 87–122.

Sacerdoti, E. D., 1977, *A Structure for Plans and Behavior* (New York: Elsevier, North Holland)

Sage, A. P., 1991, *Decision Support Systems Engineering* (New York: Wiley); 1992, *Systems Engineering* (New York: Wiley).

Schank, R. C., and Abelson, R. P., 1977, *Scripts, Plans, Goals, and Understanding* (Hillsdale, NJ: Lawrence Erlbaum Associates).

Schuppe, T. F., 1988, *A Methodolgy for Predicting Pilot Workload* (Report No. AFIT/CI/NR 88-169).

Senders, J. W., 1977, Axiomatic models of workload. In N. Moray (Ed.), *Mental Workload, its Theory and Measurement* (New York: Plenum Press), pp. 263–267.

Sheridan, T. B., 1992, *Telerobotics, Automation, and Human Supervisory Control* (Cambridge, MA: MIT Press).

Sheridan, T. B., and Hennessy, R. T., (Eds.), 1984, *Research and Modeling of Supervisory Control Behavior* (Washington, DC: National Academy Press).

Siegel, A. I., and Wolf, J. J., 1969, *Man–machine Simulation Models* (New York: Wiley).

Swain, A. D., and Guttman, H. E. (Eds.), 1980, *Handbook of Human Reliability Analysis with Emphasis on Nuclear Power Plant Applications.* Technical Report NUREG/CR-1278 (Alburquerque, New Mexico: Sandia National Laboratories).

Valayas, J. M., and Levary, R. R., 1987, Validation of simulation models using decision theory. *Simulation,* **48**(3), 87–92.

Wherry, R. J., 1976, Human operator simulator: HOS. In T. Sheridan and G. Johannsen (Eds.), *Monitoring Behavior and Supervisory Control* (New York: Plenum Press).

Wickens, C. D., 1984, *Engineering Psychology and Human Performance* (Columbus, OH: Charles E. Merril); 1985, The multiple resource model of human performance: Implications for display design. *AGARD Conference Proceedings,* **371,** 17-1–17-6; 1987, *Pilot Performance Models: Resource Management and Time-sharing* (Champaign, IL: University of Illinois); 1989, Resource management and time-sharing. In J. I. Elkind, S. K. Card, J. Hochberg, and B. M. Huey (Eds.), *Human Performance Models for Computer-aided Engineering* (Washington, DC: National Academy Press).

Wickens, C. D., and Liu, Y., 1988, Codes and input modalities in multiple resources: A success and a qualification. *Human Factors*, **30**.

Wise, J., and Asiala, C., 1977, *F-18 Human Engineering Task Analysis Report (Fighter Task Analysis—Part I)*, MDC A4276-1 (St. Louis, Mo: McDonnell Aircraft Co.).

Wise, J., Asiala, C., and Loy, S., 1977, *F-18 Human Engineering Task Analysis—Part II: Timeline and Workload Analysis (Fighter Task Analysis)*, MDC A4276-2 (St. Louis, Mo.: McDonnell Aircraft Co.).