Task-Analytic Models of Human Operators:

Designing Operator-Machine Interaction*

Christine M. Mitchell

Center for Human-Machine Systems Research

School of Industrial and Systems Engineering

Georgia Institute of Technology

Atlanta, GA 30332-0205

(404) 894-4321

(404) 894-2301 (fax)

cm@chmsr.gatech.edu

Introduction

Human operators are essential to ensure the safe, effective, and efficient operation of complex dynamic systems. Examples of such systems include process control plants, power plants (e.g., nuclear, fossil fuel, etc.), increasingly automated manufacturing systems, the flight deck of modern commercial aircraft, and other aerospace systems.

Operators are often the 'glue' that makes complex dynamic systems work. Operators provide bridges between islands of automation that typically constitute the control portion of a system (Jaikumar, 1986). Organizational structure and economic pressures often result in control systems defined by a collection of loosely coordinated subsystems some of which are automated. Operators are often the most important mechanism that enables various subsystems to act together as a coherent whole. Operators transfer essential state data, configuration information, or system constraints from one subsystem to another.

Operators provide an essential back-up to the control system as a whole and to individual subsystems. This is an important operator function as automated control systems are often brittle (Woods, Johannesen, Cook, & Sarter, 1994). That is, automated subsystems may be effective within the specified domain of control, but fail quite seriously when encountering unanticipated circumstances or circumstances which, for financial and organizational reasons, are not included in the system specifications. Examples of operators performing such functions abound in every complex system.

Due to limitations in span of control and the brittle nature of current automation, control systems still rely on extensively on human ingenuity, creativity, and flexibility to ensure safe and effective operation. Although human error accounts for a significant proportion of errors in complex systems (U.S. GAO, 1966), without human operators most systems could not operate effectively, or, sometimes, could not operate *at all*.

Given the critical role of the human operator, the lack of attention and the limited resources allocated to ensuring that the human operator is able to effectively carry out the needed functions is a very serious limitation of most current design processes. There are virtually no resources dedicated to designing an effective role for the human operator or to ensuring the availability of appropriate tools so that operators can effectively carryout their control functions. Even if conducted, traditional human factors assessments or human interface reviews often focus on low-level issues such as environmental or physiological factors or features of computer-based workstations such as window management, color, or font size.

The thesis of this paper is that the role of the human operator in the control of complex dynamic systems should be as carefully and rigorously specified as the system hardware, software, and functionality. To accomplish this objective, models of operator function are needed to characterize the role of the operator and the associated activities used to carry out the control functions. Such models rigorously define an envelope of expected or implied operator activities. Furthermore, such models provide necessary knowledge to address operator-related issues including the allocation of functions between human and computer controllers, the design of 'intelligent' displays and operator workstations, operator decision support systems, and adaptive, potentially embedded, training systems. Finally, operator models and their related computational implementations may provide a repository for the accumulation of operator expertise as new generations of operators replace personnel with decades of experience. Accumulating, organizing, and making available accumulated experience and expertise is a critical component if computer-based tools are to be effective.

The sections that follow discuss the importance and use of models of operator function in complex dynamic systems. The first discusses the general characteristics of such models and their uses. The second summarizes one modeling methodology, the operator function model, that has been developed and extensively used to describe the role of the human operator and to guide the development of operator displays, aids, and training systems.

Models of Operator Function

Human-machine systems engineering is an interdisciplinary effort that seeks to understand, describe, and prescribe the activities of humans interacting with complex systems. With roots in modern and optimal control theory, human-machine systems engineering highlights the interesting fact that though models of human reasoning, decision making, and other sophisticated cognitive processes are very difficult to construct or validate, human performance in control of a complex dynamic system can in fact be modeled quite well. Human operators are, and thus can be successfully modeled as such, well-trained and well-motivated individuals who in the execution of their professional functions typically carry out their tasks in predictable ways.

Sheridan (1976; 1992), one of the foremost leaders in the formation of human-machine systems engineering as an engineering discipline, distinguishes it from other areas concerned with humans and their interaction with equipment, machines, or systems as follows.

The 1940s and the 1950s saw the emergence of *human factors* (*ergonomics* in Europe), first in an empirical 'knob and dials' form concentrating on the human-machine interface...In contrast with human-factors engineering with its emphasis on interface, human-machine systems analysis considers characteristics of the entire causal 'loop' of decisions, communications, control, and feedback through the human operator's physical environment and back to the operator (p. 7-8).

Thus, human-machine systems analysis, models, and design are intended to complement traditional engineering analysis, models, and design methods used to specify system hardware, software, and overall functionality. Figure 1 depicts an overview of the critical issues in the design of operator-machine interaction.

The Role of the Operations Team. Models of operator function are essential to *understand* what an operator or an operations team is expected to do. It is frequently the case, lacking models of operator function, that designers of control systems specify the type and number of operations personnel with little concept of what they will actually do. The response when questioned is often: The operators are there to do whatever is necessary to 'make the system work.' In other words, operators provide the *glue*, automation *back-up,* and on-line *debugging* to ensure that the system operates safely and effectively.

Without models that clearly delineate expectations, there is no mechanism to ensure that operators *can* effectively carry out the implied functions. Many operators (airline pilots being the most vocal) argue that most *human error* in such systems is in fact *design error*--that is, the control system was not designed to be effectively operated by a human being (FAA, 1996; Woods et al., 1994). Consider, for example, the uselessness of the dozens of audio alarms that characterize critical incidents in many control systems (e.g., power plants, aircraft, manufacturing systems).

Without models of operator function, *allocation* of functions between human and computer controllers happens by default. Computer control is designed to carry out all functions that current technology and organizational resources permit; the human operator is responsible for whatever remains. This strategy makes no systematic attempt to utilize the strengths or to compensate for the limitations of the human beings who are ultimately responsible for effective system operation. At the most fundamental level, this design strategy, or essentially the lack of a coherent design strategy, wastes expensive and versatile human resources and, thus, constitutes poor engineering design.

Operator Workstations and Displays. A survey of most control room workstations through which operators monitor the system and carry out essential control functions provides a demonstration of the problems that arise when a model of operator function is lacking. Rasmussen (1986) characterizes such displays as representing one-sensor-one-display technology. Displays typically contain, in unrefined form, data at the sensor level of detail. Operators become data gatherers, filters, and transducers, transforming raw sensor data into useful control information.

Historically, there were few alternatives to one-sensor-one-display design: technology permitted nothing else. Current computer-based displays and inexpensive workstations, however, provide the technology for displays that can tailor information given the operator's current goals and the system's current state. Reducing the information processing burden associated with data-intensive computer-based control systems may reduce the operator's overall cognitive workload and thus reduce error-prone behavior.

At this time, given inexpensive and powerful workstation technologies, poor displays exist because designers do not know how to specify display *content*. They have little knowledge or formal specifications of the operator's information or control needs. Designers lack a design methodology.

'Intelligent' displays may be characterized as displays that provide the *right* information (filtered by current operator goals), at the *right* level of detail (useful information as opposed to raw data), at the *right* time (contextualized based on system state). Designing such displays requires a characterization of operator control functions that is linked to information needs based on evolving system state. A model of operator function provides an initial specification to define the collection of display pages and their content. Rather than representing machine functions (e.g., all sensor information associated with the power system), displays based on a model of operator function link display pages and contents to the needs of operator control. Dynamic displays, that is, displays that tailor displayed information to the context of changing system state, require software that explicitly links information to operator control activities and system state. The design of such software requires explicit specification of what information is needed, when, and how to derive it from system state and sensor data.

It is important to note that this discussion is about information/control requirements, i.e., *workstation semantics*, not visual/interaction form, i.e., *workstation syntax*. Given a design methodology, specification of workstation semantics can and should be a engineering design task comparable to hardware and software engineering. Definition of workstation syntax, e.g., choice of icons, colors, interaction modes, is primarily an art, and likely to remain so. Research both in the real world and in the laboratory shows that when compared for well-trained operators, workstation semantics whose definition is based on a model of operator activity provides a lasting enhancement to and stability of operator performance (Mitchell & Saisi, 1987; Thurman & Mitchell, 1995); whereas, enhancements to workstation syntax may result in no improvement or they may reduce learning time but have performance enhancements that disappear as operators become increasingly well trained (Kirlik, Kossack, & Shively, 1994; Pawlowski, 1990).

Operator Aids, Assistants, and Associates. As knowledge-based technology becomes more sophisticated, the urge to enhance operator performance, reduce operator error, or consolidate the number of operations personnel by adding intelligent aids to the system is irresistible. To date, most intelligent systems that function as operator aids have limited utility and little long-term success. NASA space systems have many examples of intelligent assistants that operators use only when management wants a demonstration. The Pilot's Associate program has yet to be tested under conditions remotely resembling realistic operational scenarios.

One hypothesis is that many of these systems can be characterized as 'black-box' automation. They give the operator advice, reminders, and offers of assistance that may fail to meet the operators' current needs or which operators do not trust. Operators do not understand how such systems work or the limitations of offers of help. The only clue many expert systems provide about how they arrive at a conclusion is a 'rule trace-back' that can be read easily only by software developers; though even developers would be loath to explain a specific decision in the time frame necessitated by the evolving state of dynamic systems. In addition, operator aids, just as any other piece of software, are brittle--in the face of unplanned events or circumstance that are not represented in the software, such systems tend not to fail gracefully.

A (human) operations team is characterized by a high degree of collaboration and interaction that dynamically formulates, clarifies, and modifies control goals and associated control activities. Human interaction with black-box automation fails to provide operators with the ability to easily answer essential questions: Why did it recommend that? Under what assumptions was this recommendation made? What if the current state fails to match exactly the circumstances assumed in the formulation of the recommendation? Woods and colleagues (1994) have proposed an alternative paradigm for designing intelligent aids, 'glass-box' systems. Such systems are inspectable and repairable by operators in real time. They support a dialogue that emulates the natural and highly effective interactions of a team of trained human operators.

Glass-box automation is indeed an attractive design goal. The design, implementation, and operation of such a system is not, however, straight forward. Billings (1991), in his landmark treatise on human-centered automation lays out as an axiom the requirement that intelligent operator aids support *mutual intent recognition* between the operator(s) and the intelligent system (s). The ability to understand each other, from a computational point of view, at the very least requires a detailed model of operator function. The intelligent assistant can use the model both to interpret operator actions and to formulate timely and relevant offers of assistance or advice. The shared model of operator function supports, in computational form, a flexible dialogue that can be conducted at multiple levels of abstraction. It is a framework for collaboration that is explicit, inspectable, repairable, and maintainable.

Intelligent and/or Embedded Training Systems. Training personnel to operate complex systems is an increasingly difficult but critically important function. Modern organizational life is characterized by high personnel turnover. The days are gone when employees with thirty years of experience provide the expertise and corporate memory that form the backbone of flexible system control. In NASA satellite control systems, 100% employee turnover occurs on the average of every twelve months.

Exacerbating the lack of stable human expertise is the fact that emerging technologies, (i.e., hardware, software, and applications), change, almost continuously, the structure and function of the underlying control system. Even operators with lengthy tenure with a given system are confronted with, usually minor, but sometimes significant, system changes on a regular basis. Flight management systems on modern aircraft change almost monthly. NASA control systems are updated with new releases every six months.* Thus, for some conditions, every operator is a *trained novice* rather than a true *expert* (Chappell & Mitchell, 1995).

Intelligent tutoring systems (ITS) are often proposed as one strategy to address training problems. Such systems provide a stable repository of system knowledge (descriptive, procedural, and operational). Linked to a simulator, computer-based training systems provide a safe environment in which operators can practice new skills or explore the limits of the newest release of control software. With the addition of an 'intelligent' tutor, computer-based training systems can tailor instruction to the needs of individual operators.

Two challenges in ITS design are specification of an expert model that guides instruction and a student model that tracks the evolution of student knowledge and helps to tailor instruction to an individual student's needs. Combined, the expert and student models give an ITS the ability to allow a student with limited instructional needs to progress quickly through the curriculum or to provide additional instruction for a student experiencing problems in a specific area. Although the literature has many candidate modeling structures for intelligent tutoring systems, a model of operator function is a viable candidate for the expert model in a training system that teaches students how to control a complex dynamic systems.

One successful strategy for designing a student model is by formulating it as an overlay model on the expert model. That is, the student model has the same structure as the expert model. As training progresses, the student model is annotated to indicate areas in which the student is successful or may need addition assistance. The pedagogical component of an ITS compares the expert and student model to formulate instruction precisely tailored to the needs of an individual student.

Thus, a model of operator function may provide the structure and knowledge necessary to design and control an intelligent tutor. In addition, if comparable models of operator function are used for displays, aids, and tutors, an integrated system might be conceived in which a student evolves from novice to expert with the system that controls training gradually becoming a well-understood and trusted assistant.

Summary. There are several strategies for modeling system and operator functions that can be used to guide the formal description of an operator's role and form the basis of the design of operator tools and training systems that, both in theory and in fact, make operators more effective. One such methodology is presented below.

Operator Function Model (OFM)

Discrete control models and models of operator function using discrete control modeling constructs have been successfully used to *describe* operator behavior in a range of complex systems (Miller, 1985) and to *prescribe* operator functions by representing the interrelations between dynamic system state and operator functions, subfunctions, control actions, and information needs related to operator activities (Mitchell and Miller, 1986; Mitchell, 1987, 1996). The latter, operator function models (OFM), have been proposed as a modeling tool that provides a dynamic, task-analytic structure that can be used by system designers to define a user interface that is based on operator rather than hardware function (Mitchell, 1987, 1996).

The operator function model is proposed as an alternative to traditional task analysis techniques used by human factors engineers (McCormick, 1976; Sanders and McCormick, 1987). Most frequently used in the study of human error in nuclear power plant control rooms (INPO, 1983), task analysis is typically a static and somewhat ill-defined methodology. McCormick (1976), in his classic text on human factors, summarizes the task analysis procedure in one global step: "Identify and list all the human operations performed and their relation to system tasks (p. 24)."

To more effectively describe and structure operator activities in control of a complex dynamic system, a more rigorous structure is required. The model or representation must be dynamic and analytic. A dynamic representation requires that operator activities be modeled within the context of changing system state. An analytic representation is one that can be easily coded into software and readily used to characterize the semantics a user interface, operator aid, or tutor requires.

The Structure of the Operator Function Model: Network, Hierarchy, and Heterarchy. An operator function model attempts to represent in mathematical form how an operator might decompose a complex system into simpler parts and coordinate control actions and system configurations so that acceptable overall system performance is achieved. The model represents basic issues of knowledge representation, information flow, and decision making in complex systems. Miller (1985) suggests that the network structure can be thought of as a possible representation of an operator's internal model of the system plus a control structure which specifies how the model is used to solve the decision problems that comprise operator control functions.

Mathematically, the operator function model is defined in terms of a network of finite state automata. The structure of the model is a general hierarchic/heterarchic scheme which structurally accounts for coordination of operator activities and focus of attention. Figure 2 depicts a generic operator function model.

The model's hierarchy helps to organize complexity with higher level nodes specifying the purpose for which lower level activities are undertaken. Nodes at the top level represent major operator functions; nodes at the bottom define operator actions. The general framework depicts each function decomposed into a collection of subfunctions, subfunctions are decomposed into a number of tasks, and tasks are accomplished by one or more operator actions. The exact number of levels in the operator function model, however, varies within and across applications. At times, for example, the system being modeled does not require a subfunction level; at others, there is a hierarchy of subfunctions.

The model's heterarchy depicts the concurrent nature of supervisory control: at any given time the operator may be coordinating multiple activities. The arcs in the model connecting nodes at the same level represent system triggering events or the successful completion of operator activity. Arcs can be thought of as either pre-conditions or initiating events. Heterarchically, arcs may define the expected temporal flow of control activity.

The next-state transition functions associated with the finite-state automata at each node in the hierarchy may be non-deterministic: a map from a current state into a collection of *possible* next states. A non-deterministic transition reflects the flexibility an operator has in coordinating and executing activities comprising effective control as well as system constraints over allowable operator activities. The collection of feasible next states denotes the choices an operator can make about the next activity to pursue. States not included in the feasible set indicate possible operator errors.

Deriving an Operator Function Model. Constructing an operator function model is typically iterative and, depending on the modeling purpose, may proceed bottom-up or top-down. Some descriptive modeling proceeds bottom-up, with the specification of system outputs first. Several such applications of the operator function model base their structure on configuration of system controls or switches, e.g., Miller (1985) and Mitchell and Miller (1986). Thus, switch settings and the associated operator actions required to configure them form the lowest level nodes of the model. This structure is very useful for quantifying operator behavioral data. Modeling proceeds by aggregating and abstracting actions into groups that attempt to explain why an individual action or group of actions is undertaken. The process continues until a set of plausible high-level operator functions is formulated.

The development of prescriptive models to guide design typically proceeds top-down, beginning with the high-level specification of operation control functions. Functions are decomposed in constituent parts. Modeling may then jump to the lowest level--operator actions. Actions are typically not switch setting, but input to a computer interface through which the operator controls the system. The modeling proceeds iteratively, and both top-down and bottom-up: lower level nodes define how and under what conditions higher level activities are carried out, and bottom-up; higher level nodes define when and why lower level activities are executed.

Control Actions: Physical and Cognitive. Control actions represented in the operator function model may take one of two forms: manual or cognitive. Examples of manual actions include: "set switch A to position 1" or "replace component A with some available component of the same class." Cognitive actions represent operator activities such as monitoring and situation assessment, e.g., "assess system state and determine if component A is operating properly."

Cognitive actions comprise an important component of the supervisory controller's overall role in the system. The supervisory controller is often passive, monitoring an automated process that typically controls and adjusts itself without the need for human intervention. Monitoring, however, does not mean that the operator is not doing anything; rather, it means that a well-trained operator engages in systematic examination of displayed variables and periodic assessment of system state. Cognitive actions do not typically alter system state, change system switch configurations, or necessarily require any physical operator movement. Depending on the specific syntax of the operator workstation, cognitive actions may require the operator to examine a piece of displayed information or, in order to obtain the necessary information, request one or more display pages.

Information Sources and Operator Commands: Links to Operator Actions. Cognitive actions include activities such as information gathering, integration and processing, and decision making. In addition to representing both cognitive and manual actions, the operator function model often links a set of likely operator information needs to a cognitive action node, e.g., identifying a replacement component requires examination of displays showing current hardware status and schedules. In a computer-based, multi-page display workstation, execution of the set of relevant information requests is likely to be the only observable indication that the operator is currently carrying out the necessary cognitive activities.

The use of operator actions that are both manual and cognitive together with a linkage between cognitive actions and information needs is an important characteristic of the operator function model. Modeling cognitive as well as manual actions enables the model to reflect more accurately operator interaction in a supervisory control environment. Used predictively, the model can identify the information needed to undertake a cognitive activity. Used descriptively, the model can attempt to explain why a series of information requests was made. Used inferentially, an on-line implementation of the model may be able to infer and explain current operator action or the lack of occurrence of a seemingly necessary operator action.

In addition to linking needed information to cognitive action nodes, the operator function model may link specific operator commands to manual operator actions. The linkage between manual operator actions and either a specific command or set of commands is similar to the linkage between cognitive action nodes and information nodes. The operator function model uses a semantic interpretation of action at the action node level; command nodes define the syntax needed to carry out the intended action.

Summary. The rigorous definition of detailed operator activities and system interaction represented in the operator function model provides operator workstation designers with a detailed understanding of the human component of the system. This representation is as well-defined as typical functional specifications of system hardware and software. The operator function model allows meaningful answers to critical automation questions such as the following: What activities constitute an operator function? How is a function carried out? What choices does an operator have to bring about a desired state? What operator actions are likely to be errors, given a current operator function and system state? Is the operator too busy; perhaps not busy enough? What is the effect on overall the operator function if a specified subfunction is automated?"

Operator errors, either potential or actual, can be analyzed by means of an operator function model. With a formal understanding of what the operator should do, when, and why, unexpected or erroneous actions can be examined to see if they are indeed simply accidents or random mistakes by the operator, or if they are logical consequences of a poorly designed human-machine interface.

Equally important, the operator function model provides a basis for friendly or 'intelligent' user-interface design. From a user's point of view, an interface is friendly if it knows what the user wants and can help the user achieve current goals. From the computer's point of view, a functional definition of 'friendly' requires an interface that has a detailed and comprehensive representation of the user's goals and activities together with a specification of how and when each relates to system state. The operator function model has this breadth and level of detail. It is intended to be implemented in the human-computer interface as a supervisory program that interprets user commands and information queries and provides adaptive responses.

The basic operator function model structure is flexible enough to meet the diverse needs of a variety of real system contexts and applications. The operator function model has been applied to the analysis of information requirements and activity in ship navigation (Lee and Sandquist, 1993); manufacturing control system design (Dunkler et al., 1988); troubleshooting in electronics assembly (Cohen, Govindaraj, and Mitchell, 1992); navigation in advanced aviation systems (Callantine and Mitchell, 1994; Verfurth, 1991); design of visual displays (Mitchell and Saisi, 1987; Thurman and Mitchell, 1994); and the development of a blackboard architecture for dynamic intent inferencing (Rubin, Jones, and Mitchell, 1988) and operator aiding (Bushman, Mitchell, Jones, and Rubin, 1993; Jones and Mitchell, 1995) ; and the design of both expert and student models for an intelligent tutoring system that teaches satellite ground control operations (Chu, Mitchell, and Jones, 1995).

References

Billings, C. (1991). Human-centered aircraft automation: A concept and guidelines (NASA Tech. Memo. 103885). Moffett Field, CA: NASA Ames Research Center.

Bushman, J. B., Mitchell, C. M., Jones, P.M. and Rubin, K. S. (1993). ALLY: An operator's associate for cooperative supervisory control systems. *IEEE Transactions on Systems, Man, and Cybernetics*, 23(1), 111-128.

Callantine, T. and Mitchell, C. M. (1994). A methodology for understanding how operators use modes of automation. *Proceedings of the 1994 IEEE International Conference on Systems, Man, and Cybernetics,* San Antonio TX, October 1994.

Chappell, A. R. & Mitchell, C. M. (1995). Addressing the trained novice/expert performance gap in complex dynamic systems: A case-based intelligent tutoring system. In *Proceedings of the 1995 IEEE International Conference on Systems, Man, and Cybernetics* (pp. 4557-4562). Vancouver, BC, Canada.

Chu, R. W. and Mitchell, C. M., and Jones, P. M. (1995). Using the operator function model/OFMspert as the basis for an intelligent tutoring system: Towards a tutor-aid paradigm for operators of supervisory control systems. *IEEE Transactions on Systems, Man and Cybernetics*, 25(7), 1054-1075.

Cohen, S. M., Govindaraj, T., and Mitchell, C.M. (1992). Analysis and aiding the human operator in electronics assembly. In M. Helander and M. Nagamachi (Eds.), *Design for Manufacturability: A Systems Approach to Concurrent Engineering and Ergonomics*. London: Taylor & Francis, 361-376.

Dunkler, O., Mitchell, C. M. Govindaraj, T., and Ammons, J. C. (1988). The effectiveness of supervisory control strategies in flexible manufacturing systems. *IEEE Transactions on Systems, Man and Cybernetics*, 18(2), 223-237.

FAA (1996). *The interfaces between flightcrews and modern flight deck systems.* Report Human Factors Team Report. Washington, D.C.: Federal Aviation Administrations.

INPO, (1983). Control Room Design Review Task Analysis Guideline, Nuclear Utility Task Action Committee, INPO 83-046 (NUTAC), December.

Jones, P. M. and Mitchell, C. M. (1995). Human-computer cooperative problem solving: Theory, design, and evaluation of an intelligent associate system. *IEEE Transactions on Systems, Man and Cybernetics,* 25(7), 1039-1053.

Jaikumar, R. (1986). Postindustrial manufacturing. *Harvard Business Review*, *November-December*, 69-76.

Kirlik, A., Kossack, M. F., & Shively, R. J. (1994). *Ecological task analysis: Supporting skill acquisition in dynamic interaction.* Report Center for Human-Machine Systems Research, School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA.

Lee, J. and Sanquist, T. (1993). A systematic evaluation of technological innovation: A case study of ship navigation. *Proceedings of the 1993 IEEE International Conference on Systems, Man, and Cybernetics, Vol. 1*, 102-107, Le Touquet, France.

McCormick, E. J. (1976). *Human factors in engineering and design.* New York: McGraw Hill.

Miller, R. A. (1985). A system approach to modeling discrete control performance. In Rouse, W. B. (Ed.), *Advances in Man-Machines System Research*, Vol. 2. Greenwich, CT: JAI Press.

Mitchell, C. M. and Miller, R. A. (1986). A discrete control model of operator function: A methodology for display design. *IEEE Transactions on Systems, Man, and Cybernetics*, 16(3), 353-369.

Mitchell, C. M. (1987). GT-MSOCC: A domain for research on human-computer interaction and decision aiding in supervisory control systems. *IEEE Transactions on Systems, Man, and Cybernetics*, 17(4), 553-572.

Mitchell, C. M. (1996). GT-MSOCC: Operator models, model-based displays, and intelligent aiding. In W. B. Rouse (Ed.), *Human-technology interaction in complex systems* (pp. 233-293). Greenwich, CT: JAI Press Inc.

Mitchell, C. M. and Saisi, D. S. (1987). Use of model-based qualitative icons and adaptive windows in workstations for supervisory control systems. *IEEE Transactions Systems, Man, Cybernetics*. 17(4), 594-607.

Pawlowski, T. J. (1990). *Design of operator interfaces to support effective supervisory control and to facilitate intent inferencing by a computer-based operator's associate*. Unpublished doctoral dissertation, Center for Human-Machine Systems Research, School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA.

Rasmussen, J. (1986). *Information processing and human-machine interaction: An approach to cognitive engineering*. New York: North-Holland.

Rubin, K. S., Jones, P.M. and Mitchell, C. M. (1988). OFMspert: Inference of operator intentions in supervisory control using a blackboard architecture. *IEEE Transactions on Systems, Man, and Cybernetics*, 18(4), 618-637.

Sheridan, T. B. (1976). Toward a general model of supervisory control. In T. B. Sheridan and G. Johannsen (Eds.), *Monitoring behavior and supervisory control.* New York: Plenum Press, 271-281.

Sheridan, T. B. (1992). *Telerobotics, automation, and human supervisory control.* Cambridge, MA: MIT Press.

Thurman, D. A. & Mitchell, C. M. (1995). A design methodology for operator displays of highly automated supervisory control systems. In *Proceedings of the 6th IFAC/IFIP/IFOR/SEA Symposium on Analysis, Design, and Evaluation of Man Machine Systems* . Boston, MA.

U. S. Government Accounting Office (GAO) (1996). *Human factors: Status of efforts to integrate research on human factors into FAA's activities.* Report (GAO/RCED-96-151 Human Factors). Washington, D.C.: U.S. Government Accounting Office.

Verfurth, S. C. (1991). Modeling the pilots and constructing an intent inferencer for a Boeing 727 cockpit. M.S. thesis, Center for Human-Machine Systems Research, School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA.

Woods, D. D., Johannesen, L. J., Cook, R. I., Sarter, N. B. (1994). *Behind human error: Cognitive systems, computers, and hindsight*. Wright-Patterson Air Force Base, OH: Crew Systems Ergonomics Information Analysis Center.

Acknowledgments

This research has involved many people over the years whose participation, critical insight, and encouragement have made it far richer. In particular, the assistance and insights of T. Govindaraj, Richard Robison, Patty Jones, Jim Bushman, Rose Chu, Dave Thurman, and Alan Chappell are gratefully acknowledged. This work would not have been possible without the continuing stimulation and support, both intellectual and financial, of Ev Palmer and NASA Ames Research Center. Finally, many thanks for the support of Walt Truszkowski and other NASA Goddard Space Flight Center researchers, and to the NASA Goddard satellite ground controllers who introduced me and many generations of graduate students to the wonders and challenges of real-time control.

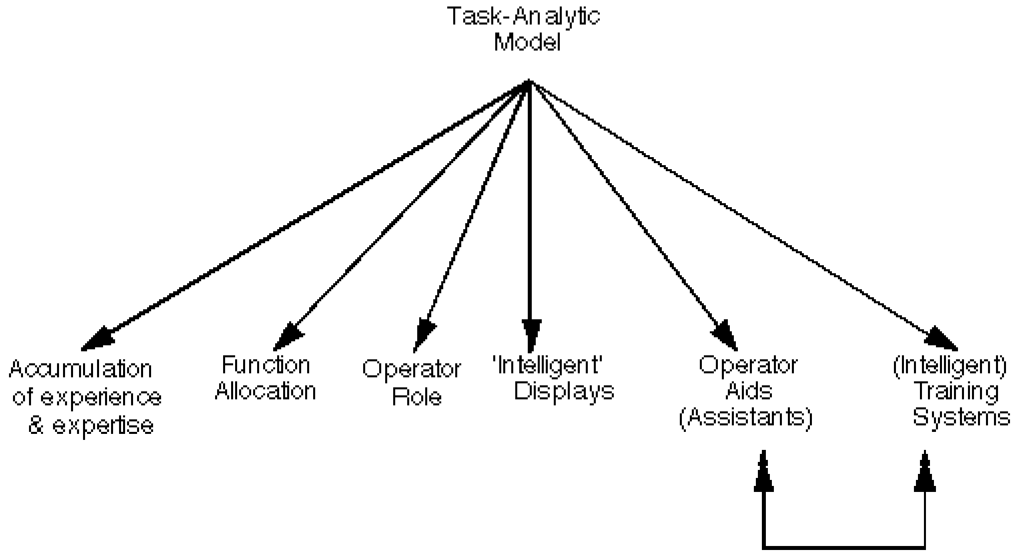

Figure 1. Design of Operator-Machine Interaction

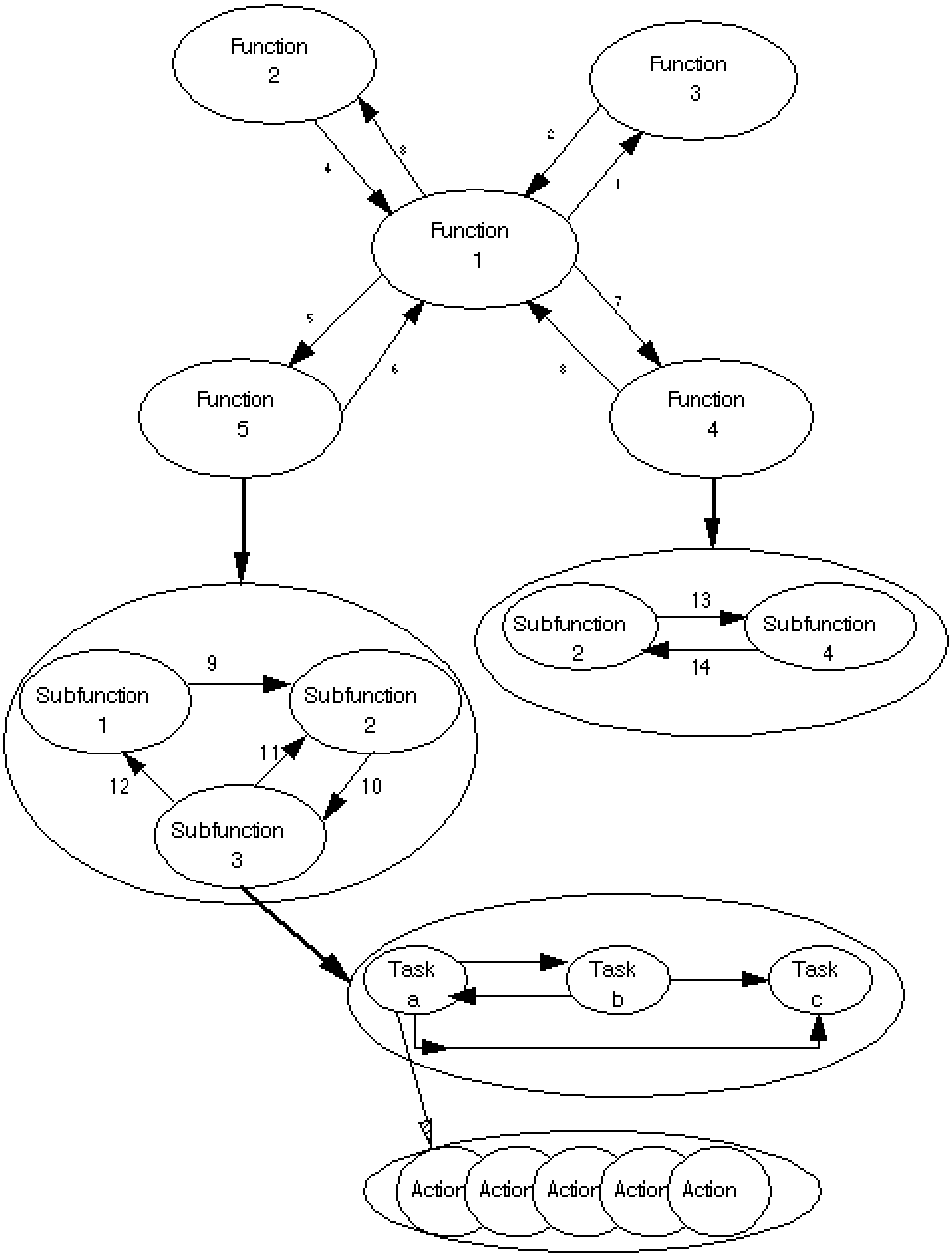

Figure 2. Structure of the Operator Function Model (OFM)