NASA-CR-193472

177367
P. 22

Cognitive Engineering Models:

A Prerequisite to the Design of Human-Computer Interaction in Complex Dynamic Systems

NAG9-422

Christine M. Mitchell

Center for Human-Machine Systems Research
School of Industrial & Systems Engineering
Georgia Institute of Technology
Atlanta, Georgia 30332-0205
404-894-4321
cm@chmsr.gatech.edu

P. Polson (ed.) ***Human Computer Interface Design: Success Cases, Emerging Methods, and Real-World Context.*** **New York: Morgan Kaufman, to appear.**

June 1993

(NASA-CR-193472) COGNITIVE ENGINEERING MODELS: A PREREQUISITE TO THE DESIGN OF HUMAN-COMPUTER INTERACTION IN COMPLEX DYNAMIC SYSTEMS Final Technical Report, 1 Jul. 1990 - 30 Nov. 1993 (Georgia Inst. of Tech.) 22 p

N94-12558

Unclas

G3/60 0177367

Introduction*

This chapter examines a class of human-computer interaction applications, specifically the design of human-computer interaction for the operators of complex systems. Such systems include space systems (e.g., manned systems such as the Shuttle or space station, and unmanned systems such as NASA scientific satellites), aviation systems (e.g., the flight deck of 'glass cockpit' airplanes or air traffic control) and industrial systems (e.g., power plants, telephone networks, and sophisticated, e.g., 'lights out,' manufacturing facilities).

The main body of human-computer interaction (HCI) research complements but does not directly address the primary issues involved in human-computer interaction design for operators of complex systems. Interfaces to complex systems are somewhat special. The 'user' in such systems--i.e., the human operator responsible for safe and effective system operation--is highly skilled, someone who in human-machine systems engineering is sometimes characterized as 'well trained, well motivated' (Baron, 1984). The 'job' or task context is paramount and, thus, human-computer interaction is subordinate to human-job interaction. The design of human interaction with complex systems, i.e., the design of human-job interaction, is sometimes called cognitive engineering (Woods and Roth, 1988; Rasmussen and Goodstein, 1988).

In Helander's (1988) seminal work, *Handbook of Human-Computer Interaction,* Rasmussen and Goodstein distinguish between human-computer interaction and cognitive engineering. Though both address human-machine systems, human-computer interaction is approached from the point of view of the computer and information science. Cognitive engineering, on the other hand, is approached from the point of view of an analysis of human work performance and the potential benefits and problems which may follow from the introduction of computer-based interfaces.

Rasmussen and Goodstein characterize HCI research as bottom-up and technology-driven.

> Basically, HCI is concerned with the interaction of users with computers in terms of the syntax of communication languages *irrespective of the context of work* (emphasis added) in which the systems are used, that is of the semantic aspects of work. Clearly this approach is important for the development and optimization of 'application programs' such as word-processors, graphics packages and spread-sheets (p.176).

Rasmussen and Goodstein define cognitive engineering as the design of human-computer interaction tailored for the work environment, specifically for operators of complex systems. The human-computer interaction issues in such systems are top-down and problem-driven.

> The quality of human-computer interaction in these cases can only be judged with reference to the ultimate system goals and constraints such as productivity and safety. A top-down approach to

---

* The help of Dr. Marianne Rudisill of NASA Johnson Space Center is gratefully acknowledge. Her extensive comments and editing suggestions have, I believe, made this a stronger and more readable paper.

the analysis, design and evaluation of the entire system is mandatory (p. 177).

Cognitive engineering and human-computer interaction are closely tied. As Woods and Roth (1988) note, the "...need for a cognitive engineering occurs because the introduction of computerization radically changes the work environment and the cognitive demands placed on the worker. (p. 3)" The world for which the cognitive engineer designs will be one in which the operator performs her/his task through the window and with the tools of the computer workstation.

Rasmussen and Goodstein argue that the design of the human-computer interaction in complex systems is dependent upon the existence and quality of models of cognitive function in complex environments. They note that, because of its bottom-up, technology-driven orientation, much human-computer interaction research can be carried out empirically, with behavioral sudies. Whereas, cognitive engineering design, because of the high-risks and high-costs associated with complex systems, must rely on predictive models of the human-system interaction. These models, cognitive engineering models, represent higher level cognitive functions, such as problem solving, decision making, and supervision of automation at lower levels of system control.

Cognitive engineering models are essential to the design of human-computer, and human-system, interaction for operators of complex systems. These models guide the design process by representing the *semantics* of the work environment. So in addition to determining the *form* of information (i.e., human-computer interaction), cognitive engineering through its predictive models of human-system interaction addresses the dynamic problem of *what* information to present *when*, and at what *level of detail*.

This chapter presents additional background describing the type of systems of interest--supervisory control systems, a discussion of the characteristics of cognitive engineering models for operators of supervisory control systems, and a specific modeling approach, the operator function model (OFM), used to design human-computer/system interaction in supervisory control systems. The chapter concludes with a set of applications of the operator function model to design human interaction in various applications, e.g., intelligent displays, operator aids, and human interaction in predominantly automated control systems.

Supervisory Control Systems

A supervisory control system is a complex, dynamic, predominantly automated, system. NASA space systems, including mission operations on the ground and in space, are examples of supervisory control systems. The human operator in such systems is a supervisory controller who monitors and fine-tunes computer-based control systems and is responsible for ensuring safe and efficient system operation. In such systems, the potential consequences of human mistakes and errors may be catastrophic. The probability of such events is very low; thus, system design cannot be based on direct empirical evidence from system operation and accidents, but has to be judged by predictive models of human-system interaction. Furthermore, automation tends to transfer the human from psycho-motor tasks that can be formally described to higher level cognitive tasks. In a supervisory control system, cognitive functions include supervision, problem solving, and decision making; conventional behavioral studies are often inadequate to represent these functions. Thus, models of cognitive functions in complex systems are needed to describe human performance and to form the theoretical basis of operator workstation design. Workstation design includes specification of operator displays, controls, decision support aids, and intelligent tutors.

Cognitive function models are essential as they provide the *rationale* for workstation design and the *intelligence* for operator aids.

The next section presents a set of salient quotes that forms a backdrop for operator models in supervisory control systems. A proposed set of basic tenets that underpins research on human interaction with supervisory control systems follows. Next, some background on the disciplines of human-machine systems engineering with respect to related disciplines follows. Given this background, the chapter summarizes the features of the operator function model (OFM) methodology, developed and used at the Georgia Institute of Technology's Center for Human-Machine Systems Research.

Background: Old Wine in New Bottles

Salient Quotes on Models of Operator Function in Supervisory Control Systems:

On the purpose of operator models (Sheridan and Ferrell, 1974, p. 2):

> The question is often raised whether or not it is possible to predict human performance sufficiently well for purposes of engineering design. Isn't human behavior inherently too variable to be quantitatively modeled? The answer depends upon what one wishes to predict. Frequently, the circumstances for which the prediction is desired are ones in which the person is, in large measure, *trying* to act rationally and skillfully in accordance with physical constraints and social and personal norms. People behave in this way when they are operating vehicles, using tools, running machines, or perhaps even when following office procedures. In all such tasks, consistent and predictable behavior is a prerequisite for effective performance and the operator is rewarded for such performance by other persons or by the task outcome itself. The individual, though capable of a tremendously wide range of behavior and sensitive to very subtle inputs, both present and past, finds that it pays in high constrained situations to organize his own behavior so that it is regular and so that it takes into account relevant available information.
>
> The kind of behavior just described is not appropriate in all situations, and hence models of it will be limited in their range of applications. Even then, the measures used to describe performance will not capture its true richness...We have to be content to describe and predict at a much more mundane level. Our frequent use of terms such as *operator* and *performance* instead of *person* or *behavior* is meant to emphasize the engineering context and the relatively narrow range of human experience which it encompasses.

On the context for a human-machine systems model (Baron, 1984)

> (A human-machine systems model)...embodies the idea that to model human performance, one must model the system in which that performance is embedded. Human behavior, either cognitive or psychomotor, is too diverse to model unless it is sufficiently constrained by the situation or environment; however, when these environmental constraints exist, to model behavior adequately, one must include a model for that environment (p. 6).

On the expectation of what models will provide to analysts/designers/engineers (Singleton et al., 1971 quoting R. B. Miller in *The Human Operator in Complex Systems*):

...the first objective...: to describe and illustrate the human factors orientated system approach to design for the benefit of research workers and potential users.

"With respect to the first topic, I still find that many people who should know better seem to expect magic from analytic and descriptive procedures. They expect that formats can be filled in by dunces and lead to inspired insights. This silly notion persists in spite of my many adjurations in print to the contrary.

....I have noticed with increasing trepidation the assumption that an "analytic model" such as an equation of human or man-machine behaviour is somehow the equivalent of a system design or even a system design specification. These may be powerful guiding tools for design, but I believe even the **aspiration** for making an analytic model based on any set of functions and/or components is a "fata morganna" (a mirage, so called because (it is) supposed to be the work of Morgan le Fay, Webster's). On this I may certainly be wrong, or seriously misled by my experience and logic. I am unaware that the issue has been seriously explored. Should I be right, we might ponder the tremendous intellectual and financial resources going into theoretical models (p. xiv, 8 December 1964)."

Human-Machine Systems and Cognitive Engineering

Human-machine systems (HMS) research has its roots in engineering: mechanical, industrial, and aeronautical. Applications all examine the operator interface to complex, real-time, and, often, high risk systems. Examples include aircraft (military and civilian), power plants (often nuclear), space systems (i.e., satellite ground control, manned-space systems such as the shuttle and space station), and, most recently, manufacturing. The initial goal of human-machine systems engineering was to develop robust operator models, models with the same levels of fidelity as the machine models, e.g., aircraft dynamics. The vision was to have a complete human-machine system model/simulation that could accurately predict system behavior and would give quantitative predictions of human behavior as well as quantitative evaluations of proposed system designs. In the sixties and seventies, the crossover model and the optimal control model (Baron, 1984) were good examples of this type of model. It is important to note that these operator models described continuous tracking behavior carried out by the human operator in the control of systems such as fully manual aircraft or driving an automobile. These initial control theory models showed great promise for a comprehensive human-machine system model (Wickens, 1984).

The introduction of relatively inexpensive and extremely powerful digital computers in the late sixties and early seventies changed the nature of control systems, and, as noted earlier, changed the role of the human operator from continuous manual controller to supervisor of multiple computer-controlled subsystems. Thus, the human-machine systems engineering community refocused the behavior that they were trying to model and, since the continuous control theory models were not adequate, changed the modeling methodologies they were attempting to use. Initially, there were many models that used emerging tools from operations research (e.g., queueing theory models of air traffic control and cockpit display sampling (see for example Rouse, 1977; Walden & Rouse, 1978; Chu & Rouse, 1979)). More recently, human-machine systems models use computational structures from artificial intelligence (AI), e.g., expert systems models of operator system control (e.g., Knaeuper and Rouse, 1985). Currently, computational models represented with AI formulations (e.g., case base reasoning, or blackboards) are predominant (e.g., Rubin et al., 1988).

In addition to the change in modeling tools, a portion of the human-machine systems research community changed its goal: rather than pursuing the development of an analytic/computational human operator simulation, (i.e., a quantitative, predictive model), the goal was more conceptual: task-analytic representations that could be used for design and for the 'intelligence' in operator aids. Thus, the goal is no longer to produce a black box human operator simulator , but to produce a useful description of the operator-task-system interactions[1] . Given this type of model, the discipline of human-machine systems engineering meets the disciplines of cognitive engineering and human-computer interaction. The human-machine systems model provides task and operator knowledge which the designer uses to define the supervisory control computer interface to facilitate safe and efficient system control.

Cognitive engineering models are a class of models that make explicit the types of information required to design the semantics of the human-computer interaction for a supervisory control system. Cognitive engineering is an emerging discipline with its foundations in human factors, computer science, cognitive psychology, and artificial intelligence (Rasmussen, 1986; Woods and Roth, 1988). The approach is problem-driven and system-oriented. Cognitive engineering models of the human operator in complex systems explicitly represent the domain of application, task constraints, and the flexibility inherent in human interaction with a complex system. Cognitive engineering models need to be dynamic and reflect the work environment as perceived by the individual, given the current system state and current system goals. Such models must represent at least three properties of the supervisory control system and the human operator supervising the system. In particular the model must represent (1) *what* changes to the system the operator wants to make; (2) *why* the changes should be made, with respect to system goals and current state, and, finally, (3) *how* the needed changes to the system can be made, i.e., the operator activities undertaken to bring about the desired state. In addition, in most supervisory control systems, there is not one, strictly deterministic, way of undertaking a desired task. Rather, there is a range of acceptable ways of accomplishing the same task. Often the specific operator actions are, to some degree, at the discretion of the decision maker. A cognitive engineering model must have structures that represent the task, operator goals and related activities, together with the range of choices available to the operator.

Basic Tenets for Models of the Human Operator in Complex Systems

1. The Need for Models that Represent the Domain of Application

Over the years leading researchers in the field have made a series of statements that provides a foundation for the theories and models of human interaction in the control of real-time, dynamic systems. First, human-machine systems engineering rests on the assumption that research and models of human-machine systems must *explicitly* represent the system itself. Context-free research, e.g., research into the 'best' menu structure, the best number or set of colors for display screens or 'domain general' problem solving, is not typically useful in enhancing operator or system performance in complex systems in which operators are experienced and well motivated. There are two rationales that underlie this assumption. The first is nicely summarized by Baron

[1] HOS, human operator simulation, is a counter example and is typical of the 'traditional' methods and goals of HMS modeling.

(1984) in the quotation above--essentially the model must also include the system in which the human operator behavior occurs. The second is that most research that is purported to be context free in fact has implicit assumptions about the domain of application that restricts its utility and generalizability. In fact, a great deal of the research presented under the rubric of Human-Computer Interaction at both the national Human Factors and CHI meetings assumes that the domain is office automation or programming. Very little of this research is applicable to an environment which is dynamic (i.e., the state changes whether the human operator does something or not) or complex (the cost of errors may be catastrophic), or addresses the needs of operators engaged in a function that is multi-task, multi-objective, and multi-person (Baron, 1984). For example, although the cluttered desk metaphor is ubiquitous, most designers of supervisory control systems know that they will have multiple display monitors (workstations typically contain somewhere between five and eight monitors). Thus, the issue is not so much how to manage a set of overlapping windows on a desktop as much as how to manage seven or eight sets of monitors, with potentially hundreds of display pages, together with data and information that may be needed all the time or only under certain circumstances.

2. *Models Require a Hierarchic Structure*

As systems become more complex, the number of operator-controlled modes and system-operating conditions increases, thus increasing the combinatorial complexity of the control problem. As a result, operators, both individually and as teams, develop ways to organize and reduce the complexity of the system and operator control task. This complexity reduction and system organization may be represented by means of a network hierarchy (Miller, 1985). The network is one way of representing task-specific knowledge, its organization, and the inherent system constraints that reduce apparent system complexity to limits that a human operator, with memory and cognition limitations, can manage in order to perform the supervisory control task. Thus, models of human-machine interaction in complex systems must reflect the hierarchic organization of the control and problem solving processes.

3. *Decision Aiding Must Retain the Human in the Decision Process*

Advances in computer technology and artificial intelligence provide new computational tools that greatly expand the potential to support decision making in the supervisory control of complex work environments (Woods, 1986a). The most frequent use of this technology, however, is often inconsistent with human skills.

> ...the primary design focus is to use computational technology to produce a stand-alone machine expert that offers some form of problem solution...(Thus), the interface design process focuses on features to help the user *accept* the machine solution (Woods, 1986b; p.87).

Woods notes that the primary issue in such systems is user acceptance of the proposed solution and that system designers will go so far as to suggest that the system provide the user with placebo-like interaction in order to facilitate user acceptance of the machine's recommendations. For example, some systems allow the user to report facts considered important to the knowledge-based system, although the system is not designed to use interactively input data.

Woods (1986b) identifies three problems with such systems. First, when the machine gives only its solution to a problem, the decision maker may not have the authority to override machine output *in practice* as well as in theory. Since the only practical options are to accept or reject system output, there is a great danger of what Woods calls the 'responsibility/authority double-bind' in which the user always either rejects or accepts the machine solution. The former discards the enhancements that intelligent decision support may add to overall system effectiveness; the latter abrogates the responsibility and purpose of the human decision maker in the system.

The second problem with such designs is that it is not clear that people are skilled at discriminating correct from incorrect machine solutions. The effectiveness of human decision makers in system control may depend on intimate involvement in the decision process rather than only evaluation of the decision product. Research in other supervisory control domains shows that there is an 'optimum' level of control system automation beyond which a human cannot effectively transition from the role of a relatively passive monitor to an active system controller (Bergeron, 1981).

Woods identifies the potential loss of cognitive skill as the third problem. Humans are retained in systems to compensate for the limitations of automation. A user who depends almost exclusively on the recommendations of the machine expert may be ill-prepared for the occasions when the machine expert fails and his/her skill is essential to safe and effective system operation.

Recent research provides experimental data that demonstrate some of the problems with 'decision making' decision support. In a series of experiments at Georgia Tech, advice-giving systems consistently failed to improve overall system performance (Knaeuper and Morris, 1984; Resnick et al., 1987; Zinser and Henneman, 1988). The primary reason that these systems failed to enhance system performance is that users either did not ask for or did not take the advice. In one instance in which the machine-based system automatically recommended the next operator procedure, a pilot study showed that in order to dispel user animosity, the aid had to be 'toned down' (Knaeuper and Morris, 1984). In the two other studies, although advice was free (i.e., neither system had either an implicit or explicit penalty for requesting advice), subjects rarely asked for advice.

These results raise interesting questions about the efficacy and style of decision support. In all three experiments, the aid explicitly gave advice, provided reminders, and generally gave the impression that it was omniscient at the task. Yet the human-computer interaction and system performance metrics did not show that advice-giving enhanced overall system effectiveness.

There is other research that suggests that decision support may not always be fruitless. Another Georgia Tech experiment used a computer to provide dynamically adapted system status information. Information content and form was based on a domain-specific model of the human-machine interaction that tailored and grouped displayed information based on the current system state and current operator functions (Mitchell, 1987). This aiding system resulted in improved system performance across a variety of measures and did not have any user acceptance problems (Mitchell and Saisi, 1987)[2].

The differences between these sets of experiments provide insight into the more general issue of aiding. The experiment in which decision support had a positive effect used the computer to aid the user's decision making *process*. The model of human-machine interaction was embedded in the workstation and provided system information at various levels of abstraction, with both type and level of abstraction estimated based on a model of operator function and information about current system state. The workstation provided an initial view into the controlled system based on a 'best guess' about the user's needs. Additional information, however, was always available at the user's request and the decision process always remained under the user's control.

---

2 The operator function model (OFM) will be discussed in detail in the sections which follow.

Decision support systems that aid the user in the process of reaching a decision, rather than making or recommending a solution are proposed as an alternative to the typical decision aiding paradigm (Woods, 1986b; Rasmussen and Goodstein, 1987; Mitchell and Saisi, 1987; Rubin et al., 1988; Vicente, 1987). The basic principle that underlies a decision aiding design is that automation and machine intelligence should enhance or extend human decision making capabilities, not replace the decision maker (Woods, 1986b).

In a recent article on decision support in the supervisory control of high-risk industrial systems, Rasmussen and Goodstein (1987) summarize this position succinctly.

> Rather than continuing their efforts to make the preplanning (i.e., automation) of responses and countermeasures more and more complete and thus restrict the operator's own initiative, designers should take advantage of modern information technology to make available to operators their own conceptual model and their processing resources so as to allow the operators to function as their extended arm in coping with the plant. Such an interactive decision-making activity would thus benefit from this simultaneous availability of the design basis, up-to-date knowledge of the plant status, and accumulated operational experience (Rasmussen and Goodstein, 1987; p. 663).

Current research programs attempting to develop electronic or computer-based associates explicitly address the design of decision *aiding* systems. The pilot's associate project is a research effort that addresses the operational issues of decision support in real-time decision making environments (Chambers and Nagel, 1985; Rouse et al., 1987). The intent of this program is to produce a support system architecture that enhances human abilities, overcomes human limitations, and complements individual human preferences (Rouse et al., 1987).

OFMspert[3], a similar effort for a satellite ground control systems, uses a blackboard architecture to infer operator intentions based on a normative model of operator function (Rubin et al., 1988). Although OFMspert has been quite successful at inferring operator intentions for a laboratory task (Jones, 1988; Jones et al., 1989), the next step in the development of an operator's associate, determination of the style and substance of interaction, is very difficult. Given a representation of operator intentions, OFMspert must interact with the user, providing information and/or assistance. Bushman's (1989) research consists of a preliminary implementation and evaluation of such a system. Empirical results show that a team comprised of OFMspert and a human operator controlled the system as effectively as a team comprised of two human operators (Bushman et al., 1993).

4. Decision Support Must Be Based on a Model of Operator Interaction

A detailed, application-specific model of operator interaction is necessary to provide effective decision support. An effective decision support system is one which amplifies the operator's decision making skills and effectiveness; the system is 'flexibly available', depending on the user's needs and preferences, and can both match the user's current problem representation, and give timely, context-dependent advice, explanations, and reminders. This level of sophistication necessitates a model of human interaction with the controlled system that can provide a context for aiding and can interpret operator intentions in the process of carrying out prescribed procedures or in undertaking problem solving in novel situations. Such models must capture the salient

---

[3] OFMspert (operator function model expert system) will be discussed in detail in the sections which follow.

features of the system as they relate to the operator's tasks, structure operator functions in ways that are cognitively compatible with the way that the operator might, in fact, represent the system (i.e., hierarchically), and permit an understanding of specific operator actions, both physical (e.g., a display request or new switch setting) and cognitive (e.g., information selection and integration). Thus, the underlying intelligence about decision support is embedded in a model that is normative and interpretive. The normative portion, given system events, sets up expectations for operator tasks and functions. The interpretative portion 'understands' operator actions in the context of pursuing hypothesized operator functions, subfunctions, etc.

These tenets have significant implications for human-system interface design. They may provide the basis for models of interaction with the system; these models may in turn be used to support principled design of the system's interface.

There are a range of candidate methodologies for such a model. One which has had several successful applications for intelligent displays and on-line operator's associates is described below. The operator function model (OFM) and its computational implementations (OFMspert) are proposed as candidate models.

Operator Function Model (OFM)

Currently, there several candidate cognitive engineering models (Jones and Mitchell, 1987 and 1989). They include the Rasmussen abstraction/aggregation hierarchy and decision ladder (Rasmussen, 1986), the goal-means network (Woods and Hollnagel, 1987), the problem behavior graph (Newell and Simon, 1972), the GOMS model by Card et al. (1983), with specific application to real time (i.e., time constrained) tasks (John et al., 1990) and the operator function model (OFM) (Mitchell, 1987). These models are very similar in high-level goals and general flavor differing primarily in details that are linked to the specific domains in which they evolved and the applications for which they were initially used. For example, the Rasmussen, Woods, and Hollnagel models were initially defined in the area of continuous process control such as nuclear power; whereas, Mitchell's model was developed for discrete systems such as satellite ground control and automated manufacturing system control. In addition, the former models initially were *descriptive*, attempting to characterize actual operator activities, particularly in situations involving human error. The operator function model represents *normative* operator behavior--expected operator activities given current system state.

The operator function model provides a flexible framework for representing operator behavior in the control of a complex, predominantly automated, dynamic systems. The operator function model is a representation of how an operator (or team of operators) might decompose and coordinate system control functions. Mathematically, the operator function model is a heterarchic-hierarchic network of finite-state automata (Figure 1). Network nodes represent operator activities defined as operator functions, subfunctions, tasks, and actions. The network is a hierarchy that represents operator functions as the highest level nodes and decomposes an individual operator function into subfunctions, tasks, and actions. Actions may be manual or cognitive. Manual actions consist of system commands or explicit information display requests. Cognitive actions include information gathering, information processing, and decision making. At each level of the hierarchy there may be a heterarchy, i.e., a collection of activities that, given the corresponding systems events, are undertaken concurrently. The heterarchy accounts for the coordination of the operator activities and the operator's dynamic focus of attention. Network arcs are enabling conditions, system events or the results of operator activities, that initiate or terminate operator activities.

The operator function model has several features designed to represent the inherent flexibility of operator activities in complex systems. The state transition functions are non-deterministic, i.e., given a current state, a set of *possible* next states is identified, no attempt is made to specify the exact next activity, but merely a set of *plausible* activities given the current situation. Another feature is context-sensitivity; subfunctions, tasks, or actions may be performed in support of more than one higher level activity; which one depends on current system state. In Figure 1, action 2 depicts the OFM's context sensitivity; the action may have been undertaken to support either task 1 or 2. This context-sensitivity is an important attribute in applying the OFM to decision support.

The operator function model is a prescriptive model of human performance in supervisory control. Given system enabling events, it defines the functions, subfunctions, tasks and actions to which the operator might reasonably be expected to attend. Used predictively, the operator function model generates expectations of likely operator intentions and actions in the context of current system state. In this way, the operator function model provides the 'intelligence' needed for knowledge-based displays and aids.

Applications of the OFM

An operator function model was developed for supervisory control tasks at NASA's Multisatellite Ground Control Center (MSOCC) at Greenbelt, Maryland (Mitchell, 1987); this model was used to redesign the existing and planned operator interfaces. Figure 2 illustrates a portion of this model. Using its OFM, a new design for display semantics and workstation configuration for the operator of this system was proposed. The conventional workstation (emulating existing and planned NASA designs) consisted of three monitors and over a hundred pages of hardware-oriented status data (e.g., system state (up or failed) and data and error block counts (e.g., 14897/143)). The data were linked to the physical configuration of the system, and reflected a one-sensor-one-display design philosophy (Figure 3). Using the OFM, the workstation was redesigned with two monitors, one supporting the operator tasks of monitoring and fault detection and the other supporting fault compensation (Mitchell and Saisi, 1987). The information displayed on one of these monitors was in the form of icons that reflected *operator* as opposed to hardware function. The operator function model showed that the operator's primary role was to ensure data integrity by monitoring the rate and quality of data flow. As a result, icons that qualitatively displayed flow rate and error block accumulation replaced the raw data and error block counts. Icons were hierarchical consisting of both high-level and low-level icons. High-level icons aggregated and abstracted current system state to facilitate rapid and accurate monitoring and fault detection. Lower level icons decomposed the system into a more physical representation to facilitate fault identification. The second monitor supported fault compensation. The workstation included a set of operator 'help' commands defined by the high level OFM operator functions. Each help command used the knowledge encoded in the OFM to dynamically select information about equipment status, satellite schedules, and system constraints that was likely to be useful to the operator given current system state. Empirical comparison of the model-based versus the conventional workstation/interface showed that the model-based interface significantly enhanced overall system effectiveness over a range of system performance measures (Mitchell and Saisi, 1987).

The OFM was also used to model predominantly automated manufacturing systems in order to systematically identify operator control functions. This model was used for human-system interface design. Figure 4 depicts an operator function model for the supervisory controller of a flexible manufacturing system. The OFM for cell schedule management is given in Figure 4a and the OFM for inventory management is given in 4b. This model was used to design the operator control functions together with the displayed information (Dunkler et al., 1988). Empirical evaluation showed that the supervisory control system which included the operator undertaking the functions defined in the OFM operated significantly better than a fully automated scheduling and control system.

OFMspert (Operator Function Model Expert Systems

A second application of the OFM is as the knowledge base for an operator's associate. Figure 5 depicts the OFMspert (Operator Function Model expert system) architecture (Rubin et al., 1988). OFMspert uses the problem solving capabilities of a blackboard (Nii, 1986) and operator knowledge derived from an OFM to build and maintain a current best hypothesis about operator intentions given current system state. Operator intentions are represented hierarchically on the blackboard as functions, subfunctions, tasks, and actions (Figure 6). OFMspert's abilities to adequately hypothesize operator intentions and understand operator actions have been extensively validated with previously collected subject performance data and more recent verbal protocols (Jones, 1988; Jones et al., 1989). Given a structure to understand, OFMspert was enhanced with system control and user interaction capabilities. Then, operator intentions inferred from OFMspert were used to control the interface. Ally, OFMspert with control functions, was empirically shown to be an effective operator associate (Figure 7). No significant system performance differences were found between teams of two human operators and teams comprised of Ally and a human operator (Bushman, 1989). This research offers strong theoretical and empirical evidence for the possibility and effectiveness of the notion of decision support via a stand-alone knowledge-based system that augments rather than replaces human operator skills. In addition, it demonstrates how such a knowledge-based system may be used to control and enhance the human-system interface.

More recently, the OFM and OFMspert structures have been used to model pilots in civil aviation (Figure 8). An OFMspert for a Boeing 727 crew was built and evaluated (Smith et al., 1990; Verfurth, 1991). Although the model's intent inferencing was accurate, its dependency on data that could not be obtained digitally in real time showed that pilot intent inferencing is infeasible in aircraft with low levels of automation. The project, however, did suggest that an OFMspert application was likely to be much more successful in the next generations of more automated aircraft. A robust OFMspert for the flight deck offers the promise that human-computer interaction of 'smart' knowledge-based systems might be 'human-centered' (Billings, 1991), i.e., characterized as an aid, rather than autonomous system, for pilots on the flight deck.

Recently completed research, used the OFM and OFMspert as structures for the student and task models in an intelligent tutor (Chu, 1991; Chu and Mitchell, 1993). Using OFMspert, the task and student models controlled the student-computer interaction with respect to both display and dialog. Recently evaluated with NASA satellite ground controllers, the tutor was judged to reduce the training for operators from three months to 7 to 15 hours. The system is now a required portion of NASA orientation for both operations and management personnel.

A related project used the OFMspert Action Interpreter (ACTIN) as the basis of real-time cooperative problem solving system (Jones, 1991; Jones and Mitchell, 1993a and 1993b). Empirical evaluation demonstrated the effectiveness of an OFMspert-based aid that assisted the operator in task and information management during time-critical satellite control. As with previous applications, OFMspert was used to guide the design of the interface and, in real time, manage the content of the interface so as to present information and assistance to the operator that was 'intelligent'--aid at the right time, in the right form, and at the correct level of detail.

Conclusion

This paper summarizes operator modeling from the supervisory control perspective and discusses ways in which the operator models may be used to enhance the human-system interface. The paper proposes basic tenets for the design of decision support systems for operators responsible for the safety and effectiveness of complex, dynamic systems. Among other issues, this paper posits the belief that 'generic' human-computer interaction research is of marginal utility and that effective decision support requires application-specific models of operator function and system structure. Furthermore, the paper suggests that decision support must amplify the human decision process rather than replace the human decision maker. Next, the paper attempts to distinguish between human-machine systems engineering models and related models in psychology, computer science, etc. Given this background, the operator function modeling methodology and a range of its application are described.

The operator function model shows much promise as a 'macro' or engineering model of human behavior that can complement 'micro' or cognitive science models. The operator function model in particular, and the wider set of cognitive engineering models in general, provide a way to capture the salient features of the entire system, i.e, exogenous and endogenous events, system hardware, software, and constraints, as well as the goal-directed behavior of the well-trained, well-motivated operator in complex engineering systems. Together with micro models representing the specific details of how and why individual activities are selected and undertaken, human-machine and cognitive engineering models provide powerful tools to understand and design displays, controls, and decision support systems essential for effective operator supervision of complex systems.

It is the thesis of this research that effective human-computer interaction, defined within the context of a supervisory control system, requires a detailed cognitive engineering model in order to identify the displays (i.e., the correct information, in the correct form, at an appropriate level of detail) and controls (i.e.., what to do, when, and how). When considering the design of human-centered automation, an emerging concept in high risk systems, the design and functioning of intelligent displays and aids necessitates a detailed, robust model of the human functions and their relation to the changing system state.

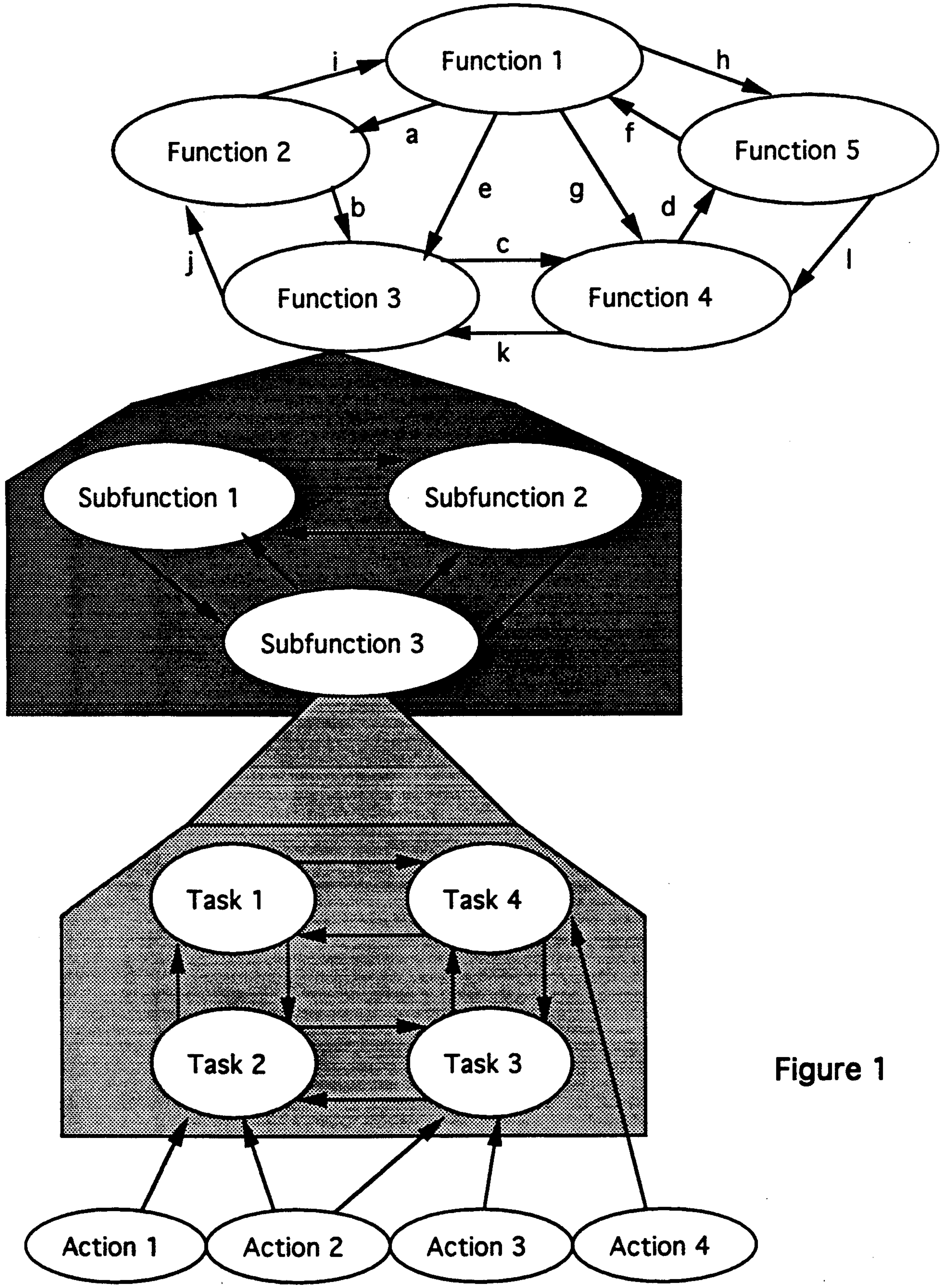


Figure 1

A Generic Operator Function Model

OFM.generic.OFM1

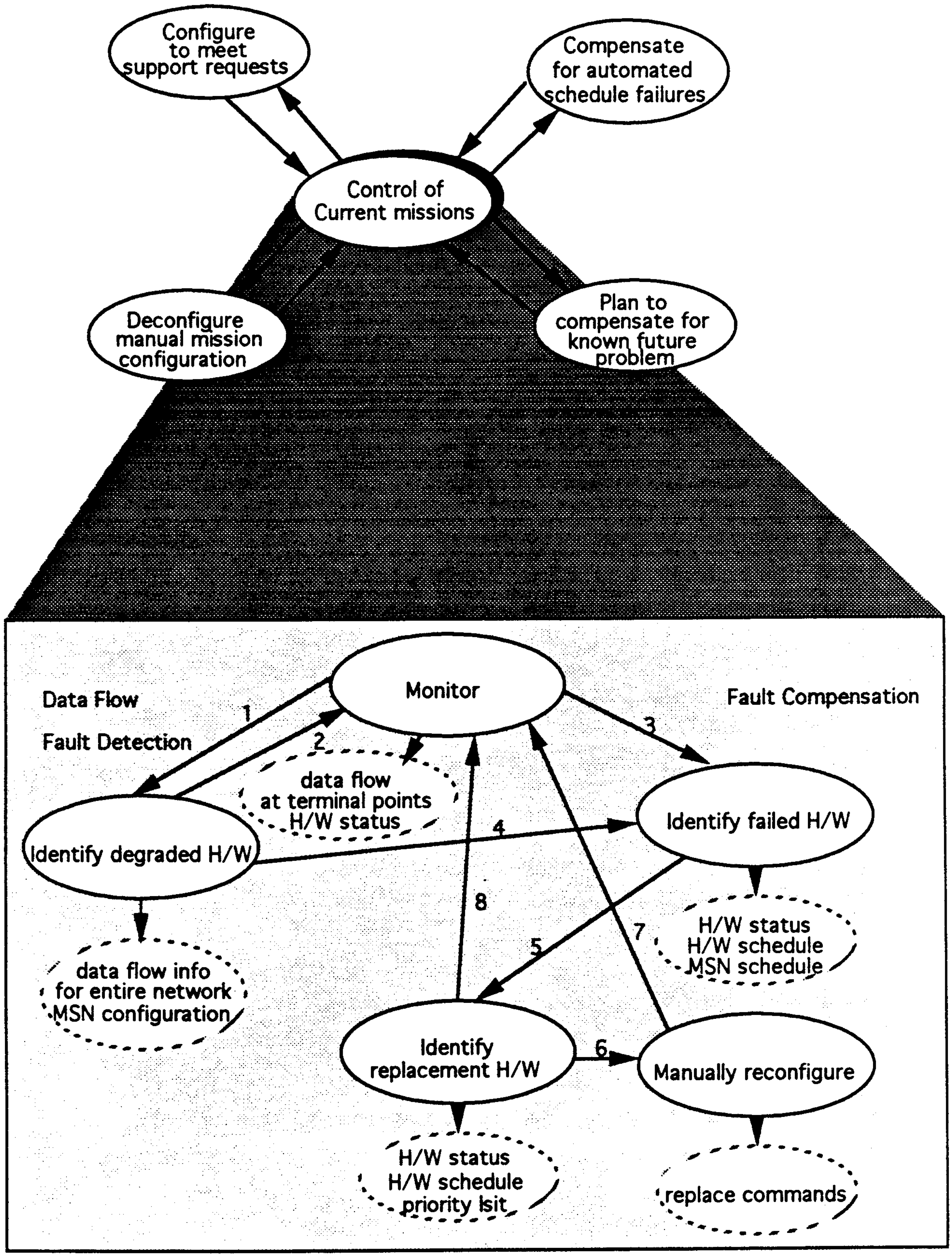


Figure 2. GT-MSOCC OFM

ofm.ccm

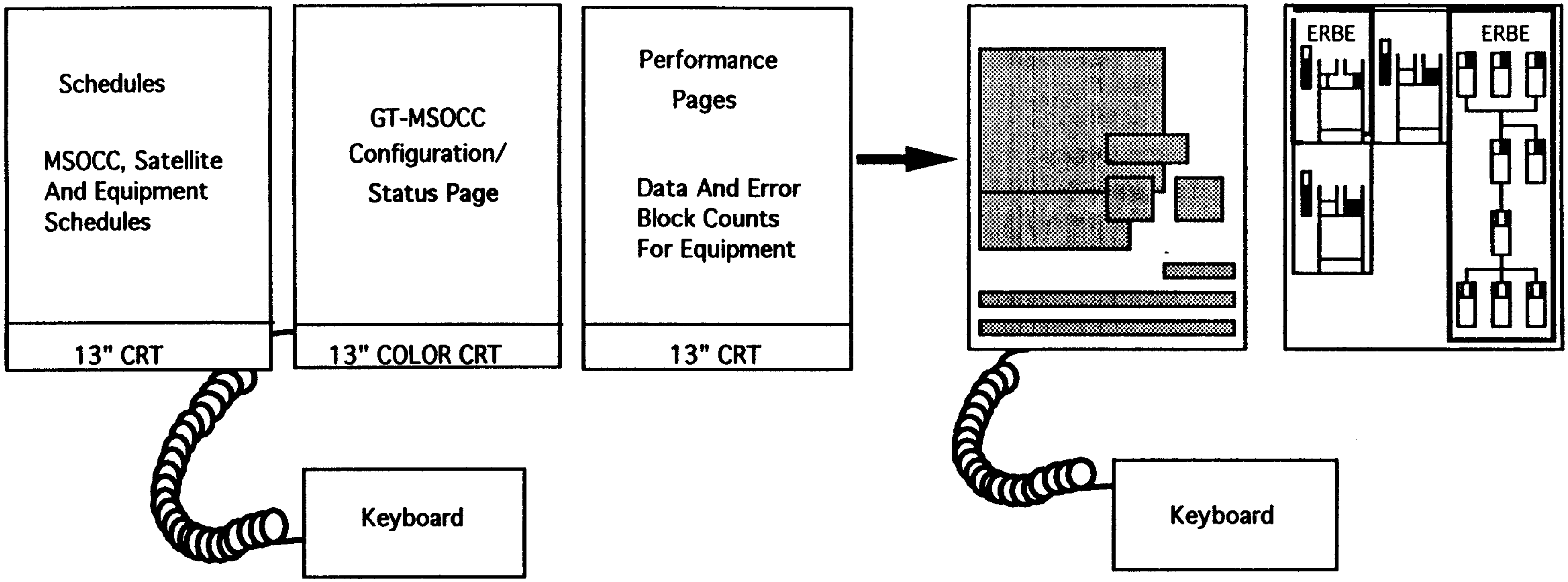
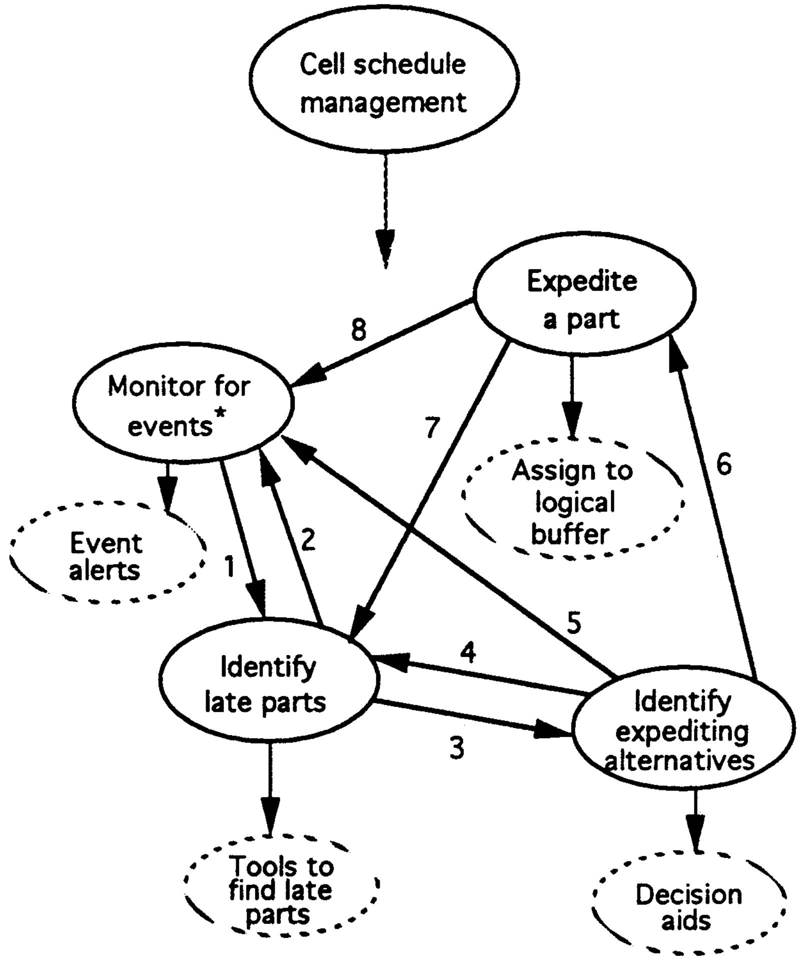


Figure 3. Conventional Vs. Model-Based (OFM) Workstation

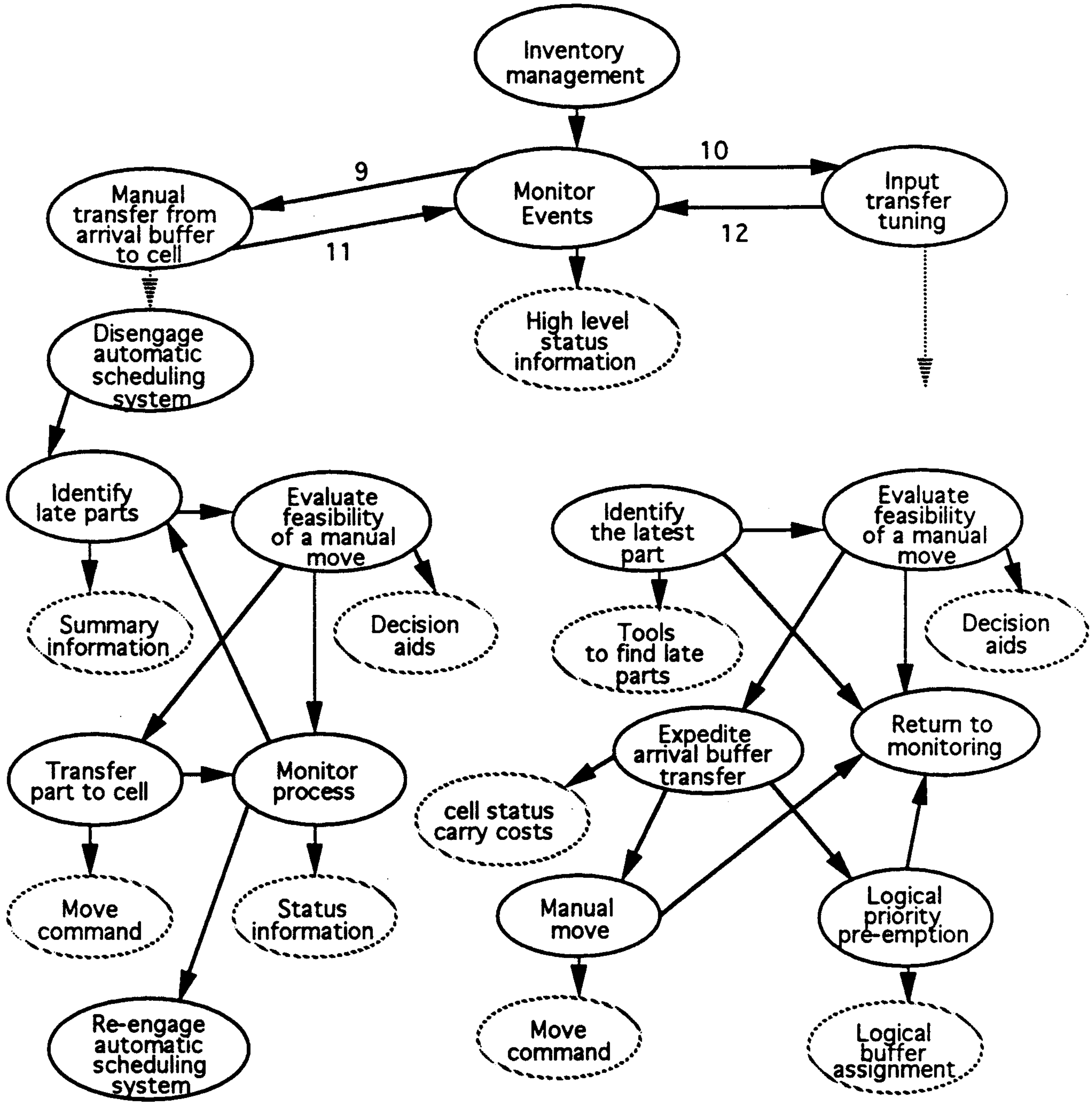
msocc2.workstation.comparison



* Events include machine failures, WIP arrivals, due date changes for parts contained in the cell, and schedule pre-emption by operator.

1. A critical event has occured
2. No late parts currently contained in the cell.
3. One or more late parts found.
4. Decision not to expedite the late part with more late parts to consider.
5. Decision not to expedite the late part with no other late parts to consider.
6. Decide to expedite a late part.
7. Completed expedite action.
8. End of task.

Figure 4a. Operator Function Model of Cell Schedule Management



9. Events that initiate fully manual transfer from arrival buffer to cell include workstation failures or a large number of late parts contained in the arrival buffer.

10. Events that initiate tuning include WIP departures, part completions, and arrival buffer arrivals.

11. When the operator re-engages the automatic scheduling and control system, the manual transfer subfunction is completed.

12. Input transfer tuning is completed when the operator either physically moves or logically prioritizes a part in the arrival buffer or decides that pre-empting the automated schedule is not feasible.

Figure 4b. Operator Function Model of Inventory Management

GT-FMS.OFM.inventory

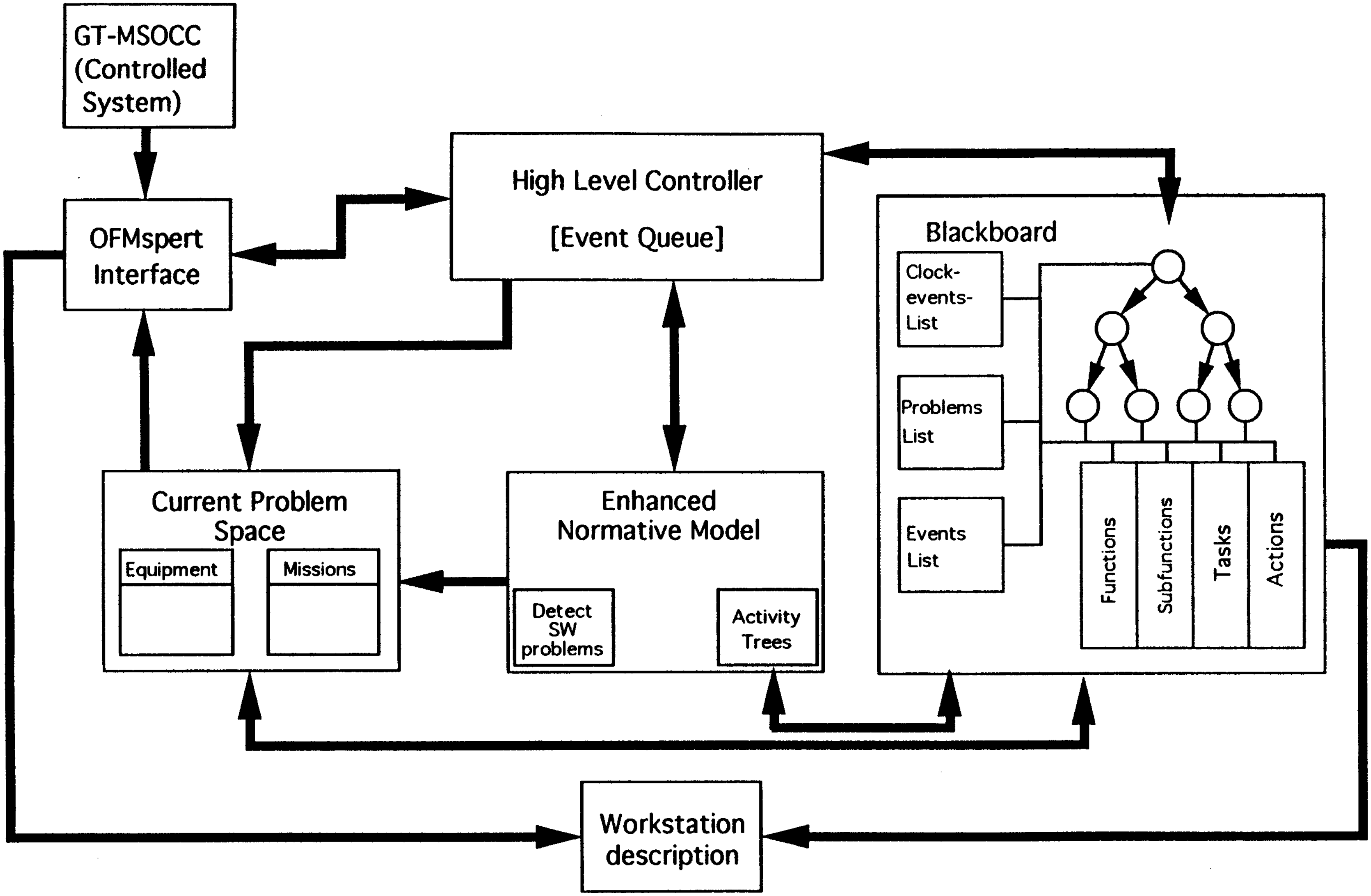


Figure 5. OFMspert Architecture

ofmspert.msocc.architecture

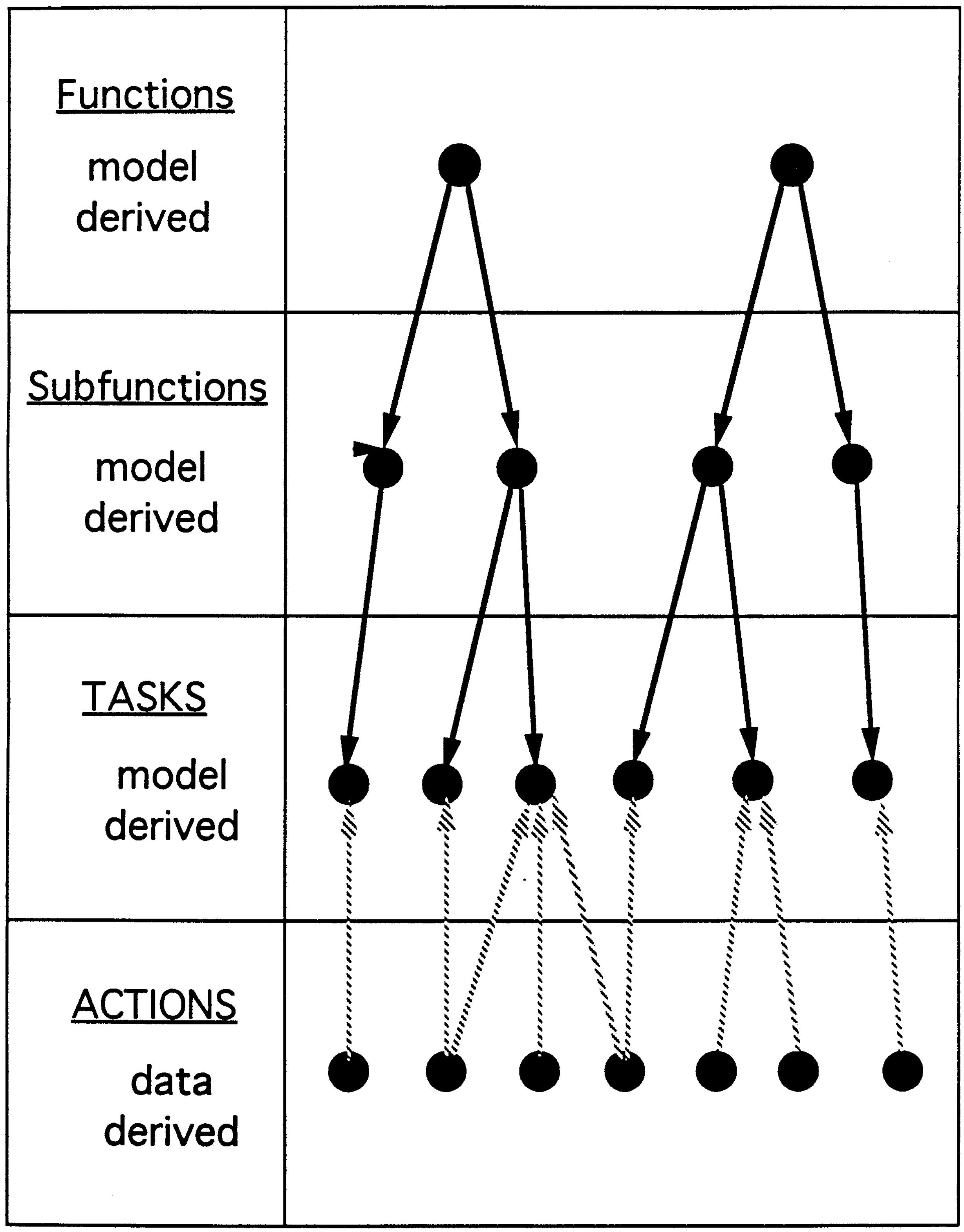


Figure 6. ACTIN (Actions Interpretter)
A Dynamic, Hierachic Representation of
Operator Intentions

ofmspert.actin.2

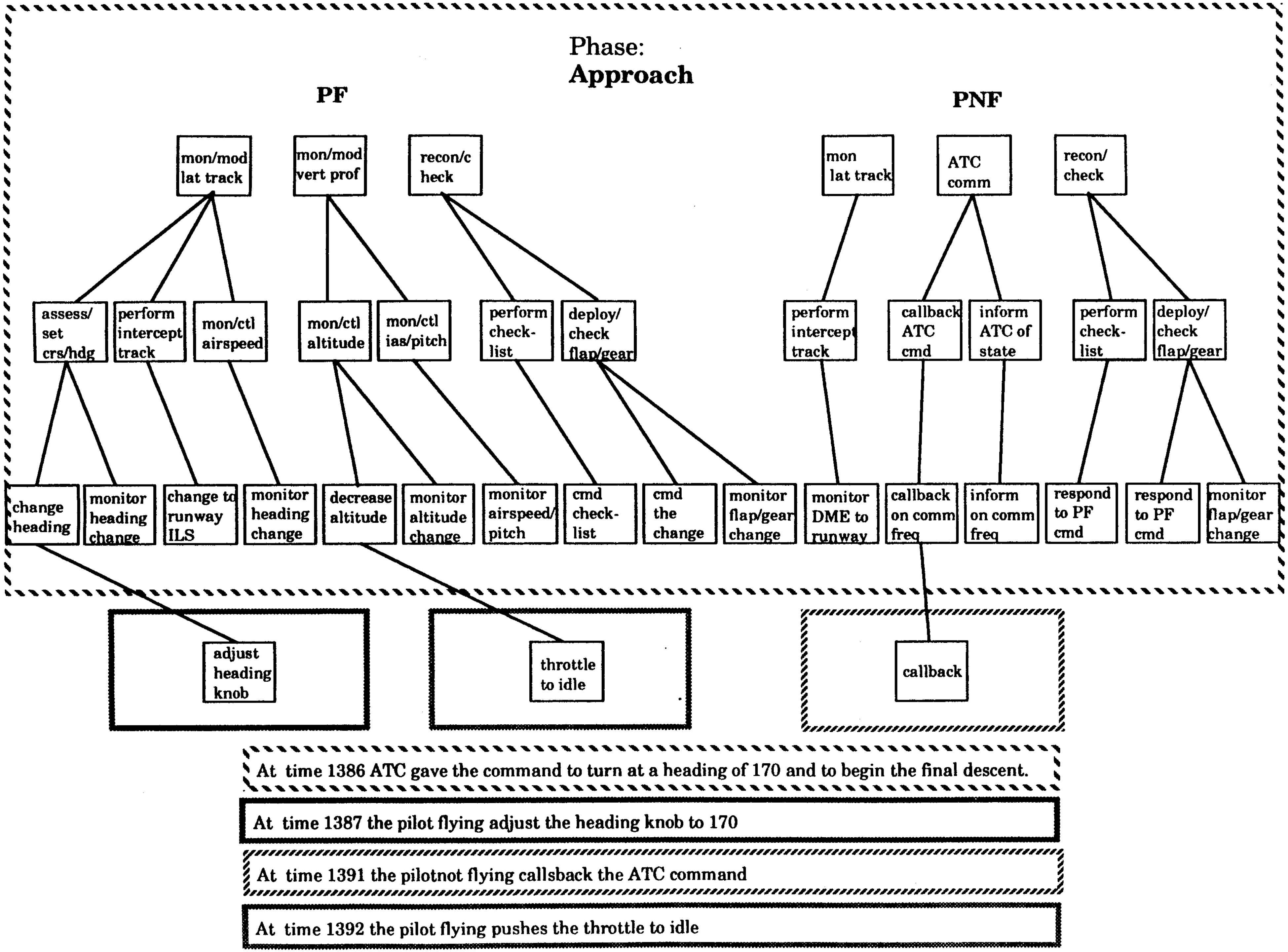


Figure 8. OFMspert of 727

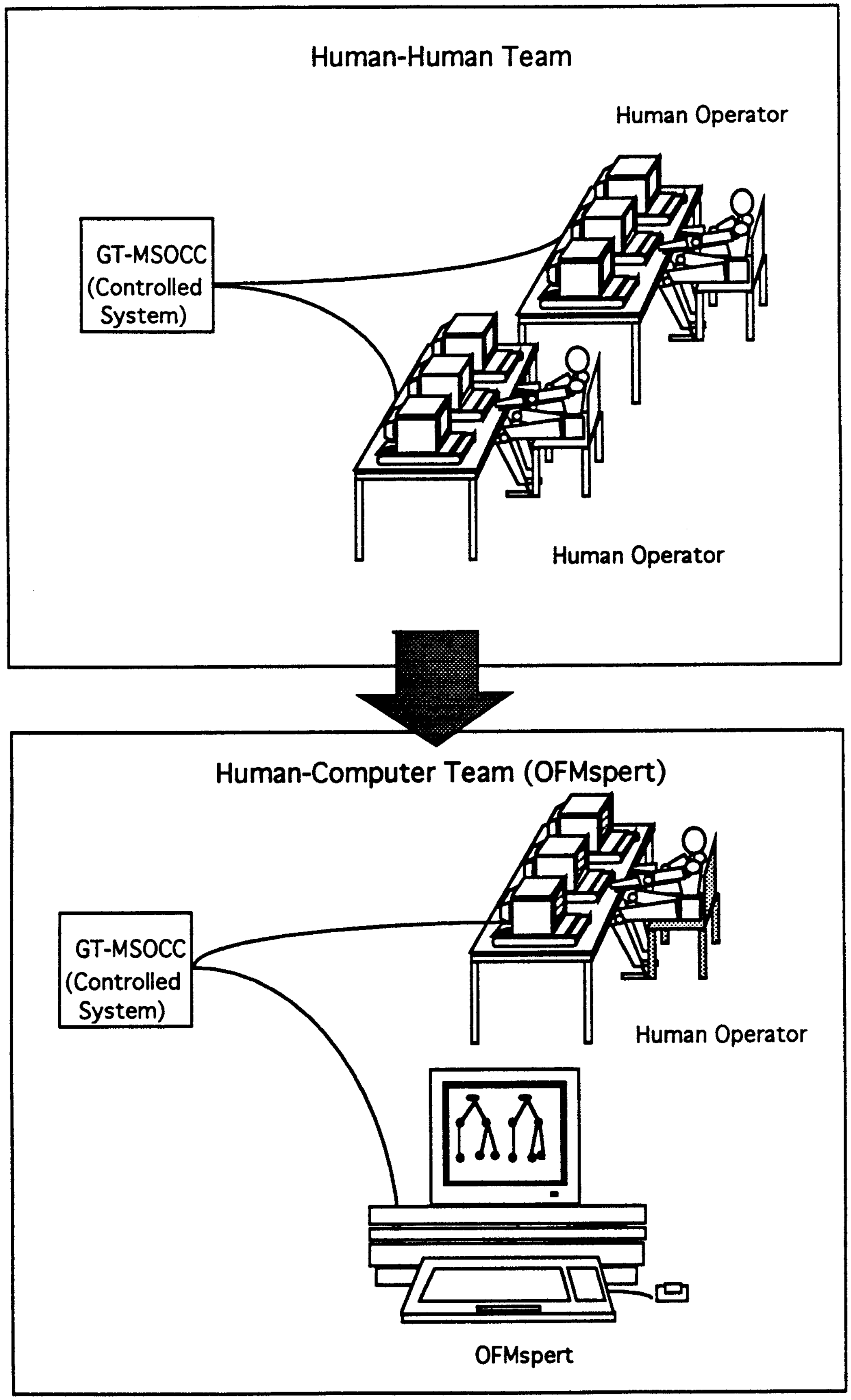


Figure 7. OFMspert: An Operator's Associate