Cognitive Engineering: Issues in User-Centered System Design

EMILIE M. ROTH
Roth Cognitive Engineering

EMILY S. PATTERSON
Cognitive Systems Engineering Laboratory
Institute for Ergonomics
Ohio State University

RANDALL J. MUMAW
Boeing Commercial Airplane Group

To appear in: Roth, E. M., Patterson, E.S. & Mumaw, R. J. Cognitive Engineering: Issues in User-Centered System Design. In J. J. Marciniak (Ed.), *Encyclopedia of Software Engineering*, 2nd Edition. New York: Wiley-Interscience, John Wiley & Sons.

INTRODUCTION

Suppose you were assigned the task of designing software to help automobile mechanics troubleshoot engine malfunctions. How would you approach the problem to ensure that you developed a useful and usable system? Or, suppose you were asked to develop computer-based procedures to replace the paper-based procedures that operators now use to monitor and control a paper-mill process. Or, suppose you were asked to build an information system to support customer service personnel in fielding phone inquiries. How would you know what information to include in the computer database or knowledge base? On what basis would you design the human-computer dialogue structure? How would you know you have developed a usable system that aids users in accomplishing their tasks and leads to improved performance? These questions do not have simple answers. In this article, we introduce some basic concepts from an emerging field called cognitive engineering that is designed to address these types of questions.

Cognitive engineering is an interdisciplinary approach to the development of principles, methods, tools, and techniques to guide the design of computerized systems intended to support human performance (Norman, 1981; Woods and Roth, 1988a; 1988b). In supporting human performance, we are concerned with cognitive functions such as problem solving, judgment, decision making, attention, perception, and memory. The basic unit of analysis and design in cognitive engineering is a cognitive system, composed of human and machine agents in a work domain that is delineated by roles, work and communication norms, artifacts, and procedures. Cognitive engineering draws on the disciplines of cognitive psychology, cognitive science, computer science, human-computer interaction, human factors, and related fields. The goal of cognitive engineering is to develop systems that are easy to learn, are easy to use, and result in improved human-computer system performance.

Experience with the introduction of new technology has shown that increased computerization does not guarantee improved human-machine system performance. Poor use of technology can result in systems that are difficult to learn or use, can create additional workload for system users, or in the extreme, can result in systems that are more likely to lead to catastrophic errors. Personal catastrophes have been caused by unintentional, but unrecoverable, keystrokes that wiped out entire file structures (Norman, 1983). More serious catastrophes, for example, fatal aircraft accidents, have occurred because cockpit automation switched the flight mode without the pilots' knowledge (Lenorovitz, 1992; Sarter, Woods, and Billings, 1997; Sarter and Woods, 2000). Cognitive engineering attempts to prevent these types of design failures by taking explicit consideration of human processing characteristics in the context of the task.

The guiding tenet of cognitive engineering is that consideration of the users and the tasks they will be performing with the aid of a computer system should be central drivers for system design specification (Hollnagel and Woods, 1983; Norman and Draper, 1986). Human-computer interface design is not to be viewed as peripheral to the primary concerns of software engineering. Instead, a user-centered, or practice-centered, system design approach is embraced in which the questions that drive design include the following:

- What are the goals and constraints in the application domain?
- What range of tasks do domain practitioners perform?
- What strategies do they use to perform these tasks today?
- What factors contribute to task complexity?

- What tools can be provided to facilitate the work of domain practitioners and achieve their goals more effectively?

Thus, design is viewed as a means to create a tool that best supports the user in task performance. The benefits of this approach are computer-based tools and aids that are more likely to be successful when deployed because they are solving the appropriate problem.

In this article, we introduce some of the basic concepts of cognitive engineering. We use examples to illustrate some of the common design pitfalls that have led to poor human-computer systems and underscore the importance of adopting a cognitive engineering approach. We then focus on one major topic in cognitive engineering: cognitive task analysis techniques for assessing task demands and identifying human-computer system requirements.

This article is not an attempt to cover all topics in cognitive engineering. Rather, we focus on concerns and techniques that show clear distinctions between cognitive engineering and other approaches, and that are likely to have the most impact on the quality of the human-computer system design. For additional perspectives on cognitive engineering and more in-depth coverage of specific topics, the reader is referred to Card, Moran, and Newell (1983); Helander (1988); Helander, Landauer and Pradhu (1997); Hollnagel, Mancini, and Woods, (1988); Norman and Draper (1986); Norman (1988); Rasmussen et al. (1994); Schneiderman (1987); Dowell and Long (1998); Vicente (1998); Hollnagel (1998); Hoffman and Woods (2000); Woods and Tinapple (1999); Woods, Christoffersen, and Tinapple (1999); and Woods and Roth (1988a, 1988b).

THE COGNITIVE ENGINEERING APPROACH

Rapid advances in computer technology have produced both a need and an opportunity for cognitive engineering. A need for cognitive engineering arises because computerization often radically changes the work environment and the cognitive demands placed on the worker. In too many cases, the introduction of computer technology has placed greater requirements on human cognitive activity. Thus, while there are typically reductions in physical workload, mental workload has increased (Weiner, 1989). One significant change is that in many work environments computerization has shifted the operator's role from manual control of simple systems to supervisory control of highly complex, automated systems. For example, aircraft pilots now supervise automated flight systems via multi-display computerized cockpits; process control operators have shifted from manual control of systems with analog displays and controls to supervisory control of highly automated systems via computerized interfaces; and even coal miners now operate highly sophisticated, automated coal-extraction systems. This shift in the role of operators has placed greater emphasis on their ability to understand the operation of complex systems that accomplish tasks under their supervision, to access and integrate relevant information rapidly, and to monitor and compensate for system failures.

These experiences have created a growing recognition of the need to take into account human information-processing requirements in computer system design. Further, system developers are beginning to realize that systems that provide access to more data or that automate more processes do not necessarily simplify the user's task or guarantee improved performance. Due to the failures of the standard approach to the application of technology, there has been a gradual shift toward a problem-driven approach in which the requirements and bottlenecks in task performance drive the development of computer-based systems that support human operators.

The opportunity for cognitive engineering arises because computational technology offers new kinds and degrees of machine power that greatly expand the potential to facilitate and augment human cognitive activities — activities such as monitoring, situation assessment, plan generation and adaptation, and decision making. Capabilities range from advanced data visualization techniques (Card, Robertson, and Mackinlay, 1991; Card, Mackinlay, and Schneiderman, 1999) to intelligent agents that work cooperatively as part of human-intelligent agent teams (Guerlain, Smith, Obradovich, Rudmann, Strohm and Smith, 1999, Fischer, Lemke, Mastaglio and Morch, 1991; Roth, Malin and Schreckenghost, 1997), to groupware that supports multi-person collaborative work (Patterson, Watts-Perotti and Woods, 1999; Jones, 1995; Stefikand et al., 1988), to head-mounted displays that create virtual-reality environments (Ellis, 1991). The question that system designers continue to face is "How should the capabilities provided by these new technologies be deployed to assist human performance in a particular application?"

A Problem-Driven, Tool-Constrained Approach to Design

Smooth and effective introduction of technology requires taking a problem-driven approach, which entails both an analysis of the goals and activities of users and an understanding of the factors that contribute to good and poor performance, such as large numbers of interruptions by nuisance alarms. This analysis is used to define the kind of solutions that are needed to meet the cognitive demands of the world, such as maintaining an accurate picture of plane trajectories in air traffic control, to help people function more expertly, and to eliminate error-prone activities from the total cognitive system (humans and intelligent machines working jointly). The results of this process can then be deployed in many ways, including new information representation aids, advisory systems, training systems, and new automation.

There are numerous examples of cases in which the introduction of new technology creates unanticipated obstacles to performance because of a failure to take a problem-driven approach. A common pitfall is to provide excessive flexibility, e.g., number of features, configuration options available, because the technology makes it easy to do. Familiar examples come from the consumer electronics field (televisions, videocassette recorders, microwave ovens, and thermostats), where there is a strong incentive to add just one more feature. The result is feature-laden products that come with thick manuals that need to be read to perform the simplest operations. As a result, many consumers are unable to perform the most basic tasks.

Experience with multi-windowing computer systems provides another salient, and more directly relevant, example of the potential pitfalls of excessive flexibility (Woods, 1993; Woods and Watts, 1997). One pitfall is to rely on the flexibility of multi-windowing systems to substitute for detailed analysis of information display requirements. The premise is that users can call up and configure many data views in parallel, tailoring the display to their particular needs at a given point in time. However, this strong reliance on the skills of users to configure displays optimally represents a failure of design. Handing off too much of this task to users creates extra data-management tasks that shift the burden of task analysis and display configuration onto users. The user's limited attention resources are shifted to the interface in order to identify desired data, navigate to the necessary location in the display space, and configure coordinated views. The CRT can become cluttered with windows, causing problems in determining where to focus attention and making it easy for the user to miss new events. These navigation and interface-management activities place high demands on user memory rather than capitalize on the interface as an external representation aid.

Woods and Cook and their colleagues (Cook and Woods, 1996; Cook and co-workers, 1991; Obradovich and Woods, 1996) have been studying the introduction of new computer-based technologies in medical applications. They have documented numerous examples of human-computer interface problems that are symptomatic of a technology-driven approach. The following example illustrates the problem of excessive flexibility.

Cook and Woods (1996) studied the introduction of a new operating-room patient-monitoring system for cardiac anesthesia. A physician-observer charted anesthesiologist activities and monitored use of the new device during 20 heart surgery cases. The new monitoring system integrated what was previously a set of individual devices into a single CRT display with multiple windows and a large set of customization options for display of data. However, this flexibility created the need for new interface management tasks. It forced serial access to highly interrelated data because related sets of data could not be viewed in one window and created a requirement to unclutter displays periodically to avoid obscuring important data channels. The interface failed to support domain tasks adequately. For example, during the operation there is a need to monitor blood pressure and cardiac output in parallel. The new system required that users managed the serial display of these data, an additional task that arose during a critical period in the operation.

As a result, physicians were observed to constrain the display of data into a fixed spatially dedicated default configuration. The full flexibility of the device was not employed. In addition, anesthesiologists had to plan device setup and interaction carefully so that device reconfiguration took place during low-criticality periods, minimizing the need for interaction at high-workload periods.

This medical device example argues for the importance of taking a problem-driven approach in human-computer system design. It is necessary to understand the cognitive demands of the domain and the sources of performance bottlenecks to ensure that the introduction of new technology will reduce mental workload rather than exacerbate it. The following sections provide a framework for this analysis.

The Cognitive System Triad

The basic unit of analysis and design in cognitive engineering is a cognitive system, composed of interacting human and machine agents in a work setting. For example, in software design, the software cannot be evaluated independently of the tasks that the user needs to perform or how users will interact with the software to accomplish the tasks. Woods and Roth (1988a) coined the term "cognitive system triad" to emphasize that three interconnected elements, which are shown in Figure 1, determine the quality of performance on a task: the challenges to be met in an *external world* or domain of interest, the expertise and sources of error of *human and machine agents* who act on the world, and the *artifacts* or external representations through which the agents experience and learn about the world.

Characteristics of the domain contribute to the cognitive demands that are required for competent performance. Most important are those characteristics or factors that increase cognitive complexity. Primary factors include how goals are achieved, the number and complexity of task elements to be controlled or manipulated (e.g., managing air traffic is more complex than maintaining a steady temperature in a refrigerator), interactions and constraints that need to be considered in determining actions (e.g., a small number of independent components are easier to control than are multiple, interdependent components), hazards in the world to be avoided (e.g., avoiding mountains in helicopter flight), the temporal characteristics or dynamics of the domain (whether things change slowly or quickly), coupling between systems (e.g., the loss of an electrical bus affects mechanical systems on the space shuttle), uncertainty, and risk (Woods, 1988). These factors largely determine the cognitive demands and the range of cognitive situations that agents are likely to face in trying to perform domain tasks.

The information-processing characteristics of the agents, especially human agents, are also an important determinant of performance. A fundamental goal of cognitive engineering is to translate knowledge of human information-processing characteristics, such as what is needed to attract attention to unexpected data, into principles and techniques for human-computer interface design. More specific examples are developing principles for graphical display design that capitalize on human perceptual characteristics (Cleveland, 1985; Sanderson, Haskell, and Flach, 1992; Woods, 1984), developing models of human performance that enable explicit consideration of human memory and attention processing constraints in system design (Card, Moran, and Newell, 1983; Kieras and Polson, 1985), and developing principles for the design of error-tolerant systems based on analysis of human error (Brown and Newman, 1985; Norman, 1983; Woods et al., 1994).

The external representation of the domain in supporting artifacts affects performance by making certain information or manipulations of information more accessible at the expense of others. It is a fundamental scientific finding that how a problem is represented affects the cognitive work that is needed to solve the problem, referred to as the representation effect (Zhang and Norman, 1994, Norman, 1993). For example, digital watches make it easy to determine the current time with a high degree of precision. Analog watches, although more difficult to read accurately, provide a better sense of duration (how long something takes). As a result, analog watches are more effective in teaching children the concept of time.

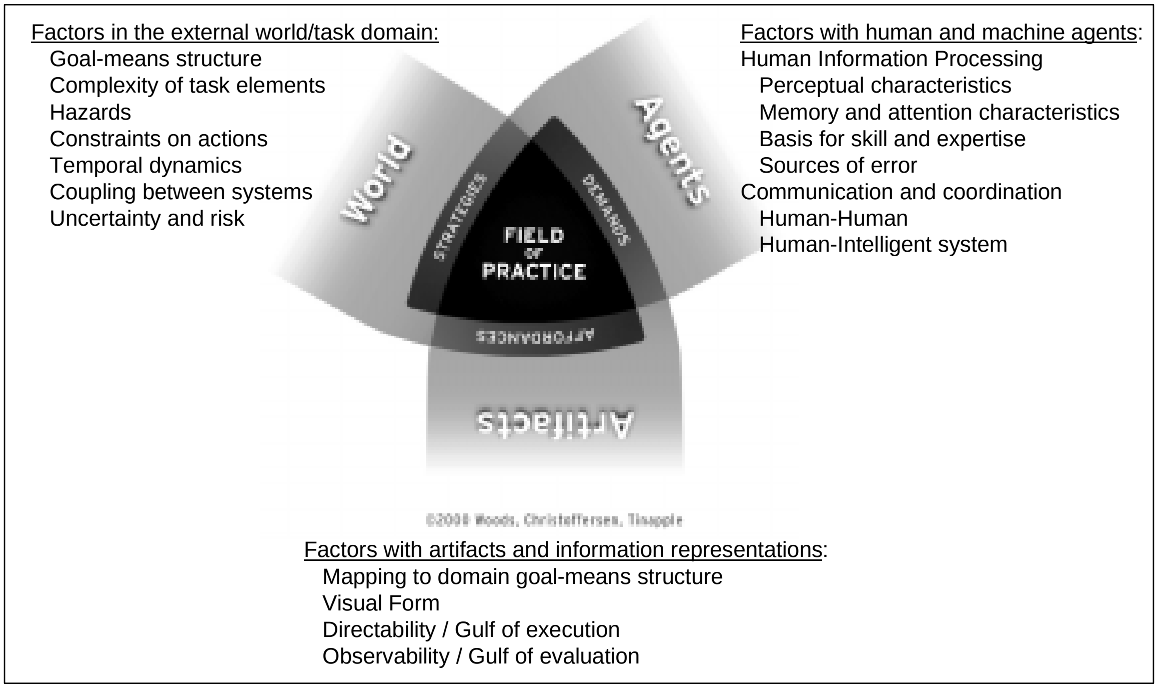

Figure 1. Cognitive system triad.

Ideally, the representation provided by a computer interface should facilitate extraction of information required for task performance. Figure 2 provides an example of a configurable display representation that makes critical domain information visually apparent. This display is used in process control applications to support rapid determination of process state. In this case, there are eight parameters that need to be monitored to assess whether the process is operating normally. The parameters are represented as spokes of an eight-sided geometric shape. When the parameters are all at the desired value, a regular octagon appears. When there are disturbances in parameters, the octagon is distorted in a readily detectable manner. This display capitalizes on human perceptual capabilities to make process state information readily apparent. Compare this approach with a digital readout display of all eight values, where extensive monitoring and mental computation would be required by the operator to extract the same information.

<Figure 2. Configural display that reveals higher-level data patterns>

In order to make software a "team player" that effectively supports humans in performing tasks, *interactions* among the three elements of the cognitive triad also need to be considered. First, there are *strategies* that agents employ to meet the challenges of a domain. For example, if users are printing out spreadsheets and highlighting sections that would change with new inputs, we can design features that electronically highlight areas of a spreadsheet that would be affected when data in a cell is changed. Second, the ways that the challenges of the domain are represented by the artifacts will make it easier or harder to meet them. In other words, the artifacts can make solutions to problems transparent or difficult to see by providing *affordances*, or naturally emerging representations that point to solutions. For example, multiplication is much more difficult with roman numerals than with arabic representations. Third, we can observe how agents interact with current and new artifacts in order to better understand the *demands* of the domain that need to be better met. For example, if air traffic controllers are printing out flight plans for aircraft and cocking the printouts that correspond to planes which will require interventions later, we can learn that a demand of the domain is to remember many actions that need to be done in the future.

Cognitive engineering, when done successfully, is directed at all the elements of the triad and interactions between the elements that are always present in a field of practice. Inattention to any element can produce a human-computer system that fails. The following sections further illustrate the cognitive engineering approach and potential pitfalls through more detailed discussion of the elements of the Cognitive System Triad.

Understanding the Functional Structure of Domain Tasks

In designing and evaluating computer aids one must understand the structure of domain tasks. This includes the range of tasks to be performed by the user, as well as the kinds of complexities that can arise that make the task hard. For example, in considering the functionality of text editors, one must consider the tasks that are performed using a text editor, e.g., transcription, data entry, and composition. It is only by analyzing the range and complexity of tasks to be accomplished that one can understand what computational features need to be provided to support performance. Similarly, the temporal structure of a domain needs to be considered. In event-driven domains, many systems designed to reduce workload have failed because they reduced workload during slow-paced periods at the expense of adding tasks during escalating, fast-paced periods that followed unexpected events (Woods and Patterson, 2001).

The importance of understanding the structure of domain tasks was highlighted in a study of alternate designs for a satellite communications system (Mitchell and Saisi, 1987). Satellite communication systems coordinate the use and monitor the effectiveness of computer and communications equipment shared by various satellites. During real-time communication with the satellite, the system is responsible for ensuring the integrity of the data flowing through system equipment. The operators of the satellite communication system provide a supervisory control function. Their primary role is to monitor the system and compensate for any problems encountered by the automated scheduling and control system. Problems can include hardware failures, software failures, and manual configuration requests for unscheduled spacecraft support.

The original operator workstation design followed traditional approaches to organization of data. Namely, displayed data consisted of the entire set of measurable system variables organized by hardware class or function. The idea was to provide a single layer of displays that contained the most detailed system description that would be needed in any situation. Operator performance with this traditional display system was compared with an alternative workstation display structure that was designed based on a model of operator functions.

As an example, one operator function is to ensure data flow and data quality from each transmitting satellite. To do this, operators are interested in data about the set of hardware that supports a particular satellite transmission. Accessing this information with the original display system involved a great deal of switching between display pages and mental computation since a variety of equipment types supported each satellite. The re-engineered workstation organized the data around the operator functions. The data were presented at several levels of abstraction and aggregation that varied depending on the operator's task. Display pages were tailored to operator functions rather than hardware functions, and graphical icons were used. Mitchell and Saisi compared operator performance with the original display structure and re-engineered workstation design and found large and significant performance improvements with the new interface, including both reductions in errors and productivity improvement.

The study by Mitchell and Saisi illustrates an important feature of cognitive engineering. Cognitive engineering places an emphasis on understanding the domain tasks in terms of the goals of users and the methods available for them to achieve those goals. This function-based approach defines the tasks that will need to be supported and the information and computational processing requirements needed to support operator functions. There are a variety of methods and tools that have been developed for deriving this information. Mitchell (1987) presents a methodology for deriving operator function models that define the dynamic information needs of users given the operator's current task. A number of additional approaches are discussed below in the section on Cognitive Task Analysis.

The Total Cognitive System: Humans and Intelligent Machines

As we said above, an agent in the Cognitive Triad may be either a human or an intelligent machine. In many cases, human and machine (e.g., expert systems, automated systems) will be required to work jointly. Thus, in designing and evaluating a computer-based aid or tool, the "system" of interest is not solely the computer software and hardware, but rather the total complement of human and machine elements that are involved in performance of domain tasks (Hutchins, 1995b; Woods, Roth, and Bennett, 1990). The objective is to maximize the overall performance of this joint human-computer cognitive system, which is not always the same as maximizing the performance of the software when considered in isolation.

A common pitfall is focusing too narrowly on optimizing the performance of the machine element without regard for the role humans are expected to play. This approach can result in suboptimal performance when viewed from the perspective of the total human-computer system (Sorkin and Woods, 1985; Lee and Moray, 1992). This was illustrated by Sorkin and Woods (1985) in the context of design of alarm systems and automated decision systems in general. In most systems, the function of alarms is to alert people to potential malfunctions. The role of the human in this system is to confirm/disconfirm the existence of a malfunction and take appropriate action. Sorkin and Woods showed that in evaluating the effectiveness of alarm systems it is not sufficient to examine the success rate of the alarm system in detecting malfunctions. Rather it was more important to consider the impact of the alarm system on the performance of the human monitor. Alarm systems that have the highest "hit" rate in detecting malfunctions do not necessarily lead to best performance of the total human-computer system. Typically, a system with a higher hit rate also has a higher false alarm rate (alarms falsely indicating a malfunction when none exists). A high false alarm rate leads to distrust, and humans can fail to react when an alarm occurs. Therefore, the alarm system that will optimize the performance of the total human-computer system is the alarm system that is designed from explicit consideration of the role of the human in the system and the effect of alarms (both hits and false alarms) on human performance.

There is growing evidence that a machine agent can alter users' cognitive and decision processes in ways that lead to worse performance than if the person were working unaided.

There have been several empirical studies that compared the performance of individuals performing cognitive tasks with and without the aid of a decision-aid that generated a problem solution for the user. A consistent finding is that the availability of this type of decision-aid alters people's information gathering activities, and narrows the set of hypotheses that they consider, increasing the likelihood that they will miss critical information or fail to generate the correct solution in cases where the intelligent system's recommendation is inaccurate. As an example, a study of airline pilots and dispatchers showed that in a scenario where the computer's recommendation was inaccurate, the generation of a suggestion by the computer early in the person's own problem solving produced a 30% increase in inappropriate plan selection over users of a version of the system that provided no recommendation (Layton, Smith and McCoy, 1994).

In numerous cases, failure to consider the total human-computer system has led to suboptimal designs. Poorly thought out use of automation has produced several salient examples (Bainbridge, 1987; Norman, 1990; Weiner and Curry, 1980). As we stated earlier, the introduction of automation can shift the human's role from active controller to supervisor of automated systems. The effectiveness of the total human-computer system depends on the ability of the human to monitor performance of the automated system, to gauge accurately when the automated system is not performing adequately, and to take over manual control when needed. Too often, little attention is paid to providing human monitors the information they need to carry out this role. Norman (1990) provides several examples from the domain of aviation; one compelling example concerns an engine failure whose symptoms were masked by the functioning of the autopilot. A China Airlines Boeing 747 suffered a slow loss of power from its outer right engine. This malfunction would have caused the plane to yaw to the right, alerting the crew to a problem, were it not for the autopilot. The autopilot, sensing the engine imbalance, compensated and maintained level flight without alerting the crew to a problem. As a result, the engine malfunction was not detected immediately. The autopilot eventually reached its limit, and as it did, the plane rolled and went into a vertical dive. At that point, the crew did not have enough time to determine the cause of the problem or take action.

This example illustrates an opaqueness, a lack of transparency, in the automated system. That is, the autopilot masked an existing problem with the engine, failed to inform the pilot of its progressive compensation to keep the plane level, and failed to indicate that it was approaching its compensation limit. This lack of feedback, or opaqueness of the automated system, remains a concern and potential source of error even in the "glass cockpits" of more modern aircraft.

Sarter and Woods (1992, 1994, 1997, 2000; Sarter, Woods, and Billings, 1997) have conducted an extensive set of studies examining pilot interaction with cockpit automation and have found that pilots have significant difficulty tracking and anticipating the behavior of the automated systems. In one survey, they found that the majority of pilots (67%) reported that they continue to be surprised by the behavior of the flight management system (FMS), which is the automated system that flies the plane based on pilot and designer instructions (Sarter and Woods, 1992). The FMS changes modes in response to pilot commands, but also responds automatically to situational factors. For example, the FMS will shift automatically from climb mode to altitude hold mode after a target altitude is reached. The FMS interface does not provide pilots with adequate feedback on these transitions and on the system's past, present, and future status and behavior. As a result, pilots are often left wondering, "What is the system currently doing Why?" and "What will it do next?" (Weiner, 1989).

These studies by Sarter and Woods found that pilots sometimes missed mode changes that occurred without direct pilot intervention, especially during the high workload descent and approach phases. These difficulties with system and mode awareness reduced the pilots' ability to "stay ahead" of the aircraft. Consequently, when the pace of operations increased (e.g., in crowded terminal control areas), pilots tended to switch to a less automated means of flight control, which could be seen as a failure of design.

Norman (1990) has argued that the failure of automated systems to provide adequate monitoring and feedback information largely results from a lack of appreciation by designers of the need for this information. When everything is functioning well and there is no need for human intervention, then providing information on system state may not be necessary. However, inevitably equipment does fail, and unexpected events do occur (cf. Roth, Bennett, and Woods, 1987). In any complex task or environment one should expect, and design for, the possibility of unanticipated events. Because humans are assigned the task of coping with unanticipated situations, it is critical that they be provided the information and control mechanisms required to recognize and respond to the situation.

A 'total cognitive system' approach that considers the joint role of multiple human and machine agents changes the questions from "how to compute better solutions?" to "how to determine what assistance is useful, and how to situate it and deliver it in the interface?". Roth, Malin, and Schreckenhost (1997) reviewed three alternative metaphors for deployment of machine intelligence in support of human problem-solving and decision-making:

- intelligent systems as "cognitive tools" that can be utilized by practitioners in solving problems and accomplishing tasks;
- intelligent systems as "members of cooperative person-machine systems" that jointly work on problems and share task responsibility;
- intelligent systems as "representational aids" that dynamically structure the presentation of information to make key information perceptually salient.

These approaches are not intended to be viewed as mutually exclusive, but rather complementary metaphors that provide converging insights into the features required for intelligent support of human problem-solving and decision-making tasks. What they fundamentally have in common is a commitment to viewing the user as the central actor, and the intelligent interface as a means for supporting the user's cognitive processes. Among the common themes that emerge are:

- the idea that machine intelligence should be deployed in service of human cognitive activity;
- the importance of understanding the demands of the domain, and identifying what aspects of human cognitive performance require support as input to the design of the system architecture and user displays;
- the importance of providing an external representation that supports human performance and facilitates human and machine cooperative activity by providing a shared frame of reference.

In reviewing the literature on intelligent aids for a variety of tasks and domains, a common path of design evolution seemed to emerge. In many cases systems began as stand-alone solution generators, but over time, evolved into user-directed aids. A concrete example is the SAGE system (Roth and Mattis, 1990) that automatically generates integrated information graphics (e.g., charts,tables, network diagrams). A recent extension has been to embed SAGE in an interactive data exploration system called SageTools that assigns the user a more active role in specifying and creating graphical displays (Roth, Kolojejchick, Mattis, and Goldstein, 1994). Sagetools provides sketch tools that allow users to create designs or partial design specifications;

supports browsing and retrieval of previously created data graphics to use as a starting point for designing new displays; and utilizes knowledge-based design techniques to complete partial design specifications. The progression from SAGE, which is an automated graphics generator, to SageTools, which takes a user-directed approach, mirrors the progression from stand-alone automated problem-solvers to human-centered 'cognitive tools' or 'cooperative team members' that has been seen in other decision-aiding contexts, and provides a concrete example of a 'human-centered' intelligent system.

Creating Effective External Representations

The computer, or system, interface provides a "window" on the domain. It controls how well the domain is understood, and its effectiveness on the ease with which users can extract information about the domain and take actions to perform domain tasks. To create effective representations, the designer must decide what domain properties are to be communicated, choose the specific domain data that are tied to these properties, and map the data into the properties of a visual form so as to make the relevant information directly visible to the viewer. In the Mitchell and Saisi (1987) study, for example, the analysis of operator functions indicated that the information needed for monitoring data flow was primarily qualitative. That is, the operator needed to know that actual flow rate was approximately equal to expected flow rate. A graphical representation was used to make the relationship between actual flow rate and expected flow rate visually apparent.

Norman (1986, 1988) argues eloquently for the need to create a bridge between the user's understanding of task goals and the interface representation that is offered for achieving those goals. Norman employs two concepts called the "gulf of execution" and the "gulf of evaluation." The user starts with goals and intentions related to domain tasks (e.g., "I want to erase this line of text"). These intentions must then be converted to a set of commands or actions that manipulate the computer system to achieve these tasks—perhaps a string of character commands or a direct manipulation-type action. The "gulf of execution" is the degree of conversion required to turn intentions into working commands. When the conversion is extensive (character strings used to erase), the gulf may be too large to span and performance fails. Simpler conversions (strongly analogous actions as in direct-manipulation interfaces) allow for an easy bridging between intents and system actions. After an action is taken, the user is interested in feedback with respect to goal achievement, e.g., "Have I successfully erased the line of text?" or, more generally, "What is the system's state?" This assessment, the "gulf of evaluation," is based on how well the interface represents critical properties of the domain.

The designer can bridge these two gulfs by constructing an interface representation that is organized around meaningful task units. Norman (1986) provides an elegant example based on evolving designs of one everyday artifact: the bathtub faucet. From the user's perspective, the main goals are to control water temperature and rate of flow. Early faucet designs had two faucets and two spouts, one for cold water and one for hot water, which made the mapping between the user's goals and the actions available for achieving those goals complex. To make the water hotter while keeping total rate of flow constant required simultaneous manipulation of both faucets; further, with two spouts, it was difficult to determine if the correct outcome had been reached. Later designs maintained two faucets but mixed the water through a single spout, which simplified the control problem somewhat. However, there remained a link between flow rate and temperature controls. More recent "single control" faucet designs provide a direct one-to-one mapping between the user's goals of controlling temperature and rate of flow. One dimension of movement of the control, usually turning, affects temperature, and a second, orthogonal movement, usually pulling, affects rate of flow.

Vicente and Rasmussen (1990; 1992) describe an approach to interface design that they call *ecological interface design*. This approach captures the idea of providing effective mappings between domain goals and user-interface characteristics. The goal of ecological interface design is to reveal goal-relevant information about the work domain through the computer interface in such a way as to take advantage of the powerful capabilities of human perception and action.

The ecological interface design framework consists of three prescriptive design principles for building interfaces:

1. Develop a goal-means structure for representing domain goals and the means available for achieving them. This representation provides a normative description of the knowledge and information required for successful performance. As such, it provides a basis for specification of the content and structure of the interface. (Techniques for establishing this type of goal-means representation are described in the Cognitive Task Analysis section of this article.)
2. Map the semantics of the domain derived from the goal-means structure onto graphical elements of the interface in order to reveal goal-relevant information in a way that supports direct perception.
3. Support natural interaction by creating direct-manipulation interfaces that allow users to act directly on displayed objects (see also Schneiderman, 1987).

Vicente (1992) illustrated the ecological interface design approach using a thermal-hydraulic process control application. Analysis of the domain revealed the need to provide both functional and physical representations of the thermal-hydraulic process in order to reinforce multiple views of the domain and to support multiple modes of reasoning (Rasmussen, 1986). Vicente compared the effectiveness of two alternative interfaces for the thermal-hydraulic application: one display provided a physical representation of the hydraulic process. For example, the display included a graphical representation of the pumps and valves in the system and their interconnections. This is similar to the traditional approach used in process control applications. The second interface, based on the ecological interface design approach, provided graphical visualization of higher - order functional relations as well. For example, the display included configural graphics that visually depicted mass and energy balances. This allowed operators to readily detect process disturbances and diagnose the source of the process disturbance (e.g., a change in inlet water temperature; a leak). Vicente found clear performance improvements with the ecological interface design that provided graphical visualization of functional relationships.

There is a growing literature on design of graphical interfaces that capitalize on human perceptual properties to reveal domain semantics. Useful resources include Bennett and Flach (1992), Cleveland (1985); Sanderson, Haskell, and Flach (1992), Tufte (1983, 1990), and Woods (1991).

Summary

In this section, we tried to show that, in too many cases, the introduction of new technology has created increases in cognitive workload and unanticipated obstacles to performance because of a failure to take a problem-driven approach. In many of the domains that have seen significant changes in the level of automation over the last few decades, there has been a shift from manual control of simple systems to supervisory control of complex, automated systems. This shift in the role of the human has placed greater emphasis on their ability to understand the operation of complex systems, to access and integrate relevant information rapidly, and to monitor and compensate for system failures.

In cases where it may be reasonable for users to shift to more of a supervisory role, it is critical to avoid the common pitfall of reducing the transparency in the system. Often, too little attention is paid to providing human monitors the information they need to gauge accurately when the automated system is not performing adequately and to take over manual control when needed. The human must remain engaged in the system at some level to be able to respond appropriately when automated components fail. As we have seen, severe consequences can result when too little attention is given to the information requirements of the person on the scene.

In other cases, the technological support has not been developed with an understanding of domain tasks in terms of the goals of users and the methods available for them to achieve those goals. A function-based or goal-decomposition approach defines the tasks that will need to be supported and the information and computational processing requirements needed to support operator functions.

We described several of the pitfalls that can occur when this approach is not adopted. First, when an intelligent system is designed without taking the needs of the human partner into account, the intelligent system can take actions that are unobservable to the human partner that needs to intervene in certain situations. In these cases, the operator may fail to intervene or may be unaware of changes to the mode of operation, which can lead to high-consequence failures in certain situations. Similarly, if the reasoning of the intelligent system is opaque to the human partner, it is difficult for the operator to know when the machine reasoning is correct or incorrect, leaving that person responsible for the machine’s recommendations without a basis for knowing whether or not to trust them.

A second pitfall is the development of a system interface that does not reflect the organization of domain tasks or make critical task information easily accessible. Ideally, the interface structures displays to fit job functions and presents domain information in such a way as to take advantage of the powerful capabilities of human perception and action. The skilled designer creates effective representations by deciding what domain properties are to be communicated, choosing the specific domain data that are tied to these properties, and mapping the data into the properties of a visual form so as to make the relevant information directly visible to the viewer.

Finally, system developers often fail by providing excessive flexibility in system use. This flexibility often shifts the burden of task analysis and display configuration onto users. The user’s limited attention resources are shifted to the interface in order to identify desired data, navigate to the necessary location in the display space, and configure coordinated views. These data- and display-management tasks can interfere with the more critical tasks that the system was developed to support.

ASSESSING COGNITIVE COMPLEXITY WITH COGNITIVE TASK ANALYSIS: TECHNIQUES FOR IDENTIFYING HUMAN-COMPUTER SYSTEM REQUIREMENTS

The previous section used the cognitive system triad to introduce a number of the basic concepts of cognitive engineering. A major emphasis of this section was the importance of performing up-front analyses of the demands of the domain and the requirements for effective joint human-computer system performance. In this section we describe some of the techniques and tools that have been developed for performing these up-front analyses.

What these techniques share in common is an attempt to provide a formal specification of the knowledge and cognitive processing requirements for competent performance of domain tasks. Because of their emphasis on analyzing the cognitive requirements of task performance, they fall under the general heading of "cognitive task analysis methods" (Hoffman and Woods, 2000; Roth and Woods, 1989; Potter, Roth, Woods and Elm, 2000; Schraagen, Chipman and Shalin, 2000). Cognitive task analyses identify the dimensions of task complexity and the demands imposed by the world that any intelligent agent would have to deal with. They provide a formal specification of the range of problems that arise in the domain and the factors that contribute to problem difficulty. This enables an informed decision about the kind of support systems that need to be built, the range of problems that need to be addressed, and the computational tools that need to be adopted.

There are three basic approaches to cognitive task analysis. One approach relies on an analysis of the application domain to uncover the cognitive demands inherent in domain tasks. These are usually based on some form of goal-means decomposition. That is, the domain application is analyzed in terms of the goals or functions that need to be achieved for success and the means that are available to achieve those goals. From this analysis one can derive an assessment of the range and complexity of tasks facing the user. This provides the basis for specification of the content and format of displays and controls. This approach is most effective in domains where the goals and means for achieving those goals are well understood and documented. For example, this technique has been used to design computer-based displays for process control applications where the controlled process and the available control mechanisms were well understood (Easter, 1987; Roth, Lin, Kerch, Kenney and Sugibayashi, in press).

In the second approach, empirical techniques are used to analyze how people actually go about performing the task (either in the actual task environment or in a simulated task environment). This approach enables discovery of the knowledge and strategies that domain practitioners utilize. It is particularly effective in domains where the nature of the task, the methods used by practitioners to accomplish the task, and the factors that complicate task performance, are less well understood. It is particularly well suited for identifying factors, both in terms of knowledge and strategies, that account for differences in performance between expert practitioners and those less skilled.

The third approach is to build a computer model that simulates the cognitive activities required to perform the task. The runnable simulation provides an explicit description of the knowledge and processes that users need to utilize to perform the task. It provides an objective basis for comparing alternative system designs in terms of the amount of knowledge and processing required to operate the system.

Each of these approaches is described in more detail below. For expository purposes we present the three approaches as distinct; however, they should be viewed as complementary rather than as alternative methods. In practice, a mix of analytical and empirical techniques are required for a thorough cognitive task analysis (cf. Roth and Woods, 1988; Roth, Woods and Pople, 1992; Potter, Roth, Woods and Elm, 2000).

Analytical Approaches to Cognitive Task Analysis: Defining the Problem Space

Various analytical methods have been developed to map out the range and complexity of tasks inherent in a domain. These are generally based on some form of functional or goal-means decomposition. One approach focuses on an analysis of the goals and structural constraints inherent in the domain (Rasmussen, 1986; Rasmussen, Pejtersen, and Goldstein, 1994; Vicente, 1999). For example, in the case of engineered systems, such as a process control plant, functional and physical representations are developed that characterize the purposes for which the engineered system has been designed, and the means structurally available for achieving those objectives. These representations provide the basis for defining roles that humans will play in the system, and their information display and control requirements. Jens Rasmmussen is one of the most influential proponents of this approach. He has developed an elaborate theoretical framework for characterizing engineered systems at multiple levels of abstraction (Rasmussen, 1986; Vicente, 1999). Several variants on the Rasmussen approach to goal-means representation have been developed. Examples include those by Woods and Hollnagel (1987), Roth and Woods (1988), and Vicente and Rasmussen (1992). Alternative goal-means representation techniques have been developed on the basis of discrete control models of operator tasks. These techniques focus more directly on representing the functions to be performed by domain practitioners and the methods and control actions available to them for performing these functions. An example of this approach is the Operator Function Model developed by Mitchell and Miller (1986; Mitchell, 1987).

Goal-means decomposition methods focus on building a model of the domain problem space (Newell and Simon, 1972). A goal-means representation is constructed by first structuring domain tasks in terms of the goals to be accomplished, the relationships between goals (e.g., goal-subgoal relations, mutually constraining or conflicting goals), and the means available to achieve the goals (e.g., alternative methods available, preconditions, side effects, preferred order). Appended to the goal-means representation is a description of the information and physical controls that are either actually available (given a particular human-computer interface) or potentially available (given a proposed human-computer interface) to the person for assessing current state, detecting and resolving goal competition, and determining courses of action.

The goal-means representation serves as a framework that is used to describe the kinds of cognitive situations that arise in the course of carrying out domain tasks. It provides a basis for identifying the information and cognitive processing requirements for achieving domain goals under different conditions. The results of the goal-directed analysis define the *requirements for competent performance.* It provides the basis for assessing:

- The kinds of problem-solving situations that arise in the domain
- What people must know and how they must use that knowledge to solve these problems

- What information they need to be provided to support them in identifying goals, selecting among alternative methods to achieve those goals, and carrying out the detailed steps entailed by a given method.

The goal-means decomposition aids in the identification of complex situations that will be difficult to handle. It also provides a basis for developing new representations and decision aids that reduce cognitive burdens and facilitate performance. For specific examples of the application of goal-means decomposition methods to the design of computerized performance aids see Mitchell and Saisi (1987), Potter et al., 2000, Roth et al., in press, Woods and Roth (1988c), and Vicente (1992).

Empirical Approaches to Cognitive Task Analysis: Analyzing Good and Poor Performance

The goal-means analysis approach is best suited to domains where the goals and specific methods for achieving those goals are well specified and documented. This is often the case when designing an interface to a engineered system such as a process control plant, where the goals and methods available for achieving those goals are strongly constrained by the physical characteristics of the underlying process. In those cases the goal-means representation can provide the framework for guiding empirical investigation of how domain practitioners respond to task demands. In other cases the goals, constraints, and methods developed by domain practitioners to meet those constraints may be less well understood. Examples include electronics troubleshooting or air traffic control where the goals and constraints are less well defined and the strategies developed by domain practitioners are less well understood. In those cases, an empirical approach that relies on analysis of how practitioners actually perform domain tasks may be used in place of, or as a complement to, an analytical approach.

The focus is on identifying what practitioners do, both successfully and erroneously, given the demands of the task and the available external resources (displays, procedures, decision-aids). By observing and analyzing highly skilled, as well as less-skilled practitioners as they perform tasks, one can develop a characterization of the requirements for skilled performance. A number of approaches have been developed for studying the performance of domain practitioners ranging from ethnographic or field studies where practitioners are observed in their natural work domain (Hutchins, 1995a; Mumaw, Roth, Vicente and Burns, 2000; Suchman, 1987); to critical incident analyses (Klein, Calderwood, and MacGregor, 1989; Klein, 1998, Militello and Hutton, 1998); to analyses of performance under simulated conditions (Roth and Woods, 1988; Sarter and Woods, 2000), to observation of practitioner performance under highly controlled conditions (Lesgold and co-workers 1986; 1988; Means and co-workers, 1988).

The common element in these approaches is an attempt to develop performance models that specify the knowledge and information-processing strategies used by domain practitioners. This includes analysis of ineffective strategies that lead to poor performance (i.e., a model of error), as well as analysis of the adaptive strategies that have been developed by skilled or "expert" practitioners to cope with task demands (i.e., a model of skill). The analysis also yields a description of the effect, both positive and negative, of external aids and displays on practitioner information processing.

Analysis of errors or performance breakdowns reveals which aspects of the task are the largest contributors to poor performance, and thus, where performance-aiding efforts should be concentrated. Analysis of expert performance defines candidate strategies that may be useful to transfer to less experienced practitioners, for example, through training or online advice.

There are a variety of empirical data collection methods that can be employed to examine practitioner performance. These include informal interviews, retrospective analyses of actual incidents, observation of practitioners as they attempt to perform domain tasks, and formal behavioral studies that attempt to uncover the problem-solving strategies underlying task performance under highly controlled conditions (e.g., Chi, Glaser, and Farr, 1988; Lesgold et al., 1986; 1988; Means et al., 1988). Below we provide two examples of empirical studies that illustrate some of the techniques available. The choice and combination of methods employed depend on the depth of modeling required and the resources available. Detailed discussions of the pragmatics of performing cognitive task analyses and descriptions of particular techniques can be found in Cooke (1994), Roth and Woods (1989), and Jordan and Hendersen (1995).

A study by Lesgold et al. (1986) provides an illustration of empirically based Cognitive Task Analysis. This study investigated a complex troubleshooting task, diagnosing navigation electronics breakdowns in the F-15 tactical fighter plane, in order to develop an intelligent tutoring system (Lajoie and Lesgold, 1989). One important feature of their methodology was an attempt to define the range and complexity of problem-solving tasks that arise in the domain. Through a variety of interview and analysis techniques, they identified a range of difficult problems that could be handled only by highly skilled troubleshooters, such as when the test station that was used to aid diagnosis failed. From this analysis, a classification scheme for problems was developed that guided the selection of problems for study. Identification of sample cases to present to study participants that are representative of the range of complexity inherent in the domain is one of the most difficult but critical aspects of empirically-based cognitive task analyses in order to ensure that the cognitive factors of interest are investigated.

After a set of domain problems was identified, experts were asked to solve these problems using a "thinking aloud" methodology. Lesgold et al. used a second expert as the problem presenter. This expert proposed a fault in the avionics system, and through a previous analysis, had thought through the symptoms that would be present as a result of the fault. The initial symptoms were presented to the "test subject" expert, who was asked to isolate the fault, while "thinking aloud." That expert performed troubleshooting by asking the problem presenter about the results of diagnostic tests that might be executed. Lesgold et al. structured the protocol of the solver by focusing on several types of information:

- Assumptions about the system's state (faults present, etc.)
- Hypotheses about where the fault lies
- Intended action to test that hypothesis and the result expected from the test
- Any inferences or shifts in assumptions due to feedback as tests are performed

This analysis identified the knowledge and information used by the expert to select actions and make decisions at each branch point in the problem space. In addition, Lesgold and co-workers tried to understand the mental representation experts used to guide their problem solving. When experts were compared to less-skilled solvers, there were clear differences in the experts' understanding of the full avionics system. The experts had a deeper understanding of functioning and causal relationships within the system that allowed them to draw inferences about causation and form expectations about useful test points. Cognitive scientists refer to this representation as a mental model (Gentner and Stevens, 1983).

The Lesgold and co-workers (1986) study illustrates a structured problem-solving protocol approach that examined practitioner performance under semi-realistic conditions to uncover the different knowledge representations and troubleshooting strategies employed by expert and less-skilled practitioners. Based on the findings of this cognitive task analysis, a computer-based learning environment called Sherlock was developed that compares favorably with traditional short-term instructional treatments. It is estimated that 20-24 hours of training in this environment will produce learning equivalent to about four years of on-the-job experience (Lajoie and Lesgold, 1989).

Other techniques examine practitioner performance under controlled conditions that do not closely resemble the task as practiced; that is, components of the task are converted to memory or reaction-time tasks to reveal details of practitioner information processing that would not otherwise be available for observation. For example, Means and co-workers (1988) used a memory reconstruction technique to derive information about how air traffic controllers perceive and organize patterns of aircraft on a radar screen. They had air traffic controllers control traffic in simulated exercises. At the end of the exercise, the controllers were asked to indicate on a sector map the position of all aircraft that appeared in the sector when the exercise ended. Controllers were then asked to identify the aircraft that formed meaningful groups. Means and co-workers found an almost perfect correlation between the order in which aircraft were recalled and the groupings controllers identified. When the controllers were asked to label groups, they produced labels that reflected functional relationships between aircraft (crossing over a fix, airport arrivals, etc.). That is, these groups described relationships between aircraft that captured how they functioned in the airspace. Aircraft in high altitude overflight sectors functioned differently than aircraft in a terminal control area. They had different expected behavior and different control needs. Thus, experienced controllers used the characteristics of the sector to identify control needs and group aircraft into meaningful functional sets that shared control needs.

Most important in these results is that the controller groupings were not strongly tied to spatial proximity. That is, two aircraft that were near each other on the radar screen were not necessarily in the same group. The functional relationships were more important. This type of finding has implications for the way aircraft might be coded on a radar screen. For example, if color coding were to be used to group aircraft, coding should be determined by membership in the functional groups as opposed to spatial location or direction of flight.

Cognitive Modeling Approaches to Cognitive Task Analysis: Tracing the Information Processing Flow

A third approach to cognitive task analysis has been to develop cognitive models that represent the knowledge and information processes that are presumed to be required to perform domain tasks (Newell and Simon, 1972; Card, Moran, and Newell, 1983). These cognitive models often take the form of runnable computer simulations of practitioner performance that actually perform the domain tasks (Kieras, 1988; Olson and Olson, 1990). Cognitive simulations have been developed for activities as diverse as text editing tasks (Kieras and Polson, 1985; Bovair, Kieras, and Polson, 1990; Howes and Young, 1991); comprehension of written instructions (Catrambone, 1990); the performance of telephone operators using a new workstation (Gray, John and Atwood, 1993)); NASA space shuttle countdown activities (John, Remington and Steier, 1991); aircraft pilots (Corker, 1993); nuclear power plant operator performance during emergencies (Roth, Woods, and Pople, 1992; Woods and Roth, 1995); communication in emergency call centers (Pavard and Dugdale, 2000) and Submarine Officers (Ehret, Gray, and Kirschenbaum, 2000).

Building a runnable computer program forces the modeler to describe mechanisms in great detail. An examination of the extent of knowledge and processing required by the computer simulation to perform the domain task provides a measure of the cognitive complexity imposed on domain practitioners. In addition, although a simulation is relatively costly to develop, once it is generated, changes to parameters and scenarios can be easily investigated without impacting operations, which is particularly important in environments with high consequences for failure such as aviation.

Cognitive simulation models have been used for many purposes, ranging from improving our scientific understanding of how the human mind works (Rosenbloom and Newell, 1993, Anderson, 1993) to comparing the usability of alternative computer interfaces (Bovair, Kieras, and Polson, 1990, Catrambone, 1990) to designing building layouts to improve communication between personnel in emergency call centers (Pavard and Dugdale, 2000).

A large set of applications have been developed around the GOMS model (Card, Moran and Newell, 1983). (The acronym GOMS stands for goals, operators, methods and selection rules). GOMS models have been successfully applied to predict the usability and learnability of computer interfaces. John and Kieras (1996a,b) describe the current family of GOMS models and the associated techniques for predicting usability, and the types of human-computer interface design projects to which they have been successfully used.

There is currently an increasing interest in developing cognitive simulations that can be used as a test-bed during early design phases to predict the impact of alternative system designs on human performance before they are developed and fielded. Elkind, Card, Hochberg and Huey (1990) and Pew and Mavor (1998) provide a broad overview of cognitive modeling approaches for use in computer-aided engineering. One example is the Man-machine Integration Design and Analysis System (MIDAS), which is an extensive set of computer simulations of human perceptual, cognitive, and motor processes for use in modeling pilot performance in modern aircraft (Corker, 1993). MIDAS is being developed under the sponsorship of NASA-AMES and the U. S. Army Aeroflight Dynamics Directorate. It is intended to provide designers of advanced aircraft with a computational test-bench that they can use to evaluate alternative human-computer configurations early in the design process before any physical prototypes are built.

Cognitive simulations have also been used to explore how designs might impact cooperative work between human agents. For example, Pavard and Dugdale (2000) developed a computer simulation of an emergency call center using an object-based software tool called SWARM. With the simulation, they investigated how different physical layouts of positions in a new emergency call center that integrated firemen and medical personnel for the first time might affect (1) noise levels, (2) the ability to overhear what others are saying, and (3) the level of awareness of others' activities in the center. In their simulation, each agent was modeled with an X,Y spatial coordinate, a role (e.g., firefighter, nurse, physician, or phone receptionist) a dynamic level of "busyness", a dynamic set of incidents of which the agent was aware, and time spent waiting to interrupt others. The environment in which the agents worked was modeled by the level of noise, which was a function of the number of people speaking and the intensity with which they were speaking (based on the urgency of the incident), the position of a loudspeaker and its volume level. Each incident was modeled separately and included the number dialed by the caller (in France, 15 is currently the number for the medical personnel and 18 for the firefighter), the time received, the duration of the call, the type of incident, the time delay until a dispatched vehicle (e.g., fire truck or ambulance) arrives on the scene, and the duration of the report from the dispatched vehicle. Another object was used in the simulation to represent whether or not receptionists were aware of which specific personnel were responsible for taking reports from dispatched vehicles for incidents that had already begun. If not, the receptionist would interrupt other personnel to find out the information, based on the other agents' interruptibility and the ability to have others overhear indirect requests to the room given the noise level.

Based on the findings from this simulation and other cognitive task analysis methods, principled design decisions were made regarding such issues as

- the spatial layout of consoles in the main room,
- which personnel would be responsible for addressing follow-up calls,
- how personnel would identify the personnel responsible for past calls about an incident (e.g., whether the information should be displayed through specialized software, by telephone, or simply by broadcasting in the room),
- whether to use noise-reducing equipment (e.g., whether noise-reducing foam or glass partitions are required and whether glass partitions should be flexibly configurable or in dedicated locations).

Cognitive simulation can provide a complementary approach for cognitive task analysis. Cognitive simulations can aid in cognitive task analyses by revealing the knowledge and reasoning required to respond to the task demands successfully. They provide a tool for understanding the extent to which the environment supports the task confronted by the problem solver. For example, Roth, Woods, and Pople (1992) showed how one cognitive simulation built as a model of some of the cognitive processes involved in dynamic fault management could be used in conjunction with other sources of data to uncover the cognitive demands of a task, to identify where errors are likely to occur, and to point to improvements in the human-computer system. The cognitive simulation provided an objective means for establishing some of the cognitive activities required to handle the emergency event successfully. As such, it provided a tool for validating and extending the cognitive task analysis that was performed based on discussions with instructors, review of procedures, and observations of crews in simulated emergencies.

Summary

One important output of a cognitive task analysis is an accurate assessment of the dimensions of task complexity and the cognitive and coordination demands imposed by the domain that any agent, human or machine, will have to handle. Appreciating the range of complexity that the human-computer system will confront is necessary to identify what computational mechanisms are necessary for this application and to specify the range of cases that the system will have to handle.

While we have presented goal-means decomposition, empirical analysis of practitioner performance, and cognitive modeling as three distinct approaches, they are better viewed as three complementary methods. The analyst needs to determine the techniques that are most appropriate given the resources available, the structure of the domain, and the availability of highly skilled practitioners.

In practice, some degree of up-front analysis of the goal-means hierarchy that characterizes the domain is required to guide empirical studies of practitioner performance. This is necessary to identify a set of sample cases to present to domain practitioners that are representative of the range of complexity inherent in the domain. It is also needed to interpret the behavior of domain practitioners and draw more general conclusions (Roth and Woods, 1988). In turn, empirical analyses of practitioner performance serve to enrich the goal-means representation and clarify the range of practitioner strategies that have been developed to cope with domain demands.

Taken in combination, the analyses reveal the sources of task difficulty and enable identification of options to produce a better match between the cognitive demands of the task and the available resources. They serve to discriminate between difficulties that derive from the inherent structure of the domain (e.g., the fact that there are multiple competing goals that must be balanced) from those that arise because of characteristics of the current interface (e.g., the form of information presentation). They provide the basis for specifying what new information, representations, and advice should be provided.

Similarly, cognitive simulations depend on a detailed specification of knowledge and processing required to perform domain tasks. This is usually derived from a combination of analytical techniques and empirical observation of practitioner performance. In turn the cognitive simulation provides a means for validating and extending the cognitive analysis derived from these other techniques.

CONCLUSIONS

Advances in technology offer the opportunity to greatly facilitate and augment human cognition. Capabilities range from multimedia display capabilities, to intelligent decision-support systems, to groupware for collaborative work, to virtual-reality environments. The question that system designers continue to face is "How should the capabilities provided by these new technologies be usefully deployed to improve human performance in a particular application?"

Experience with the introduction of new technology has shown that increased computer power does not guarantee improved performance. Inappropriate application of technology can result in systems that are difficult to learn and difficult to use. These lessons, in some cases, have had devastating results.

Cognitive engineering attempts to prevent these types of design failures by taking explicit consideration of characteristics of the *users* and the *domain tasks* that they will be performing. In this approach, analysis of human-computer interface requirements is viewed as a central driver of design specification, rather than a peripheral activity to be relegated to the end of the design process.

In designing and evaluating computational aids one must take into account the performance of the joint human-computer cognitive system. This, in turn, is a function of three mutually constraining elements of the cognitive system triad: the demands of the domain, the information-processing characteristics of the practitioner, and the external representation through which the practitioner experiences and interacts with the domain. Cognitive engineering, when done successfully, is directed at all three elements of this triad.

Cognitive task analysis methods can be used to analyze the three elements of the cognitive system triad and their impact on total cognitive system performance. The product of the analysis is a description of the cognitive demands imposed by the domain, the knowledge and problem-solving strategies required to meet those demands, and the reasons for poor performance (e.g., limitations in knowledge, ineffective strategies, inappropriate problem representation) observed in the current environment. This enables an informed decision about the kind of support systems that need to be built, the range of problems that need to be addressed, and the computational tools that need to be adopted.

ACKNOWLEDGMENTS

We would like to thank Kim Vicente and two anonymous reviewers for their insightful comments on a previous draft of this article. We would also like to acknowledge the contributions of David Woods, with whom we collaborated on earlier papers on cognitive engineering, and who continues to define and push the frontiers of this emerging field. David's writings and stimulating discussions have had a significant impact on the ideas presented in this article.

BIBLIOGRAPHY

J.R. Anderson, *Rules of the Mind*. Hillsdale, NJ: Lawrence Erlbaum Associates, Inc., 1993.

L. Bainbridge, "Ironies of Automation," in J. Rasmussen, K Duncan and L. J. Leplat, eds., *New Technology and Human Error,* John Wiley & Sons, Inc., New York, 271-284, 1987.

K.B. Bennett, and J.M. Flach Graphical displays: Implications for divided attention, focused attention, and problem-solving. *Human Factors, 34*. 513-534 (1992).

S. Bovair, D. E. Kieras, and P. G. Polson, "The Acquisition and Performance of Text-Editing Skill: A Cognitive Complexity Analysis", *Human-Computer Interaction 6*, 1-48 (1990).

J. S. Brown and S. Newman, "Issues in Cognitive and Social Ergonomics: From Our House to Bauhaus," *Human-Computer Interaction 1*, 359-391 (1985).

S. K. Card, T. P. Moran, and A. Newell, *The Psychology of Human Computer Interaction*, Lawrence Erlbaum Associates, Hillsdale, N.J., 1983.

S. K. Card, G. G. Robertson, and J. D. Mackinlay, The Information Visualizer: An Information Workspace", *CHI'91 Conference Proceedings, Association for Computing Machinery Special Interest Group on Computer-Human Interaction,* 181-188 (1991).

R. Catrambone, "Specific Versus General Procedures in Instructions," *Human-Computer Interaction 6,* 49-94 (1990).

M. T. H. Chi, R. Glaser, and M. J. Farr, *The Nature of Expertise,* Lawrence Erlbaum Associates, Hillsdale, N.J., 1988.

W. S. Cleveland, *The Elements Of Graphing Data,* Wadsworth, Monterey, California, 1985.

N. J. Cooke Varieties of knowledge elicitation techniques. *International Journal of Human-Computer Studies* (1994), 41, 801 – 849.

R.I. Cook and D.D. Woods. Adapting to new technology in the operating room. *Human Factors, 38(4)*, 593-613 (1996).

R. I. Cook, S. S. Potter, D. D. Woods, and J. S. McDonald, "Evaluating the Human Engineering of Microprocessor Controlled Operating Room Devices," *Journal of Clinical Monitoring 7,* 217-226 (1991).

K. M. Corker "An Architecture and Models for Cognitive Engineering Analysis of Advanced Automation Control Environment", *Proceedings of the American Nuclear Society Topical Meeting on Nuclear Plant Instrumentation, Control and Man-Machine Interface Technologies* (1993).

J. Dowell and J. Long, "Target Paper - Conception of the cognitive engineering design problem," *Ergonomics 41(2)*, Taylor and Francis, 126-139, 1998.

B. Pavard and J. Dugdale, (2000) "The application of a cognitive engineering methodology to agent based simulation in the social sciences." *Proceedings of the workshop on Agent-based Simulation 2000*. Passau. Germany. SCS Publishing.

J. R. Easter, "Engineering Human Factors Into The Westinghouse Advanced Control Room", *Nuclear Engineering International*, 35~8 (1987).

J. Elkind, S. Card, J. Hochberg, and B. Huey, eds., Human *Perfor*mance *Models for Computer Aided Engineering,* Academic Press, New York, 1990.

S. R. Ellis, "Nature and Origins of Virtual Environments: A Bibliographical Essay," Computing *Systems* in Engineering 2, 321-347 (1991).

B.D. Ehret, W.D. Gray, and S.S. Kirschenbaum. Dontending with Complexity: Developing and Using a Scaled World in Applied Cognitive Research, Human Factors, 42, 8 – 23, 2000.

J. Elkind, S. Card, J. Hochberg & B. Huey(Eds.) Human Performance Models for Computer Aided Engineering, New York: Academic Press, 1990.

G. Fischer, A.C. Lemke, T. Mastaglio, and A.I. Morch, "The role of critiquing in cooperative problem solving*." ACM Transactions on Information Systems*, *9, No. 3*, April 1991, 123-151.

D. Gentner and A. Stevens, Mental *Models,* Erlbaum, Hillsdale, N.J. 1983.

W. D. Gray, B. E. John, and M. E. Atwood Project Ernestine: A validation of GOMS for prediction and explanation of real-world task performance. *Human Computer Interaction, 8*, 237-309 (1993).

S. Guerlain, P.J. Smith, J. Obradovich, Heintz, S. Rudmann, P. Strohm, J.W. Smith, J., Svirbely, and L. Sachs, "Interactive Critiquing as a Form of Decision Support: An Empirical Evaluation." *Human Factors 41(1)*, 72-89 (1999).

M. Helander, ea., *Handbook of Human-Computer* Interaction, Elsevier Science Publishers B. V., North-Holland, 1988

M. G. Helander, T. K. Landauer and P. V. Prabhu,, Handbook of Human-Computer Interaction, Second Edition, Elsevier, 1997.

R. R. Hoffman and D. D. Woods, "Studying Cognitive Systems in Context", *Human Factors, 42*, 1-7, 2000.

E. Hollnagel and D. D. Woods, "Cognitive Systems Engineering: New Wine in New Bottles," *International Journal of Man-Machine Studies 18*, 583-600 (1983).

E. Hollnagel, "Commentary on 'Conception of the Engineering Design Problem' by John Dowell and John Long", *Ergonomics 41(2),* Taylor and Francis, 160-162 (1998).

E. Hollnagel, G. Mancini, and D. D. Woods, eds., *Cognitive Engineering in Complex Dynamic Worlds,* Academic Press, New York, 1988.

E. Hutchins, *Cognition in the Wild*. Cambridge, MA: MIT Press, 1995a.

E. Hutchins, "How a cockpit remembers its speed." *Cognitive Science, 19*, 265-288, 1995b.

A. Howes and R. M. Young, "Predicting the Learnability of Task Action Mappings," *CHI'91 Conference Proceedings, Association for Computing Machinery's Special Interest* Group *on Computer-Human Interaction*, 113-124 (1991).

B. E. John and D. E. Kieras (1996a). Using GOMS for User Interface Design and Evaluation: Which Technique? *ACM Transactions on Computer-Human Interaction*, 3 (4), 320- 351.

B. E. John and D. E. Kieras (1996b). The GOMS Family of User Interface Analysis Techniques: Comparison and Contrast. *ACM Transactions on Computer-Human Interaction*, 3 (4), 287-319.

B. E. John, R. W. Remington and D. M. Steier, *An Analysis of Space Shuttle Countdown Activities:* Preliminaries *to a Computational Model of the NASA Test* Director, technical report of the Computer Science Dept., Carnegie Mellon University, CMU-CS-91-138, 1991.

P.M. Jones, "Cooperative work in mission operations: Analysis and implications for computer support." *Computer Supported Cooperative Work, 3,* 103-145 (1995).

B. Jordan and A. Henderson (1995). Interaction Analysis: Foundations and Practice. *The Journal of the Learning Sciences*, 4, 39-103.

D. E. Kieras, Towards a Practical GOMS Model Methodology for User Interface Design," in M. Helander, ea., *Handbook of Human-Computer Interaction,* Elsevier Science Publishers, 135-157, 1988.

D. E. Kieras and P. G. Polson, "An Approach to the Formal Analysis of User Complexity," *International Journal of Man-Machine Studies* 22, 365-394 (1985).

G. A. Klein, R. Calderwood, and D. MacGregor, Critical Decision Method for Eliciting Knowledge," *IEEE Systems, Man, and Cybernetics SMC-19,* 462-472 (1989).

G. A. Klein, *Sources of power: How people make decisions*. Cambridge, MA: MIT Press, 1998.

S. Lajoie & A. Lesgold (1989). Apprenticeship training in the workplace: Computer coached practice environment as a new form of apprenticeship. *Machine-Mediated Learning*, 3, 7-28.

C. Layton, P. J. Smith and E. McCoy (1994). Design of a cooperative problem-solving system for en-route flight planning: An empirical evaluation. *Human Factors*, 36, 94-119.

J. Lee and N. Moray, "Trust, Control Strategies And AllocationOf Function InHuman-Machine Systems," *Ergonomics, 35,* 1243-1270, (1992).

J. M. Lenorovitz, "French Government Seeks A320 Changes Following Air Inter Crash Report," *Aviation Week and Space Technology* 30-31 (March 2, 1992).

A. M. Lesgold, H. Rubinson, P. Feltovich, R. Glaser, D. Klopfer, and Y. Wang, "Expertise in a Complex Skill: Diagnosing Xray Pictures," in M. T. H. Chi, R. Glaser, and M. Farr, eds., *The Nature of Expertise,* Erlbaum, Hillsdale, N.J., 311-342, 1988.

A. Lesgold, S. Lajoie, R. Eastman, G. Eggan, D. Gitomer, R. Glasser, L. Greenberg, D. Logan, M. Magone, A. Weiner, K. Wolf, and L. Yengo, *Cognitive Task Analysis to Enhance Technical Skills Training and Assessment*, University of Pittsburgh, Learning Research and Development Center, Pittsburgh, Pa., 1986.

B. Means, R. Mumaw, C. Roth, M. Schlager, E. McWilliams, E. Gagne V. Rice, D. Rosenthal, and S. Heon, *ATC Training Analysis Study: Design of the Next Generation Air Traffic Controller Training System*, HII Technical Report, HumRRO International, Inc. Alexandria, Va., 1988.

C. M. Mitchell, "GT-MSOCC: A Domain for Research on Human-Computer Interaction and Decision Aiding in Supervisory Control Systems," *IEEE Transactions on Systems, Man and Cybernetics* SMC-17 (4), 553-572 (1987).

C. M. Mitchell and R. A. Miller, "A Discrete Control Model of Operator Function, A Methodology for Information Display Design," *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-16, 343-357 (1986).

C. M. Mitchell and D. Saisi, Use of Model-based Qualitative Icons and Adaptive Windows in Workstations for Supervisory Control Systems, *lEEE Transactions on Systems, Man, and Cybernetics*, SMC-17, (4), 573-593 (1987).

L.G. Militello, and R. J. B. Hutton, Applied Cognitive Task Analysis (ACTA): A practitioner's toolkit for understanding cognitive task demands. *Ergonomics Special Issue: Task Analysis*, vol 41, # 11, 1618-1641, 1998.

R. J. Mumaw, E. M. Roth, K. J. Vicente, K. J. & C. M. Burns, There is more to monitoring a nuclear power plant than meets the eye. *Human Factors*, vol 42, # 1, 36-55, 2000.

A. Newell and H. A. Simon, Human *Problem Solving*, PrenticeHall, Englewood Cliffs, N.J., 1972.

D. A. Norman, *Steps Toward* A Cognitive Engineering, Technical Report, Program in cognitive Science, University of California at San Diego, 1981.

D. A. Norman, "Design Rules Based On Analyses Of Human Error," Communications *of the* ACM 26, 254-258, 1983.

D. A. Norman, "Cognitive Engineering," in D. A. Norman and S. W. Draper, eds., User *Centered System* Design: *New Perspectives* on *Human-Computer* Interaction, Lawrence Erlbaum Associates, Hillsdale, N.J., 31-61, 1986.

D. A. Norman, *The Psychology of Everyday Things*, Basic Books, New York, 1988.

D. A. Norman, *Things that make us smart*. Reading, MA: Addison-Wesley, 1993.

D. A. Norman, "The 'Problem' with Automation: Inappropriate Feedback and Interaction, Not 'Over-Automation'," *Philosophical Transactions of the Royal Society of London B327*, 585-593, 1990.

D. A. Norman and S. W. Draper, User *Centered System* Design: *New Perspectives* On Human-Computer Interaction, Lawrence Erlbaum Associates, Hillsdale, N.J., 1986.

J.H. Obradovich and D.D. Woods. Users as designers: How people cope with poor HCI design in computer-based medical devices. *Human Factors*, *38(4)*, 574-592 (1996).

J. R. Olson and G. M. Olson "The Growth of cognitive Modeling in Human Computer Interaction Since GOMS," Human Com*puter* Interaction 6, 221-266 (1990).

E. S. Patterson, J. Watts-Perotti, and D. D. Woods, "Voice loops as coordination aids in space shuttle mission control." *Computer Supported Cooperative Work: The Journal of Collaborative Computing, 8(4)*, 353-371 (1999).

R. W. Pew and A. S. Mavor (Eds.) Modeling Human and Organizational Behavior. Washington, D. C.: National Academic Press, 1998,

S. S. Potter, E.M. Roth, D.D. Woods, & W. Elm. Bootstrapping Multiple Converging Cognitive Task Analysis Techniques for System Design. In Schraagen, J.M.C., Chipman, S.F., & Shalin, V.L. (Eds.) *Cognitive Task Analysis*. Mahwah, NJ: Lawrence Erlbaum Associates, Inc., 317-340, 2000.

J. Rasmussen, Information Processing, and Human-Machine Interaction: An Approach to Cognitive Engineering, Elsevier Science (North-Holland), New York, 1986.

J. Rasmussen, A. Pejtersen, and L. Goldstein, *Cognitive Systems Engineering*. New York: John Wiley & Sons, Inc., 1994.

L. Rosenbloom and A. Newell, *The Soar Papers: Research on Integrated Intelligence, Volume One*, 1993, MIT Press.

E. M. Roth, L. Lin, K. Kerch, S. J. Kenney, & N. Sugibayashi (in press) Designing a first-of-a kind group view display for team decision making: a case study. In Salas, E. & Klein, G. (Eds) *Linking Expertise and Naturalistic Decision Making*. Mahwah, New Jersey: Lawrence Erlbaum Associates, Inc.

E. M. Roth, J. T. Malin and D. L. Schreckenhost, Paradigms for Intelligent Interface Design, In M. Helander, T. K. Landauer, P. Prabhu (Eds) *Handbook of Human-Computer Interaction*, Elsevier Science B. V., Amsterdam, The Netherlands, 1997. (pp. 1177 – 1201).

E. M. Roth, K. B. Bennett, and D. D. Woods, "Human Interaction With an 'Intelligent' Machine," *International Journal of Man-Machine Studies 27*, 479-525 (1987).

E. M. Roth and D. D. Woods, "Aiding Human Performance: I. Cognitive Analysis," *Le Travail Humain 51 (1),* 39-64 (1988).

E. M. Roth and D. D. Woods, "Cognitive Task Analysis: An Approach To Knowledge Acquisition Of Intelligent System Design," in G. Guida and C. Tasso, eds., *Topics in Expert System Design*, Elsevier Science, New York, 233-264, 1989.

E. M. Roth, D. D. Woods, and H. E. Pople, "Cognitive Simulation As A Tool For Cognitive Tasks Analysis," *Ergonomics 36,* 1163-1198 (1992).

S. F. Roth, J. Kolojejchick, J., Mattis, and J. Goldstein. (1994). Interactive graphic design using automatic presentation knowledge. In *Human Factors in Computing Systems CHI'94 Conference Proceedings* (pp. 112-117). New York, NY: ACM/SIGCHI.

S. F. Roth and J. Mattis (1990). Data characterization for intelligent graphics presentation. In *Human Factors in Computing Systems CHI '90 Conference Proceedings* (pp. 193-200). New York, NY: ACM/SIGCHI.

P. M. Sanderson, I. Haskell, and J. Flach, "The Complex Role Of Perceptual Organization In Visual Display Design Theory," *Ergonomics 35*, 1199-1210 (1992).

N.B. Sarter, and D.D. Woods, "Pilot Interaction with Cockpit Automation: Operational Experiences with the Flight Management System (FMS)." *International Journal of Aviation Psychology, 2(4),* 303-321, (1992).

N.B. Sarter, D.D. Woods, and C.E. Billings, "Automation Surprises." In G. Salvendy (Ed.), *Handbook of Human Factors and Ergonomics* (2nd edition) (pp. 1926-1943). New York, NY: Wiley, 1997.

N.B. Sarter, and D. D. Woods, "Pilot Interaction with Cockpit Automation II: Experimental Study of Pilots' Models and Awareness of the Flight Management System." *International Journal of Aviation Psychology, 4,* 1-28 (1994).

N. B. Sarter and D.D. Woods. "How in the world did we get into that mode?" Mode error and awareness in supervisory control. *Human Factors, 37*: 5-19 (1995).

N.B. Sarter and D.D. Woods, "Teamplay with a Powerful and Independent Agent: A Full-Mission Simulation Study," *Human Factors, 42(3),* 2000, 390-402.

Schraagen, J.M.C., Chipman, S.F., & Shalin, V.L. (Eds.) *Cognitive Task Analysis*. Mahwah, NJ: Lawrence Erlbaum Associates, Inc, 2000.

B. Schneiderman, *Designing the User Interface: Strategies for Effective Human-Computer Interaction,* Addison-Wesley, Reading, Mass., 1987.

R. D. Sorkin and D. D. Woods, "Systems With Human Monitors: A Signal Detection Analysis," *Human-Computer Interaction 1*, 49-75 (1985).

M. Stefik, G. Foster, D. G. Bobrow, K. Kahn, S. Lanning, and L. Suchman, "Beyond the Chalkboard: Computer Support for Collaboration and Problem Solving in Meeting," in I. Grief, ea., *Computer-supported Cooperative Work: A Book of Readings,* Morgan Kaufmann Publishers, Inc., California, 1988.

L. Suchman, *Plans and Situated Actions: The Problem of Human Machine Communications*, Cambridge University Press, Cambridge, England, 1987.

E. R. Tufte, *The Visual Display of Quantitative Data*, Graphics Press, Cheshire, Conn. 1983.

E. R. Tufte, *Envisioning Information*, Graphics Press, Cheshire, Conn., 1990.

K. J. Vicente, *Cognitive Work Analysis: Toward Safe, Productive, and Healthy Computer-based Work*, Lawrence Erlbaum Associates, Mahwah, NJ, 1999.

K. J. Vicente, "An evolutionary perspective on the growth of cognitive engineering: the Riso genotype. *Ergonomics 41(2)*, Taylor and Francis, 156-159, 1998.

K. J. Vicente and J. Rasmussen, "Ecological Interface Design: Theoretical Foundations," *IEEE Transactions on Systems, Man, and Cybernetics* 22, 589-606 (1992).

K J. Vicente, "Memory Recall in a Process Control System: A Measure of Expertise and Display Effectiveness," *Memory & Cognition* 20, 356-373 (1992).

K J. Vicente and J. Rasmussen, The Ecology of Human-Machine Systems II: Mediating 'Direct Perception' in Complex Work Domains," *Ecological Psychology* 2, 207-250 (1990).

E. L. Weiner, *Human Factors of Advanced Technology ('Glass Cockpit) Transport Aircraft*, NASA, Washington, D. C., 1989, Technical Report 117528.

E. L. Weiner and R. E. Curry, "Flight-Deck Automation: Promises And Problems," *Ergonomics 23*, 995-1011 (1980).

D. D. Woods, "Visual Momentum: A Concept To Improve The Cognitive Coupling Of Person And Computer," *International Journal of Man-Machine Studies 21*, 229-244 (1984).

D. D. Woods, "Coping with Complexity: The Psychology of Human Behavior in Complex Systems," in L. P. Goodstein, H. B. Andersen, and S. E. Olsen, eds., *Mental Models, Tasks and Errors*, Taylor and Francis, London, 128-148, 1988.

D.D. Woods, L.J. Johannesen, R.I. Cook, & N.B. Sarter. *Behind Human Error: Cognitive Systems, Computers, and Hindsight*. Dayton, OH: CSERIAC, 1994.

D.D. Woods. The price of flexibility in intelligent interfaces. *Knowledge-Based Systems*, 6:1-8 (1993).

D. D. Woods, The Cognitive Engineering of Problem Representations," in J. Alty and G. Weir, eds., *Human-Computer Interaction and Complex Systems*, Academic Press, London, 169-188, 1991.

D.D. Woods, K. Christoffersen and D. Tinapple. Complementarity and Synchronization as Strategies for Practice-Centered Research and Design. *Plenary Address, 44th Annual Meeting of the Human Factors and Ergonomics Society and International Ergonomic Association*, August 2, 2000. <http://csel.eng.ohio-state.edu/iea2000/>

D. D. Woods and E. Hollnagel, "Mapping Cognitive Demands in Complex Problem-Solving Worlds," *International Journal of Man-Machine Studies* 28, 257-275 (1987).

D.D. Woods and E. S. Patterson. "How Unexpected Events Produce an Escalation of Cognitive and Coordinative Demands." In *Stress, Workload and Fatigue*. P. A. Hancock and P. A. Desmond (eds.) Lawrence Erlbaum, Hillsdale NJ, 2001, pp. 290-304.

D. D. Woods and E. M. Roth, "Cognitive Systems Engineering," in M. Helander, ea., *Handbook of Human-Computer Interaction,* Elsevier Science Publishers B. V., North-Holland, 1988a, pp. 3~3.

D. D. Woods and E. M. Roth, "Cognitive Engineering: Human Problem Solving with Tools," *Human Factors 30,* 415-430 (1988b).

D. D. Woods and E. M. Roth, "Aiding Human Performance: II. From Cognitive Analysis To Support Systems," *Le Travail Humain 51*(2), 139-171 (1988c).

D. D. Woods and E. M. Roth, "Symbolic AI-Based Computer Simulations as Tools for investigating the Dynamics of Joint Cognitive Systems," in P. C. Cacciabue, B. Pavard and J-M Hoc, eds., *Simulation of Cognition in Dynamic Environments,* Hillsdale, NJ: Lawrence Erlbaum Associates,1995.

D. D. Woods, E. M. Roth, and K. B. Bennett, "Explorations in Joint Human-Machine Cognitive Systems," in W. Zachary and S. Robertson, eds., Cognition, *Computing and Cooperation*, Ablex Publishing, Norwood, N.J., 1990, pp. 123-158.

D.D. Woods and D. Tinapple. W3: Watching Human Factors Watch People at Work. Presidential Address, *43rd Annual Meeting of the Human Factors and Ergonomics Society*, September 28, 1999. <http://csel.eng.ohio-state.edu/hf99/>

D. D. Woods and J. C. Watts, “How not to Have to Navigate Through Too Many Displays”, in M. G. Helander, T. K. Landauer, P. V. Prabhu, *Handbook of Human-Computer Interaction*. Second Edition, Elsevier Science B. V., Amsterdam, The Netherlands, 1997, pp 617 – 650.

J. Zhang, and D.A. Norman, "Representations in distributed cognitive tasks." *Cognitive Science, 18*, 87-122, 1994.