CHAPTER 10

User-Centered Systems Engineering Framework

LEE SCOTT EHRHART and ANDREW P. SAGE

10.1 INTRODUCTION

Human systems integration (HSI) is a systems engineering effort employed across the life-cycle process for the engineering of systems to ensure the incorporation of such critical human factors as usability, reliability, manning, training, and safety within the deployed system. Even when regulations require HSI plans[1] as part of the system acquisition process, process reviews within the U.S. Department of Defense (DoD) indicate a failure in making and implementing these plans (cf. DoD, 1994). Many traditional systems engineering efforts result in the generation of rich and precise hardware/software specifications and implementations with only meager representation and accommodation of users and their tasks (Ehrhart, 1994). Crucial information about users and the tasks they must perform is often lost somewhere between initial problem description and final detailed design specification. As a result, technologies introduced to streamline organizational processes and to facilitate other human activities often create new bottlenecks instead. Efforts in reviewing the literature in new product development and associated decisions notice few instances of concern for human issues in product development (Krishnan and Ulrich, 2001).

The recent emphasis on quality management in systems engineering (Sage, 1992, 1995; Sage and Rouse, 1999) reflects growing concern over the high cost of systems that either:

- Fail to adequately address the functional needs of the operational environment or
- Fail to support the users' successful access to that required functionality (Ehrhart, 1994).

The first issue noted is essentially one of system *utility*; the second issue is that of system *usability*. Both have critical implications for task performance and mission success. A

Handbook of Human Systems Integration, Edited by Harold R. Booher.
ISBN 0-471-02053-2

goal of HSI activities is to orchestrate the application and introduction of new technologies to effectively support individual and team performance to meet organizational needs.

Successful implementation of systems for human users requires an understanding of what it means to provide technology in support of purposeful action in organizations. This includes the formulation, analysis, and interpretation of decision-aiding requirements and the engineering of human–computer cooperative systems to address the identified utility and usability requirements. Systems engineering process models and the associated architectures that lead to system development must address these requirements effectively to enable the design and engineering of useful and usable systems for human interaction (Sage, 1987). The needed efforts encompass various aspects of the problem domain and require evolving technological solutions, each with major human interaction and integration facets. Requirements documents are text-based models of the operational need; software and hardware designs are text and graphic models of the solution path proposed. Prototypes are also models, representing the current design of the system being developed. In between are many more models created as part of artifacts such as data structures, drawings, and charts. Structuring, evaluating, and refining these models highlights gaps in the requirements or conceptual design and alerts the systems engineering team responsible for requirements and conceptual design to critical human–machine factors that effect performance.

This chapter presents frameworks for user-centered systems engineering for HSI. We discuss frameworks for definition and development of systems that emphasize methods for creating, structuring, and applying models and processes needed to identify and address HSI issues across all phases of the development life cycle. Our hope is that this chapter will enable those responsible for HSI to address the wide scope issues that affect systems integration issues affecting humans, technologies, and organizations. Thus, we provide approaches that will enable determination of the value and impact of effective and ineffective user interfaces on systems integration. We address the diversity of users and tasks and their impact on the design of interfaces for HSI. We discuss different system development life cycles, including those particularly applicable to HSI, and show how HSI issues can be incorporated into systems engineering process life cycles. The references provide detailed information concerning sources available on these subjects.

10.1.1 HSI Players and Interactions

There are many stakeholders involved in HSI issues. Figure 10.1 illustrates five of these stakeholder groups and their roles. HSI methods and processes need to engage all the development stakeholders: operational end users of the system, as well as the organizations and enterprises for whom the system is to be engineered. These stakeholders also include those in systems engineering and management, who are responsible for technical direction and communications relative to the process of engineering the system, and the detailed implementation specialists responsible for detailed design production. Our concern is primarily with the first three groups. Their support needs are identified in Table 10.1.

The management, cognitive, and behavioral sciences include many advocates for holistic approaches to understanding the multiple facets of human–machine collaboration in organizations. These crosscut disciplinary interests and organizational functions may be called "transdisciplinary" endeavors (Somerville and Rapport, 2000). For example, enterprise management interests drive process modeling and improvement efforts for software process improvement (Humphrey, 1989), process reengineering (Hammer and Champy, 1993; Hammer and Stanton, 1995; Yu et al.; 1996, Sage 1995, 1999); and

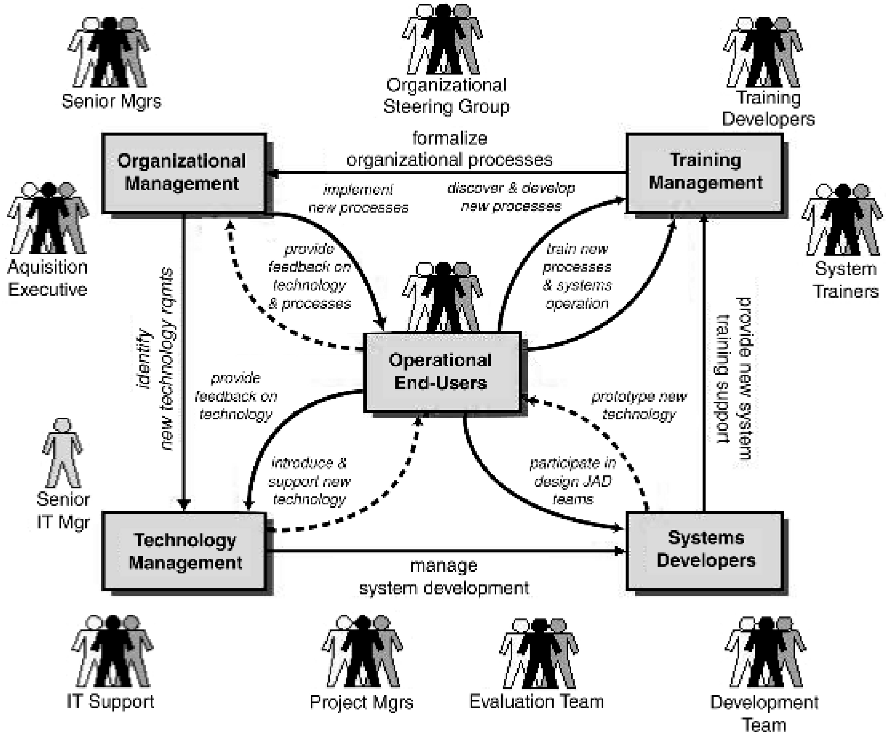

Figure 10.1 Stakeholder roles in human systems integration.

TABLE 10.1 HSI Application to Stakeholder Issues

Group	HSI Support Needs
Operational end users of deployed system	• Focusing on critical factors in decision tasks to make better decisions faster • Understanding and managing information flows and cognitive workload • Facilitating distributed collaboration and cooperative problem solving across multiple users and systems
Organizations and enterprises who acquire the system	• Identifying and expressing organizational requirements • Co-evolving organizational processes with information technology introduction • Synchronizing information operations across functional boundaries
Systems engineering and management for development effort	• Identifying and representing the human information processing and decision support requirements • Identifying and addressing potential sources of error in the decision system • Relating operational needs to system concepts for effective human–computer cooperative problem solving and decision making • Identifying and managing development risks in evolutionary development of complex decision systems

information technology enabled change (Manzoni and Angehrn, 1998). Training and education in organizations is supported by research on situated learning (Suchman and Trigg, 1991), action research (Checkland and Holwell, 1998), and learning organizations and knowledge management (Choo, 1998; Senge, 1990; Senge et al., 1994, 1999). In addition, the cognitive and behavior sciences have contributed user-centered design (Norman and Draper, 1986), decision-centered design (Andriole and Adelman, 1995; Ehrhart and Aiken, 1991; Woods and Roth, 1988), collaboration support (Olson and Olson, 1991), participatory design (Greenbaum and Kyng, 1991), and a broad range of approaches to enhance human information processing in systems and organizations (Sage, 1990). These approaches model humans and technology support as "organic" to information processing, knowledge-creating, and decision-making processes within organizations. Handbooks of human factors and ergonomics (Salvendy, 2001) and systems engineering and management (Sage and Rouse, 1999) generally address these issues. Although systems engineering and systems management necessarily cross boundaries to connect these disciplines, the state-of-practice is often still multidisciplinary, rather than interdisciplinary and transdisciplinary.

10.1.2 Cognitive Systems Engineering

An extremely important interface for HSI and systems engineering for decision systems is enabled through cognitive systems engineering (CSE). Cognitive systems engineering is often spoken of as the practice of the engineering user-centered, decision-focused, information-technology-based systems. The CSE concept provides approaches to the engineering of systems that have major human–machine cooperative problem solving and organizational decision-making requirements. There are three major imperatives:

- Model organizations as *decision systems* to better understand their aiding and training requirements
- Focus on the operational end users—their processes, organization, environment, technology support requirements, and training
- Drive the organizational decision system design to permit the co-evolution of organizational structure, advanced information technology, user and team tasks/processes, and the training design to ensure the successful integration of technology

The CSE approach synthesizes tools and methods across multiple disciplines—including artificial intelligence, cognitive science/psychology, sociology and organization science, systems engineering, and operations research—to provide both the scientific base and applied technologies necessary to support research and development. For both the designer and manager, incorporating CSE activities into the development process assures a better match to operational needs by capturing a more robust set of functional and nonfunctional requirements. This understanding supports informed decision making when design tradeoffs must be made during development life cycle.

10.1.3 Systems Engineering Life Cycle

The traditional systems engineering process is comprised of an iterative, multiphase process providing essential guidance in engineering effective systems. The essential phases

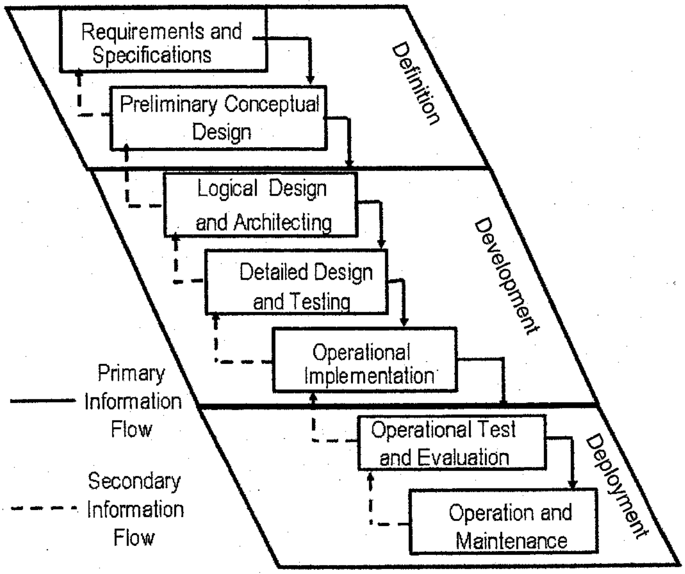

Figure 10.2 Typical structure of a systems engineering life cycle.

in a systems engineering life cycle involve definition, development, and deployment as suggested in Figure 10.2 and may be described in terms of seven constituent phases as follows.

System Definition

1. Requirements and Specifications The first part of a systems engineering effort results in the identification of user requirements and the translation of these into technological specifications for a product, process, or system. The goal of this phase is the identification of client and stakeholder needs, activities, and objectives for the functionally operational system. This means that information is a necessary ingredient and results in the mandate to obtain, from the client for a systems engineering effort, a set of needs and requirements for the product, process, or system that is to result from the effort. This information requirement serves as the input to the rest of the systems engineering process. This phase results in the identification and description of preliminary conceptual design considerations for the next phase. It is necessary to translate operational deployment needs into requirements specifications so that these needs may be addressed by the system design and development efforts. Thus, information requirements specifications are affected by, and affect each of the other design and development phases of, the systems engineering life cycle.

2. Preliminary Conceptual Design and High-Level System Architecting The primary goal of this phase is to develop several concepts that might work and are responsive to the specifications identified in the previous phase of the life cycle. The preliminary conceptual design selected must be one that is responsive to user requirements for the system and associated technical specifications. Rapid prototyping of the conceptual design is clearly desirable for many applications as one way of achieving an appropriate conceptual design. Several potential options are identified and then subjected to at least a preliminary evaluation in order to eliminate clearly unacceptable alternatives. The surviving alternatives are next subjected to more detailed design efforts, and more complete functional and physical architectures or specifications are obtained. It is at this phase that the enterprise, functional, and physical architectures are initially identified. Functional analysis approaches are particularly useful in this phase of effort.

System Development

3. Logical Design and Physical Architectural Specifications This phase results in an effort to specify the content of the system product in question and to provide more detail to the associated high-level functional and physical architectures that were identified in the previous phase. Specifications are translated into detailed representations in logical form such that system development may proceed. This logical design product (sometimes called a functional architecture) and the product architectural specifications are realized in terms of the physical architecture (sometimes called engineering architecture) of the system that will ultimately be implemented.

4. Detailed Design, Production, and Testing The goal of this phase is a set of detailed design specifications that should result in a useful system product. There should exist a high degree of user confidence so that a useful product will result from detailed design, or the entire design effort should be redone or abandoned. Another product of this phase is a refined set of specifications for the operational deployment and evaluation phases of the life cycle. Again, design alternatives are evaluated and a final choice is made, which can be developed with detailed design testing and preliminary operational implementation. This results in the implementation architecture for the system. Utilization of this implementation, or detailed design architecture, results in the actual system. Preparations for actual production and manufacturing are made in this phase.

5. Operational Implementation An implementation contractor produces the system here, often in an outsourced manner. A product, process, or system is implemented or fielded for operational evaluation. Preliminary evaluation criteria for final acceptance of the system are obtained and then modified during the following two phases.

System Deployment

6. Operational Test and Evaluation (and Associated Modification) Once implementation has occurred, operational test and evaluation of the system can occur. The system design may be modified as a result of this evaluation, leading, hopefully, to an improved system and, ultimately, operational deployment. Generally, the critical issues for evaluation are adaptations of the elements present in the requirements specifications phase of the systems engineering life-cycle process. A set of specific evaluation test requirements and tests are evolved from the objectives, and needs are determined in the requirements specifications. These should be such that each objective and critical evaluation component can be measured by at least one evaluation test instrument. If it is determined, perhaps through an operational test and evaluation, that the resulting systems product cannot meet user needs, the life-cycle process reverts iteratively to an earlier phase, and the effort continues. An important by-product of system evaluation is determination of ultimate performance limitations for an operationally realizable system. Often, operational evaluation is the only realistic way to establish meaningful information concerning functional effectiveness of the result of a systems engineering effort. Successful evaluation is dependent upon having predetermined explicit evaluation standards.

7. Operational Functioning and Maintenance The last phase includes final acceptance and operational deployment. Maintenance and retrofit can be defined either as part of this phase or as additional phases in the life cycle; either is an acceptable way to define the system life cycle for system acquisition or production. Maintenance can include reengineering of the product or system or retirement and phase-out.

With only cursory examination, this process would seem to confine evaluation activities primarily to the last stages of development. This is correct in the sense that it is at this phase that the formal operational test of the deployed system is conducted. However, configuration management efforts include evaluation and verification efforts at all phases of the life cycle. Adelman (1992) is among many that recognize that judgments and decisions pervade every phase of the systems engineering process. The results of analysis and evaluation, represented as iteration or feedback loops in most models, provide input to support development objectives at each phase and determine whether those goals have been achieved. This continuous evaluation is a critical component in requirements-driven design.

The early phases of system development are characterized by the greatest degree of uncertainty. As a result, as much as 80 percent of the mismatch between what the user wanted and what the developers delivered has been traced to shortfall in the definition of requirements (Boar, 1984). Barry Boehm's (1981, 2000) research indicates that the cost to fix these discrepancies may range as high as 100 times the cost had correct requirements been identified during the requirements analysis phase. Furthermore, empirical evidence from a number of studies reveals dramatic increases in error correction costs the later in the development cycle the error is found (Daly, 1977; Boehm, 1976; Fagan, 1974). The requisite rework leads to cost overruns and schedule slippage. Conversely, approaches to development that eliminate rework and postdevelopment modification promise productivity improvements from 30 to 50 percent (Boehm, 1987). For this reason, the search for cost-effective system performance improvement methods has focused on improving the quality of requirements identification and representation methods.

10.2 MODELS FOR HSI

As we have noted, there are three essential life-cycle phases in engineering a system: definition, development, and deployment. Models are especially useful in implementing these phases, especially in the definition phase and the early portion of the development phase. There are three generic types of models that are most useful here: conceptual models, requirement models, and prototyping models. Each of the phases of systems engineering is an iterative refining process in which formulation, analysis, and interpretation—and associated modeling and evaluation—interact continually. In the early phases, especially during system definition, models may be largely informal, conceptual expressions of the system engineer's architectural view of the system and its context. Evaluation of existing system operations supports the early stages of concept definition that, in turn, may form the first system model. Implicit in this model is some representation of the system's purpose as it relates to organizational goals and the identification of criteria by which the achievement of those goals is recognized.

As definition progresses, the current system model is analyzed in terms of the perceived deficiencies, or shortfalls, between what the system provides and what the organization needs. This process leads to the definition of yet another model—a set of requirements for the next-generation system and the criteria by which alternative architectures and designs, or system models, will be evaluated with respect to those requirements. Evaluation and modeling continue to play a key role in supporting decisions throughout the iterative process of architecting and systems design engineering. Even in the early life-cycle phases, evaluation is still being performed upon models in the form of system prototypes. Finally, evaluation of operational systems is accomplished in the early phases of system deployment based on the assumption that the evaluation criteria, established in the form of measures of performance (MOPs) and measures of effectiveness (MOEs), accurately represent (or model) the relationships between component, subsystem, and system performance and the larger purpose for which the system is intended.

The HSI approach uses models to conceptualize the user, tasks, and system supports. To effectively incorporate human systems engineering into systems engineering processes requires a framework for integrating and extending the multiple models that support understanding, representing, and translating the user's role in the human–machine system in terms of tasks performed, knowledge required, context of use, and organizational objectives. Ultimately, the level of detail chosen must be determined by the information required. Effective models may be characterized in terms of several key desirable characteristics:

- The level of detail is adequate to support evaluation of principal factors of interest at the current phase in the life-cycle process.
- The issue representation scheme and mode are appropriate to the question at hand.
- The assumptions regarding the nature and relationship of pertinent system variables can be supported by valid sources (historical data, acknowledged experts, output from other validated models).
- The resulting model is understandable to the responsible analysts and the critical reviewers.

A number of approaches for modeling based on human factors and systems engineering concerns are discussed in Salvendy (1997, 2001) and Sage and Rouse (1999). We will now turn our attention to a number of modeling issues in systems engineering as they specifically relate to HSI. These issues are covered for the following major sections: systems definition, system requirements, system conceptual and architectural design, prototyping and implementation, and system evaluation.

10.3 SYSTEM DEFINITION

10.3.1 System Definition Goals

The problem definition phase of systems engineering serves two purposes. First, the definition phase determines the scope of the proposed system in terms of what is needed and technically feasible. Second, this initial phase establishes the goals and objectives for the system development effort to follow. System definition is accomplished by examining three general types of information:

- *System Context* Who will use the system, what they are trying to do with it, under what conditions it will be used, etc.
- *Constraints* "Built in" requirements for inputs, outputs, interconnection, environmental tolerances, etc.
- *Technological Opportunities* Leverage points where technology may be applied with greatest benefit

During the initial portion of the system definition phase, the systems engineering team gathers information needed to understand the functional objectives of the user enterprise for the system to be engineered. Information drawn from various organizational documents and discussions with the user enterprise help to sketch the system boundaries and develop a profile of the system context as defined by:

- *Users* Experience, training, organizational roles
- *Tasks* High-level functions, performance goals, decision task characteristics (timing, criticality)
- *Organizational Context* Organizational goals, missions, control structures, communication modes
- *Environmental/Situational Context* When, where, how, and under what conditions will the system be used

This information comprises the operational needs that the new system must meet. The various dimensions of the system context each generate constraints on the system that must be explored during the requirements phase and addressed in the design. Moreover, constraints involving human performance, hardware, and software interact. For this reason, it is essential that the human factors unit of the systems engineering team coordinate with the other members of the team during these early definition efforts in order to consider these interdependencies. Initial decisions regarding the system concept trade off these technological opportunities (i.e., what *might* be done) against the system context and constraints (i.e., what *must* be done). The impacts of human user model both affect and are affected by the other hardware and software issues.

10.3.2 Models for System Definition

The early portions of the system definition phase provide the initial suggestions that guide the more detailed requirements identification and analysis and the subsequent technological specifications that follow. For this purpose, the most useful outputs from this early part of the definition phase are preliminary models, such as concept maps and functional decomposition diagrams, which define the central constructs of the system and indicate relationships between them. One of the most difficult aspects of the initial part of the system definition phase is the internal (and sometimes external) pressure to "define" in terms of solutions. Jumping to solution thinking during this phase may focus the subsequent requirements identification activities on a subset of the problem while neglecting other equally relevant aspects. This "tunnel vision" early in the development can lead to one of the most common sources of error—defining the wrong problem and then proceeding to solve it.

System definition activities focus on understanding the current ("as-is") organizational activity and developing goals and descriptions of the desired ("to-be") processes, functions, and technology support. System definition and the associated requirements identification activities vary widely in the granularity of representations that are required. The same system may use different modeling methods for different development efforts. Some models are suitable for extension and elaboration as the system concept evolves, while others are more narrowly focused with limited application. Several methods specifically address the semantic aspects of domain knowledge and are useful to the human systems engineer. For example, concept mapping is an informal technique for modeling relationships and interdependencies. The method was developed in the field of educational psychology and has been applied successfully to the acquisition and modeling of knowledge requirements for decision support systems (Seamster et al., 1997; Vennix et al., 1994; Klein, 1993b). Kieras (1988) developed a similar set of goal-task models to structure cognitive learning tasks. This method was used to identify and structure the cognitive requirements for embedded training in tactical information systems (Williams et al., 1989). Cognitive mapping (Montazemi and Conrath, 1986) is a more formal technique that evolved in the field of artificial intelligence. It focuses on modeling cause-and-effect relationships for process or behavior understanding and has been adapted to create computable cognitive architectures in neural networks (Senge and Sterman, 1994; Zhang et al., 1992). Soft systems methodology (SSM), developed by Peter Checkland (1981, 1998) in the 1970s, has been applied to action research and information systems development in medical, industrial, military, and other governmental organizations. The method uses informal models as a means to explore purposeful action within an organization and the necessary information support required.

Figure 10.3 presents a model, drawn from Ehrhart (1994), of a simple decision task in relating incoming information and the human information interpretation process. This example models aspects of the tanker duty officer's (TDO's) tasks in the Air Operations Center. The TDO is responsible for providing air-refueling support to all scheduled missions that require refueling. Replanning is required when new missions are created, existing missions rerouted, or air-refueling resources change. The TDO performs replanning tasks as indicated by his own assessment of the evolving situation and as tasked by other duty officers. For the HSI team, this model helps to identify the elements, or key variables, that need to be presented to the user such as current, planned, and required resources and operational situations. It also indicates that the user is basing part of the interpretation of this information on the potential change in information values across time. The model is annotated with HSI issues, such as potential error sources, experience, and aiding requirements.

Byrd et al. (1992) surveyed 18 requirements analysis and knowledge acquisition techniques that facilitate problem domain understanding in terms of information requirements, process understanding, behavior understanding, and problem frame understanding. They emphasize that none of the methods is suitable for eliciting and modeling all the dimensions of domain knowledge. The key to effective problem definition is finding a means for creating and relating *multiple* models, or views, of the problem. When the problem is complex and multidimensional, the design team needs methods specifically designed to facilitate interdisciplinary thinking. For example, multiperspective context

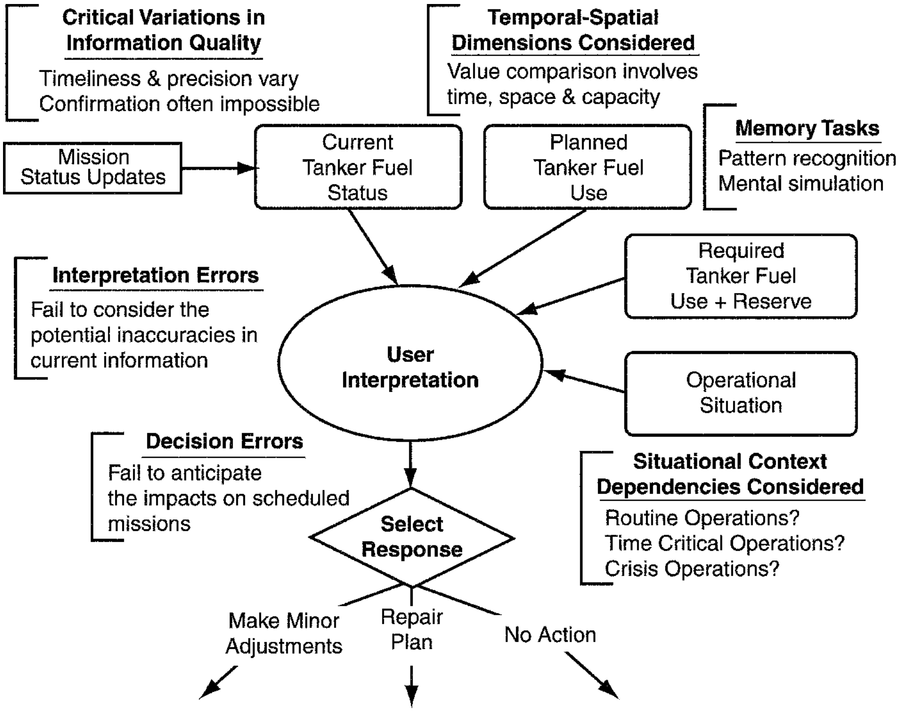

Figure 10.3 Simple decision task with HSI annotations.

models, such as those described for problem analysis in Davis (1993), assist in creating informal models for review and iteration with the sponsors and operational users. Similarly, Zahniser (1993) describes the creation of *N*-dimensional views of the system developed by cross-functional development teams. The process is designed to encourage innovative thinking and bring multidisciplinary experience to bear on system development problems.

System definition models help to organize the system goals and objectives to guide the developers in the requirements identification phase. For the human engineering team, the most relevant issues are those aspects of the problem definition that address the functional roles and activities that are modeled for the human users. Using the initial high-level function allocation, the design team must begin to identify and analyze the human task requirements and the associated implications for HSI.

10.4 SYSTEM REQUIREMENTS

Although the deployment efforts to bring about a technical solution to problems are often cited and blamed for performance failures, the results of several studies of software-intensive systems traced the majority of errors in delivered systems to the *pre*deployment phases (Davis, 1993). Thus, the greatest leverage on improving the product integrity in human–machine systems is to be gained by adopting a systematic method for improving the predeployment or preimplementation processes and products. This may best be accomplished by obtaining a more comprehensive *understanding* of the users, tasks, and operational context; a more accurate *representation* of the technical requirements; and a more effective *translation* of requirements into development specifications, architectures, and system designs.

10.4.1 Requirements Identification, Analysis, and Representation Goals

During the portion of the definition phase that concerns requirements identification and analysis, the systems engineering team focuses on deepening and extending the knowledge represented in the system definition models with respect to the human users and their task support needs. System design and development requirements provide a focal point for integrating the information gathered on the users, problem-solving tasks, and the operational environment in order to guide design and development decisions. These requirements include not only the human–machine interaction requirements that define the operation of interfaces, but also the cognitive task requirements (CTRs) that define the supports for the user's decision task performance. Particularly in cases where the decision tasks are complex and must be performed in a dynamic, time-stressed environment, the operation of the interface must not distract the user from the primary tasks involved in accomplishing the organizational goals. The systems engineering architecture and design team uses the cognitive and interaction task information to determine the most beneficial human–machine task allocation, information representation modes, display formatting, and information interaction protocols.

During the requirements analysis phase of development, the CTRs can be identified and defined as part of the normal requirements identification activities. The goal during this phase is to gain an understanding of the functional tasks that the human user(s) must perform and how the user, organization, and situation define and impact those tasks. Using the high-level conceptual models from the early system definition activities and the evolving hardware and software requirements, the team develops models of information flows, task allocations, and organizational procedures for decision making. At this point, it is useful to observe the way the organization currently addresses the problem and interview representative users to expand and correct various preliminary functional, procedural, and dependency models.

User-Centered Requirements Framework The CSE framework includes a user-centered requirements framework (Ehrhart, 1997) that expands upon information obtained during the system requirements analysis. This should be modeled to include a representation of the user's cognitive tasks, as implied by the information flows or prescribed by operational procedures, and the interpretation of analysis of that model with respect to the user's information requirements and the possible sources of cognitive errors. The CTRs are constructed through the process of evolving and relating models that profile the user and organization. They describe the environmental and situational contexts and define the various cognitive tasks involved in accomplishing the functional tasks assigned to the human–computer decision component, as shown in Figure 10.4.

10.4.2 Models for Requirements Capture and Analysis

A CTR represents either the nature of the *input* required for a human decision-making task or the content of the *output* required from that task. Thus, initial objectives in the

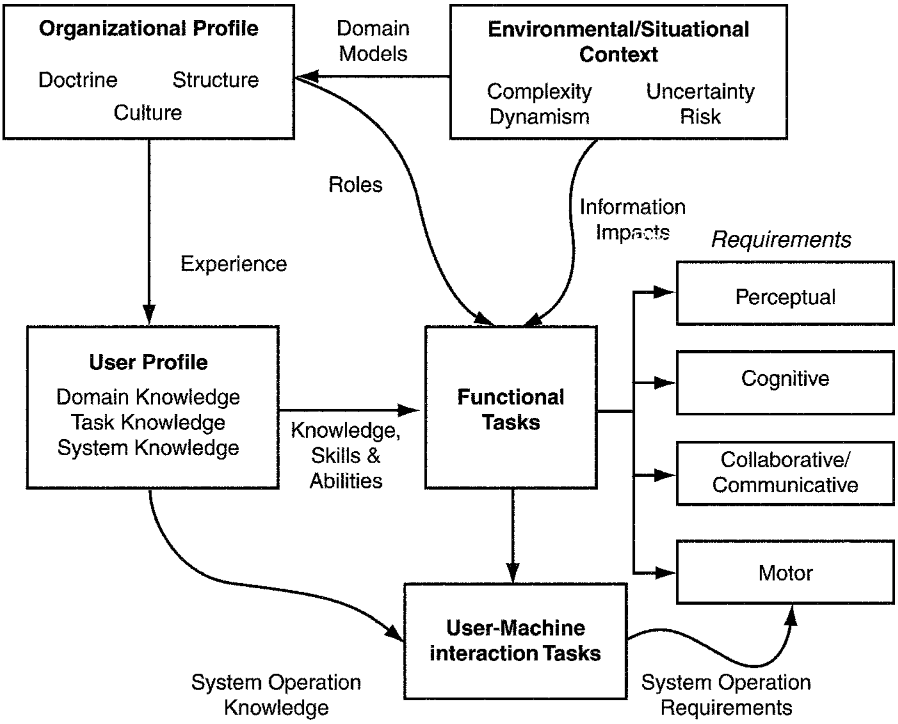

Figure 10.4 Relationship between elements of user-centered requirements model.

requirements phase are to identify the kinds of cognitive tasks that the users may be required to perform and to examine the factors that may affect performance. If a task affects decision performance, it is necessary to find out what characteristics of the task do so. Meister (1981) identifies five task dimensions that may affect performance:

1. Functional requirements (cognition, perception, etc.)
2. Complexity
3. Mental workload
4. Temporal factors (pace, duration, sequence, etc.)
5. Criticality

Cognitive task taxonomies, such as those found in Fleishman and Quaintance (1984) and Rasmussen et al. (1990) can be used as a filter to identify and categorize basic cognitive tasks with respect to these dimensions. In addition, Andriole and Adelman (1995) present a taxonomic discussion of human information processing and inferencing tasks with respect to the potential cognitive errors associated with each.

As the team reviews the context diagrams, functional decomposition diagrams, and straw-person storyboards, descriptions of activities can be examined for verbal constructs that indicate human user actions. For example, in systems where the human user must *monitor* a situation and *interpret* evolving events, the software designers may view the inputs to the user as updates to a database. From the user's perspective, however, this requirement has implications not only for interface operation design but also for the information presentation design. In order to interpret those updates, the changes must not only be visible to the user but also presented within a meaningful context. Using the concepts of analogical representation and causal reasoning, this context might include some mapping of relationships between key factors, tracing of changes in relevant factors over time, and/or models of a goal state to which certain parameters should conform. At this point, the information presentation and interaction requirements continue to be identified from the user's perspective without specifying the design solution.

The simple model shown in Figure 10.4 raises numerous questions for the support system (aiding) design, as well as the training and staffing design, such as:

- How often must the status information be updated?
- How does the user need the information presented to comprehend the meaning of the change?
- Does the user ever need to know or review resource trends going back several updates? If so, is the current direction of the design implying that the user will retain this in his or her memory or keep notes off-line?
- Does the user make these interpretations routinely? Occasionally? Rarely?
- How does the change in current resources relate to the operational situation?
- Will the user have experienced a wide or narrow range of interpretation situations?
- What situational contingencies might negatively affect the user's accurate interpretation of these factors?
- How does the task/decision impact the mission? How critical is it? How rapidly must the decision be made? Where and how will it be disseminated?

These questions and others may need to be addressed in the design and coordinated with the other development teams involved in the effort to engineer the system. To answer them requires understanding not only the structure of information flows but also the way in which that information is used. Thus, it is important to address relevant issues identified in cognitive research regarding users, tasks, organizations, and situational context.

10.4.3 Profiling the Operational Context

More often than not, performance issues that affect organizational operations strongly relate to human performance issues (Stolovitch and Keeps, 1992). Models of the situational context, or decision environment, should capture and represent the conditions that impact decision making. They should also capture the effects of agents and events that are internal and external to the organization itself. The models in this section provide several perspectives for modeling situational context and interpreting potential impacts on decision making. Table 10.2 highlights important factors, associated characteristics, and human performance issues. Due to their considerable interaction with the decision tasks and users, similar issues are addressed with respect to the characteristics of the users, organizations, and tasks.

Context Categories and Situational Response Meister (1991) presents a categorization of situational contexts in terms of four possible levels of determinacy that

TABLE 10.2 Environmental/Situational Context Profiles

Factor	Characteristics	Human Performance Issues
Determinacy	• Ranges from *determinate* (highly predictable with only one probable outcome) to *indeterminate* (no outcome can be identified as significantly more probable). • Describes degree to which all variables affecting given problem or situation are known and understood. • Interacts with complexity and dynamics of environment.	• *Task allocation and aiding impacts:* dynamic demand on user's perceptual and cognitive resources • *Information design:* level of detail; representation in discrete or symbolic formats; importance of structure in information presentation • *Personnel assignment and training requirements:* more stochastic environments require expertise acquired through rich experience; experiential learning difficult in highly mutable environments
Structure and boundedness	• *Structure:* Describes extent to which crucial information for task performance is known, available, and quantifiable. • *Boundedness:* Describes extent to which problem is constrained, may be represented in reliable fashion, and is tractable for human information processing.	• *Task allocation and aiding impacts:* tractability impacts on user cognitive workload; structural impacts on aiding concept • *Information design:* unstructured environments may require symbolic representation to support understanding of the qualitative aspects of the task • *Personnel assignment and training requirements:* reduced structure demands greater breadth of understanding; procedure-oriented training may not develop adequate task knowledge
Complexity	• Defined by number of interconnected components or aspects and degree of interdependence between them. • Interdependence critically impacts "fault tolerance" of procedural designs.	• *Task allocation and aiding impacts:* task control may be distributed and require "what-if" tools to predict effects of proposed actions on related elements • *Information design:* users need representations of dependencies to understand effects of possible events on chain of dependent factors • *Personnel assignment and training requirements:* users need training to understand structure of operational context and requirement to examine "ripple effects" of their actions

roughly equate to the degree to which the domain is well-bounded and predictable. The situational context may be considered *determinate* when the given situation or initial condition has only one significantly probable outcome. This highly predictable context for decisions includes common mechanical systems, some highly institutionalized social systems, and certain control systems. *Moderately stochastic* situations have only a limited number of qualitatively similar outcomes with a significant probability of occurrence. In this context, prediction of outcomes remains tractable as in the case of genetic processes or system variability due to variable dimensions in the component parts. *Severely stochastic* situations have a large number of qualitatively similar outcomes with a significant probability of occurrence. While event outcomes in these situations remain predictable, they are computationally intensive and beyond the range of unaided human computation. Severely stochastic situations involving human agents also have qualitative aspects that increase the difficulty of response and outcome prediction. *Indeterminate* situations provide so little information about possible outcomes that no outcome can be identified as significantly more probable. Meister cites psychotic human behavior and some political alliances as examples of indeterminate contexts.

These “environments” rarely exist in discrete form in practice; system users generally perform tasks simultaneously across a range of environments. For example, flying an aircraft requires interacting with multiple environments. The aircraft systems perform within determinate to moderately stochastic ranges. Air speed and altitude are absolute values with narrowly defined meanings for certain tasks. Other parameters (i.e., fuel consumption) represent calculated values for which there are ranges of accuracy. Outside the cockpit, the aircraft pilot must interact with severely stochastic weather conditions that may affect the aircraft in unpredictable ways. When the aircraft involved is a military aircraft, the pilot must also respond to the indeterminate environment of the battlefield.

In more determinate contexts, the operational goals focus on applying well-understood procedures to respond effectively to a highly constrained set of triggering events. The users seek to maintain operational consistency and control to meet routine performance demands. Errors occur when responses are too rigid to react to major changes and/or novel events or when users apply inappropriate procedures in a changed environment. Since control is maintained by manipulating key factors to create predictable outcomes, users need detailed information about situation inputs, user actions, and outcomes. As environmental/situational uncertainty increases, users must make efficient use of resources in a succession of varying short-term situations to rapidly and effectively exploit opportunities. The emphasis is on flexibility and adaptive, creative responses in the face of novel events. Errors occur when latency between recognition of situation and internal readjustment results in a lack of effective control. In addition, the high degree of uncertainty and ambiguity in novel events makes the application of experiential learning more difficult. Detailed data is often less valuable than symbolic representations of functional relationships to convey structure and assist users in recognizing and interpreting common aspects of novel events. Overviews and aggregated displays can support pattern matching and provide externalized mental models. Since extremely stochastic and indeterminate environments are often complex and dynamic, it is important to support users with multiple levels of abstraction to meet adaptive cognitive control requirements.

In addition to the determinacy of the situational context, it is useful to understand and model the degree of structure as well as the boundedness and complexity inherent in the situational context and typical decision tasks. Several researchers discuss the interaction of these factors (Fleishman and Quaintance, 1984; Meister, 1991; Rasmussen et al., 1994) and their implications for aiding the user. The structural characteristics of the decision context and tasks should be considered in the selection of the analytical methods that form the basis of the decision aid design as well as the interaction routines that facilitate the human–computer cooperation.

The degree of structure in a decision domain characterizes the typical situations and decision tasks in terms of the extent to which information on the key variables is available and quantifiable. For example, *highly structured* contexts are those where all critical information is readily available and quantifiable for accurate manipulation. In *semistructured* contexts, the key variables may be quantified without losing critical information or making difficult assumptions; however, often some of the critical information is unavailable. In this case, the uncertainty surrounding the decision involves "known unknowns" that may have to be inferred if further information cannot be obtained. Finally, *unstructured* contexts involve qualitative variables that may not be legitimately quantified. In addition, there may be "unknown unknowns," that is, critical information that is either not available or not represented in the user's model of the situation or task.

Closely related to determinacy and structure, "boundedness" incorporates the degree to which the key variables constrain the problem to make it tractable. The representativeness and reliability of the variables also contribute to the boundedness of the problem domain. A *closed* domain may be constrained and described accurately with variables that require minimal cognitive demands to manipulate. When the domain is *semibounded*, the variables may only be generally representative and reliable. The associated uncertainty is manageable only by highly trained and motivated experts. The *open*, or unbounded, context involves variables that may not be well-understood and/or reliable. The resulting uncertainty exceeds human ability to absorb and manipulate.

The degree of complexity characteristic in the domain is interwoven in the concepts of both structure and boundedness. Woods (1988) defines complexity in a domain or a system in terms of the number of interconnecting parts or subsystems and the degree of interdependence between them. Using a structural model of situational context, complexity may be further delineated with respect to the number of hierarchical levels (vertical complexity) and number of parts or subsystems per level (horizontal complexity). In *simple* domains, both the vertical and horizontal complexity is low and the critical variables in the situation do not interact. In a system context, this absence of interdependence results in component functioning unaffected by performance of other system parts. In *moderately complex* domains, the degree of vertical and horizontal complexity increases and there is greater interdependence between the variables involved. In moderately complex domains, performance of functions may be enhanced or degraded by the performance or nonperformance of other subsystems. *Complex* domains and systems involve many hierarchical levels extended by many interdependent parts and subsystems. The functions of a complex system cannot be performed if other subsystems perform poorly or not at all. The inherent complexity of the situational context plays a significant role in the user's ability to mentally simulate the consequences of a proposed response. From a design perspective, simplifying domain complexity may eliminate critical information with unpredictable results.

Effects of Situational Context on Task Performance Situational context is an important variable in several models of human information processing and decision making. For example, Rasmussen's (1986) skills–rules–knowledge (SRK) model has three levels of cognitive control based upon situational contingencies and user knowledge. *Skill-based* control comprises the highly integrated, automatic sensory–motor responses that occur with little conscious effort. Efficient control in this mode is dependent upon experience and a predictable environment. In *rule-based* responses, the user is consciously aware of taking a sequence of steps to attain a goal that may not be explicitly formulated. As a result, the user can accurately describe the procedure or rule triggered by the situation, but often cannot explain the situational cues that triggered the rule. In novel situations or unfamiliar environments, the user does not have readily understandable cues to trigger procedural responses and must use additional cognitive resources to analyze the situation. Situation assessment in *knowledge-based*, or model-based, control is used to formulate an explicit goal and identify procedures to attain the goal. When reasoning identifies an appropriate rule or procedure, control drops back to the rule-based level. The decision-making effectiveness in this mode depends upon the quality of the user's "mental model" of the situational context.

Understanding the situational context can provide insight into the potential cognitive demand placed on the human user. For example, in simple, primarily determinate contexts, the decision-making efforts focus on optimizing the outcome by manipulating the initial conditions. This generally involves skill-based actions and some rule-based control. Semistructured, moderately stochastic contexts tend to induce attempts to manipulate initial conditions using primarily rule-based control. Since the possible outcomes are bounded, efforts often focus on optimizing the expected value of outcomes. In severely stochastic contexts, precisely manipulating initial conditions cannot control outcomes. Furthermore, detailed planning and reliance on preplanned procedures (rules) are less useful due to the unpredictability of complex evolving situations. In this case, combinations of knowledge- and rule-based response control efforts focus on preparation for unfavorable outcomes and maintaining an ability to recognize and rapidly exploit opportunities. Decision responses in a complex, indeterminate situational context rely primarily on knowledge-based control. Effective performance depends upon knowing enough about the situation and the domain to classify it. The highly unpredictable nature of these contexts requires an intuitive approach based upon well-developed mental models of the domain and environment to protect against disastrous response errors.

Situation assessment and mental models also drive Klein's (1993a) recognition primed decision (RPD) making model of expert decision making in dynamic situations. The RPD model describes decision-making behaviors comparable to the rule-based and knowledge-based behaviors described by the SRK model. When forced to respond quickly in an unfamiliar situation, the expert user attempts to identify aspects of the situation similar to previously experienced situations. In simple recognition situations, matching the current situation to a previously experienced analog automatically indicates the appropriate course of action in terms of the procedure to follow. In more complex recognition situations where there is no readily available analog addressing the key features of the situation, users must also reason about possible courses of action. This reasoning involves mental simulation of the possible outcome(s) of a particular course of action based upon the user's mental model of the situational context and ability to manipulate the network of interdependencies. The resulting cognitive demands lead to a satisfying, rather than optimizing, strategy in which the user selects the first course of action that appears to satisfactorily attain the goal.

Crises form a special case in situational contexts that impact the users, organization, and decision tasks. Hermann (1972) defines a *crisis* as a situation that:

- Presents a *threat* to one or more important goals of the organization.
- Permits only a very *short decision time* before situation changes significantly.
- Involves novel or unanticipated events that *surprise* the system users.

Threat or risk to the organization plays a central role in such domains as international politics, corporate management, and military operations. In each case, the situational context is dynamic and complex. The normal states of these environments range from moderately stochastic to indeterminate. The systems designed to cope with normal operations also must support rapid response to unanticipated events.

10.4.4 Profiling Organizations

The situational context surrounding a judgment or choice situation forms the external environment for decision making. The structure, function, and purpose goals of the organization provide the internal environment. The contingency task structure of a decision situation, which represents the external and internal environment surrounding a task needing attention and the experiential familiarity of the person or group undertaking judgment and choice with the task and the external and internal environment, will influence the mode of judgment and choice that is used. Another relevant variable, organizational culture, is dominantly influenced by the shared values and group behavior norms that shape human and organizational progress (Harrison and Huntington, 2000). Systems designed to support decision making within organizations must take into account not only the technical facets of the hardware, software, and communications architectures with which they cooperate but also the structure of the human organization in which they function. This involves understanding the organizational culture and how it directly, or indirectly, impacts and is impacted by the individual users, their tasks, and the contingency task structure surrounding decisions that must be made. Table 10.3 presents likely characteristics and human performance issues associated with leadership and authority, communication, and decision-making aspects of an organizational profile. Organizational policy, whether implicitly or explicitly communicated to a person or group attempting to exercise judgment, provides not only procedural guidelines for structured tasks but also conceptual perspectives and strategy objectives that must be considered. Organizational response to issues is generally evolutionary, emergent, and adaptive, and the resulting organizational systems share these characteristics. The engineer of organizational systems must be sensitive to the associated redefinition effects of new systems on the organization and its culture and doctrine. This subsection presents methods for profiling organizations and modeling the relationship of the organization to the other dimensions of systems engineering for human interaction.

Methods for Profiling Organizations In a seminal work, Kotter and Heskett (1992) identify shared values and group behavior norms as the two major ingredients of organizational culture. Values are virtually invisible and are difficult to change. Norms that result from values are easier to identify and to change. However, any attempt to change norms, without an accommodating change in values, is likely to produce very unsatisfactory results. A multistage development approach illustrates how organizational cultures often emerge.

1. The top management in a new organization attempts to implement a strategic vision to support organizational strategy.
2. The deployment is successful and organizational personnel are guided by the new vision and strategy.

TABLE 10.3 Organizational Profile

Factor	Characteristics	Human Performance Issues
Leadership and authority	• Describes span of control, degree of centralization, style of interaction, and flexibility of organizational authority structures.	• Less flexible organizational structures depend upon specific role assignments and narrowly focused training to ensure reliable task performance; more flexible organization structures require cross-training to ensure adaptable responses.
Communication	• Describes chain of interaction required to affect control and obtain feedback on actions taken.	• Organization's structural complexity impacts communication speed and may impact performance across all phases from planning through execution. • Feedback delays due to complex communication chains may result in overcorrection when information on the results of actions is delayed.
Decision making	• Organizational "decision systems" require creation and sharing of common understanding of task domain, goals, and methods for achieving goals.	• Information presentation and interaction requirements must support creation and communication of shared mental models. • Training must include an understanding of task domain, characteristics patterns, roles, and responsibilities within and across functional boundaries.

3. The successful deployment leads to organizational success that continues over the years.
4. A culture results that reflects the vision, strategy, and experiences of organizational leadership.

Many have noted the fact that cultures develop when people interact over a sustained period of time and when they are successful in producing desired results. The longer the initial solution works, the deeper the particular culture becomes imbedded in the organization. Any number of external threats and opportunities may challenge the then prevalent organizational culture. The extent to which it can adapt to future needs determines the extent to which the organization will survive as an excellent organization. The notion of adaptation is key here. Figure 10.5 represents an extension of the ideas presented to illustrate adaptive and maladaptive organizational behavior.

In a landmark work, Edgar Schein (1992) identifies 10 phenomena that exist in a culture. On the basis of these and additional stability and integration requirements, he defines a group or organizational culture as a pattern of shared basic assumptions the group learned as it solved its problems of external adaptation and internal integration. A successful organizational culture is one that has worked well enough to be considered valid and taught to new members as the correct way to perceive, think, and feel in relation

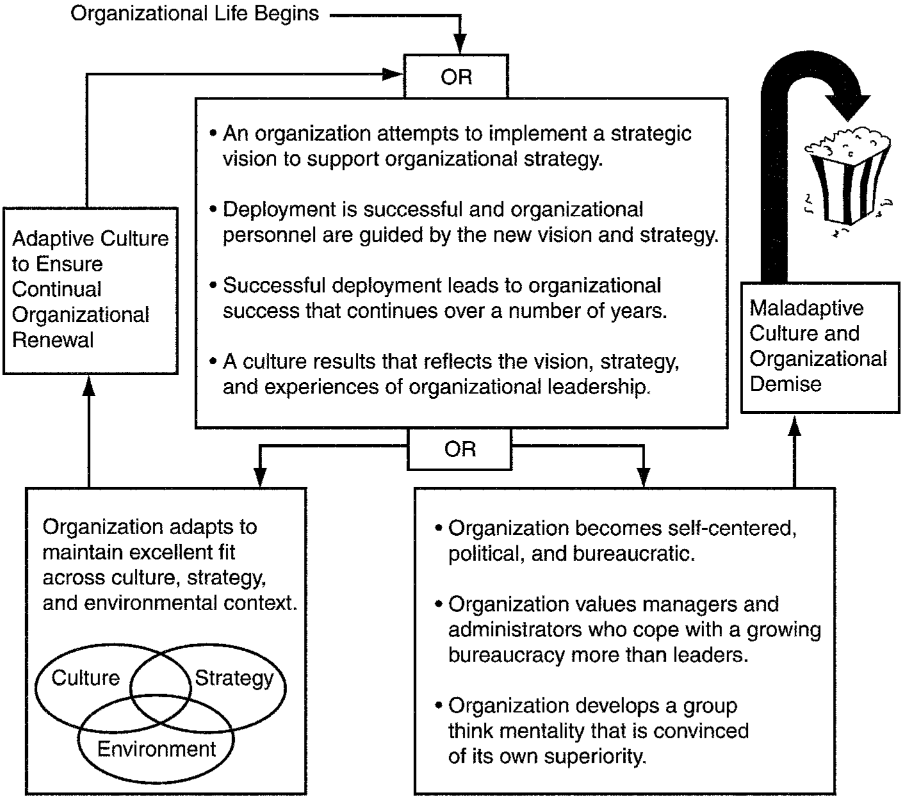

Figure 10.5 Adaptive culture creation and emergence in organization.

to organizational problems. There are three major elements in this definition: *socialization issues*, including the process of how one learns; *behavior issues*; and *issues of subcultures*, and the extent to which they will develop. From this perspective there are causal dynamics involved in culture and leadership. Leaders initially create cultures when creating groups and organizations. Once a culture exists, it will determine the criteria for leadership. A leader in a dysfunctional culture must either change it, such that the group survives, or the culture will ultimately govern the leader.

Schein suggests three levels at which culture may be studied. At the unconscious level of basic underlying assumptions there are often unarticulated beliefs, thoughts, and feelings that represent the ultimate top-level source for the resulting values and organizational structures and processes. At the level of espoused values, formal statements of organizational objectives, purposes, and philosophies may be found. At the level of artifacts, which are comprised of organizational structures and processes, or functions, the organization attempts to implement its espoused values. We can most easily observe the cultural product that is embedded in an organization's structure. With potentially a little more difficulty, we can observe organizational artifacts as represented by processes or organizational functions. It is more difficult to examine the espoused value for the organization and even more difficult to determine the actual values from the espoused values. It is also difficult to measure cultural facets at the level of espoused purpose and inherent values. However, it is important that this be done. Measurements at the levels of process and structure possibly allow for good inferences about espoused values, but will not easily lead to information about actual values. To obtain these, we need observations of espoused values and the processes and structure implied by these, and the products of the actual value system in terms of processes, structure, and organizational products.

Six observable factors, or observables, are identified as primary means to embed culture in an organization. These factors are:

1. Critical factors that leaders measure and control
2. Approaches taken in response to crises and critical incidents
3. Observed criteria for resource allocation
4. Activities of leaders as role models, teachers, and coaches
5. Observed criteria, used in practice, to allocate organizational rewards and status
6. Observed criteria, used in practice, for recruitment of new organizational members and outplacement of existing organizational members

There are a number of supportive mechanisms that relate to organizational structure and processes, symbols and rituals, and formal statements of organizational purpose and philosophy. The formation of subcultures is an important aspect of organizational leadership. Subgroup and subculture formation is an inherent likelihood that results from the differentiation process that invariably occurs as an organization expands and grows. Differentiation may be functional, geographical, divisional, hierarchical, or may result across products or markets. Subcultures are, in no sense, always harmful. They may be supportive or harmful to an organization's mission depending upon how they are grown and how they mature. Thus, it is important to be able to characterize the factors that will lead to organizational growth, maturity, decline, and rebirth and the role of technology, especially the role of information technology, in supporting these changes. This creates a major need for organizational profiling.

In relevant work concerning profiling, Burton and Obel (1995) formalize the interactions between the organizational features, external environment, and technology use to generate prescriptive advice for organizational design. The underlying guidelines also may be used to project technology needs based upon the interaction of such organizational factors as structure, coordination and control, size, and strategy and such environmental factors as ambiguity, uncertainty, and complexity.

French and Bell (1973) present a hierarchical framework for developing an understanding of organizational functioning based upon information regarding organizational culture, climate, processes, and goals. The framework permits study of the organization as a whole and provides methods for examining and relating the subsystems, teams, and individual functional roles. At each level in the hierarchy, the analyst may select from a range of knowledge elicitation techniques to characterize activities and model the relationship of that level to the rest of the organization. At the top level of the hierarchy, investigation focuses on the organization as an entity with a common mission and power structure. It may also include the relevant external organizations, groups, or forces and lateral associations that control or interact with the organization. Investigation methods include questionnaires, interviews, focus groups, and examination of organizational documents that concern such relevant aspects as policies and standards.

There are a number of related studies. Salama (1992) reviews organizational "biographies," or histories of the development and activities of an organization, and shows that they provide insight into organizational culture. Questions related to culture, climate, and attitudes also are relevant at the team or group level functioning within an organization. The analysis reported here seeks to discover answers to such questions as:

- What are the major problems of a group or team?
- How can team effectiveness be improved?
- How well do the member/leader relationships work?
- How does the team relate to organizational goals? Do members understand this relationship?
- How well are team resources employed?

Individual interviews, using techniques such as concept mapping, followed by group review and discussion aid in identifying and refining models of team/group functioning. The models developed may be used in conjunction with more detailed cognitive task analysis to link team structure and function to the specifics of the task and environment.

Most models developed to describe organizational structure and functioning assume additional meaning when an understanding of the organizational culture augments them. Robbins (1990) identifies 10 dimensions that define organizational culture. These include *structural features* such as control, integration, interaction patterns, and rewards; *management characteristics* such as direction and support; *organization responses* such as conflict tolerance and risk tolerance; and *individual characteristics* such as initiative and identification. A strong organizational culture communicates the organization's model of appropriate behaviors to the individual members and increases their identification with the organization. An organization is said to have a strong culture when the core values of the organization are clearly understood, intensely held, and widely shared. The resulting unit cohesion prevents breakdowns in procedures in high-stress, crisis situations and is critical for effective performance. For this reason, technologies introduced into an organization must facilitate and not interrupt the flow of communication and interaction that supports team cohesion. A strong organizational culture also can have negative effects on decision making, such as the social pressure for uniformity and failure to question weak arguments common in "groupthink" situations (Janis, 1982, 1989).

The concepts of *collective cognition* and the *collective mind* propose to describe the purposeful interaction characterizing team performance in situations requiring a high level of continuous reliability (Weick, 1995; Weick and Roberts, 1993). The collective mind is evidenced by the manner in which the team members structure and coordinate their actions with respect to a shared mental model of the system. The research of Weick and Roberts (1993) examined the effects of variations in the individual models and coordination of actions in aircraft carrier flight deck operations. As team members increased the conscious interrelating of their actions within the system, they improved their comprehension of unfolding events and reduced the incidence of error. The researchers present a model of collective cognition that relates actions (contributions), the shared mental model (representations), and the coordination of actions within the system (subordinations). In related research, Schneider and Angelmar (1993) investigated collective cognition in organizations and proposed a cognitive framework based on structure, process, and style that is applicable to the individual, group, and organizational levels of analysis.

Examining the formal and informal lines of communication in an organization provides additional information on the means by which control is exercised in an organization. Harrison (1985) discovered that patterns of interaction defined through communication between the hierarchical levels of an organization establish a shared understanding about levels of influence in decision-making processes and how such influence may be exercised. Moreover, the definition of participation through interaction dominated the perceptions of subordinates, regardless of the management style reported by their superiors. The results indicate the importance of actively supporting interaction between levels of the organization where decision-making effectiveness depends upon intraunit participation. Thus, it is not surprising to see a variety of literature in this area relating to the culture of work organizations and organizational design and functioning (Trice and Beyer, 1993) and works on organizational design in the management science literature appearing over the years (Nystrom and Starbuck, 1981; Galbraith, 1977, 1995; Nadler and Tushman, 1997). The subject also plays a prominent role in works that concern such subjects as systems engineering (Sage, 1992) and systems management (Sage, 1995).

Organizational Responses to Situational Contexts and the Role of Contingency Task Structures Organizations and systems must be designed for effective response in both routine operating conditions and problem situations (Meister, 1991). Organizations develop routine (or standard) operating procedures to guide responses in relatively stable, predictable environments. Although specific tasks may involve some risks there is usually low threat and adequate response time. In this context, users respond to problems arising in their sphere of responsibility according to specific guidance from superior authority. These procedures permit a high degree of control and consistency across all organizational levels to ensure organizational objectives are met. The longer decision horizons permit subordinate users to defer responses when situations exceed the scope of their responsibility. The reduced threat allows users to reduce their workload through the use of various cognitive shortcuts or heuristics. Janis (1989) suggests that the cognitive shortcuts used in routine decision making provide more efficient responses than the conscious pursuit of more precise decisions through formal reasoning efforts. Efforts such as this were at the heart of the cognitive mode model of Janis and Mann (1977), which indicated that individuals search for information to:

1. Enable recognition of a potentially challenging opportunity.
2. Enable determination of potential losses if the present course of action is continued.
3. Enable determination of potential losses if a change is made to a new but familiar course of action.
4. Enable determination of whether it is reasonable to find a better course of action than the familiar ones already considered and initially dismissed as improper.
5. Ascertain if familiar courses of action not previously considered are acceptable.
6. Ascertain whether the remaining time until the decision must be appropriate to formal rational deliberation.
7. Support a formal formulation, analysis, and interpretation of the issues and the resulting vigilant search, processing, and deliberation.

Crisis conditions trigger shifts in organizational communication and control patterns (Hermann, 1972; Meister, 1991). Organizations designed to operate effectively in dynamic, high-threat environments must adapt rapidly to crisis conditions and novel situations. Communication delays may impair information gathering and decision implementation. For this reason, users must respond to novel problems arising in their sphere of responsibility during a crisis with only general guidance from superior authority. There is some evidence that more loosely coupled organizational structures with built in redundancy and informal interaction are necessary to respond effectively in complex, dynamic, high-threat environments (Pew, 1988). With training and experience in crisis operations, users gain experiences to develop a wide range of creative responses; however, their focus on the immediate problem may result in a satisfying response that does not meet organizational objectives.

Hermann (1972) describes the effects of crisis situations on three organizational dimensions: leadership and control, communication, and decision making. The leaders' attitudes toward rank and authority are critical determinants of subordinates' willingness to raise issues that appear to challenge the prevailing hypothesis. Conversely, weak or inexperienced leaders may be influenced in crisis situations by subordinates to make incorrect decisions (Janis, 1989). In crisis operations, there is typically a marked increase in communication with internal and external agencies. The increased intrateam communication may lead to a general air of confusion (and potentially panic) and increase the impulse to action.

When routine operations constitute the majority of organizational experience, users have little opportunity to develop a wide range of responses and may be ill prepared for sudden shifts in the environment. This can have disastrous effects for response coordination. For example, Helmreich (1988) cites National Aeronautics and Space Administration (NASA) and National Transportation Safety Board (NTSB) studies implicating crew coordination in more than 70 percent of aircraft crashes. Often such cases involved minor malfunctions, simple errors, or erroneous assumption that were compounded through inattention or incorrect judgments by a team into nonrecoverable crises. Human–human and human–machine miscommunications, poor use of available support resources, and inadequate situation assessment are the major contributing factors to the resulting failures (Helmreich, 1988).

Designers are rarely able to observe the functioning of organizations during crisis or intense periods of activity. Research indicates that organizational performance during crisis operations may be enhanced through aiding designs that support improved situation assessment and facilitate communication based upon shared mental models (Orasanu and Salas, 1993). The organizational models developed to guide design should explore the human user requirements associated with both crisis and routine operations. The knowledge acquired through these models is used to determine appropriate human–machine task allocation, design information presentation, and develop interaction routines. The organizational models should also provide structures to link user and task profiles. When utilized for the engineering of systems, these principles should lead to appropriate designs for cognitive task performance (Orasanu and Shafto, 1999).

10.4.5 Profiling Users

The functional roles of system users within an organization often are developed in conjunction with the profile of the organization. The HSI systems engineering team also needs to develop a profile of typical users' knowledge and experience. In certain organizations, such as military units, this information may be assumed in part by the functional definition of the position. For example, an aircraft commander may be assumed to have a minimum number of flying hours, to have completed specific training, and passed certain qualifying examinations. System designers need information that may not be assumed automatically from job descriptions. To design the information presentation and interaction routines that coordinate the performance of human–computer cooperative decision making, the design team must develop a profile of the user's experiential familiarity and knowledge of the domain, tasks, and systems involved or the contingency task structure.

Dreyfus and Dreyfus (1986) identify six levels of knowledge that a user may progress through in developing expertise. These levels (novice, advanced beginner, competent, proficient, expert, and master) provide a more detailed picture of the role of expertise in cognitive tasks. Intended in large part initially to support the design of training aids, a subject of much contemporary interest (Salas and Cannon-Bowers, 2001), the Dreyfus model describes the differences that various knowledge/skill/expertise levels make in influencing the mental functions employed in decision-making tasks and the associated mental attributes of the person exercising judgment. The mental functions involved in decision-making tasks include: differences in ability to recognize similarity in environmental and task features, differences in the way task components are conceptualized and recognized, and differences in the decision strategies employed. Figure 10.6 is an interpretation of how the various transitions occur at various levels of proficiency in this model.

The ability to make similarity judgments is essential for rapid recognition of prototypical situations and analogical reasoning for unfamiliar situations (Beach, 1992; Klein, 1993a). Tasks and situations are perceived as decomposed attributes at lower levels of

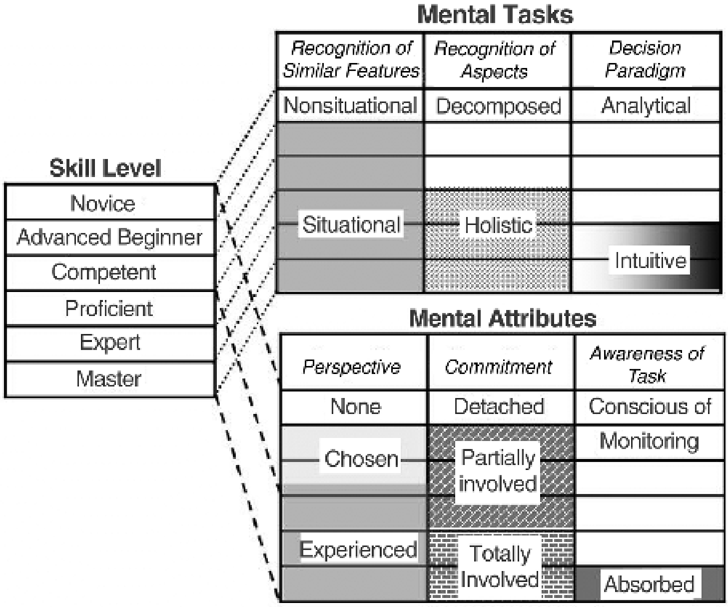

Figure 10.6 Interpretation of the Dreyfus model of decision style.

proficiency or as a whole at the highest levels. Expertise also factors in the decision strategy employed. Lower levels of expertise usually require analytical strategies to manage the problem perceived as parameters or attributes. The holistic models that characterize higher levels of expertise facilitate intuitive strategies. Hammond (2000) has long been concerned with a cognitive continuum model where cognition varies from analytical to intuitive as a function of experiential familiarity with the contingency task structure. Thus, decision strategy selection is based upon the attributes of the task and task situation and expertise of the decider. Clearly, these models are related and address the combination of factors that determine decision strategy.

Several resources are available to guide the system designer in modeling human users (Meister, 1991; Senders and Moray, 1991), and psychology is often viewed as a requisite science of system design (Meister, 1991; Carroll, 1997). Table 10.4 characterizes user experiential familiarity and knowledge in each dimension as low, medium, and high and provides a simple representation of the continuum of knowledge and experience that usually exists as a mixture of expertise—deep in some areas and broad in others. The same system user may have different expertise and knowledge levels across domain, task understanding, and systems control ability.

The user's knowledge of the specific functional tasks to be performed generally interacts with domain knowledge. For example, a user may have considerable knowledge and experience with the situational contexts that characterize the domain but may have never performed the specific tasks now assigned. In such cases, the user may understand intuitively *what* must be done to accomplish a goal but not know *how* to do it. If a system user is not able to distinguish between relevant and irrelevant information needed to perform a task, the associated cognitive workload will increase. A moderate level of task knowledge supports task performance, and the amount of sustenance is based on the users' experience and familiarity with the task at hand and the associated facility with procedures for system use in accomplishing this task. This knowledge permits the user to trade-off performance quality in order to maintain a reasonable workload and still attain the desired goal. At the highest levels of task knowledge, the user demonstrates flexible, intuitive task performance. Depending upon the level of their domain knowledge, the users can rapidly recognize prototypical situations and adapt their task performance in an appropriate resourceful response to task requirements.

There can be several manifestations of low system knowledge and expertise. For example, when new technology is introduced, the user may have knowledge and experience with the domain and functional tasks but may have had little or no experience using the new system itself. Depending upon their role in the organization, a user may only have used a few system functions while remaining largely unaware of its other capabilities. Both of these knowledge levels result in a limited, often fragmented, knowledge of system operation. As a result, the user usually has an insufficient mental model of the system and may be confused by any errors in system operation that result from its use. The resulting increase in cognitive workload may greatly impair performance in tasks at which the user is otherwise proficient.

The competent user has a moderate knowledge of system functions and the interaction routines required to exercise those functions. The user understands the operation of commonly used system features and can operate various interfaces in order to accomplish the required tasks. The competent user's mental model of the system provides an adequate foundation to allow them to learn from operational errors. The master, or "power" user, has a strong, accurate mental model of the relationships between himself, the machine, and

TABLE 10.4 User Knowledge and Experience Profiles

User Knowledge	Low	Moderate	High
Contingency task structure (domain)	• Limited, fragmented models of domain • Very limited ability to recognize prototypical situations or interpret novel situations	• Recognizes some prototypical situations and can use reasoning to respond to unfamiliar situations.	• Rapidly recognizes prototypical situations and can intuitively interpret novel situations based on similarities to other prototypical situations.
Functional task understanding	• Adequate for most routine operations; may be too brittle to handle novel and crisis situations	• Adequate for all routine operations and some novel situations.	• Intuitively able to interpret task outcomes in novel situations.
Systems control ability	• Novice or casual user; limited, fragmented knowledge of system operation	• Competent user; understands operation of all commonly used system features; successfully learns from operational errors.	• Master (“power user”); strong, accurate mental model of system relationships between self, machine, and tasks to perform.

the tasks each performs. This system model permits the user to coordinate fluid operation of the interface such that the system operation tasks are "transparent." The user is, thus, freed from the additional cognitive load associated with system operation and is able to focus directly upon the functional tasks at hand. This level of facility is critical in situations where tasks must be performed rapidly and under pressure.

User knowledge is a function of training, experience, and level of interaction with the system. Even well-trained, experienced users who rarely interact with the system cannot maintain their fluency with the system due to this infrequent interaction. When technology plays a crucial role in the organization's mission, the information presentation and interaction routines selected for the human–computer interface (HCI) must support the anticipated variation in user knowledge across all three dimensions. Where performance reliability is critical, the HCI design must make up the deficit in the user's system knowledge. Depending upon technological feasibility and the goals set for the system, it may also attempt to address deficit knowledge of the domain and tasks. Finally, the system's HCI design should provide the means for the users to extend their knowledge and improve their performance.

10.4.6 Profiling User Tasks

The task analysis is the usual focal point of HSI requirements models; however, there is no general method for capturing and analyzing tasks that fully addresses the range of task factors and questions. The process that leads to profiling user tasks involves application of task analysis findings to the engineering—including design, development, and evaluation—of the target system. The specific features of the task analysis choices should drive the selection of a suitable method for these analyses. Stammers et al. (1990) identify a range of task analysis methods defined by such representation techniques as hierarchical, network, and flowchart methods or by such content entities as cognitive and knowledge description, taxonomies, and formal grammars. Table 10.5 adapts the definitions from Stammers et al. (1990) and Meister (1985) to depict the advantages, disadvantages, and examples for several methods that can be used to compare task analyses.

It is important to note that every cognitive task performed by the human–computer cooperative decision system and supported by system design is impacted by the user, the organizational structure and goals that define the role of the system user, and the situational environment that provides the context for user judgments and decisions. During the identification and analysis phase, the HSI team must gain sufficient knowledge about the multiple dimensions of the requirements in order to be able to model their interactions and implications for system design. The activities involved in capturing and modeling the situational context, the organizational user, and task profiles are not necessarily discrete or sequential. These analyses occur largely in parallel and often represent shifts in focus as the system evolves over time, rather than separate discrete efforts.

Modeling Tasks to Determine Requirements From the perspective of system users, the functional tasks encompass the activities that the human user performs to fulfill their roles in supporting the organization's mission. Functional tasks include not only the human–machine cooperative tasks and decision-making activities but also human–human communication activities. These tasks are separate from the system operation (user–machine interaction) tasks that constitute the focus of most traditional human factors

TABLE 10.5 Task Analysis Methods Comparison

Method	Examples	Advantages	Disadvantages
Hierarchical methods	Hierarchical task analysis (HTA); GOMS (goals, operators, methods, and selection) (Card et al., 1983); GQM (goal, question, metric) (Basili and Weiss, 1984)	• Provides systematic and complete structure for analysis • Broadly applicable • Well-documented; easy to learn and apply	• Difficulties with representation of parallel activities • Limited representation of cognitive factors
Network methods	Time event charting, link analysis (Chapanis, 1959); petri nets (Perdu and Levis, 1993)	• Models temporal order; can represent parallel activities • Analysis may be performed quickly given right data • Relatively easy to learn	• Limited applicability and narrow focus • Does not consider underlying cognitive or behavioral relationships
Knowledge description and cognitive methods	Cognitive task analysis (Seamster et al., 1997; Rasmussen et al., 1990); task analysis for knowledge description (TAKD) (Johnson et al., 1984)	• Provides methods for characterizing cognitive tasks not found in other TA methods • Consistent structure for representing task information	• Difficulty assuring completeness in highly cognitive tasks • Requires considerable analyst skill in knowledge elicitation • Expert sources may not be able to adequately verbalize knowledge
Taxonomic methods	Fleishman and Quaintance (1984) provide a descriptive survey of many human behavior and performance classification methods.	• Explicit categorization of task information; variety of uses • Well-documented; relatively easy to learn and apply	• Difficult to assure completeness and definition of mutually • Potential for inconsistent allocation of task elements
Formal grammar methods	Task action grammar (TAG) (Payne and Green, 1986)	• Rigorous, formal specification of procedural task information • Provides a mapping of tasks to actions for HCI dialog	• Narrowly focused, inflexible structure • Does not model relationships between task elements
Flowchart methods	Job process chart (Tainsh, 1985); decision models	• Models parallel user/system tasks and information flows • Well-documented; relatively easy to learn and apply	• Difficult to assure completeness and definition of mutually exclusive categories • Potential for inconsistent allocation of task elements

engineering. For example, an air traffic controller has functional tasks that include using computer-based support to track aircraft in flight and on the ground, making decisions about control options and communicating directly with the aircraft personnel. Each of these broad categories of functional tasks must be considered in the development of the computer system that supports the controller.

The early problem definition models provide an initial framework for task definition. During the requirements modeling phase, tasks are iteratively defined using a combination of top-down and bottom-up analysis methods. Andriole (1986) and Ehrhart (1993) describe a variety of task analysis methods useful for investigating both the functional and interface operation tasks in decision-aiding systems. There are a number of other resources that describe techniques for capturing and modeling the cognitive aspects of decision making (Sage, 1992; Andriole and Adelman, 1995; Klein, 1998; Senge and Sterman, 1994; Zachary, 1988). Task profiling and requirements identification activities focus on four areas:

1. Identification and modeling of the sequencing and dynamics of the tasks.
2. Identification and characterization of decision-critical information regarding the situation elements external to the system (support systems, physical environment, threats, etc.).
3. Identification of the ways that users interact with all of this information to explore situations, develop hypotheses, generate options, make choices, and implement their decisions.
4. Identification of the information presentation and interaction requirements of the alternative analytical methods proposed to support tasks and decision processes.

10.4.7 Functional Tasks in HSI

The general characteristics of the functional tasks involved in HSI are very important as they lead to identification of the cognitive characteristics of decision-making tasks. One of the principal goals of the task analysis models is to identify and characterize the key variables in the task inputs, outputs, and feedback that define the tasks and affect task performance. The characteristics of these variables and their interrelationship have implications for task allocation, flow of control, information presentation and interaction design, as well as hardware, software, and communications requirements. As the tasks and their associated variables are identified, the individual variables must be characterized vis-á-vis these various dimensions and related in order to model the dependencies, information flows, etc. The relationships defined then provide building blocks for design of information presentation and interaction design.

Functional Task Characteristics The HSI design efforts often begin with information from organizational job descriptions or the functional role models developed in the organizational profile. In addition to the individual task parameters discussed previously, the designer must develop a profile of the overall shape and flow of various tasks. This profile considers such principal human functions as discrimination, communication, and interpretation that are required in order to complete the tasks. In addition, the combination, or cumulative effect, of tasks is examined in terms of such factors as complexity, loading, pacing, and criticality. The overall complexity of the human users' tasks is a function of the number of interdependent factors or subtasks involved in the overall task. The level of complexity is highly correlated with task difficulty (Meister, 1991). Also related to complexity, the overall task load describes the demands placed on users by such factors as the number of concurrent tasks, interactions, and their sequencing. Meister distinguished task "load" from task "stress" by the absence of an element of fear or anxiety. Finally, both the pacing and criticality of task performance must be understood to assess their impacts on timing, accuracy and precision, prioritization, and attention requirements.

In addition to the overall profile of the functional tasks, the HSI design team must also discover and model the relationships between the elemental aspects of tasks, such as variables, constants, actions, and processes. A simple system model interrelating task elements allows the designer to categorize each element in terms of whether it is:

1. An input to the task
2. A response activity
3. A process output from the task
4. Feedback on action(s) that have been taken

This broad categorization helps to identify the characteristics of elements that are relevant to the task flow. To be especially useful, these elements must be defined further.

Input Characteristics The task input characteristics incorporate the concepts of triggering events (stimuli) and task information that the user must sense, perceive, attend to, and interpret in order to generate a response. For instance, the task stimuli or input information may vary over time in a predictable or random fashion. This variation can affect not only stimulus detection but also the user's ability to recognize and identify the stimulus. Stimuli with numerous patterns of variation task users' long-term memory and create additional cognitive workload as they attempt to match features against remembered patterns. The duration of the stimulus relative to the task time and other tasks occurring simultaneously has ramifications for the user's attention and short-term memory resources. When the stimulus occurs only briefly or changes while occurring, it may be necessary to store and redisplay stimuli for examination. When users can neither control nor predict the occurrence of stimuli, they may fail to detect occurrence or recognize significance. Moreover, where task-relevant stimuli are mixed with such irrelevant stimuli as a "noisy" environment, the user may fail to detect the relevant stimuli or mistake irrelevant stimuli as relevant and thereby experience a "false sensation." In addition to the added workload, an abundance of irrelevant stimuli can create confusion and seriously degrade performance (Meister, 1991). This provides strong motivation for proper provision of cognitive support for decision making (Lerch and Harter, 2001).

Response Characteristics The requirements and characteristics of the user's response are closely related to the output characteristics. For example, task allocation strategies and feedback design depend upon how often the user must respond and what the response frequency and precision must be. The difficulty of attaining these response goals becomes a function of the number of component elements incorporated in the task output unit and the output workload. Very low levels for goal attainment difficulty may affect the user's attention and interest (Meister, 1991). In contrast, very high levels of difficulty may indicate tasks out of the range of human performance. These factors also have emotional consequences in terms of motivation, frustration, and stress. The cognitive demands associated with the content of the decision-making task are discussed in the section on decision task characteristics.

Once the broad tasks are identified, the designer must look at the subtasks or procedures that comprise those tasks. For example, the number and interdependency of the procedural steps required in the response to produce one task output unit also impacts task complexity. The precision required in responding has implications for both the information presentation precision and the means by which the user formulates the response. Tasks and subtasks that must be performed more or less simultaneously create extra demands on attention and cognitive resources (Wickens and Hollands, 1999). This issue must be addressed in task allocation strategies if these are to be appropriate. Finally, in addition to task precision parameters, the designer needs to take into consideration how closely the user must adhere to prescribed procedures. Tasks requiring absolute adherence to a strict procedure may be candidates for automation. At the very least, the sequencing of valid actions will have to be controlled in the HCI design through the use of constraints and affordances (Norman, 1986).

Output Characteristics From the system's perspective, the functional tasks are the outputs of a human–system cooperative response process. For this reason, the identification and analysis often begins with desired outputs. One of the first issues to be resolved is what constitutes an output unit. An output unit may be a single task, such as the assignment of a single entity to a service unit, or a composite task composed of a number of elements or component tasks, such as would result from planning a series of activities for multiple actors. Task volume, or throughput, is measured in terms of the number of output units produced during a period of time. In some cases, the duration of the output unit is also an issue. For example, an operator may have to maintain some signal or machine state for a set period of time or until an appropriate feedback signal is received. In this case, the workload associated with task output is a function of the number of task output units produced during a set time period and the duration the output is maintained.

The HSI designer must be concerned with several issues brought about by the task output characteristics. For example, the output volume required has implications for human attention, workload, and short-term memory capacity that must be considered in human–computer task allocation decisions (Ehrhart, 1990; Gardiner and Christie, 1989; Huey and Wickens, 1993). The number and format of elements composing the output task unit have implications for the level of detail that must be addressed, manipulated, and then sent as output. The duration over which the task output unit must be maintained also impacts attention, memory, and workload by limiting resources available to respond to incoming tasks; therefore, it must be considered in task allocation schemes. Finally, the level of workload associated with output requirements affects not only the task allocation design but also impacts the cognitive resources required to maintain the level of vigilant performance required.

Feedback Characteristics Feedback during task performance informs the user on the appropriateness and efficacy of the response. In continuous tasks, feedback becomes part of the input for the next response cycle. Feedback on task performance may be characterized in terms of pacing factors such as feedback lag and the ratio of reaction time to feedback lag. When there is no feedback or feedback is greatly delayed, task performance may be impaired (Rasmussen, 1986). In addition, the absence of usable feedback impedes experiential learning (Gardiner and Christie, 1989). Delayed feedback is often misinterpreted or incorrectly associated with the wrong response causing the user to construct invalid causal models of the task and domain (Brehmer, 1987; Reason, 1990). When the user's reaction time must be faster than the feedback returned, the delay in feedback may lead to overcorrection in the mistaken belief that the response had no effect. Feedback is also important with respect to the number of subtasks involved in making choices based on feedback on the outcome of the previous response. When feedback is variable in quality or delayed, the effects propagate through a network of dependent choices making the reliability of task performance unpredictable.

Table 10.6 summarizes the impacts of the functional task characteristics. The HSI team can use information developed to raise issues regarding task allocation between user and system, team interaction designs, staffing cycles, training designs, and knowledge and experience required for effective team and individual performance.

TABLE 10.6 Impacts of Functional Task Characteristics

Characteristics	Elemental Task Features	Potential HSI Issues
Input	• Input variability • Input duration • Occurrence regularity • User's control of input	• Stimulus detection and identification; long-term memory • Impacts on attention and short-term memory requirements • Stimulus detection; attention • Stimulus detection and identification; workload and frustration
Response	• Goal attainment difficulty • Response precision • Response frequency • Number simultaneous subtasks • Number and interdependency of procedural steps • Degree of procedural • Degree of procedural adherence required	• Task complexity factor, impacts on user frustration and motivation levels • Impacts information display precision and response input mechanisms • Demand on attentional resources; response input mechanisms; task allocation strategies • Demand on attentional focus; response input mechanisms; response feedback design • Task complexity; impacts on short-term memory • Impacts on level of autonomous control extended to user; attention requirements
Output	• Number output units • Number elements/output unit • Duration output unit maintained • Output workload	• Task allocation; short-term memory; attention • Identification of appropriate level of information detail • Impacts on attentional or short-term memory resources; task allocation design • Impacts of extended vigilance on attentional or short-term memory resources
Feedback	• User's control of response lag • Feedback lag • Reaction time/feedback lag ratio • Number of choice subtasks	• Task allocation strategies • Attention; impacts on short- and long-term memory • Impact of feedback on performance quality • Impact of feedback on performance quality; short-term memory

10.4.8 Cognitive Decision Task Characteristics and Error Sources

Many authors have identified cognitive tasks in decision making. Although specific terminology varies across authors, these cognitive tasks are commonly described in terms of the following four generic activities:

1. *Information processing*—to collect and organize decision information
2. *Inferencing*—to interpret information for situation assessment
3. *Judgment*—to identify a suitable response
4. *Mental simulation*—to plan the execution of the chosen response

Each activity has further cognitive implications in terms of attention and memory demand or workload and such potential errors as biased interpretation or inappropriate use of heuristics. For example, overloading human attentional and memory resources impacts situational awareness, triggers accuracy/effort trade-offs, and influences judgment and choice strategies (Andriole and Adelman, 1995; Janis, 1989; Payne et al., 1993; Reason, 1990; Svenson and Maule, 1993). Decision task profiling helps to identify aspects of task and task sequence that must be supported in the design of information presentation and interaction routines.

As noted, decision tasks may be characterized and modeled using a variety of methods (Andriole and Adelman, 1995; Sage, 1992; Ehrhart, 1993; Fleishman and Quaintance, 1984). One of the most commonly used general models for decision making in complex, dynamic situations is Wohl's (1981) stimulus–hypothesis–option–response (SHOR) model that was initially proposed for tactical air combat decision making. The SHOR model's four generic elements, representing the aspects of the decision cycle, are subdivided into the cognitive functions or activities involved in each:

1. *Stimulus*—the detection/recall, manipulation, display, and storage of the decision data (i.e., situational context and variable inputs)
2. *Hypothesis*—the creation, evaluation, and selection of alternative perceptions or interpretations of the stimulus
3. *Option*—the creation, evaluation, and selection of feasible response alternatives to the hypotheses
4. *Response*—the planning, organization, and execution of the selected response option

The SHOR model provides a useful framework for identifying the characteristics, potential sources of error, and support requirements associated with decision tasks. Using the task characteristics identified in Table 10.7, the designer identifies potential decision task-related design issues, such as task allocation, information presentation, decision/task aiding, training, and staffing. We discuss each of these four characteristics further since they are central to the implementation of the SHOR model.

Decision Stimulus The decision stimuli constitute the primary inputs into hypothesis generation and evaluation for situation assessment efforts. The stimulus phase of decision making is concerned with initial data gathering and processing. Performance during this phase is determined by the quality of monitoring, focus of attention, and such processing

TABLE 10.7 Impacts of Decision Task Characteristics

Characteristics	Elemental Task Features	Potential HSI Issues
Stimulus	• Vigilance level required • Stimulus detection difficulty • Stimuli and decision cue detail • Qualitative vs. quantitative characteristics • Rate and volume of incoming information • Reliability and representativeness of stimulus variables	• Impacts on attention; short-term memory; task allocation and workload; staffing cycles • Stimulus alerts and display • Information display and interpretation; error characteristics • Information display and interpretation • Impacts on short-term memory; dynamic task allocation • Problem perception; judgment and reasoning errors; training and experience required
Hypothesis	• Situation novelty • Number of possible hypotheses • Decision horizon (time allowed) • Inferencing required	• Problem perception; long-term memory, judgment and reasoning errors; training and experience required • Problem perception; cognitive workload; judgment and reasoning errors; training and experience required • Attention, memory, workload, judgment and reasoning errors; aiding requirements • Cognitive workload, reasoning errors; training and aiding required
Option	• Number of possible options • Option evaluation tractability • Potential for goal shifts or conflict • Option value assessment difficulty • Outcome uncertainty	• Attentional focus; memory; information processing; judgment and reasoning errors • Memory and information processing workloads; judgment and reasoning errors; aiding design; training and experience required • Memory load; feedback required; judgment and reasoning errors; templating for rapid recognition of new goal requirements; training for adaptive response • Aiding for option understanding and comparisons; training and experience required • Feedback information requirements; inferencing requirements; aiding required
Response	• Planning required • Coordination required • Execution control required	• Memory; reasoning; decision horizon; training and aiding requirements • Memory; organizational structure; decision horizon; staffing and training requirements; communication design • Memory; organizational structure; decision horizon; staffing and training requirements; monitoring aids

activities that bring meaning to data gathered as filtering, aggregation, and correlation. In addition, performance depends upon memory of the evolving context, previous experiences, and training to identify relevance and code stimuli.

In addition to the pacing and volume characteristics of the inputs discussed in the previous section, the data inputs to the decision task must be examined in terms of their impacts on attention, memory, cognitive workload, and information processing. Situational awareness requires varying levels of vigilance depending upon the dynamics of the environment. Therefore, the attentional requirements associated with a decision task may require little active monitoring, monitoring at intervals, or continuous monitoring of the situation. The low monitoring requirements of typically stable or very slowly changing situations may result in poor situational awareness when the stimulus event occurs. When continuous monitoring is required, human fatigue can result in loss of attentional focus. Monitoring at set or random intervals incurs additional cognitive workload as the user may be required to maintain a working memory of the sequence of signals or events monitored in order to create an accurate mental model of the evolving situation. Monitoring at intervals is often involved in divided attention tasks and may require a rapid mental reorientation each time attention is refocused (Wickens, 1987). Attention is also related to the degree of difficulty in detecting the stimuli. Stimuli that are very difficult to detect, either due to inherent characteristics or the presence of other stimuli, may not attract attention during monitoring. In these cases, stimuli may require machine monitoring for detection or enhancement to facilitate perception or focus attention.

Over and above the cognitive resources demanded by the attention requirements, the pacing and volume of incoming decision data place demands upon the user's short-term memory. The designer must evaluate these impacts in terms of whether the typical memory demands exceed the capability of proposed users. At the lowest levels, the pace and volume of incoming information are manageable by the average trained user. As the demands increase, only highly motivated experts can manage the flow of information. The expert uses domain and task knowledge to cluster information in meaningful "chunks" rather than as discrete elements (Badre, 1982). At the highest levels, the volume of information overloads human ability to absorb and manipulate. At this point, machine monitoring and preprocessing is required to aggregate information into more manageable forms.

One of the key issues the design team must examine is the appropriate level of abstraction, or the proper level of detail, that is required in information presentation in order to permit the user to effectively interpret the decision data. Rasmussen and his colleagues (1986, 1994) categorize three levels of abstraction for decision inputs: signals, signs, and symbols. Signals are sensed information directly representing time–space data about the environment. Signs are indirect representations of the state of the environment derived from the pattern of physical signals. Signs serve to trigger learned behaviors or rules for response. Symbols are conceptual, rather than physical, structures that represent functional properties and relationships. Signs, or indicators, carry with them a context that triggers not only interpretation but also expectation. When the situational context differs from the learned context, as in novel situations, it may not be possible to correctly interpret the available information as signs. Symbols represent the more abstract conceptualization of domain relationships necessary in causal reasoning to interpret unfamiliar situations. Forcing users to work with information at the wrong level of abstraction can either overburden them with unmanageable detail or provide them with insufficient information to adequately assess the situation.

Conceptual foundations have been developed recently for human–machine interface design, based primarily on supporting these three cognitive levels. In general, humans use skill-based knowledge, rule-based knowledge, and formal-reasoning-based knowledge in an attempt to keep processing effort at the lowest cognitive level that trustworthy performance of the task requires. The ecological interface design construct attempts to minimize the difficulty of controlling a complex system while, at the same time, supporting the entire range of activities that specific users may require. Vicente and Rasmussen (1992) suggest that the usual approach to interface design, which is generally based on a direct manipulation interface (DMI), fails to consider that:

1. Practical problem solving can take place at various levels of abstraction in a hierarchical problem domain representation.
2. The same interface can be interpreted in different ways.
3. The way in which information is interpreted triggers qualitatively different modes of information processing, each requiring a different type of computer support.

Vicente and Rasmussen (1992) indicate that human errors may be related to: problems of learning and adaptation, interference among competing control structures, lack of resources to avoid error, and entrenched human variability. These four categories of human error are impacted somewhat differently as a function of whether skill-based, rule-based, or formal-reasoning-based approaches to performance are used. The ecological interface construct suggests that an interface design must take these factors into account if it is to be a viable aid that supports human interaction. Ecological interfaces are related to direct manipulation interfaces, direct perception interfaces, object displays, and graphics-based displays. To ensure that these requirements are met satisfactorily, interface designers must be concerned with the best way of describing or representing the complexity of the domain and the best way of communicating this information to the system operator. These concerns translate directly into more specific questions—one relating to the problem side of design and one relating to user resources for the designer of systems. They may be stated as follows: Problem side—What is the best way of presenting the complexity inherent in the problem or issue at hand? User side—What is the most effective resource that the operator has for coping with complexity and how can this be best utilized?

The reliability and representativeness of the input information affects the extent to which the variables may be understood and correctly interpreted. Moreover, when information is incomplete or ambiguous, users may focus on irrelevant information and inappropriate causal explanations (Reason, 1990). Users may be unaware that critical information is missing and need reminders or models that call attention to missing, imprecise, or ambiguous values in relevant stimuli. Strategies for analytical support and information presentation require an understanding of which data elements may vary in information reliability and how potential variation may affect interpretation.

Hypothesis Formation During hypothesis formation, the user seeks to bring an order to the information collected by creating, evaluating, and selecting a causal explanation or assessment of the possible situation that would account for the collected data. Several factors characterize the decision tasks during the hypothesis phase. First, the degree to which decisions are made in familiar or unfamiliar conditions affects the reasoning that must be supported and extent to which functions may be automated. For example, routine situations may be handled with procedural reasoning or automated to reduce workload. In contrast, decision making in highly uncertain environments requires support for interpreting unfamiliar situations. In complex, dynamic environments, human decision-making errors often stem from failure to consider processes across time, such that evolving and emerging trends are neglected and form a tendency toward thinking in causal tree representations rather than causal network representations.

The decision tasks should also be characterized in terms of the number of feasible hypotheses that commonly may be generated to explain the available information. In well-bounded domains with few possible hypothesis alternatives, situation assessment is usually performed with rule-based, procedural reasoning. Errors in hypothesis evaluation in such instances result from selecting an inappropriate evaluation rule or a flawed evaluation rule (Reason, 1990). In situations where the number of feasible explanations for stimuli may be large, users may use cognitive shortcuts to rapidly reduce complex hypothetical relationships into loosely integrated general hypotheses. In such cases, the hypotheses may never be adequately integrated for evaluation purposes, and the evaluation will be consequently flawed.

Another dimensional characteristic of the hypothesis phase tasks that must be identified is the time allowed for hypothesis generation, evaluation, and selection. Planning and forecasting tasks have longer decision horizons and do not require rapid hypothesis evaluation; however, the delays in feedback can affect the quality of the causal models used to interpret decision inputs. The shortened decision horizon in time-critical tasks increases the effects of user experience, attention, and workload. The more robust mental models developed with experience increase the user's ability to focus attention on relevant information, reducing workload to evaluate complex stimuli in shorter periods of time (Shanteau, 1992; Rouse et al., 1992, 1993). Real-time decision making may require almost instantaneous situation assessment. In addition to experience level and attention focus, decision performance may depend upon vigilance levels maintained and the speed of feedback (Edland and Svenson, 1993; Janis and Mann, 1997; Janis, 1989).

The stress associated with shorter decision horizons results in general narrowing of perceptual focus ("tunnel vision") or issue fixation, rendering decision makers less capable of dealing with multiple stimuli/issues (Helmreich, 1988; Janis, 1989; Orasanu and Salas, 1993). This tends to result in a decrease in the number of information sources used in situation assessment and the number of alternative courses of action considered. In addition, there is often a failure to critique the microdecisions that aggregate to a larger, central decision. The frequency of action or decisions increases as users feel "impelled" to action.

The nature and amount of inference that is required to interpret situational data impacts the quality of hypothesis evaluation. Situational and presentation contexts affect not only the detection of stimuli but also their cognitive interpretation. In cognitive tasks, the context in which stimuli occur appears to have greater significance than its physical attributes. For example, Lockhead (1992) found context and sequence were the primary factors affecting similarity judgments in recognition and categorization tasks. In other research, Edgell et al. (1992) discovered a context effect in the perception of cue salience for probability judgments. The sequence, or presentation order, of decision stimuli has also been found to affect their interpretation in expert situation assessment tasks (Adelman et al., 1993). In a series of experimental studies, researchers found that experts constructed different causal explanations for event sequences depending upon presentation order. The explanations provided indicated that the significance experts attached to a particular decision cue differed based upon its sequential context.

The human ability to perceive and interpret information based upon context is an essential strength in situation assessment. When decisions must be made in high-threat, dynamic environments, contextual interpretation permits the user to make accurate assessments intuitively and respond rapidly. Context, however, has also been a factor in misinterpretation and disastrous decisions. For example, the erroneous shooting of the Iranian Airbus in 1988 by the USS *Vincennes* was, in part, due to the context under which the available information was interpreted (Helmreich, 1988; Klein, 1998). Similarly, Pentagon investigations revealed that the April 1994 shooting of two U.S. Army UH-60 Black Hawk helicopters by U.S. Air Force F-15C fighters occurred when the fighter pilots misidentified the helicopters as Russian-made Hind helicopters flown by the Iraqis. Expectation may have been a contributing factor in the misidentification. The fighter pilots had not been briefed that allied helicopters would be in the area (Harris, 1994). Other cues, such as negative identification friend or foe (IFF) response and Airborne Warning and Control System (AWACS) communication, increased the expectation that the helicopters were either unknown or hostile and may have influenced visual identification.

Option Generation, Evaluation, and Selection The objective of the option generation, evaluation, and selection phase is to seek a feasible response to the hypothesized situation. Several characteristics of tasks during option generation, evaluation, and selection bear examination during task modeling. Many of the same factors affecting hypothesis generation and evaluation, such as situational context, boundedness, and tractability, also influence the performance of the option phase tasks. The number of potential responses to a situation affects the boundedness of option evaluation. Furthermore, when there are many feasible options to a situation, users may shift from option to option without sufficient evaluation or attempt to oversimplify (Janis and Mann, 1977; Dörner, 1987; Janis, 1989). Information volume and problem boundedness also affect tractability and may cause the workload in the option evaluation task to exceed human manipulation abilities. The goal variability inherent in the environment impacts option evaluation based on the rapidity and predictability of the variation and resulting option conflicts. In multistage, evolving decisions, a change in goals may supersede previous subchoices. Such shifts require rapid reprioritization and reevaluation of current options against higher level goals (Klein, 1993b). Feedback timeliness also becomes more critical as goals shift rapidly.

The difficulty of option evaluation tasks is judged by the extent to which outcome values are well understood and easy to determine. In bounded and semibounded domains with well-understood outcome values, users may employ rule-based evaluation. Higher levels of evaluation difficulty become less tractable for unaided evaluation. At this point, the decision making may be unacceptably delayed as users wrestle with the possible consequences of possible courses of action. Inference is required where outcome values are uncertain. In complex environments, the network of uncertainties rapidly becomes intractable for human evaluation, leading users to simplify with insupportable inference leaps (Dörner, 1987; Hogarth, 1987). Users may also avoid committing to any option, often waiting to see if changing events force or suggest a choice (Janis and Mann, 1977; Janis, 1989).

Response Planning and Execution The response planning and execution phase involves planning, coordination, and execution control as required to carry out the course of action option that has been selected. Plans are essentially hypotheses based on a network of causal assumptions about the sequence of steps that will bring about the desired goal. Simple responses based on experiential familiarity involve little or no planning. Skill-based control evokes reactive responses based on experiences with similar situations; rule-based control triggers procedural plans; and formal reasoning based control will usually require explicit use of analytic procedures. Moderate levels of planning feature manageable levels of effort using ad hoc or prepackaged plans. Complex responses usually require extensive planning or replanning involving the reevaluation of goals and adjustment of control structures.

Reason (1990) categorizes cognition-based plan failures that result from properly implemented action plans not accomplishing the intent that led to their implementation as mistakes,[2] that is, errors of intention. Reason suggests three basic sources of these planning failures, including: errors in the working database, such as stimulus phase errors; errors in mental operations, such as hypothesis and option phase errors; errors in the properties of the schema or a misguided act of poor planning. Reason traces these errors to characteristics of the human planner based upon limits of attention and memory and a powerful urge to accept seemingly rational explanations that bring order to complex, chaotic situations. The response coordination requirements are determined as a function of the size, complexity, and dispersion of the network of the agents that must be coordinated. These elements depend upon the organizational factors discussed previously and the time available for a response. Coordination tasks are communication intensive. The effectiveness of coordination is dependent upon experience, training, shared task and situational models, and flow of communication.

Execution control is defined in terms of the number and interdependency of the actions required in the planned response. As such, control is closely related to coordination. Multiphased, interdependent responses increase the coordination effort required to track the status of the evolving response. Moreover, the network of dependencies increases the difficulty of tracking all the possible consequences or "ripple effects" of actions taken. If feedback is delayed, it may be associated with the wrong phase and result in confusion and overcorrection (Meister, 1991). Finally additional cognitive resources, requiring attention and memory, are demanded to handle the wider range of control and potential goal shifting in multiphase responses.

10.4.9 Relating Cognitive Task Characteristics to Task Models

Investigating the situational, organizational, user, and task dimensions helps to identify the specific aspects of the decision tasks that should be considered in the design of the decision information presentation and interaction routines. The user's cognitive tasks emerge as part of describing the sequence of steps involved in performing a task or procedure. As discussed in the previous section, the tasks in decision making involve such generic cognitive functions as information processing, inference formation, judgment, and mental simulation. The situational context, organizational structure and culture, the user's experience and training, and the inherent features of the task influence each of these functions.

As the task models are developed, the designer can begin to explore the cognitive requirements involved in successful task performance. As the design team refines models of the problem domain and tasks, the issues raised may be compiled for later distillation and structuring. Figure 10.7 is based on the example TDO decision tasks model presented earlier and summarized in Figure 10.4. It indicates some cognitive support issues that might surface during requirements identification and modeling. For example, if identification of the status updates to current resources reveals a variation in the timeliness and reliability of the data, this fact must be considered in presentation of that information to the user. The reliability will also be a consideration in determining the analytical method used to track and compare the change in resources. Additionally, since current and projected status of refueling assets, missions, and available fuel are also multidimensional constructs involving time, location, and capacity/range, the combination of those dimensions must be presented in a form that is meaningful to the user and representative of the underlying relationship. The situational context involving operations of a routine, exercise, or crisis nature is also a factor in the decision to reassign, reroute, or cancel refueling missions. Where constraint-based planning tools are employed, the user will generally need support for modifying the constraints to meet operational circumstances. This is the case, since projecting changes in the complex network of tanker and fuel-receiving missions exceeds human short-term memory capabilities and requires aiding to trace the ripple effects of change across the schedule. The task models also provide means for projecting possible errors related to human performance (Fields et al., 1995; Reason, 1990).

An understanding of the TDO's tasks also suggests requirements for training and staffing. For example, the proposed operational process the new system must support incorporated assumptions about the TDO's knowledge and experience both in the air-

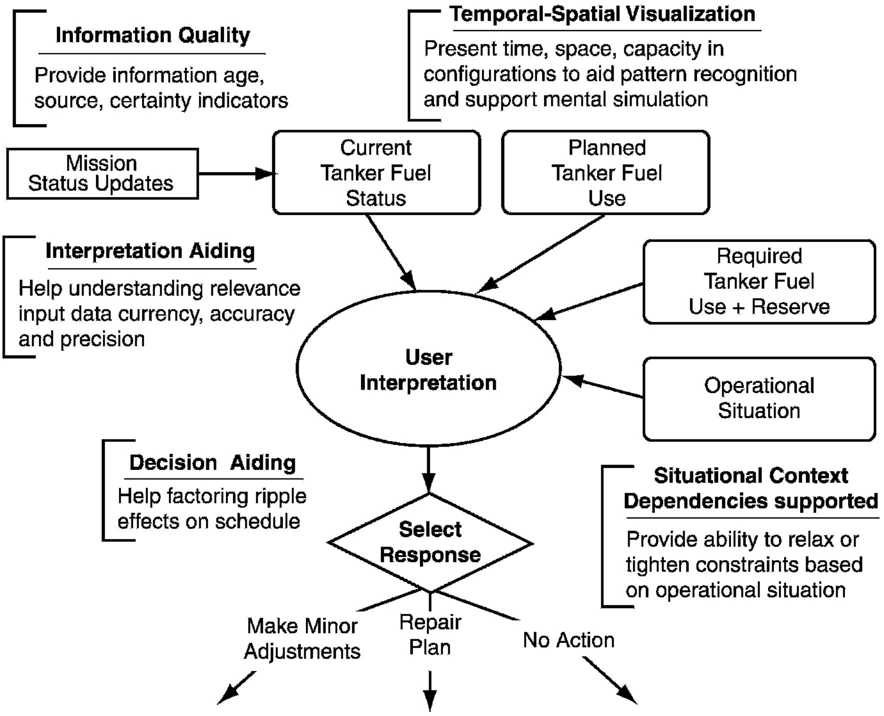

Figure 10.7 Potential impacts of user/task/context on task aiding.

refueling domain and in performing the execution control tasks. These assumptions must be feasible with respect to both personnel projections and training regime. The user will need training both on the new task processes and roles and the capabilities and limitations of the aiding technology. Figure 10.8 indicates possible training and staffing impacts suggested by the task model.

The models represented in Figures 10.7 and 10.8 provide the means to characterize various aspects of the decision domain and tasks in terms of readily observable, broad criteria. Location of the domain and tasks within certain parameters suggests possible sources of cognitive demand and user error. These potential problems are evaluated in terms of system support and expressed as cognitive task requirements. Figure 10.9 presents an example of how the issues raised while analyzing the decision task requirements suggest cognitive task requirements (CTRs).

Contemporary system design teams have a range of software tools that facilitate the creation of rich representations of task requirements. For example, modeling software that permits hypermedia links provides the means for "annotating" the basic task models with such additional models as one based on situational context, text-based descriptions, audio clips from interviews, and even field video of task performance under realistic conditions (Ehrhart and Aiken, 1990). The process of building and reviewing these models helps to identify the cognitive characteristics of each task. These cognitive characteristics, in turn, raise performance and HSI issues that should be included in the requirements documentation to assure their inclusion in the design and implementation of the system.

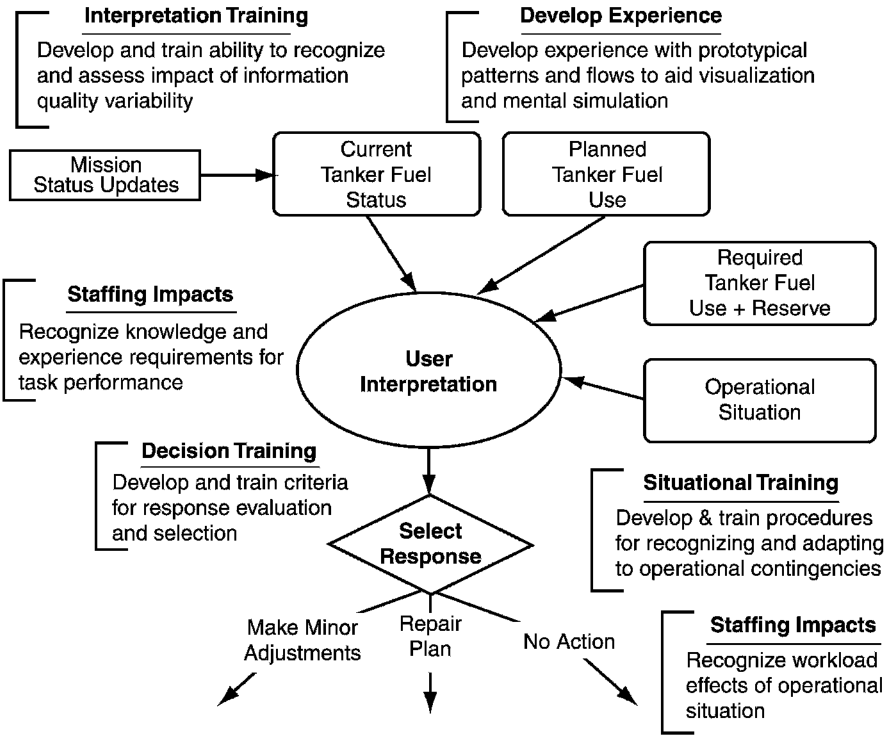

Figure 10.8 Potential impacts of user/task/context on training and staffing.

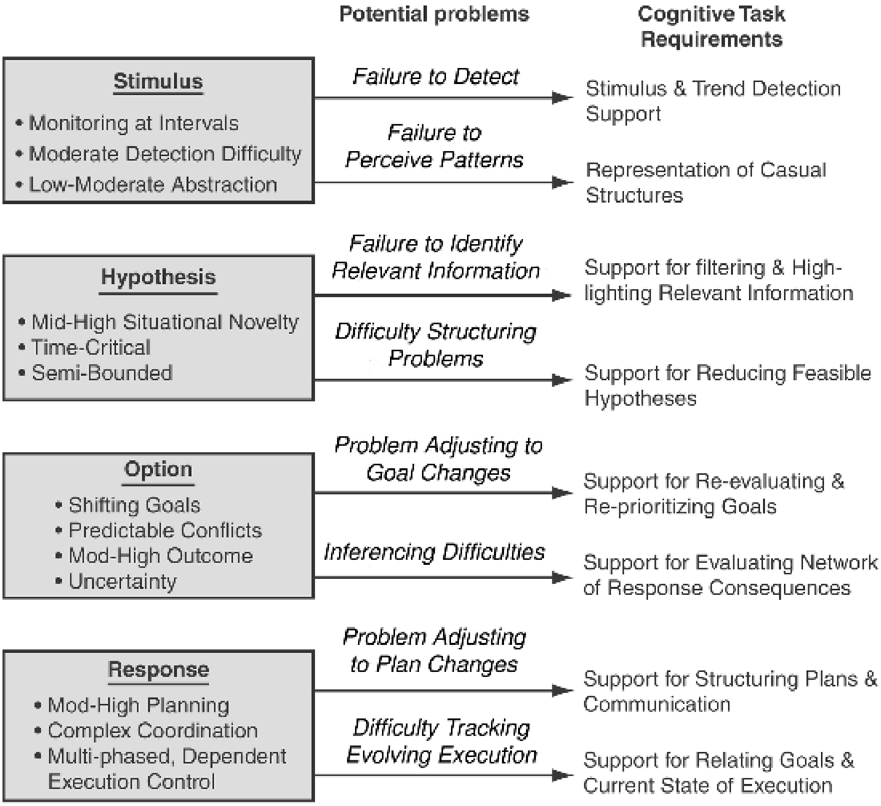

Figure 10.9 Mapping decision task characteristics to cognitive task requirements.

10.5 SYSTEM CONCEPTUAL AND ARCHITECTURAL DESIGN

During the conceptual and architectural design phase, the systems engineering team begins to interpret the cognitive task requirements in terms of cognitive systems engineering design principles, such as those presented in Gardiner and Christie (1987) and Rasmussen et al. (1994). This allows interpretation of the cognitive demands that characterize certain tasks and task situations in terms of the impacts on information presentation and human–computer interaction. When coupled with basic human factors guidelines for system design, the cognitive systems engineering design principles help to identify technological solutions that support the cognitive task requirements and which also conform to the identified hardware and software requirements and specifications. For example, selectively focusing attention is a coping strategy invoked when the user is overwhelmed by large amounts of information. This information processing strategy may be associated with such biases as fixation on one problem element or overemphasis of cues that support the current hypothesis. The design principles that address “selective attention” provide reminders of the “larger world” to avoid tunnel vision and means for directing the user’s focus to the most relevant information. The design goals for implementation of these principles include:

- Providing an overview or "establishing shot" to expand the decision-makers perspective
- Exploiting common representational analogies, such as maps and models to highlight the relationships between domain factors

As new requirements and related design "goals" are identified and understood, they can be integrated into the developing system concept. Rather than occurring in a rigid sequence, this process continues iteratively as requirements surface and prototype concepts are proposed. In this fashion, the prototype design evolves as the incarnation of the designers' hypotheses regarding the decision-making activities and interaction requirements.

Figure 10.10 illustrates formulation of design goals based on informal requirements knowledge as embodied in situational, organizational, user, and task models; formal requirements specification; and standards and guidance literature concerning cognitive systems engineering principals and human factors. The resulting conceptual design and architecture concept is a configuration of features including the information presentation methods, interaction routines, and the hardware and software technologies that support them. Each feature must be traceable to the requirements and specifications documentation. The specific incarnation of the feature and its configuration in the design should be traceable to the higher level design goals, principle(s), and guideline(s) that defined or suggested it. This dual traceability ensures that the proposed design adequately meets requirements and helps the systems engineering design team make better use of the technology options available to them.

For purposes of generalization, the discussion of design goals presented here, as well as the principles and guidelines underlying them, is restricted to the higher level design goals

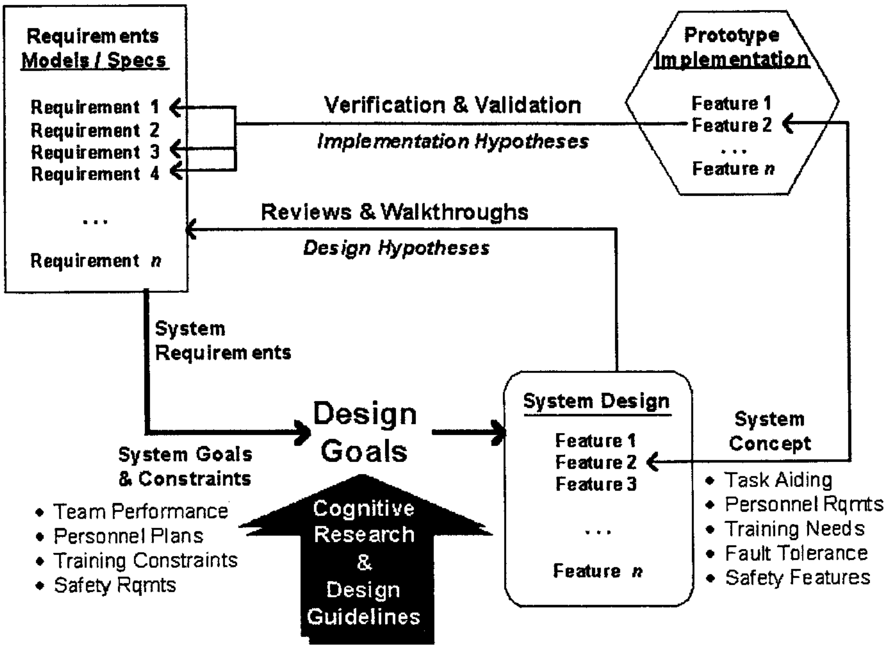

Figure 10.10 Tracing user/task requirements.

as contrasted with detailed design considerations needed to implement a physical realization of the system. The design practitioner is directed to the human factors and decision support (Sage, 1991) literature for more detailed presentations. Among more recent efforts, the following references are particularly noteworthy:

- *Principles and Guidelines* Salvendy (1997), Wickens and Hollands (1999), Sheridan (2002), Hackos and Redish (1998), Gardiner and Christie (1987), Preece et al. (1994), Rasmussen et al. (1994), Shneiderman (1997), and Smith and Mosier (1986).
- *Empirical and Experimental Evaluations* Andriole and Adelman (1995), Brannick et al. (1997), Guzzo and Salas (1995), and Svenson and Maule (1993).
- *Theoretical Foundations* Card et al. (1983), Dreyfus and Dreyfus (1986), Janis (1989), Klein et al. (1993), Meister (1991), Norman and Draper (1986), Rasmussen and Vicente (1989), Sage (1991, 1992), and Senders and Moray (1991).

The remainder of this section surveys the cognitive systems engineering principles and design guidance that relates to the situational, organizational, user knowledge, and task characteristics developed in the profiles. Each category is discussed in terms of information requirements, support for potential performance errors, and possible systems engineering design goals.

10.5.1 Design Goals Associated with Situational/Environmental Context

Vicente and Rasmussen's (1992) ecological interface design (EID) model presents two environment-related design goals based on Rasmussen's (1986) model of cognitive control. First, the interface design should not force the user to use a higher level of cognitive control than required by the task. Empirical evidence suggests that the skill-based and rule-based levels of cognitive control produce the most efficient response, provided the user has correctly interpreted the situation through possession of sufficient experiential familiarity. In addition, there is evidence that users attempt to reduce task demand by relying on the cognitive shortcuts provided by the lower levels of control (Hammond, 2000; Rasmussen, 1993; Rastegary and Landy, 1993). Second, the interface should support all three levels of control: skill-based, rule-based, and formal-knowledge-based. This goal reflects the user's requirement to operate in the multiple environments that make up complex domains.

In determinate environments, the principal design goal is providing support for users to help them rapidly select an effective response to a relatively unchanging and predictable environment (Meister, 1991; Rasmussen, 1986). In such an environment, the limited highly structured set of cause-and-effect relationships permits response automation in situations when very rapid response is required. Users generally need detailed displays that present specific values for relevant parameters, such as altitude and air speed in aircraft. When these values must be considered together, the display should either integrate them or present them in sufficiently close proximity that the user can compare the readings almost simultaneously (Vicente and Rasmussen, 1992). Interactions should be designed to allow the user to act directly on the display to manipulate the time–space signals in an appropriate and timely manner.

In moderately stochastic environments, the user needs to understand the effects of variability in some parameters and the interaction of the parameters. In some cases, the display of some individual parameter values may be integrated into a single display for interpretation as signs rather than as signals. There is empirical evidence that indicates the use of "configural displays" improves performance by allowing users to extract critical data relationships from both the low-level parameter values and the high-level constraints (Bennett et al., 1993). Woods and Roth (1988) indicate that the strength in configural displays lies not only in the economy of representation but also in the emergence of certain domain features. It is important, however, that displays representing complex domains not reduce the complexity below the level of the fundamental parameters and their interdependencies. This is in keeping with Ashby's law of requisite variety (Ashby, 1956).

In severely stochastic and indeterminate environments, the design goals focus on providing the means to make most efficient use of resources in a succession of varying, short-term situations. Users must be able to rapidly develop creative, adaptive responses to effectively exploit opportunities and avoid disasters. This requirement suggests the need for displays that represent the causal relationships and make use of goal-relevant domain models. The representation of causal networks provides externalized mental models that relieve the user of the cognitively demanding tasks involved in comprehending the causal factors underlying a situation and the network of consequences associated with options (Rasmussen and Vicente, 1989). As such, these displays help to support the mental simulation required for the intuitive response patterns suggested in Klein's recognition primed decision-making model (Klein, 1993a). Table 10.8 summarizes the design goals related to the situational and environmental contexts that the human–computer cooperative decision system must operate.

10.5.2 HSI Design Goals Associated with Organizational Contexts

Response selection and coordination within an organizational context involves synchronizing multiple perspectives, synthesizing intraorganizational information, and recognizing relevant patterns in evolving situations in order to formulate an appropriate response. The design goals associated with the organizational context focus on: responding to interdependencies of organizational structure, facilitating communication, incorporating accepted doctrine, and supporting the shared mental models required for effective organizational response.

In organizations characterized by complex interdependent structures, the performance of one unit or subsystem affects the performance of the others. The extent of this effect

TABLE 10.8 Design Goals Summary—Situational/Environmental Context

- Support all three levels of cognitive control: skill-based, rule-based, and knowledge-based.
- Support skill-based control with displays and interaction methods that allow users to directly manipulate the signal-level parameters of the problem.
- Support rule-based control with displays that map structure and constraints of environment. Model structural relationships and make domain variables salient through design and highlighting.
- Support knowledge-based (or model-based) control with domain models that help to relate problem parameters to goals. Model causal relationships and make goal-relevant information salient through design and highlighting.

may range from enhancing or degrading functional performances to a tightly coupled relationship where one function cannot be performed if the other fails. In either case, the users responsible for the performance of a function within an interdependent structure must maintain some awareness of the organizational functions that support their functional responsibility, as well as the organizational functions that are affected by their decisions. Users must consider these causal factors in contexts where knowledge-based control is used to adapt to complex, dynamic environments. Depending upon the tasks supported and degree of interdependence within the organization, design goals for organizational structure may include models that relate the dependent network of supporting functions for diagnostic reasoning and situational awareness. In addition, causal models can provide reminders of the potential consequences of decisions for other organizational functions. Finally, models may present the flow of coordination and control involved in implementing decisions within the organization.

The shifts in organizational response that occur during crisis situations present challenges that may also require attention in conceptual system design. For example, if decision making is performed in a distributed environment, the user may have to cope with failure of communication links that provide updates to critical information. The design of information presentations must provide indications of the data elements affected. The interaction design may include methods for reorganizing the display of information that may be needed when there are changes in data reliability. The system design may also have to accommodate shifts in decision-making autonomy under crisis conditions. In these cases, the standard operating procedures and channels of authorization may be replaced by a set of high-level goals and constraints, such as military rules of engagement, to permit faster, semiautonomous responses. Based upon the information gathered in requirements, the information presentation design and interaction control should be adaptable to these conditions.

Wellens (1993) presents an information-processing model for multiperson and human–machine decision making in a distributed decision-making environment that addresses some of the problems of communication design. This model incorporates the concept of communication bandwidth, representing degree of richness in communication, associated with the modes of interaction and communication that are supported by the system. For example, video-conferencing should provide all the cognitive content of face-to-face discussion but should "filter" out some behavioral facets that may be counterproductive if retained. This "filtering" is not at all due to any function of the electronic medium; rather it is due to the participants' tendency to focus on rational presentation of factual information without additional emotional behaviors. Despite the intuitive appeal of increasing communication bandwidth, Wellen's experimental research with dynamic situational awareness in team decision making indicated that increases in information richness were not always associated with improved situational awareness. This result seems to be due largely to the time pressures and additional filtering required in an information-rich media.

The design goals for supporting communications in organizational contexts should lead to an understanding of who must share information, what information and knowledge must be shared, and how it must be communicated. Knowledge management and knowledge sharing are very important contemporary research issues (Von Krogh et al., 2000; Sage and Small, 2000), and this provides yet another important dimension to the complexity of HSI issues. Within this high-level construct, information interaction design concepts should strive to maintain an appropriate distance and directness in the communication between members of the team or organization. As such, the design should facilitate the integration of users who must cooperate and not interfere with their cooperative tasks.

Much of the strength in shared mental models appears to be task, training, and communication dependent. Rouse et al. (1992) state that the current empirical evidence is insufficient to form a coherent theory of team-based design. In fact, there seems to be some evidence that technology interferes with shared mental models. For example, Duffy (1993) cites the loss of "backchannel communication" as a potential negative effect of introducing technology in team processes. The communication that occurs in the background of the primary communication provides team members the opportunity to question, clarify, and confirm their understanding of the situation. This secondary communication is a critical part of avoiding errors due to miscommunication. As another illustration of this, the investigation of the Black Hawk helicopter shooting indicated that some of the members of the AWACS team knew before the shooting that the helicopters were U.S. Army Black Hawks, but the information did not get communicated to the pilots of the F-15Cs (Harris, 1994).

Group or team situational awareness involves sharing of common perspectives between two or more individuals regarding current environmental events, including their meaning and anticipated future (Wellens, 1993). The HSI designs to support shared mental models should incorporate not only the advantages of multiple perspectives but also the power of shared knowledge and training. This shared knowledge includes doctrinal concepts and common representations of both abstract and concrete organizational information (Kahan et al., 1989). Table 10.9 summarizes the design goals associated with the organizational context.

10.5.3 Design Goals Associated with User Profiles

The user profile characterizes predicted levels of knowledge, experience, and training that the users are likely to have with respect to three knowledge areas: the domain, the functional tasks, and the operation of the system. The effects of this knowledge generally conform to models of beginners with low amounts of experiential familiarity and expertise, competent practitioners with moderate levels of these, and experts with high-level knowledge and expertise. Individual system users will typically demonstrate a range of competency across the three knowledge areas. The three knowledge levels have a number of common features, regardless of the area of knowledge involved. As with cognitive control, the predicted knowledge levels of the prototypical user must be supported for each area. Each knowledge level is discussed below with design goals for each area of

TABLE 10.9 Design Goals Summary—Organizational Context

- Provide models of interdependencies in organization to aid user in assessing causes of situations and effects of choices.
- Provide means for users to adapt to shift in organizational response during crisis situations. Encourage consideration of organizational doctrine through use of goal- and constraint-based displays.
- Facilitate all necessary and useful communication between decision participants with information display and interaction concepts that support team interaction.
- Support sharing of team or unit mental models to foster effective task coordination.

knowledge. The HSI design goals identified here were synthesized from Dreyfus and Dreyfus (1986), Rasmussen (1986), Rasmussen et al. (1994), and Senders and Moray (1991).

At the lowest level of expertise, the user may not recognize critical cues regarding the situation, task, or system state. In addition, the user usually has only limited ability to reason about the cues provided. In novel situations this limitation may induce confusion and error. The beginner often lacks confidence and may be slower to respond and reluctant to commit to action. Finally, lower expertise is associated with a limited goal framework that increases the probability of errors of intent.

Where domain knowledge is low, users benefit from displays that are formatted as accepted domain models such as to present situational information in context and to map causal relationships. Constraints, supports, and reminders help to guide domain understanding and increase confidence in situation assessment in these low-knowledge situations. In addition, templates of prototypical domain constructs with relevant cues highlighted can assist the user in making comparisons and developing responses in novel situations.

Low task knowledge often results in an inability to handle shorter decision horizons and heavy information loads. Additional time may be lost reviewing irrelevant information or inappropriate options. As a result, the beginner has difficulty maintaining performance quality under increased task workload. Lower levels of task knowledge are characterized by limited response option generation and evaluation capabilities. Finally, the beginner has difficulty prioritizing tasks. Display and interaction supports for functional tasks are similar to those discussed for low domain knowledge. To support the beginner in developing task knowledge, the HSI design should allow the user to query the constraints and affordances that have been built into the task models. Automation strategies should be explored to relieve the beginning user from excessive cognitive workload. When feasible, adaptive "intelligent" decision aids may be appropriate to filter displays and propose options. Where this type of aiding is infeasible, organizational structures may provide the same kinds of error trapping, error flagging, and redundancy afforded in machine design.

Low system knowledge is addressed in most fundamental guidance for systems design (cf. Preece et al., 1994; Shneiderman, 1997; Wickens and Hollands, 1999). Several general guidelines apply to help reduce errors and foster system learning. First, the information presentation design should provide overview screens to help users develop a mental model of the system resources available and understand where they are in a process. Moreover, the human–machine communication should make the current state of the system implicit and the available options visible. The interaction design should include built-in constraints to prevent an unrecoverable error and alert the user to the nature of their error and current response options. Finally, Norman (1986) encourages designers to make use of natural or domain knowledge in the interaction symbology in order to allow the user to interact with the task in the most familiar terms.

Moderate levels of expertise lead to performance errors based on misinterpretation of cues due to limits of the user's domain, task, or system models. Alternatively, errors can occur when the user fixates on the most available models. Moderately experienced users have limited ability to resolve conflicts between multiple models. Finally, moderate expertise is characterized by a reliance on learned procedures and a limited ability to reason at higher levels of abstraction in unfamiliar situations.

Moderate domain knowledge may be supplemented with displays formatted as accepted domain models to present situational information in context and map causal relationships.

The associated interface and interaction design efforts should support construction of more robust mental models by providing the option to view deeper levels of explanation. Since users may fail to recognize the degree and impacts of uncertainty in situational cues, displays and interaction routines are required that make the sources and extent of domain uncertainty explicit.

Moderate knowledge of the functional task requires some of the same support described for lower knowledge levels. For example, the user's task knowledge may not be sufficiently robust to understand the effects of subtask uncertainty. Displays and interaction design should help the user to understand the source of uncertainty and explore the potential effects on task performance. Moderate levels of task knowledge also benefit from designs that make task constraints and affordances visible. In high information volume situations, the user may not have adequate schema to distinguish relevant information. The system design should provide goal or decision-oriented displays in order to focus attention on relevant information and to provide strong encouragement for error control.

Moderate system knowledge is characterized by response mode errors that are based on incorrect assumptions about the current system state. For this reason, system state, available options, and similar information should be visible or available on demand. It is also beneficial to minimize the use of similar interaction sequences that vary in effect given different operational modes. Moderate levels of system operation expertise may not provide sufficient procedural information to respond to unexpected system behavior. In addition, the competent user may become lost in complex, linked sequences of displays. Overview displays and interaction routines that help the user to trace recent steps help the user maintain orientation (Woods, 1984). Interaction and interface designs for moderate system operational knowledge should still facilitate error recovery with "undo" commands and similar recovery devices. Finally, these designs should feature multiple levels of help in order to allow the user to select an appropriate presentation depth for the information desired.

The highest levels of expertise are also associated with errors in the selection and interpretation of information and judgments regarding appropriate responses. Although users have expert levels of domain knowledge, they may exhibit inconsistencies in combining situational cues. In addition, the multiple models in their repertoire may compete, with selection triggered by availability rather than reasoned choice. Experts benefit from the option to use domain model displays or customize displays and interaction routines to match their mental models. As with the moderately experienced user, experts require designs that support the continued development of mental models and that provide the option to view deeper levels of explanation. Expert users may display overconfidence in their situational interpretation or response choice. Displays that make explicit the sources and extent of domain uncertainty continue to be useful at this level.

High task knowledge may also be associated with, and in fact plagued by, overconfidence. This stems in part from insensitivity to the potential for aggregated errors in subtasks and microdecisions performed in a multistage decision fashion and a failure to revise decisions as needed in light of new information. Due to these realities, experts will continue to benefit from being presented with constraint representations that allow for error control. They will also benefit from having optional supports and reminders available during situation assessment. These may be provided in goal-oriented displays or displays and interaction routines user-customized to match their mental models. These displays also help to promote understanding the causal network of contributing causes and consequences of action. Finally, the difficulties that expert users may have in adequately considering domain uncertainties may be reduced through use of displays that make the sources and extent of uncertainty in key variables explicit.

The users with high levels of expertise in system operation can still be confounded by illogical interaction and interface designs. In general, the rules for consistent design of information presentation and interaction routines discussed for the lower levels of expertise apply to the expert. Several additional considerations apply primarily to the higher levels of system operation knowledge. For example, expert system operators are usually very intolerant of being forced to use lengthy procedures to accomplish a simple task. Thus, appropriate designs should allow the user to tailor the interface to optimize for best performance. However, when users reach expert levels in system operation, their ability to bypass some operational sequences may result in unintended actions. For this reason, experts also benefit from the error tolerant design guidelines suggested for lower levels. Table 10.10 summarizes the design goals associated with the user's knowledge and experience in the domain, tasks, and system operation.

10.5.4 Design Goals Associated with the Task Requirements

It is very difficult to generalize about tasks outside of such very broad categories as planning and situation assessment. While broad categories provide a general framework for discussing common error sources and failure modes, the details of system design remain tied to the specifics of the actual task to be supported. Woods and Roth (1988) describe cognitive systems engineering as "problem-driven and tool-constrained." In this systems-oriented view, the requirements analysis process describes the cognitive tasks and performance context and then attempts to trace the causal factors associated with both satisfactory and unsatisfactory performance. The goal of this process is to raise issues for addressing these causal factors in both the system and the associated interaction and interface design. While there remains no adequate theoretical basis for a prescriptive approach to design, the empirical literature provides some insights into broad categories of causal factors relating to such issues as attention span, memory, and workload. Unfortunately, the response of specific designs to these factors are highly task and context dependent and, thus, often do not generalize to other tasks or contexts. A full explication of the possible system design responses is beyond the scope of this work. Instead, this section focuses on the identification of high-level design goals in the support of cognitive task performance and the information presentation and interaction solutions suggested by the empirical and experimental literatures in human factors, decision science, and cognitive psychology.

TABLE 10.10 Design Goals Summary—User Knowledge and Experience

- Provide support for predicted levels of user knowledge and experience with domain, functional tasks, and system operation.
- Provide less experienced users with reminders to support performance, constraints, and recovery routines to prevent serious errors and embedded models to promote learning.
- Provide moderately experienced users with goal- or decision-oriented displays to aid in reasoning with multiple models.
- Provide highly experienced users with ability to take shortcuts and adapt system to meet response goals.

Norman (1986) defines the means by which designers and users understand and interact with computer-based systems in terms of the construction and use of multiple models. The designer develops a conceptual model of the target system that accurately, consistently, and completely represents that system. Although Norman does not discuss user or system requirements, the designer's conceptual model is built upon a mental model of the domain and task requirements that is often imprecise, inconsistent, and incomplete. The interface design presents a system image intended to convey information about system operation. The user constructs a mental model of the system based upon interaction with the system and system image as represented in the interface. The user's mental model of the system may not match the designer's conceptual model. Note also that a user's mental model of the task domain is determined by training and experience and, thus, varies in accuracy, consistency, and completeness from one system user to the other.

This work highlights the points at which the translation of requirements to design breaks down. The completeness of a designer's conceptual model of the system is a function of his understanding of the system architecture, not the requirements of the task domain. Thus, the users' mental models of the system, constructed through interaction with the system and the interface (system image), may or may not compliment their model of the task domain. The "transparency" of the interaction, that is, the degree to which the users perceive themselves to be interacting directly with their tasks, is determined by the convergence of these various models. Mismatches in interface and interaction design reduce transparency such that the user is more occupied with the operation of the system than the performance of the task.

One of the fundamental strengths of human users is their ability to conceptualize or construct mental models of causal relationships. This ability lies at the heart of intuitive and analogical reasoning. The concept of a "mental model" appears with various definitions, taxonomic structures, and applications in the cognitive process literature. Johnson-Laird (1983, 1999) discusses mental models as analogical representations for deductive inference tasks. One of the principle contributions of this work was to emphasize the semantic aspects of thought. Gentner and Gentner (1983) propose a "structure-mapping" theory to explain the cognitive processing of analogies. Their research employed protocol analysis and experimental manipulations to demonstrate the difference in domain understanding resulting from differing causal explanations of physical phenomena. Carroll and Olson (1988) review the mental model literature and offer a practical definition of mental models. In their definition, a mental model: (1) incorporates a full and detailed structure; (2) involves an understanding of what the system contains in terms of structure, how it works in terms of function, why it works that way in terms of purpose; and (3) provides a way to simulate actions mentally before choosing the appropriate action option to perform. The concept of "running" a mental model is roughly analogous to the mental simulation activity described in Klein's (1993a) RPD making model.

When used as an analog, a mental model serves as an "advance organizer" for the interpretation of novel concepts (Mayer, 1979; Mayer and Bromage, 1980). Anderson (1983) suggests that analogy and the creation of mental associations may be the only way that people learn. Bott (1979) found that users will generate their own analogies in order to explain system behavior if none is provided for this purpose. Research indicates that an inaccurate mapping between the user's model and the actual functioning of the system can increase task complexity and result in performance errors (Carroll et al., 1988). Lehner and Zirk's (1987) experimental studies involving expert system users found that an accurate mental model of system processes was key to cooperative problem-solving performance.

Moreover, the high performance attained with an accurate mental model continued even when the user's problem-solving method was different than the expert system behavior, as will often be the case.

Decision models such as Klein's (1993a) RPD making model and Rasmussen's (1986) SRK model propose conceptualization and analogical reasoning as the means by which users respond to novel situations. Similarly, conceptualization is a common factor in all four phases of the SHOR decision paradigm. The analog selected serves to reduce cognitive demand by identifying and structuring the relevant information and filtering out the irrelevant information. The mental models associated with the proposed analog then provide the means for mentally simulating the potential outcomes of the available options. Although this model appears to explain much of what makes for expert decision performance, there are several potential pitfalls. For example, the selection of an analogy may be affected by its availability in memory due to its vividness or recent experience. The selection of and adherence to an incorrect analogy may blind the user to relevant information, generally information that contradicts the working hypothesis. Subsequent mental simulations built upon these incorrect assumptions could mislead users with respect to the potential effects of their actions. Finally, the ability to "run" complex mental models is constrained by the limitations in human working memory and information-processing capability. In highly complex domains with extensive interactions among the various factors, the mental simulation required may be intractable.

The cognitive science literature presents numerous descriptive theories and empirical studies that attest to the existence of mental models and their use in judgment and decision making (Staggers and Norcio, 1993; Mellers et al., 1998; Hastie, 2001). However, there remains no systematic method for satisfactorily harnessing the power of mental models to guide the design of interactions and interfaces for decision support. Several difficulties in the practical application or manipulation of mental models negatively impact their prescriptive value in design. In practice, mental models are fragmentary and lack discrete boundaries or formalized definitions. The incomplete and disconnected aspects of a mental model permit the incorporation of contradictory, nonrational, and invalid concepts. Furthermore, mental models of rarely used systems or procedures can deteriorate over time due to forgetting.

In complex, dynamic environments, the interaction models required for human–computer cooperative decision making must assist the user in maintaining situational awareness and understanding the short- and long-term consequences of decisions. This implies a framework of models in the mind of the user that must be represented in the interaction and interface design. These include:

- *Task Interaction Models* Representation of the current state of the target domain (situational awareness) means for acting on the domain (task variables) and means for predicting the consequences of actions on the domain (outcome simulation).
- *System Interaction Models* Representation of the current state of the system and the means to understand the actions required to perform tasks using the system.

Carroll (1987, 1997) proposes a structured methodology for designing effective interface metaphors that provides a useful starting point for developing interaction models. Extending this method to the design of interaction and interfaces for decision aiding suggests the following basic activities:

- Identify potential task domain models, such as network models for route planning.
- Describe the match between models and the domain in terms of user task scenarios, such as constraints and affordances implied by the analogy.
- Identify the potential mismatches and their implications, such as identification of the gaps or breakdowns in the analogy.
- Determine the appropriate design strategies to help users manage unavoidable mismatches.

This task profile characterizes functional tasks along four dimensions (input, output, response, and feedback) and decision tasks in terms of four decision phases in the SHOR paradigm (stimulus, hypothesis, option, and response). Although Table 10.11 also suggests potential impacts of these dimensions, coherent design for information presentation and interaction cannot be derived from the assemblage of individual "fixes." Woods and Roth (1988) refer to this as the "prosthesis approach" to design. In contrast, they suggest that the design goals of cognitive systems engineering focus on extending the human user's conceptual abilities. This concept is consistent with Zachary's (1988) approach to the design of the knowledge representation, data management, and analytical methods for decision support systems. Toward this end, the tables serve to identify the cognitive areas that the individual characteristic may impact, such as attention and situational awareness, and supports making suggestions regarding the appropriate design features that address these areas.

The human user's ability to meet the cognitive demands of decision tasks is determined by both the quality of their conceptual skills and the cognitive resources, such as attention and memory, that they bring to the task. Design goals for supporting decision tasks fall into two general categories: those related to enhancing human users' understanding, and those related to reducing the negative effects of human cognitive limits. These categories are actually two sides of the same coin, rather than distinctly different constructs. The first category involves what Woods and Roth (1988) term "conceptual tools." The conceptual tools are those features of the design that enhance the user's ability to structure the problem, formulate the goal, select a solution path, and implement the selected response. The second category addresses the limits of human attentional and memory resources that interfere with effective use of the human cognitive strengths that aid conceptualization. The attempts of a system user to cope with their own cognitive limits often result in erroneous problem formulation and option selection. On the other side, enhancing conceptualization reduces certain aspects of cognitive demand and, thus, reduces the

TABLE 10.11 Design Goals Summary—Task Requirements

- Structure problem representation to enhance the user's perception and understanding of values and relationships between key task variables.
- Provide multiperspective conceptualization aids that make abstract or nonvisible concepts and relationships visible.
- Provide decision- or goal-oriented perspectives for organizing and prioritizing tasks.
- Support mental simulation with representations of network of problem dependencies for situational assessment, option evaluation, and response coordination.
- Allocate tasks between user and computer to reduce cognitive workload and support human user's adaptive, intuitive cognitive abilities.

cognitive resources required. The HSI design goals proposed here also play such dual roles.

One of the most effective means of enhancing a user's conceptualization is to structure the problem representation to highlight the values and relationships between the relevant task variables. Woods and Roth (1988) propose that the extent to which designers can successfully structure representation is a function of three factors:

1. The designer's ability to anticipate the decision tasks and situational variables
2. The characteristics of the representation that influence decision performance
3. The degree of domain variation in the relationship between key criteria and decisions

Task analysis helps to identify the decision variables and map their relationship in the decision process. Representation impacts the user's ability to monitor, perceive, combine, and relate data to assess the situation and formulate an appropriate response. In this way, structure of the representation makes the semantics of the domain visible, such as obtained in the ecological interface designs proposed by Vicente and Rasmussen (1992). The third factor addresses the issue of representational economy and variety. In complex, dynamic environments the users' requirements for problem views may change given a variation in the situational context, such as from routine operations to crisis. In addition, representational structures may have to provide multiple perspectives on the problem.

The use of representations to aid conceptualization lies at the heart of several approaches to structuring decision variables. Treu (1992) presents examples of several structural primitives and composite structures. Each primitive is considered with respect to its effects on cognition and memory and its representation in computer-based systems. For example, node and arc structures may imply paths, scripts, spatial location, or distance. When combined with the vertical hierarchy primitive that suggests concepts of rank, ordering, and levels of abstraction, the composite structure conveys a tree or object hierarchy.

Configural or integrative displays combine the low-level syntactic data to form a high-level semantic representation. The goal of integrative displays is to facilitate the user's holistic perception of domain or situational features that are not apparent when the data elements are separated. Initially, the concept of configural displays focused on the benefits of data proximity in object displays (Carswell and Wickens, 1987). It appears to be the "emergent features" of the configural display, rather than the mapping to a recognizable object, which determines the benefits in performance (Sanderson et al., 1989). The value of integrative displays has been questioned where users must also attend to individual data elements (Bennett and Flach, 1992). Bennett et al. (1993) empirically demonstrated the benefits of configural displays to promote extraction of both high-level and low-level data.

In their EID method, Vicente and Rasmussen (1992) also incorporate integrative displays derived from an abstraction hierarchy of the work domain. Based on improvements in decision performance in an experimental study, they suggest that the representation based on the abstraction hierarchy provides a better match to the user's mental model of the work domain. These findings support the benefits of multiple problem perspectives for decision making. Support for the multiple perspective design also applies to the research in integrative displays. Coury and Boulette (1992) investigated the effects of configural displays on diagnostic tasks in conditions involving time pressure and uncertainty. Their findings suggest that accurate and timely situation assessment under all conditions of time stress and uncertainty requires both integrated and separated displays.

In cases where representation requires multiple screens, Woods (1984) proposes that the integrative construct is the level of "visual momentum" the information presentation and interaction supports. When visual momentum is low, information processing occurs in a series of unintegrated data views, and this requires the system user to reorient and search for relevant information in each view. This may add considerably to the user's cognitive workload and, as a result, may degrade performance. Woods suggests several structural features to increase visual momentum, including the "long shot" or overview display, perceptual landmarks, display overlap, spatial organization, and spatial cognition. Overviews, landmarks, and overlap provide information about the location of one display with respect to another and support the multiple perspective aspects of the Rasmussen (1986) abstraction hierarchy. Spatial organization uses such spatial orientation entities as hierarchies, paths, and maps to serve as preorganizers and aids for exploring the domain or situation. Spatial cognition refers to the use of analogical representations to provide a map to all the features of the underlying process. Woods suggests that increasing visual momentum reduces mental workload, improves data sampling behavior and identification of relevant information, and improves the cooperation between the human user and the computer-based support.

Representation structure also affects the cognitive demand associated with decision tasks. Norman et al. (1986) present strategies for cognitive layouts in windowing designs that address selective attention, multicue integration, and variable levels of cognitive control. The layout of windows to reduce the demands of monitoring multiple activities is based on a model of attention that suggests that attention works as a dynamic filter. When multiple signals are present, attention is focused on one signal while the remaining signals are attenuated. The shift of attention may be voluntary, such as when the user actively searches for necessary information, or involuntary as in the case of flashing alarms. Spatial layouts for multicue integration may achieve all or part of the integration, possibly through use of integrative displays, or much of the necessary integration may be left to the user. Although integrative displays are quite powerful, effectiveness depends upon the extent of domain-criteria variability. Leaving integration to the user is the most flexible approach for the designer but unfortunately places the burden of integration entirely on the user. Finally, the use of spatial layouts may be used to provide multiple perspectives of the problem or domain from the syntactic to the semantic, in terms of signals, signs, and symbols.

Another approach to representation is a decision-oriented or goal-oriented display. In decision-based representation, displays present problem information structured such as to aid interpretation. Similar to the concepts in integrative displays, these display paradigms shift much of the cognitive demand in data integration and interpretation from the human user to the computer. In contrast to data-oriented displays that present all available information, decision-oriented displays provide only the information that is relevant for the decision task. In an experimental study involving multiphase decisions in a complex, dynamic environment, MacMillan and Entin (1991) found that decision-oriented displays resulted in faster decisions with fewer errors. Goal-oriented displays represent the domain or situational structures that relate to desired goal or system state. These can take the form of goal and subgoal hierarchies or diagrammatic views of the system or process. Kieras (1992) employed diagrammatic displays for diagnostic tasks in control system management. Experimental investigations indicated that the causal structures and one-to-one mapping of component state to the diagram produced better diagnostic performance than the more traditional representation in which the diagram and component state values were separated.

Goal-oriented representation may also be used to support the mental simulation required to identify causes for situation assessment, evaluate consequences of options, and plan for response coordination and implementation. Woods and Hollnagel (1987) present a methodology for constructing goal–means networks that incorporate the task goals, functions (the means to achieve goals), and requirements that instantiate new goals based on what the function needs to accomplish the higher goal. Woods and Roth (1988) propose goal-oriented displays for evidence processing, situation assessment, and planning. Bainbridge (1988) discusses problem representation as structure–function and goal–means networks. These graphic representations use hierarchies and cause–effect links to support pattern recognition, planning and prediction, and semantic organizers. Bennett et al. (1997) describe the representation aiding approach to display design in some detail and provide a useful tutorial as well as a description of four alternative approaches: aesthetic, psychophysical, attention based, and problem solving and decision making.

Mental simulation is important not only for time-pressured situations but also where feedback is delayed due to the inherent response latency of the system. Hoc (1989) describes the problems that are unique to long response latencies. In such environments, the diagnosis and response to changes in a system cannot be affected in direct cause-and-effect relationships. The user cannot directly manipulate the goal variable but must manipulate it indirectly through causally related variables. Planning is complicated by uncontrolled and unanticipated interventions in the causal network and long delays in response effect and feedback. The mental simulation required for planning in this context rapidly becomes intractable without appropriate aiding.

Ball and Ord (1983) present a graphic planning aid to support the mental simulation required to predict the consequences of options in an air traffic control task. Their aid presented two problem views: the current situation with radar and a predictive display of the planned response. Bell and Ord emphasize the problem of dealing with the multiple realities of the present and predicted situational displays. Their planning aid handled these representations as discrete displays and featured both manual and computer-generated options. Experimental studies with air traffic control teams revealed decision-making problems associated with requiring the controllers to relate information from both the planning and situation displays. This design forces the user to choose between maintaining situational awareness and evaluating the consequences of his or her response.

In a cooperative decision system, the design of the cooperation in the allocation of tasks between the human user and the computer-based support is a key factor in reducing the user's cognitive task workload. This is most often accomplished by automating the attention-intensive monitoring tasks; rapid, memory, or computation-intensive tasks; or time-constrained response tasks. Task allocation also attempts to assign to the human user those tasks, such as inference and judgment that involve adaptive, intuitive, cognitive abilities. For example, in Ball and Ord's (1983) air traffic control aid, the human controller and computer shared responsibilities for monitoring and planning. The activities within those responsibilities were allocated based on the different strengths of the human controller and the computer. The human user's tasks involved pattern recognition and maintaining situational awareness; the computer was assigned responsibility for continuous updating of the situational data and detailed trend analysis.

Most cooperative decision-making task allocation strategies involve some form of static allocation where some or all of the tasks are directly assigned to either the human user or the computer support. For example, Ball and Ord's air traffic control system featured static allocation. Other research in cooperative decision making also features dynamic task allocation (Andriole and Adelman, 1995; Andriole and Ehrhart, 1990; Vanderhaegen et al., 1994). Vanderhaegen et al. (1994) also present a design for human–computer cooperative decision making that involves a dynamic activity regulation strategy based on a model of "horizontal cooperation." The concept of horizontal cooperation attempts to avoid the negative performance effects sometimes encountered with the passive user in vertical, or master–slave type, cooperation tasks (Roth et al., 1988). Horizontal cooperation places both the human and computer on the same hierarchical level and allows explicit and implicit dynamic task allocation in much the same fashion as the human–human cooperation in team decision making. One particularly intriguing feature of this design is the dynamic task demand estimation capability modeled on workload and performance assessment. Rather than attempting the more subjective task of estimating mental workload, the task demand estimator employs a weighted additive model of the functional task decomposition. In the work cited, expert controllers determined weights empirically. This task demand-modeling concept would appear to also have utility in the determination of task loading during the design phase.

This section has presented several models for interpreting the task-related aspects of HSI-related design. The high-level decision task goals proposed provide:

- A starting point for integrating the situational, organizational, and user goals
- Signposts to the determination of the more detailed design goals associated with the specific tasks

Table 10.11 summarizes the design goals that support the decision task requirements. The next section discusses implementation of the current HCI design concept in prototype form for evaluation and iterative modification.

The high-level design goals need to consider the effects of situational context, organizational context, and user knowledge and experience on the cognitive requirements of the tasks that the human–machine cooperative system must perform. Each high-level goal maps to a deeper layer of task and situation-dependent design goals. In essence, these goals provide a checklist for design, implementation, and evaluation. The design concept is a configuration of information presentation and interaction strategies that represent the designer's resolution of these high-level and specific design goals.

10.5.5 Evaluating Designs for Usability

One of the most often used terms in human–machine-related areas, especially with respect to design evaluation, is "usability." Narrow definitions of the term limit usability to the mechanics of operating the interface. Nielsen's (1993) usability heuristics exemplify this narrow definition. In a somewhat broader definition, usability may be seen as the measure of the system design's ability to support the user in accomplishing their tasks (Mayhew, 1999). This model of usability incorporates the interface operation tasks as a subset of an overall measure of the effectiveness and ease of use of the system.

Several researchers have proposed the use of so-called discount usability evaluation methods to identify areas for improvement early in design (Nielsen, 1993; Wright and Monk, 1989, 1991). Nielsen's (1993) *heuristic evaluation* approach is based upon such accepted design guidance as "use simple and natural dialog and also provide adequate feedback." It converts such heuristics into checklists of nine usability properties. Heuristic evaluation may be performed by 3 to 5 evaluators and does not involve interaction with users. Empirical evaluations using as many as 77 evaluators indicated that aggregating the responses of as few as 5 evaluators resulted in the capture of 55 to 90 percent of usability problems (Nielsen and Molich, 1990). This research also pointed out the relatively poor performance of individual evaluators. The fundamental limitation of Nielsen's heuristics is their focus on design aspects of interface operation. As designed, the checklists do not provide the means to examine the extent to which the design addresses the cognitive task requirements. A recent work by Henneman (1999) discusses skills (human factors, multimedia interaction design systems engineers) and tools (design laboratories, usability standards, and guidelines) that support a user-centered process for engineering usable systems. The author's conclusion that the key to improving the usability of new systems and products lies in the development staff and the organizational environment in which they work, appears undeniable.

10.5.6 Evaluating System Designs for Reliability

To be effective, systems must be reliable. From the user's perspective, this means the system is available upon demand with current, accurate information. The best case is 100 percent reliability; the worst case is multiple failures in critical systems. The most likely case is that there will be some disruption of services and delays in information updates. Systems fail in a variety of ways. Van Gigch (1991) lists five types of system failures:

1. Failures of structure and control, which often results from reliance on faulty controls built into the structure of the system or expecting other parts of the system to catch mistakes or take care of problems
2. Failures of technology, due to technology that does not perform as expected and that provides incorrect, incomplete, and/or imprecise information
3. Failures of decision processes through flawed assumptions and information-processing biases that effect judgment and choice
4. Failures of behavior, which generally occur through doing the wrong thing
5. Failures of evolution, due to rigid, nonadaptive human behavior

Design for reliability is, as a consequence of these failures, very important. Pecht (1995, 1999) identifies eight tasks, each requiring full systems engineering and systems management commitment:

1. Define realistic system requirements.
2. Define the system usage requirements, including the environment in which the system must operate.
3. Identify potential system failure modes and mechanisms.
4. Thoroughly characterize the system component materials and the manufacturing and assembly, or integration, process.
5. Design reliable systems within constraints posed by these.
6. Certify these processes.

7. Monitor and control these processes.
8. Manage the life-cycle process for the system to be engineered such as to improve reliability, quality, and cost effectiveness of the system to be engineered through use of this process.

The classic engineering response to reliability issues is often to build in "graceful" degradation so that failure of one subsystem does not propagate and lead to multiple failures. Information about the effects of outages in such systems is often provided in cryptic form for system administrators, but the system users are left to fend for themselves. Users need clear, understandable information about the extent to which their current information may be impaired by system outages or delays. As an illustration, it is often very helpful to provide appropriate feedback such as to reduce negative impacts on decision-maker confidence of:

- Information currency indicators
- Summary of update times and content
- Overview diagrams of systems that are affected by delays and failures

Operators may need assistance in identifying what information must be restored to bring the system up-to-date. Finally, decision makers need to be alerted when systems or networks are unavailable.

Ideally, this information would also be represented in the certainty factors for information in dependent systems. For example, if the intelligence systems supporting the enemy situation displays were impaired, the predicted or last known location could be displayed with a change in the icon that indicated its position was not based upon direct sensing or recently updated information. Without those uncertainty indications, the user may misinterpret the data provided. There are a large number of variables that affect and impact system reliability. The interaction between reliability and the closely related subject of maintainability is of major concern in implementing trustworthy systems of all types.

10.6 PROTOTYPING AND IMPLEMENTATION

A prototype is a physical manifestation of the configuration of information presentation and interaction methods, and functional capabilities and technologies, which have been proposed in the system conceptual design and architecture as potentially satisfying the user requirements for the system to be engineered. Developing prototypes during the early phases of system development provides a low-risk means for evaluating both the conceptual design and architecture and the system implementation hypotheses. At each stage in the system development effort, the represented design can be reviewed against the current version of requirements. In this way, sponsors and operational users can respond to the prototyped design to refine the requirements base and assess the utility and usability of the proposed interface for the decision tasks. Prototypes vary widely in scope and definition, from preliminary paper storyboards to functional interfaces to data. The choice of the form of prototype depends upon the questions that must be answered at the current phase of system development. For example, early in the development a prototype may be no more than a set of sample screens sketched on paper or a cardboard mockup of a control panel. More commonly, the term “prototype” is applied to early functioning versions of software and hardware.

10.6.1 Prototyping Design Concepts

Assessing the appropriateness and effectiveness of the proposed system to support the complex interactions among humans, equipment, environment, and information within the organization often requires some form of interactive prototype. Using an interactive prototype also provides useful insight for the overall development effort. The HCI design embodies most of the system concept that is “available” to the user to guide his or her mental model of the system. For example, the HCI design incorporates such critical system design factors as:

- The representation of information regarding the situational elements external to the system, such as environment and external threats and opportunities
- The representation of system states and feedback to the operator based on results of actions taken
- The allocation of tasks between the human user and the physical system as determined by the dynamics of the situation and the requirements of the analytical methods selected to support decision processes
- The modes in which users may interact with all of this information to explore situations, develop hypotheses, generate options, select among alternatives, and implement their decisions

In a requirements-driven design process, the judgments and decisions made during each phase determine the objectives of the analyses and evaluations required to support those decisions. Table 10.12 presents the relationship of prototyping objectives and the associated scope and boundaries of the prototyping effort. During each phase, the system design is considered in the context of the organizational and environmental factors that impact performance; however, these factors are represented at varying levels of detail depending upon the phase requirements. For example, during the problem definition phase and early in the requirements identification, the system design in question is modeled at a relatively high level of abstraction. The desired performance is expressed primarily in qualitative terms; the nature of the interaction with other support systems and the external environment is modeled in very low detail. As development proceeds to later phases, the specification of requirements increases in detail with respect to the system itself and its interaction with other systems in the organization and external environment. This specification, in turn, dictates the inclusion of more precise quantitative and qualitative analysis to assure the system design meets both engineering specifications and organizational requirements.

10.6.2 Prototyping Strategies

The software engineering and information systems development literatures suggest a wide variety of approaches to prototyping (cf. Andriole, 1990; Arthur, 1992; Connell and Shafter, 1989; Sage and Palmer, 1990). The selection of prototype form should be based on the goals of the current development phase and the information that must be derived

TABLE 10.12 Prototyping Goals for System Development Life-Cycle Phases

Design Phase	Prototyping Objectives	Prototype Characteristics
Problem definition and requirements identification	• Determining desirable system and HCI characteristics • Determining existing system capabilities and deficiencies • Selecting "best" of alternative system definition	• System represented at high level of abstraction • Qualitative analysis • Organizational, environmental interactions represented in minimal detail
Requirements specification and design	• Developing requirements specifications and design alternatives • Determining "best" design	• System represented in moderate to high detail • Qualitative and quantitative analyses • Organizational and environmental interactions modeled in moderate detail
Implementation	• Determining whether developmental prototype meets specifications • Providing feedback on detailed design	• System modeled in moderate to high detail • Qualitative and quantitative analyses • Organizational and environmental interactions modeled in moderate to high detail
Testing and evaluation	• Determining whether proposed design as prototyped meets system and organizational requirements	• Qualitative and quantitative analyses • High detail in system and context modeling

from the prototype. Nielsen (1993) identifies the trade-offs in prototype implementation in terms of depth of functionality (vertical prototyping) versus breadth of features (horizontal prototyping). Vertical prototyping is used in "functional" prototypes that permit the user to interact with real information; however, only a narrow range of system features is represented. In contrast, horizontal prototyping permits the presentation of the full range of system features but without the functional capability to interact with real data.

Another common prototype classification involves the extensibility of the prototype. "Throw-away" prototypes, such as paper storyboards and mockups, are used in early definition phases often before the target hardware and software have been identified. The name conveys a pejorative image of sunk costs; however, the throw-away prototype facilitates communication between development teams, system designers, sponsors, and end users. The information gathered not only contributes to design but can also be used to develop instruments for the evaluation phases. "Evolutionary" prototypes involve incremental development that attempts to represent the breadth of the system with functional depth evolving incrementally. The term, rapid prototyping, is generally used to refer to an evolutionary prototype. Interactive storyboards are commonly used as throw-away prototypes. In situations where commercial off-the-shelf (COTS) programs and computer-aided systems (software) engineering (CASE) tools may be used for development, interactive storyboards become the early forms of rapid, evolutionary prototypes. These four general approaches to prototyping are discussed in further detail below and summarized in Table 10.13.

TABLE 10.13 Prototyping Procedures

Method	Advantages	Disadvantages
Paper storyboards	• Low-cost, low-risk method for exploring requirements. • Scenarios can be reused for later evaluations of design. • Storyboards and scenarios can later be incorporated into interactive storyboards.	• Verbal descriptions in scenarios are not as vivid as visual representations. • Paper storyboards support very limited exploration of interaction. • May have less utility in identifying potential human errors.
Mockups	• Low-cost method for verifying physical layout of custom interaction hardware. • May be useful in simulating environment for exercises where full interaction is not required.	• Limited to representing surface features. • Full capture of ergonomic aspects of performance requires more expensive representation (pushable buttons, turnable knobs, etc.).
Interactive storyboards	• Useful for refining requirements and identifying potential human errors. • Provides low- to medium-fidelity environment for performing usability trials. • May be developed with low to moderate cost using COTS software.	• Will not identify throughput or information overload problems associated with data volume. • Designers must be careful to present only *feasible* design options within given hardware/software constraints.
Integrated rapid prototyping	• Useful (within limits) for evaluating performance with actual or simulated inputs. • May help prevent premature "freezing" of design.	• Moderate to high cost (some costs reduced when CASE tools provide easily modified prototypes). • Increasing fidelity is costly.

Paper Storyboards Paper storyboards provide a relatively low-cost, low-risk method for getting a preliminary feel for how the system would be used in terms of typical tasks and situations. Storyboards may be annotated, reordered, or even redesigned during requirements definition interviews. Paper storyboards are limited to representation of a set scenario with little possibility of exploring the range of interaction possible with the given design. The technique presents the sequence of screens but does not capture potential interaction errors or the cognitive workload associated with a particular design. These aspects are better addressed with interactive storyboards.

Mockups Mockups encompass a variety of nonfunctioning physical representations ranging from cardboard models of single control panels to full-scale control centers with turnable knobs and flippable switches. They are primarily used for studying the ergonomic impacts of equipment layout of physical task performance. In many cases, physical mockups are unnecessary for studying the implications of system designs since most of the visible features of interest are incorporated in interactive storyboards or prototype systems.

Where custom interaction hardware is required for user input or users must perform other physical tasks while operating the system, mockups assist in doing early evaluations of the potential workload associated with the set of system design alternatives.

Interactive Storyboards Interactive storyboards serve as a powerful means for exploring design alternatives without incurring the expense of developing a working prototype. This is particularly advantageous when the investigation is focused on evaluating several advanced interaction technologies rather than supporting the design of a specific system. Interactive storyboards are also useful for working with experts or end users to refine requirements. Subjects interact with a computer-based storyboard simulating the actual operation of the system. Interaction may take the form of informal exploration or subjects may be presented with tasks to perform using the simulated system. In the latter case, the storyboard provides a low to medium fidelity environment for assessing usability and identifying potential human errors. Verbal protocol methods may be used to elicit the cognitive processes involved in the interaction.

Where storyboards are used in requirements definition and refinement, care must be taken not to present something in storyboard form that is infeasible within the technological and resource constraints likely to be present in the operational working system. Although this method can be used to identify problems with cognitive workload due to the allocation of tasks between the human operator and the physical system, it may not task the overall system sufficiently to enable delineation of user or computer performance problems related to throughput or information overload. These issues may be addressed with operational prototypes that accept real-time data.

Integrated Rapid Prototypes Although developing prototype versions of a system is not a new concept, until recently software prototyping tended to be restricted to semioperational *beta* versions of systems under construction. As such, they represented a considerable investment in time and effort, and major changes to the design were highly discouraged. Furthermore, it was not uncommon for a cost-conscious sponsor to stop development with the prototype. If the prototype offered most of the functionality of the completed system, the sponsor would take delivery on the prototype and cancel further development. Similarly, if the prototype indicated major problems with the design or development effort, the sponsor might consider it good management to cut his or her losses at that point. For obvious reasons, developers grew reluctant to show prototypes to their clients.

The introduction of fourth-generation languages and CASE tools dramatically changed the role of prototyping in system design and development. Using the toolboxes provided in COTS system prototypes with complete interactive displays using windows and pull-down menus can now be developed very rapidly for UNIX, DOS, Macintosh, and other environments. This rapid development capability and the corresponding ease with which the software may be modified or even substantially redesigned or reengineered, makes it possible for designers to develop and use prototypes during the earliest phases of design. These early prototypes provide many of the features of interactive storyboards while reducing the possibility of presenting the user with an infeasible system concept. Nevertheless, until the system is tasked with the full volume of data expected in the target setting, actual system performance and its impacts on the users would not be fully apparent. This has important implications for reliability and validity of system and design evaluations.

10.7 SYSTEM EVALUATION

With the growth of interactive computing and its application in support of complex decision support systems of humans and machines, conceptual design prototyping has become an important tool in capturing and analyzing user requirements. Figure 10.11 illustrates potential prototype evaluation benefits. In iterative design and development processes, prototype evaluation aids in verifying and validating the working design against the requirements. Each prototyping phase culminates with some form of evaluation, and the evaluation goals vary depending upon the current development phase. Early evaluation provides a means for extending requirements and task analyses to the evaluation of the procedures embedded in the current design solution. In this manner, evaluation provides a means for acquiring information about the current version of the system design with respect to the performance characteristics and capabilities of the human–machine cooperative decision system. Finally, this process of iterative design, prototype implementation, and evaluation supports the project management planning and control processes that ensure the overall development effort stays on track both with respect to the delivery of a quality product and within the cost and schedule parameters. The cost effectiveness of incorporating an evaluation method depends not only upon the size and complexity of the project but also at which point during development the prototype evaluation is conducted (Mantei and Teorey, 1988).

Information feedback during prototyping enables iterative and evolutionary system development course correction. Early evaluation allows design modification during the initial life-cycle phases when the cost to modify a system is much lower than it will be at later phases. For the systems engineering development team, evaluation is also a discovery process. Findings from the evaluation provide input for requirements and design modification and help to set MOPs and MOEs (Sproles, 2000, 2001), benchmark targets for later system-level evaluations. Evaluation feedback informs not only the design of the

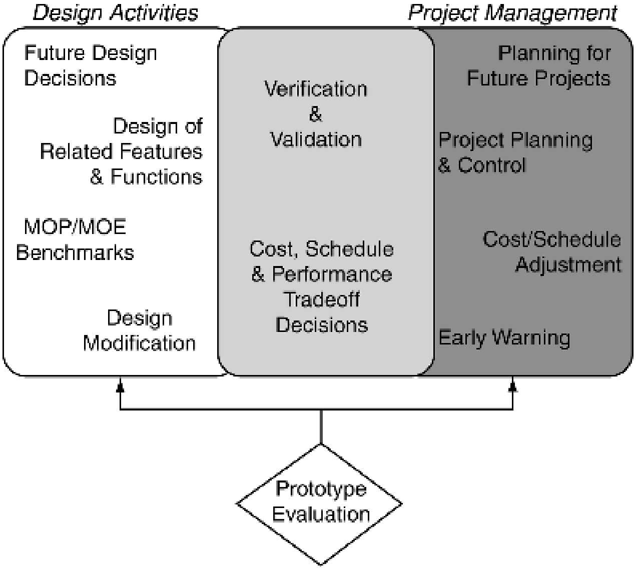

Figure 10.11 Benefits of prototype evaluation feedback to development.

particular functions and features considered, but also provides input for the design of related components. For the project manager, evaluation feedback is a critical part of project planning and control. Early evaluation flags potential problems that may require cost, schedule, or, in some cases, contract modification.

The introduction of new technology into a complex organizational system will modify its processes and the related structures and subtasks. This organizational evolution must also be mapped into the evolving system and development process. Defining cognitive requirements and evaluating their implementation in support systems is a critical part of ensuring the effectiveness of new systems. Often, of course, new technologies are introduced into organizations to cope with recognized needs for reengineered organizational processes. These statements illustrate the interactive relationships across humans, organizations, and technologies as well as relationships between these and the internal and external environment. User-centered design is an approach that enables and enhances this integration by embracing three basic principles:

- Design of aiding technology embodies the relationship of human users and computer-based aids in achieving organizational goals.
- Decomposition of functions, processes, and tasks provides measurable indicators of the extent to which specific designs fulfill system objectives.
- The utility of evaluation to the system design process depends upon the application and interpretation of performance measures in the context of a valid framework of objectives, functions, processes, and tasks as appropriate MOPs and MOEs.

Thus, the qualitative aspects of support to decision making must be included in the earliest evaluations. Designs for the complex systems supporting decision making derive conceptual requirements from models of organizational processes. The doctrine incorporated in these models, and the missions defined by the organization provide the context for identifying the functional and task requirements that structure the relationships of humans and machines. These requirements, in turn, help to determine the appropriate MOPs and MOEs that form the selection criteria for aiding designs. These can be applied through a combination of checklists, expert reviews, end-user walkthroughs, and heuristic evaluation. As early as possible, developers need the input of “real users” using the system under the most representative conditions. Pilot exercises and experiments generate a wealth of information on the complex interactions of users, processes, and system supports that can be used to assess the development paths of future systems.

10.7.1 Setting Evaluation Goals

The evaluation goals of systems engineering practitioners are often quite distinctly different from those of cognitive science researchers. System interfaces and interactions, and associated human–machine cooperative decision-making tasks, involve highly complex constructs. As discussed, the evaluation of a conceptual design and architectural prototype should track to the associated requirements. Two principal evaluations should be conducted at each level of prototyping:

- Verification of implementation of user requirements by the system
- Validation of design implementation’s effectiveness in terms of interface usability and utility

Depending upon the systems engineering phase under consideration, the evaluation scope, and the level of detail in the prototype, evaluation may range from designer-reviewed checklists and rating scales to empirical evaluations with representative users.

Computer-based interactive prototypes provide opportunities for direct observation of the human–computer decision performance. Several methods are available for examining interaction processes through automated capture and analysis of interaction protocols to facilitate the rapid data analysis required for design iteration (Smith et al., 1993). The empirical study approach builds information in a data-intensive, bottom-up fashion. While empirical evaluations can be used to determine performance benchmarks, they do not permit direct insight into the performance *requirements*.

These requirements evolve from a top-down analysis based upon the organizational and system objectives, functions, and tasks identified with those functions. An analytical framework for empirical evaluation is desirable as, without an analytical framework, the measures collected in empirical studies lack context and can misdirect decision makers. In this context, Rogers (1992) questions the desirability of the microanalysis and theoretical rigor that characterize research in cognitive psychology. Rogers suggests that applied research and, by extension, design evaluation benefits much from a macrolevel analysis that allows a parallel, symbiotic relationship with the theoretical aims of the research. This provides much support for case-study-based evaluation (Yin, 1994; Stake, 1995).

Rasmussen and Pejtersen (1993) conceptualize the well-balanced evaluation of a cognitive systems engineering design product as a combination of top-down analytical evaluation and bottom-up empirical assessments. System design often evolves through the top-down analysis of the intended purpose and identified functions. Functions are then decomposed into the procedures and tasks allocated to the machine and user, culminating in the design that maps the system's form. Bottom-up empirical evaluations first address the lower level human factors issues associated with fundamental usability and continue by evaluating the support of the cognitive requirements involved in the tasks. These human requirements interact with the system's allocation of functional requirements and the capabilities afforded by the design.

Despite some variations in terminology, this prescription for a combination of top-down analytical and bottom-up empirical evaluation is consistent with similar discussions in Meister (1985, 1991) and Adelman (1992). Meister (1985) presents a series of human performance questions grouped by development stage and indicates the various analysis and evaluation methods that supply answers. The balance between analytical and empirical evaluation approaches shifts depending upon the stage of the life-cycle process that leads to the system itself. For example, in the early stages of planning and design, there is a strong reliance on top-down analysis methods supported by the available objective data and subjective judgments. The later phases of detail design and prototype testing employ more rigorous empirical evaluation methods and well-structured subjective measures to assess performance in terms of the functional requirements outlined in earlier phases of development. Figure 10.12 illustrates the relationship of efforts at the various phases of engineering a system and evaluation objectives.

10.7.2 Selecting Evaluation Methods

Nielsen's (1993) text on usability engineering discusses usability heuristics and heuristic evaluation but does not present example checklists or sufficient information to guide the conduct of heuristic evaluation. Ravden and Johnson (1989) present a comprehensive

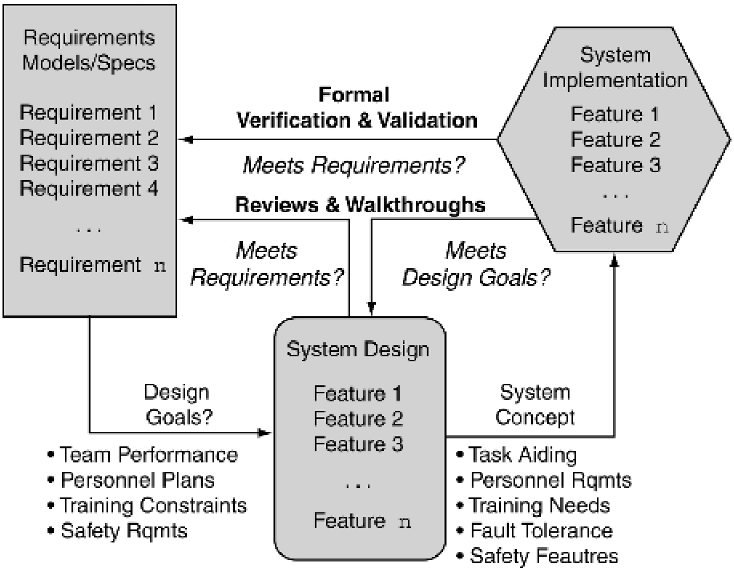

Figure 10.12 Relationship of development process inputs to evaluation goals at each development phase.

evaluation that employs nine usability criteria. Each criterion is addressed in 10 to 12 questions that may be amended to address the specific evaluation goals. Evaluators include the designer(s), representative end users, and such other technical professionals as human factors experts. The members of the evaluation team complete the checklists individually as they perform a predetermined set of exemplary tasks. The principal advantage of Ravden and Johnson's method is the potential for rapid analysis and the ready conversion of the subjective data into quantitative measures for comparison. The most significant source of overhead is in the selection and development of interaction tasks. Depending upon the goals of the evaluation, the development of simple tasks or task scenarios may entail extensive preparation.

Wright and Monk (1991) avoid some of the shortfalls in heuristic evaluation while retaining its low cost and effort features. Although they acknowledge the value of careful quantitative evaluation, they suggest that qualitative evaluation provides more cost-effective guidance for the early phases of design. Their approach, intended for the design practitioner, involves designers and users in a *cooperative evaluation* using think-aloud protocols and verbal probes. Analysis in this early phase is highly focused to capture the relevant information within cost and schedule requirements. Wright and Monk (1989) indicate that evidence in the form of either critical incidents or breakdowns is sufficient to identify system design problems. A *critical incident* is some user behavior that fails to use the functionality of the system efficiently. A *breakdown* designates any point in the interaction where the user's focus on the task is broken due to the demands imposed by the system, such as when the interface ceases to be "transparent." This approach is similar to, but in many ways different from, the retrospective analysis technique used by decision researchers (Klein, 1989).

The ability to use the systems engineering designer as the evaluator allows for the speedy, inexpensive evaluation necessary for iteration in the early stages when the conceptual design is evolving rapidly. Experimental investigations performed with design trainees indicate that satisfactory rates for detecting design problems may be achieved quickly by designers with little or no human factors background and limited training in the method itself (Wright and Monk, 1991). Rather than merely endorsing their own designs, the results of the study indicated that these designers were better at evaluating their own systems using this method than similarly experienced evaluators not associated with the design. Furthermore, the designer-evaluators uncovered more unanticipated problems than the evaluators not involved in the design. The principal limitations in the cooperative evaluation method include problems with the task altering aspects of think-aloud protocols and the potential for bias in the single designer-evaluator model. Similar to the usability evaluation method of Ravden and Johnson (1989), cooperative evaluation also requires the preparation of meaningful tasks that provide the context for the evaluation sessions.

Departing from the design-phase orientation of the classic system development life-cycle (SDLC) model, Gardiner and Christie (1987) examine the role of prototypes in addressing questions on four human system interaction design levels: conceptual, relating to the system concept; semantic, relating to the interaction concept; syntactic, relating to the interaction form; and lexical, relating to interaction details. In related work, Ehrhart (1993) presents a survey of evaluation methods useful for assessing system designs to support human–machine cooperative decision making. Gardiner and Christie's model provides some useful guidelines for trading off the time and expense required for developing a prototype against the functionality and performance achieved. In addition, it indicates the extent of evaluation support possible with a relatively small investment in so doing. Table 10.14 combines the suggestions of Gardiner and Christie (1987), Nielsen

TABLE 10.14 Prototyping and Evaluation to Match Human–System Design Requirements

Design Level	Design Evaluation Focus	Prototyping Support	Evaluation Tools and Techniques
Conceptual	• System and conceptual design concept (architecture) • Appropriate for user requirements	• Written descriptions and scenarios • Storyboards • Interactive storyboards	• Focus groups • Walk-through • Predictive models • Heuristic methods
Semantic	• Interaction concept • Broad definition of interaction, error feedback, and user support	• Interactive storyboards • Hardware mockups • Partial prototypes	• Informal user tests and observation • Walk-through • Checklists and rating scales
Syntactic	• Interaction form • Dialog parameters and interaction sequences	• Interactive storyboards • Partial (developmental) prototypes	• Formal and informal user tests • Walk-through • Controlled laboratory tests • Field tests and observation
Lexical	• Interaction detail • Specification of human–system interaction and interfaces	• Partial (developmental) prototypes • Functional prototypes	• Formal user tests • Gaming and simulation • Field tests and observation

(1993), and Ehrhart (1993) for linking the proposed evaluation focus, prototyping support, and evaluation techniques appropriate at each design level.

10.8 SUMMARY AND CONCLUSIONS

This chapter presents a framework for employing cognitive systems engineering methods to define problems, identify and represent cognitive task requirements, develop design goals, and implement and evaluate system designs for information presentation and interaction in human–machine cooperative systems. To improve HSI planning and implementation in program management requires effort among all the stakeholders. The acquisition process must be revised to require the definition and tasking of HSI responsibilities in all program management directives and acquisition program plans (DoD, 1994). System program managers must have adequate training to understand and direct HSI efforts. The HSI effectiveness should be an element in system program managers' performance ratings. Program funding must provide sufficient resources for implementation of HSI practices during all phases of the engineering of a system. These life-cycle phases must include reviews with organizational stakeholders, including representative end users, training designers, and support personnel.

NOTES

1. To improve and support HSI activities, the Defense Information Systems Agency (DISA) maintains an annotated directory of HSI design support tools and techniques developed by U.S. government agencies, NATO countries, academia, and private industry (Dean, 1998). The DoD also provides a searchable set of mandatory and discretionary HSI guidelines as part of the *Acquisition Deskbook* (DoD, 2001).
2. The other types of cognitive errors are mental memory lapses and physical action slips, which are errors that cause improper implementation of action plans.

REFERENCES

Adelman, L. (1992). *Evaluating Decision Support and Expert Systems.* New York: Wiley.

Adelman, L., Tolcott, M., and Bresnick, T. (1993). Examining the Effect of Information Order on Expert Judgment. *Organizational Behavior and Human Processes*, *56*, 348–369.

Anderson, J. R. (1983). *The Architecture of Cognition*. Cambridge, MA: Harvard University Press.

Andriole, S., and Adelman, L. (1995). *Cognitive Systems Engineering for User-Computer Interface Design, Prototyping, and Evaluation.* Hillsdale, NJ: Erlbaum.

Arthur, L. J. (1992). *Rapid Evolutionary Development: Requirements, Prototyping & Software Creation*. New York: John Wiley.

Ashby, W. R. (1956). *Introduction to Cybernetics.* New York: Wiley.

Badre, A. N. (1982, April). Selecting and Representing Information Structures for Visual Presentation. *IEEE Transactions on Systems, Man, and Cybernetics*, *12*(4), 495–504.

Bainbridge, L. (1987). The Ironies of Automation. In J. Rasmussen, K. Duncan & J. Leplat (Eds.), *New Technology and Human Error*. London: John Wiley.

Ball, R. G., and Ord, G. (1983). Interactive Conflict Resolution in Air Traffic Control. In M. J. Coombs, and J. L. Alty, (Eds.), *Designing for Human-Computer Communication*, London: Academic.

Basili, V., and Weiss, D. M. (1984, November). A Methodology for Collecting Valid Software Engineering Data. *IEEE Transactions on Software Engineering*, *10*(6), 728–738.

Beach, L. R. (1992). Epistemic Strategies: Causal Thinking in Expert and Nonexpert Judgment. In G. Wright and F. Bolger (Eds.), *Expertise and Decision Support*, New York: Plenum.

Bennett, K. B., Nagy, A. L., and Flach, J. M. (1997). Visual Displays. In G. Salvendy (Ed.), *Handbook of Human Factors and Ergonomics* (pp. 659–696). New York: Wiley.

Bennett, K. B., Toms, M. L., and Woods, D. D. (1993). Emergent Features and Graphical Elements: Designing More Effective Configural Displays. *Human Factors*, *35*(1), 71–97.

Boar, B. (1984). *Application Prototyping: A Requirements Definition Strategy for the 80s*. New York: Wiley.

Boehm, B. W. (1976, December). Software Engineering. *IEEE Transactions on Computers*, *25*(12), 1226–1241.

Boehm, B. W. (1981). *Software Engineering Economics*. Englewood Cliffs, NJ: Prentice-Hall.

Boehm, B. W. (1987, September). Improving Software Productivity. *IEEE Computer*, pp. 43–57.

Boehm, B. W. (1988, May). A Spiral Model of Software Development and Enhancement. *IEEE Computer*, *21*(5), 61–72.

Boehm, B. W. (1999, March). Making RAD Work for Your Project. *IEEE Computer*, *32*(3), 113–117.

Boehm, B. et al. (2000). *Software Cost Estimation with COCOMO II*. Upper Saddle River, NJ: Prentice-Hall.

Bott, R. (1979). *A Study of Complex Learning: Theory and Methodology*, CHIP Report 82, La Jolla, CA: Center for Human Information Processing, University of California at San Diego.

Brannick, M. T., Salas, E., and Prince, C. (Eds.). (1997). *Team Performance Assessment and Measurement: Theory, Methods, and Applications*. Mahwah, NJ: Erlbaum.

Brehmer, B. (1987). Models of Diagnostic Judgment. In J. Rasmussen, K. Duncan, and J. Leplat (Eds.), *New Technology and Human Error*. New York: Wiley, 1987.

Burton, R. M., and Obel, B. (1995). *Strategic Organizational Diagnosis: Developing Theory for Application*. Boston: Kluwer Academic.

Byrd, T. A., Cossick, K. L., and Zmud, R. W. (1992, March). A Synthesis of Research on Requirements Analysis and Knowledge Acquisition Techniques. *MIS Quarterly*, pp. 117–138.

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

Carroll, J. M. (1997). Human Computer Interaction: Psychology as a Science of Design. *Annual Review of Psychology*, *48*, 61–83.

Carroll, J. M., Mack, R. L., and Kellogg, W. A. (1988). Interface Metaphors and User Interface Design. In M. Helander (Ed.), *Handbook of Human-Computer Interaction*. New York: Elsevier.

Carroll, J. M., and Olson, J. R. (1988). Mental Models in Human-Computer Interaction. In M. Helander (Ed.), *Handbook of Human-Computer Interaction*. Amsterdam: Elsevier.

Carswell, C. M., and Wickens, C. D. (1987). Information Integration and the Object Display. *Ergonomics*, *30*, 511–527.

Chapanis, A. (1959). *Research Techniques in Human Engineering*. Baltimore, MD: Johns Hopkins Press.

Checkland, P. (1981). *Systems Thinking, Systems Practice*. New York: Wiley.

Checkland, P., and Holwell, S. (1998). *Information, Systems and Information Systems—Making Sense of the Field*. New York: Wiley.

Choo, C. W. (1998). *The Knowing Organization: How Organizations Use Information to Construct Meaning, Create Knowledge, and Make Decisions*. London: Oxford University Press.

Connell, J. L., and L. Shafter. (1989). *Structured Rapid Prototyping: An Evolutionary Approach to Software Development*. Englewood Cliffs, NJ: Prentice Hall.

Coury, B. G., and Boulette, M. D. (1992). Time Stress and the Processing of Visual Displays. *Human Factors*, *34*(6), 707–725.

Daly, E. (1977). Management of Software Development. *IEEE Transactions on Software Engineering*, *3*(3), 229–242.

Davis, A. M. (1993). *Software Requirements: Objects, Functions, and States*, rev. ed. Englewood Cliffs, NJ: Prentice-Hall.

Dean, T. F. (Ed.). (1998, October). *Directory of Design Support Methods*, Technical Report ADA355192. San Diego, CA: Defense Technical Information Center. Available: http://dticam.dtic.mil.

Dörner, D. (1987). On the Difficulties People Have in Dealing with Complexity. In J. Rasmussen, K. Duncan, and J. Leplat (Eds.), *New Technology and Human Error*. New York: Wiley.

Dreyfus, J. L., and Dreyfus, S. E. (1986). *Mind Over Machine: The Power of Human Intuition and Expertise in the Era of the Computer*. New York: Free Press.

Duffy, L. (1993). Team Decision-Making and Technology. In N. J. Castellan, Jr. (Ed.), *Individual and Group Decision Making*. Hillsdale, NJ: Erlbaum.

Edgell, S. E., Bright, R. D., Ng, P. C., Noonan, T. K., and Ford, L. A. (1992). The Effects of Representation on the Processing of Probabilistic Information. In B. Burns (Ed.), *Percepts, Concepts and Categories: The Representation and Processing of Information*. Amsterdam: Elsevier Science.

Edland, A., and Svenson, O. (1993). Judgment and Decision Making under Time Pressure: Studies and Findings. In O. Svenson and A. J. Maule (Eds.), *Time Pressure and Stress in Human Judgment and Decision Making*. New York: Plenum.

Ehrhart, L. S. (1993, March). *Tasks, Models and Methods: System-Level Evaluation of Human-Computer Interaction (HCI) for Command and Control Decision Support*, Technical Report, Contract No. F30602-92-C-0119. Fairfax, VA: C^3I Center, George Mason University.

Ehrhart, L. S. (1994, August). Cognitive Systems Engineering: Human Computer Interaction Design for Decision Support. Ph.D. Dissertation, School of Information Technology & Engineering, George Mason University, Fairfax, VA.

Ehrhart, L. S. (1997, July). Integrating Multimedia Information for Joint Theater Command and Control. In *Proceedings of the 49th IEEE National Aerospace and Electronics Conference*, Dayton, OH. New York: IEEE Press.

Ehrhart, L. S., and Aiken, P. H. (1991). Cognitive Engineering for Intelligent Control System Design: Preserving User Models in Requirements Analysis. In *Proceedings of the 1991 American Control Conference*, Boston, MA, June 26–28, 1991. Evanston, IL: American Automatic Control Council.

Fagan, M. (1974, December). *Design and Code Inspections and Process Control in the Development of Programs*, Report IBM-SDD-TR-21-572. New York: IBM.

Fields, R. E., Wright, P. C., and Harrison, M. D. (1995). A Task Centered Approach to Analyzing Human Error Tolerance Requirements. In P. Zave and M. D. Harrison (Eds.), *Proceedings of the 2nd International Symposium on Requirements Engineering (ICRE'95)* (pp. 18–26). Los Alamitos, CA: IEEE Computer Society.

Fleishman, E. A., and Quaintance, M. K. (1984). *Taxonomies of Human Performance: The Description of Human Tasks*. New York: Academic.

French, W. L., and Bell, Jr. C. H. (1973). *Organization Development: Behavioral Science Interventions for Organization Improvement*. Englewood Cliffs, NJ: Prentice-Hall.

Galbraith, J. R. (1997). *Organizational Design*. Reading, MA: Addison Wesley.

Galbraith, J. R. (1995). *Designing Organizations*. San Francisco, CA: Jossey-Bass.

Gardiner, M. M., and Christie, B. (1987). *Applying Cognitive Psychology to User-Interface Design*. New York: Wiley.

Gentner, D., and Gentner, D. R. (1983) Flowing Waters or Teeming Crowds: Mental Models of Electricity. In D. Gentner and A. L. Stevens (Eds.), *Mental Models*. Hillsdale, NJ: Erlbaum.

Greenbaum, J., and Kyng, M. (Eds.). (1991). *Design at Work: Cooperative Design of Computer Systems*. Hillsdale, NJ: Erlbaum.

Guzzo, R. A., and Salas, E. (Eds.) (1995). *Team Effectiveness and Decision Making in Organizations*. New York: Jossey Bass.

Hackos, J. T., and Redish, J. C. (1998). *User and Task Analysis for Interface Design*. New York: Wiley.

Hammer, M., and Champy, J. (1993). *Reengineering the Corporation: A Manifesto for Business Revolution*. New York: Harper Business.

Hammer, M., and Stanton, S. A. (1995). *The Reengineering Revolution: A Handbook*. New York: Harper Business.

Hammond, K. R. (2000). *Judgments Under Stress*. New York: Oxford University Press.

Harris, J. F. (1994, July 14). Downing of Copters Blamed on Blunders. *Washington Post*, *117*(221), A1, A10.

Harrison, L. E., and Huntington, S. P. (Eds.). (2000). *Culture Matters: How Values Shape Human Performance*. New York: Basic Books.

Harrison, T. M. (1985, Spring). Communication and Participative Decision Making: An Exploratory Study. *Personnel Psychology*, *38*(10), 93–116.

Hastie, R. (2001). Problems for Judgment and Decision Making. *Annual Review of Psychology*, *52*, 653–683.

Helmreich, R. L. (1988, October). *Testimony before the U.S. House of Representatives, Committee on Armed Services on the Subject of the USS VINCENNES Downing of Iranian Air Flight 655*.

Henneman, R. L. (1999, Summer). Design for Usability: Processes, Skills, and Tools. *Information, Knowledge, and Systems Management*, *1*(2), 133–144.

Hermann, C. F. (1972). Threat, Time, and Surprise: A Simulation of International Crisis. In C. F. Hermann (Ed.), *International Crises: Insights from Behavioral Research*. New York: Free Press.

Hoc, J.-M. (1989). Strategies in Controlling a Continuous Process with Long Response Latencies: Need for Computer Support to Diagnosis. *International Journal of Man-Machine Studies*, 30, pp. 47-67.

Hogarth, R. (1987). Judgement and Choice: the Psychology of Decision, 2nd Edition. New York: John Wiley.

Huey, B. M., and Wickens, C. D. (Eds.). (1993). *Workload Transition: Implications for Individual and Team Performance; Panel on Workload Transition, Committee on Human Factors, Commission on Behavioral and Social Sciences and Education, National Research Council*. Washington, DC: National Academy of Sciences.

Humphrey, W. S. (1989). *Managing the Software Process*. Reading, MA: Addison-Wesley.

Janis, I. L. (1982). *Victims of Groupthink.* New York: Free Press.

Janis, I. L. (1989). *Crucial Decisions: Leadership in Policy Making and Crisis Management*. New York: Free Press.

Janis, I. L., and Mann, L. (1977). *Decision Making: A Psychological Analysis of Conflict, Choice, and Commitment*. New York: Free Press.

Johnson, P., Diaper, D., and Long, J. (1984). Tasks, Skills and Knowledge: Task Analysis for Knowledge Based Descriptions. In *Interact'84*, Vol. 1. London: IFIP.

Johnson-Laird, P. N. (1983). *Mental Models*. Cambridge: Cambridge University Press.

Johnson-Laird, P. N. (1999). Deductive Reasoning. *Annual Review of Psychology*, *50*, 109–135.

Kahan, J. P., Worley, D. R., and Stasz, C. (1989) *Understanding Commander's Information Needs*. [Technical Report R-3761-A]. Santa Monica, CA: The RAND Corporation.

Kieras, D. E. (1988). Towards a Practical GOMS Model Methodology for User Interface Design. In M. Helander (Ed.), *Handbook of Human-Computer Interaction*. Amsterdam: Elsevier.

Klein, G. A. (1993a). A Recognition-Primed Decision (RPD) Model of Rapid Decision Making. In G. A. Klein, J. Orasanu, R. Calderwood, and C. E. Zsambok (Eds.), *Decision Making in Action: Models and Methods*. Norwood, NJ: Ablex.

Klein, G. A. (1993b, April). *Naturalistic Decision Making: Implications for Design*, Technical Report: CSERIAC SOAR 93-1. Wright-Patterson AFB, OH: Crew System Ergonomics, Information Analysis Center.

Klein, G. A. (1998). *Sources of Power: How People Make Decisions*. Cambridge, MA: MIT Press.

Kotter, J. P., and Heskett, J. L. (1992). *Corporate Culture and Performance*. New York: Free Press.

Krishnan, V., and Ulrich, K. T. (2001, January). Product Design Decisions: A Review of the Literature. *Management Science*, *47*(1), 1–21.

Lehner, P. E., and Zirk, D. A. (1987). Cognitive Factors in User/Expert-System Interaction. *Human Factors*, *29*(1), 97–109.

Lerch, F. J., and Harter, D. E. (2001, March). Cognitive Support for Real Time Dynamic Decision Making. *Information Systems Research*, *12*(1), 63–82.

Lockhead, G. R. (1992). On Identifying Things: A Case for Context. In B. Burns (Ed.), *Percepts, Concepts and Categories: The Representation and Processing of Information*. Amsterdam: Elsevier Science.

MacMillan, J., and Entin, E. B. (1991). Decision-Oriented Display Design and Evaluation. In *Proceedings of the 1991 Symposium on Command and Control Research*, June 24–25, 1991, National Defense University, Ft. Lesley J. McNair, Washington, DC. McLean, VA: SAIC, December.

Mantei, M. M., and Teorey, T. J. (1988, April). “Cost/Benefit Analysis for Incorporating Human Factors in the Software Lifecycle.” *Communications of the ACM*, 31(4), pp. 428–439.

Manzoni, J. F., and Angehrn, A. A. (1998, Winter). Understanding Organizational Dynamics of IT-Enabled Change: A Multimedia Simulation Approach. *Journal of Management Information Systems*, *14*(3), 109–140.

Mayer, R. E. (1979). Can Advance Organizers Influence Meaningful Learning? *Review of Educational Research*, *49*, 371–383.

Mayer, R. E., and Bromage, B. K. (1980). Different Recall Protocols for Technical Texts Due to Advance Organizers. *Journal of Educational Psychology*, *72*, 209–225.

Mayhew, D. J. (1999). *The Usability Engineering Lifecycle: A Practitioner’s Handbook for User Interface Design*. San Francisco, CA: Morgan Kaufmann.

Meister, D. (1981). *Behavioral Research and Government Policy: Civilian and Military R and D.* New York: Pergamon.

Meister, D. (1985). *Behavioral Analysis and Measurement Methods*. New York: Wiley.

Meister, D. (1991). *Psychology of System Design.* Amsterdam: Elsevier.

Mellers, B. A., Schwartz, A., and Cooke, A. D. J. (1998). Judgment and Decision Making. *Annual Review of Psychology*, *49*, 447–477.

Montazemi, A. R., and Conrath, D. W. (1986, March). The Use of Cognitive Mapping for Information Requirements Analysis, *MIS Quarterly*, *10*(1), 45–56.

Nadler, D. A., and Tushman, M. L. (1997). *Competing by Design: The Power of Organizational Architecture*. New York: Oxford University Press.

Nielsen, J., and Molich, R. (1990). Heuristic Evaluation of User Interfaces. In J. Carrasco Chew and J. Whiteside (Eds.) *Proceedings of ACM CHI ’90 Conference. Seattle, WA, 1–5 April 1990*. New York: ACM Press, pp. 249–256

Nielsen, J. (1993). *Usability Engineering*. San Diego, CA: Academic.

Norman, D. A., and Draper, S. W. (1986). *User Centered System Design*. Hillsdale, NJ: Erlbaum.

Nystrom, P. C., and Starbuck, W. H. (Eds.). (1981). *Handbook of Organizational Design*. New York: Oxford University Press.

Olson, G. M., and Olson, J. S. (1991). User-Centered Design of Collaboration Technology. *Journal of Organizational Computing*, *1*, 61–83.

Orasanu, J., and Salas, E. (1993). Team Decisionmaking in Complex Environments. In G. A. Klein, J. Orasanu, R. Calderwood, and C. E. Zsambok (Eds.), *Decision Making in Action: Models and Methods*, Norwood, NJ: Ablex.

Orasanu, J., and Shafto, M. G. (1999). Designing for Cognitive Task Performance. In A. P. Sage and W. B. Rouse (Eds.), *Handbook of Systems Engineering and Management* (pp. 629–658). New York: Wiley.

Payne, J. W., Bettman, J. R., and Johnson, E. J. (1993). The Use of Multiple Strategies in Judgment and Choice. In N. J. Castellan, Jr. (Ed.), *Individual and Group Decision Making*. Hillsdale, NJ: Lawrence Erlbaum.

Pecht, M. G. (1995). *Product Reliability, Maintainability, and Supportability Handbook*. Boca Raton, FL: CRC Press.

Pecht, M. G. (1999). Reliability, Availability, and Maintainability. In A. P. Sage and W. B. Rouse (Eds.), *Handbook of Systems Engineering and Management* (pp. 303–326). New York: Wiley.

Perdu, D., and Levis, A. H. (1993, September). Requirements Determination Using the Cube Tool Methodology and Petri Nets. *IEEE Transactions on Systems, Man, and Cybernetics*, *23*(5), 1255–1264.

Pew, R. W. (1988, October). Testimony before the U.S. House of Representatives, Committee on Armed Services on the Subject of the USS VINCENNES Downing of Iranian Air Flight 655.

Preece, J., Rogers, Y., Sharp, H., and Benyon, D. (1994). *Human-Computer Interaction*. Reading, MA: Addison-Wesley.

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

Rasmussen, J., Pejtersen, A. M., and Schmidt, K. (1990, September). *Taxonomy for Cognitive Work Analysis*, Technical Report Risø-M-2871. Roskilde, Denmark: Risø National Laboratory.

Rasmussen, J., Pejtersen, A. M., and Goodstein, L. P. (1994). *Cognitive Systems Engineering*. New York: Wiley.

Rasmussen, J., and Vicente, K. J. (1989). Coping with Human Errors through System Design: Implications for Ecological Interface Design. *International Journal of Man-Machine Studies*, *31*, 517–534.

Rastegary, H., and Landy, F. J. (1993). The Interactions among Time Urgency, Uncertainty, and Time Pressure. In O. Svenson and A. J. Maule (Eds.), *Time Pressure and Stress in Human Judgment and Decision Making*. New York: Plenum Press.

Ravden, S. J., and Johnson, G. I. (1989). *Evaluating Usability of Human-Computer Interfaces: A Practical Method*. Chichester: Ellis Horwood.

Reason, J. (1990). *Human Error*. Cambridge, UK: Cambridge University Press.

Robbins, S. P. (1990). *Organization Theory: Structure, Design, and Applications*. Englewood Cliffs, NJ: Prentice-Hall.

Rogers, Y. (1992). Mental Models and Complex Tasks. In Y. Rogers, A. Rutherford and P. A. Bibby (Eds.), *Models in the Mind: Theory, Perspective and Application*. New York: Academic Press.

Roth, E. M., Bennett, K. B., and Woods. D. D. (1988). Human Interaction with an “Intelligent” Machine. In E. Hollnagel, G. Mancini, and D. D. Woods, (Eds.), *Cognitive Engineering in Complex Dynamic Worlds*. London: Academic.

Rouse, W. B., Cannon-Bowers, J. A., and Salas, E. (1992, November/December). The Role of Mental Models in Team Performance in Complex Systems. *IEEE Transactions on Systems, Man, and Cybernetics*, *22*(6), 96–138.

Rouse, W. B., Edwards, S. L., and Hammer, J. M. (1993, November/December). Modeling the Dynamics of Mental Workload and Human Performance in Complex Systems. *IEEE Transactions on Systems, Man, and Cybernetics*, *23*(6), 1662–1671.

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

Sage, A. P. (1992). *Systems Engineering*. New York: Wiley.

Sage, A. P. (1995). *Systems Management for Information Technology and Software Engineering*. New York: Wiley.

Sage, A. P. (1999). Systems Reengineering. In A. P. Sage and W. B. Rouse (Eds.), *Handbook of Systems Engineering and Management* (pp. 825–932). New York: Wiley.

Sage, A. P. (Ed.). (1987). *System Design for Human Interaction*. New York: IEEE.

Sage, A. P. (Ed.). (1990). *Concise Encyclopedia of Information Processing in Systems and Organizations*. Oxford: Pergamon.

Sage, A. P., and Palmer, J. D. (1990). *Software Systems Engineering*. New York: Wiley.

Sage, A. P., and Rouse, W. B. (Eds.). (1999). *Handbook of Systems Engineering and Management*. New York: Wiley.

Sage, A. P., and Small, C. T. (2000, October). A Simulation Perspective on Knowledge Management and Sharing, and Conflict and Complexity in Social Systems Management. In *Proceedings of the IEEE Conference on Systems, Man, and Cybernetics*, Nashville TN (pp. 536–541).

Salama, A. (1992, Autumn). The Use of an Organization's Biography as a Research Method for Investigating Organizational Development. *Management Education and Development*, *23*(3), 225–233.

Salas, E., and Cannon-Bowers, J. A. (2001). The Science of Training: A Decade of Progress. *Annual Review of Psychology*, *52*, 471–499.

Salvendy, G. (Ed.). (1997). *Handbook of Human Factors and Ergonomics*. New York: Wiley.

Salvendy, G. (Ed.). (2001). *Handbook of Industrial Engineering: Technology and Operations Management*. New York: Wiley.

Sanderson, P. M., Flach, J. M., Buttigieg, M. A., and Casey, E. J. (1989). Object Displays Do Not Always Support Better Integrated Task Performance. *Human Factors*, *31*(2), 183–198.

Schein, E. H. (1992). *Organizational Culture and Leadership*, 2nd ed. San Francisco, CA: Jossey-Bass.

Schneider, S. C., and Angelmar, R. (1993). Cognition in Organizational Analysis: Who's Minding the Store? *Organizational Studies*, *14*(3), 347–374.

Seamster, T. J., Redding, R. E., and Kaempf, G. L. (1997). *Applied Cognitive Task Analysis in Aviation*. Brookfield, VT: Avebury.

Senders, J. W., and Moray, N. P. (1991). *Human Error: Cause, Prediction, and Reduction*. Hillsdale, NJ: Lawrence Erlbaum.

Senge, P. M. (1990). *The Fifth Discipline: The Art and Practice of the Learning Organization*. New York: Doubleday Currency.

Senge, P. M., Roberts, C., Ross, R. B., Smith, B. J., and Kleiner, A. (1994). *The Fifth Discipline Fieldbook: Strategies and Tools for Building a Learning Organization*. New York: Doubleday.

Senge, P. M., and Sterman, J. D. (1994). Systems Thinking and Organizational Learning: Acting Locally and Thinking Globally in the Organization of the Future. In J. D. W. Morecroft and J. D. Sterman (Eds.), *Modeling for Learning Organizations*. Portland, OR: Productivity Press.

Senge, P. M., Kleiner, A., Roberts, C., Ross, R. B., Roth, G., and Smith, B. J. (1999). *The Dance of Change: The Challenges to Sustaining Momentum in a Learning Organization*. New York: Doubleday.

Shanteau, J. (1992). The Psychology of Experts: An Alternative View. In G. Wright and F. Bolger (Eds.), *Expertise and Decision Support*. New York: Plenum Press.

Sheridan, T. B. (2002). *Humans and Automation: System Design and Research Issues*. New York: Wiley.

Shneiderman, B. (1997). *Designing the User Interface: Strategies for Effective Human-Computer Interaction*, (3rd ed.). Reading, MA: Addison-Wesley.

Smith, S. L., and Mosier, J. N. (1986, August). *Guidelines for Designing User Interface Software*, Report ESD-TR-86-278. Bedford, MA: MITRE Corporation.

Somerville, M. A., and Rapport, D. J. (Eds.). (2000). *Transdisciplinarity: Recreating Integrated Knowledge*. Oxford: EOLSS Publishers.

Sproles, N. (2000, February). Coming to Grips with Measures of Effectiveness. *Systems Engineering*, *3*(1), 50–58.

Sproles, N. (2001, May). The Difficult Problem of Establishing Measures of Effectiveness for Command and Control: A Systems Engineering Perspective. *Systems Engineering*, *4*(2), 145–155.

Staggers, N., and Norcio, A. F. (1993). Mental Models: Concepts for Human-Computer Interaction Research. *International Journal of Man-Machine Studies*, *38*, 587–605.

Stake, R. E. (1995). *The Art of Case Study Research*. Thousand Oaks, CA: Sage.

Stammers, R. B., Carey, M. S., and Astley, J. A. (1990). Task Analysis. In J. R. Wilson and E. N. Corlett (Eds.), *Evaluation of Human Work: A Practical Ergonomics Methodology.* New York: Taylor and Francis.

Stolovitch, H. D., and Keeps, E. J. (Eds.). (1992). *Handbook of Human Performance Technology: A Comprehensive Guide for Analyzing and Solving Performance Problems in Organizations*. San Francisco, CA: Jossey Bass.

Suchman, L. A., and Trigg, R. H. (1991). Understanding Practice: Video as a Medium for Reflection and Design. In J. Greenbaum and M. Kyng (Eds.), *Design at Work: Cooperative Design of Computer Systems*. Hillsdale, NJ: Erlbaum.

Svenson, O., and A. J. Maule (Eds.) (1993). *Time Pressure and Stress in Human Judgment and Decision Making*. New York: Plenum Press.

Tainsh, M. A. (1985). Job Process Charts and Man-Computer Interaction within Naval Command Systems. *Ergonomics*, *28*, 555–565.

Treu, S. (1992). Interface Structures: Conceptual, Logical, and Physical Patterns Applicable to Human-Computer Interaction. *International Journal of Man-Machine Studies*, *37*, 565–593.

Trice, H. M., and Beyer, J. M. (1993). *The Culture of Work Organizations*. Saddle River, NJ: Prentice-Hall.

U.S. Department of Defense (DOD). (1994, June 8). *Human Systems Integration Requirements for Air Force Acquisition Programs*. OAIG-AUD, Report No. 94-124. Arlington, VA: Office of the Inspector General.

U.S. Department of Defense (DOD). (2000, October 23). *The Defense Acquisition System (Directive 5000.1) and Operation of the Defense Acquisition System (Instruction 5000.2)*. Available: ric.crane.navy.mil.

U.S. Department of Defense (DOD). (2001). *Acquisition Deskbook*. Washington, DC: Office of the Secretary of Defense. Available: www.deskbook.osd.mil.

van Gigch, J. P. (1991). *System Design Modeling and Metamodeling*. New York: Plenum.

Vanderhaegen, F., Crevits, I., Debernard, S., and Millot, P. (1994). Human-Machine Cooperation: Toward an Activity Regulation Assistance for Different Air Traffic Control Levels. *International Journal of Human-Computer Interaction*, *6*(1), 65–104.

Vennix, J. A. M., Andersen, D. F., Richardson, G. P., and Rohrbaugh, J. (1994). Modeling Building for Group Decision Support: Issues and Alternatives in Knowledge Elicitation. In J. D. W. Morecroft and J. D. Sterman (Eds.), *Modeling for Learning Organizations*. Portland, OR: Productivity Press.

Vicente, K. J., and Rasmussen, J. (1992, July/August). Ecological Interface Design: Theoretical Foundations. *IEEE Transactions on Systems, Man and Cybernetics*, *22*(4), 589–606.

Von Krogh, G., Ichijo, K., and Nonaka, I. (2000). *Enabling Knowledge Creation: How to Unlock the Mystery of Tacit Knowledge and Release the Power of Innovation*. New York: Oxford University Press.

Weick, K. E. (1995). *Sensemaking in Organizations*. Thousand Oaks, CA: Sage.

Weick, K. E., and Roberts, K. H. (1993). Collective Mind in Organizations: Heedful Interrelating on Flight Decks. *Administrative Quarterly*, *38*(3), 357–381.

Wellens, A. R. (1993). Group Situation Awareness and Distributed Decision Making: From Military to Civilian Applications. In N. J. Castellan, Jr. (Ed.), *Individual and Group Decision Making*. Hillsdale, NJ: Lawrence Erlbaum.

Wickens, C. D. (1987). Attention. (1987). In P. A. Hancock (Ed.), *Human Factors Psychology.* Amsterdam: North-Holland.

Wickens, C. D., and Hollands, J. G. (1999). *Engineering Psychology and Human Performance*, 3rd ed. Englewood Cliffs, NJ: Prentice-Hall.

Williams, K. E., Reynolds, R. E., Carolan, T. F., Anglin, P. D., and Shrestha, L. B. (1989, December). *An Evaluation of a Methodology for Cognitively Structuring and Adaptively Sequencing Exercise Content for Embedded Training*. Technical Report TR89-035. Orlando, FL: University of Central Florida, Institute for Simulation and Training.

Wohl, J. C. (1981). Force Management Decision Requirements for Air Force Tactical Command and Control. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-11(9), pp. 618–639.

Woods, D. D. (1984). Visual Momentum: A Concept to Improve the Cognitive Coupling of Person and Computer. *International Journal of Man-Machine Studies*, *21*, 229–244.

Woods, D. D., and Hollnagel, E. (1987). Mapping Cognitive Demands in Complex Problem-Solving Worlds. *International Journal of Man-Machine Studies*, 26, pp. 257–275.

Woods, D. D. (1988). Coping with Complexity: The Psychology of Human Behavior in Complex Systems. In L. P. Goodstein, H. B. Andersen, and S. E. Olsen (Eds.), *Tasks, Errors, and Mental Models*. London: Taylor and Francis.

Woods, D. D., and Roth, E. M. (1988). Cognitive Systems Engineering. In M. Helander (Ed.), *Handbook of Human-Computer Interaction*. Amsterdam: Elsevier.

Wright, P. C., and Monk, A. F. (1989). Evaluation for Design. In A. Sutcliffe and L. Macaulay (Eds.), *People and Computers V.* Cambridge: Cambridge University Press.

Wright, P. C., and Monk, A. F. (1991). A Cost-Effective Evaluation Method for Use by Designers. *International Journal of Man-Machine Studies*, *35*, 891–912.

Yin, R. K. (1994). *Case Study Research: Design and Methods*, 2nd ed. Thousand Oaks, CA: Sage.

Yu, E. S. K., Mylopoulos, J., and Lesperance, Y. (1996, August). Modelling the Organization: New Concepts and Tools for Re-Engineering. *IEEE Expert*, pp. 16–23.

Zachary, W. W. (1988). Decision Support Systems: Designing to Extend the Cognitive Limits. In M. Helander (Ed.), *Handbook of Human-Computer Interaction*. Amsterdam: Elsevier.

Zahniser, R. A. (1993, October). Design by Walking Around. *Comm. of the ACM*, *36*(10), 115–123.