N93-32149

INFORMATION FOR THE USER IN DESIGN OF INTELLIGENT SYSTEMS

Jane T. Malin
NASA Johnson Space Center
Intelligent Systems Branch - ER22
Houston, TX 77058
(713) 483-2046

Debra L. Schreckenghost
The MITRE Corporation
1120 NASA Road One
Houston, TX 77058
(713) 333-0944

ABSTRACT

Recommendations are made for improving intelligent system reliability and usability based on the use of information requirements in system development. Information requirements define the task-relevant *messages* exchanged between the intelligent system and the user by means of the user interface ***medium***. Thus, these requirements affect the design of both the intelligent system and its user interface. Many difficulties that users have interacting with intelligent systems are caused by information problems. These information problems result from (1) not providing the right information to support domain tasks, and (2) not recognizing that using an intelligent system introduces new user supervisory tasks that require new types of information. These problems are especially prevalent in intelligent systems used for real-time space operations, where data problems and unexpected situations are common. Information problems can be solved by deriving information requirements from a description of user tasks. Using information requirements embeds human-computer interaction design into intelligent system prototyping, resulting in intelligent systems that are more robust and easier to use.

INTRODUCTION

Many difficulties that users have interacting with intelligent systems are caused by information problems. These problems are especially prevalent in systems used for real-time operations, where timing constraints make it essential that intelligent systems communicate effectively with their users (users in space operations are called *operators*). The following example illustrates a typical information problem in this environment.

> *Example*: A user's task is to detect event Y, which occurs when sensor A is bad and switch B is off. The intelligent system displays the current status of sensor A and state of switch B. If a change in the displayed value from either sensor A or switch B occurs before the user looks at the display, the user misses event Y.

On the surface, the problem with this system appears to be caused by "bad" user interface design (i.e., overwriting the display of data from sensor A and switch B before event Y can be detected). A closer look, however, reveals that the so-called bad user interface is merely a symptom of an underlying information problem (i.e., the information of interest is event Y, but the intelligent system does not provide that information).

This paper characterizes the information problems encountered when building intelligent systems for real-time space operations, and makes design recommendations for solving these problems. These results are based on experience gained in designing intelligent systems for space operations at the National Aeronautics and Space Administration (NASA). The authors have extended their design experience with case studies of intelligent systems built and used at NASA (Malin, et al., 1991). They have also collaborated with Woods and Potter (Woods, et al., 1991; Potter and Woods, 1991) concerning new user interface designs addressing some of these information problems.

This paper was written to assist intelligent system designers in designing for more effective communication with users, and to inform intelligent system tool builders and human factors engineers of ways to better support these system designers. The first section describes the information problems encountered in real-time space operations. The next section introduces the concept of information requirements as an approach to handling these information problems. The third section discusses the design of intelligent systems for effective communication with users. This includes describing the information needed to monitor the domain system and supervise the intelligent system, and proposing alternatives to typical user interface design approaches. The final section summarizes how these recommendations improve intelligent system reliability and usability. The topics discussed in this paper are covered in greater detail in a NASA Technical Memorandum (Malin and Schreckenghost, 1992).

INFORMATION PROBLEMS IN REAL-TIME SPACE OPERATIONS

A key observation from the case studies is that many perceived user interface problems in intelligent systems are actually information problems. These information problems result from (1) not providing the right information to support *domain tasks*, and (2) not recognizing that an intelligent system introduces *new user supervisory tasks* that require new types of information. These problems are made more difficult by failure to consider how intelligent systems operate in space environments (e.g., effect of data quality and availability on intelligent system behavior) and failure to integrate intelligent system operations with other operations. If not solved, these information problems impact intelligent system reliability and usability.

The first information problem is failure to provide the right information to support domain tasks. The most common tasks performed by intelligent systems being built today are fault monitoring, detection, and diagnosis of cause. The information needed to perform these tasks includes the important behaviors, interesting relationships, and significant changes occurring in the domain system, i.e., *monitored process* (Woods, et al., 1991). To interpret this information, the operator must also understand the behavior expected to occur, and the thresholds delimiting significant or interesting changes (i.e., transition points). For example, many of the operator's decisions during fault management require information about functional capability (what functionality has been lost, how the mission is impacted by that loss). Yet the information typically communicated by the intelligent system consists of device failures shown on schematics or listed in message logs. Such communication does not support the operator in identifying the important changes (e.g., lost functionality) and relationships (e.g, how failures impact mission goals). Considerable effort is required to use this information to make fault management decisions. Thus, common practice in communicating with the operator does not provide the information needed to make fault management decisions.

The second information problem is failure to design for user supervision. New user supervisory tasks include both monitoring ongoing intelligent system activities, and guiding and correcting the system when it malfunctions. Intelligent systems are usually not designed to be managed because it is not well recognized that they need to be managed. This omission in design arises from two misconceptions: (1) that the intelligent system is more knowledgeable than its user, and (2) that the intelligent system can be designed to prevent all errors from occurring. In fact, the typical space operations user (a flight controller) is also a domain expert. This expert user is more knowledgeable than the intelligent system and is well qualified to supervise it. The misconception that all intelligent system errors can be prevented results from unrealistic assumptions about the space operations environment and how intelligent systems operate within that environment. Due to the complexity of this environment, the behavior of the monitored process cannot always be accurately predicted, and unexpected situations occur. Because they are unexpected, the knowledge base does not address them and the intelligent system can respond anomalously. Additionally, data problems (e.g., stale, noisy, or biased data) are common in space environments and can cause intelligent system error.

Although intelligent systems can be designed for supervision and correction (Land et al., 1992), they are not typically designed that way. Often, the intelligent system provides no means for the user to respond to system errors, apart from turning the system off. If not designed for user supervision, the intelligent system can also be difficult to understand (what Abbott calls a "magical" system; Abbott, 1991) because it doesn't provide the user with necessary information about system processing. The human supervisor must understand what the intelligent system can do (its capabilities), what it is currently doing (its activities), and why (its reasoning strategies). Without such an understanding, the supervisor cannot guide and correct it. Providing this additional information to supervise the intelligent system can overload the user, however, if it is not effectively managed. Because the intelligent system is embedded in a larger support system, information from that larger system can be used when compensating for intelligent system errors (e.g., operator can use that information to take over intelligent system tasks).

INFORMATION REQUIREMENTS FOR SYSTEM DESIGN

The information problems in real-time space operations that were discussed in the previous section can be characterized as not providing the right information at the right time to support fault management tasks for the monitored process and user supervisory tasks for the intelligent system. An understanding of both of these types of tasks is necessary to determine what the "right" information is and when it should be provided. Using a description of these tasks, the designer can define the task-relevant messages exchanged between the intelligent system and the user by means of the user interface medium (i.e., the *information requirements*). These information requirements are then used in designing the system. Because they are based on a description of how the user will interact with the intelligent system, information requirements include operational considerations early in system design.

Information requirements affect the design of both the intelligent system (i.e., what information to represent) and its user interface (i.e., what information to present). Thus, intelligent system design and user interface design are not independent efforts, but aspects of a single development process. Considering human-computer interaction (HCI) as part of system development integrates user interface design into overall system design. Since information requirements affect both intelligent system and user interface design, HCI expertise is needed throughout the development of the system. Early in system development, HCI expertise is needed for describing task-level information requirements. Knowledge of display and control software and hardware is needed to design media for presenting this information, which may include early development of user interface design concepts as prototypes or storyboards. This approach also necessarily involves users early in system design to describe the task and assist in identifying information requirements.

A task-based approach to requirements definition solves many of the information problems described earlier. Identifying information requirements does not require a full task analysis, such as the GOMS analysis (Kieras, 1988). Only the high-level monitoring, control, and decision-making tasks for managing the monitored process and supervising the intelligent system need be identified (see next section for an illustration). Finer-grained task analysis techniques are not appropriate for this purpose, because they are designed for detailed user interface design at the dialogue or display level.

There is much yet to be learned about how to develop high-level task descriptions for the purpose of identifying information requirements. Formal methods will most likely evolve from approaches proposed by Rasmussen (1986) and Mitchell and Miller (1986), and from distributed agent communication approaches from artificial intelligence. In the interim, information requirements can be identified by developing and informally analyzing scenarios. Such scenarios would include the identified joint human-computer tasks, and would represent managing both the monitored process and the intelligent system. These scenarios are evaluated to identify the information that must be exchanged between the user and the intelligent system. Alternative task allocations can also be evaluated for ability to recover from intelligent system errors, to accommodate changes in task priority or workload, and to coordinate human and intelligent system activities. Both prototypes and storyboards can be used to evaluate operational scenarios.

Information requirements are the basis for selecting what information should be represented and presented for a task. This guarantees that all the needed information, and only the needed information, is provided. This solves the information problems related to magical systems and information overload. Additionally, information requirements provide a more objective and rigorous basis for evaluating a design than the usual approach in which a design is good if the users like it (what Abbott calls design by "Mikey likes it";Abbott, 1992).

COMMUNICATION BETWEEN HUMAN AND INTELLIGENT SYSTEM

Effective human-computer communication requires striking a balance. On one hand, the intelligent system must provide enough information for the user to understand its behavior (i.e., avoiding a magical system). On the other hand, the intelligent system must not provide so much information that it interrupts or distracts the user from more important tasks. The user is already overloaded with information. The problem of information overload becomes especially important in real-time environments where complex, high risk tasks are performed, and where human errors can represent serious risk.

Most of the intelligent systems studied communicate with the user in at least one of the following ways:

- *message list*: a chronological list of state and status assessments and/or action recommendations
- *annotated schematic*: a graphic representation of the physical structure of the system, annotated with sensor measurements or state/status assessments
- *explanation*: a conversational style of providing justification for an intelligent system conclusion

These typical approaches to communication are not effective at achieving balanced communication. They often do not represent the right type of information for user tasks or do not present it in a way effectively supporting real-time operations.

Balanced communication is achieved by developing a shared understanding of the ongoing situation, so that the intelligent system and user make decisions based on the same information. Such an understanding is developed over time by monitoring the same information, including the environmental events, the actions of the crew and flight controllers, and the behavior of both the monitored process and the intelligent system in response to those events and actions. Understanding the behavior of a system and its capabilities to respond to a situation requires having some "visibility" into the system. Providing visibility is providing an *unobstructed view* to the user. To be *unobstructed*, nothing should get in the way of sight (i.e., information is clearly presented, with the relevant information apparent). A *view* includes a perspective. Information for visibility into the monitored process is represented from the perspective of managing process operations and failures. Information for visibility into the intelligent system is represented from the perspective of coordinating with and managing operations and errors in this system. This section describes the types of information needed for both of these perspectives, and discusses alternatives to the traditional forms of communication.

Information for Managing the Monitored Process

Managing the monitored process requires that the operator monitor process behavior for anomalies and respond to those anomalies. To detect anomalies, the operator must have some expectation of what process behavior should be (i.e., nominal behavior). Typically, an alarm is annunciated when behavior is not within the limits defining nominal behavior. Multiple alarms may be issued shortly after the initial alarm due to failure propagation into other systems and redundant alarms. Additionally, anomalies can have serious implications for safety and mission objectives, and can impose hard timing constraints. Thus, managing information from multiple alarms increases workload just at the time when the operator can least afford it.

Responding to anomalies in space systems is a complex decision-making process, often with high stakes (e.g., potential loss of crew, failure of costly experiment). The operator must make the following decisions when an anomaly occurs:

- Determine if any action is required in response to the anomaly
- Distinguish failures from false alarms
- If more than one response is possible, select one
- When action is taken, evaluate if it is effective

A significant amount of information is needed to make these decisions. The cause of the anomaly (or a set of possible causes) must be identified. The impacts of the anomaly must be determined, including the immediate consequences to mission objectives and safety (e.g., lost functionality) and the potential for future consequences (e.g., failure propagation potential). Response options must be delineated and the "best" response enacted. The effect of response procedures must also be monitored for adverse or unexpected effects.

To support alarm management and anomaly response, the intelligent system should provide information that improves the operator's understanding of the situation. This requires focusing the operator's attention on what is diagnostically important, and quickly and clearly indicating the diagnostic content and relationships in this information. Especially when the system supports operations that change process states, it is important to call attention to events (e.g., state transitions) and procedure-driven activities (e.g., configuration changes) preceding the anomaly, as context for interpreting an anomaly. The example in Figure 1 illustrates how knowledge of preceding events can assist the operator in managing the monitored process.

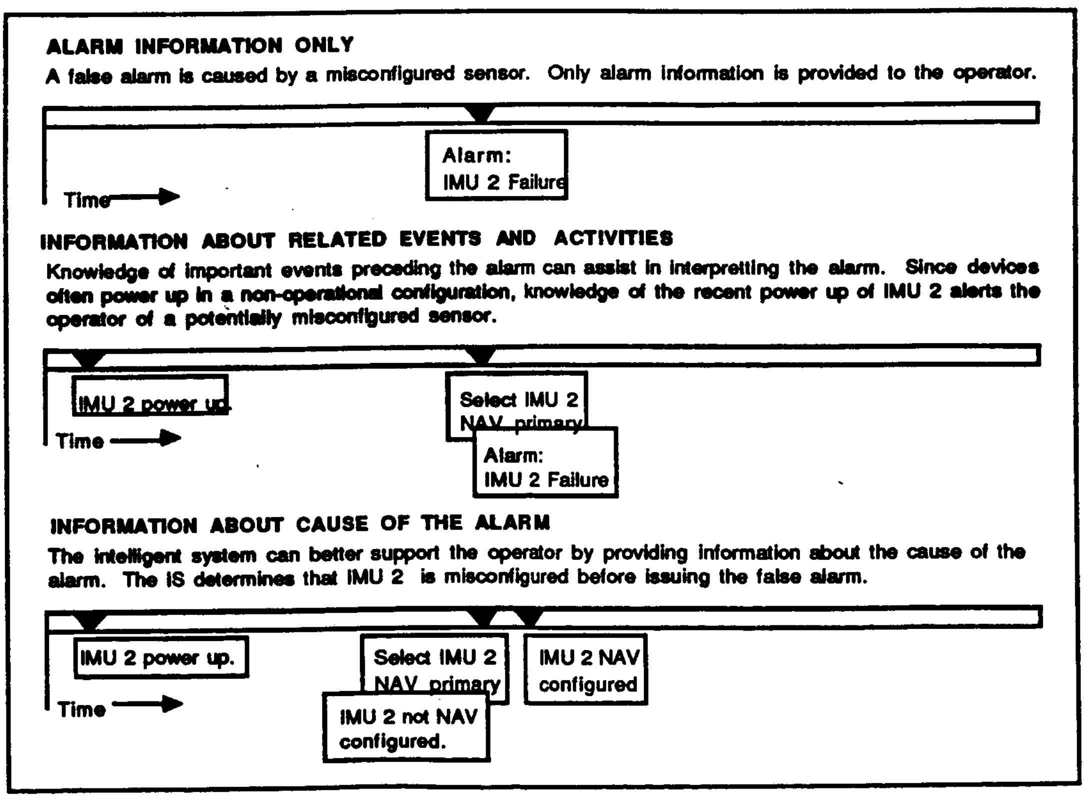

Figure 1 - Using Knowledge of Preceding Events to Improve Operator Understanding of Situation

Both message lists and schematics can obscure intelligent system information important to the operator (Woods, et al, 1991). As shown in the example, situations develop as patterns of events indicated by changing state and status. Schematics can only present the latest event (i.e., the current state). Additionally, if events are not related to physical structure (e.g., functional status), they can be difficult to present clearly using a schematic. Message lists do capture some event history, but do not represent the relationships between events (e.g., the temporal distance between events) necessary to reveal these patterns. Since chronology is the means of sorting messages, related information can become dissociated. Intelligent system designers should consider alternatives to message lists and schematics that assist the operator in seeing patterns of events as they occur. For example, Potter and Woods are investigating timelines as an alternative to message lists (Potter and Woods, 1991). Representation of functional information instead of physical information can be effective for supporting anomaly response (Malin et al., 1991).

Information for Supervising the Intelligent System

Possibly the most significant problem in intelligent system design is failure to recognize that use of an intelligent system poses new tasks for the operator. In addition to managing the monitored process, the operator must now supervise the intelligent system, including monitoring and coordinating its activities, and responding to its errors. But intelligent system are rarely designed for supervision. The traditional means of communicating with an intelligent system (message lists, schematics, and explanation) do not support the operator in monitoring its activities and understanding its reasoning. And, when system errors occur, the operator usually has few options for responding to them (the restart button or the power plug).

Designing the intelligent system for supervision means providing adequate information for the operator to understand what it is doing and to know how to respond to its errors (i.e., providing visibility into the intelligent system). Similar to providing visibility into the monitored process, providing visibility into the intelligent system means making important system behavior evident as a situation develops (i.e., conclusions and reasoning strategies). It means showing intelligent system activities in progress and how well these activities are achieving goals. It also means informing the operator of the critical evidence used to draw a conclusion and the confidence in that conclusion (hypothesis or conclusion). This information should be presented in a way that reinforces the operator's understanding of the intelligent system's reasoning strategy (Chandrasekaran, et al., 1989). Thus, managing the intelligent system means providing a lot of new information to the operator. There is a risk of overloading the operator if the system is not carefully designed to assist the operator in performing these new tasks and managing the new information needed for these tasks.

Explanation is the common approach to providing visibility into the intelligent system. Most explanation systems operate retrospectively (like help systems), requiring the operator to wait until after a situation has stabilized (and the intelligent system has reached a conclusion) before attempting to get an explanation. In real-time support environments, the operator often cannot afford to wait until system behavior stabilizes, for the safety impacts may be too great. Such event reconstruction is also not sufficient for coordinating shared human-computer activities. Additionally, the conversational style of explanation can be distracting and can contribute to information overload.

Even if the operator had time to interrupt ongoing activities for an explanation, a problem would remain with traditional approaches to explanation. Affecting the operator's behavior requires that the operator both ***understand*** the meaning and consequences of the explanation, and ***accept*** them as correct. Most explanation systems assume that failure to influence user behavior occurs because the user does not understand the explanation, and continue to provide more detailed justification directed at improving understanding. Contrary to this assumption, acceptance does not necessarily result from understanding. The user may understand the intended meaning but choose not to believe it, due to information unknown to the intelligent system or not considered by it. Or the user may believe the information but be unwilling to alter behavior, due to the belief that adverse side-effects will result or that the consequences of altering behavior are of no significance. Typical explanation approaches ignore the better information available to expert users such as flight controllers. Explanations should provide the kind of intelligent system visibility that effectively supports real-time detection of intelligent system anomalies, and even diagnosis and formulation of responses.

Thus, alternatives to explanation should both avoid the need for retrospective dialogue and support the user's supervisory task. A promising approach is for the human and intelligent system to share information and representations. Using the same information, the user can follow operational situations and compare conclusions with intelligent system assessments, as a part of normal monitoring and control operations. A good first step is to clarify intelligent system reasoning by displaying plots or tables of critical evidence supporting its conclusions, as shown in Figure 2. This example also illustrates use of the following information to support user understanding of the monitored process that parallels the data used for intelligent system assessments:

- Present information describing the situation as it develops, including both monitored process behavior and intelligent system activities.

- Establish expectations about what might happen next (e.g., annotate data with predictions).

- Identify critical transitions and regimes of behavior (both current and pending; e.g., process or task information to annotate displays, Forbus, 1991).

Such information both supports the operator's tasks and provides some on-the-job training, by reinforcing the operator's mental models of both the monitored process and the intelligent system.

A representation shared between agents is prevalent as the basis of communication in many domains, including humans advising robots using a shared task representation (Martin and Firby, 1991), distributed machine agents planning tasks using shared goals (Decker, et al., 1991), and designers developing shared mental models (Sycara and Lewis, 1991). Shared representations are achieved by designing the intelligent system's representation to correspond to the expert operator's representation for performing the management tasks. To achieve this, it may be necessary to make implicit task information explicit in the intelligent system (e.g., to support robot advice-taking in Martin and Firby, 1991).

The concept of shared representations supports development of common knowledge between the user and the intelligent system. If intelligent system conclusions can be represented in a way that is self-evident to the user, such an intelligent system can become self-explanatory. With such a system, the user would not need to analyze the details of intelligent system reasoning. Less attention and time would be needed to effectively supervise the intelligent system.

CONCLUSION

The information problems described in this paper can cause many intelligent system design problems. A solution to these problems has been proposed based on designing from information requirements. Design recommendations have been made for improving human-computer communication of this information. The impact of these problems on intelligent system reliability and usability is now described, and the benefits of solving these problems delineated.

Reliability is a critical design issue, since an unreliable and uncontrollable intelligent system can impact the safety of crew and space systems. Intelligent systems that do not perform reliably cannot successfully provide real-time decision support. Thus, solving those problems affecting reliability should be of first priority to the intelligent system designer. Information problems affecting system reliability (*shown in italics*) and recommendations for solving these problems are summarized below:

The design may fail to support the user in supervising intelligent system activities and recovering from intelligent system errors.
Designing the intelligent system for supervision means keeping the user informed of its conclusions, behavior, capabilities, and reasoning strategies in the context of

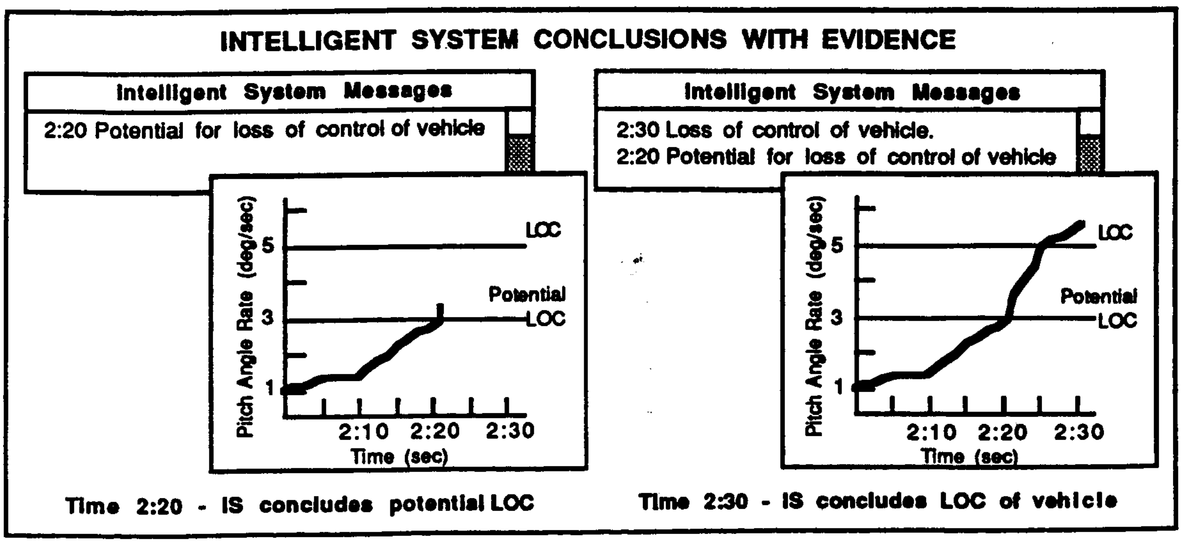

Figure 2 - Displaying Evidence Supporting Intelligent System Conclusions

ongoing events. It means providing the user with some recourse when intelligent system errors occur, such as reallocating intelligent system tasks to the user. To permit such task reallocation, information from data sources other than the intelligent system should be accessible, and independent of intelligent system conclusions.

The design may fail to handle bad or unavailable data, or unexpected situations.
Robustness to data deficiencies and unexpected situations is achieved by minimizing the bad data processed by the intelligent system and by providing capability to recover from errors. Methods include providing for data preprocessing and system self-correction (e.g., retract inconsistent conclusions), and designing for user supervision.

Intelligent system activities, reasoning strategies, and capabilities may be misunderstood and often overestimated by the user (i.e., a magical system).
To avoid building magical systems, it is necessary to provide dynamic feedback about ongoing system activities, and to reinforce the user's understanding of system capabilities. This information should be integrated into a display of the overall situation that develops over time. This permits the user to develop an understanding of system activities as they occur instead of trying to retrospectively develop such an understanding after problems occur.

Information problems also impact intelligent system usability. Intelligent systems that are difficult to use have an increased risk of user rejection and user errors. Solving problems affecting usability should improve the chances of intelligent system success. Information problems affecting system usability (*shown in italics*) and recommendations for solving these problems are summarized below:

Common practice in user interface design may increase user workload and fail to provide the important, task-relevant information (i.e., message lists, schematics, and explanation).
Deficiencies in user interface design result from misunderstanding what information is needed by the user. The user interface should be designed from a description of the task-based information requirements. This information should be presented to illustrate situations as they develop, including the behavior of both the monitored process and the intelligent system. Alternatives to explanation should clarify intelligent system conclusions and reasoning strategies, including the evidence supporting these conclusions.

The intelligent system may not be integrated with the support system.
Usually the intelligent system does not operate as a stand-alone system, but is instead embedded in a larger support system. Information from the intelligent system must be integrated with the sources of operational data, and the intelligent system displays must be integrated with other displays.

The intelligent system may not be designed for coordination with the user.
Designing for coordination requires avoiding unnecessary interruptions or interference in user activities. Changes in task allocation and dependencies between tasks (including required information exchange) represent points of coordination that constrain the design. An essential element of coordinating shared tasks is providing feedback about ongoing intelligent system activities.

Applying these recommendations improves safety and reduces cost. Building reliable intelligent systems reduces safety threats due to system error. Building usable systems reduces the potential for user error and improves user acceptance. This reduces the chance of the system not being used, and minimizes costly redesign of the system. The results are safer operations using intelligent systems and reduced cost of building these systems.

REFERENCES

Abbott, K., Internal memo, Human Automation Integration Branch, Langley Research Center, NASA, November 1991.

Abbott, K., Personal communication, Human Automation Integration Branch, Langley Research Center, NASA, 1992.

Chandrasekaran, B., Tanner, M., and Josephson, J., "Explaining Control Strategies in Problem Solving," IEEE EXPERT, spring 1989, pp. 9-24.

Decker, K. Garvey, A., Humphrey, M., and Lesser V., "Effects of Parallelism on Blackboard System Scheduling," Proc. IJCAI, Australia, August 1991.

Forbus, K., "Qualitative Physics as a Tool for Human-Computer Interaction," The Institute for the Learning Sciences, Northwestern University, Evanston, IL, 1991.

Kieras, D., "Towards a Practical GOMS Model Methodology for User Interface Design," HANDBOOK OF HUMAN-COMPUTER INTERACTION, North Holland: Elsevier, New York, 1988.

Land, S., Malin, J., and Culp, D., "DESSY: Making a Real-time Expert System Robust and Useful", Proc. Space Operations, Applications, and Research Symposium, Houston, TX, August 1992.

Malin, J., Schreckenghost, D., Woods, D., Potter, S., Johannesen, L., Holloway, M., and Forbus, K., "Making Intelligent Systems Team Players: Case Studies and Design Issues, Vol.1. Human-Computer Interaction Design; Vol.2. Fault Management System Cases," NASA TECHNICAL MEMORANDUM 104738, Johnson Space Center, Houston, TX, September 1991.

Malin, J. , and Schreckenghost, D., "Making Intelligent Systems Team Players: An Overview for Designers," NASA TECHNICAL MEMORANDUM at press, Johnson Space Center, Houston, TX, June 1992.

Martin, C., and Firby, R., "An Integrated Architecture for Planning and Learning," ACM SIGART BULLETIN, 2, 4, 1991, pp. 125-129.

Mitchell, C., and Miller, R., "A Discrete Control Model of Operator Function: A Methodology for Information Display Design," IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS, 16, 3, May/June, 1986, pp. 343-357.

Potter, S. and Woods, D., "Event Driven Timeline Displays: Beyond Message Lists in Human-Intelligent System Interaction," Proc. IEEE International Conference on Systems, Man, and Cybernetics, Charlottesville, VA, October 1991.

Rasmussen, J., INFORMATION PROCESSING AND HUMAN-MACHINE INTERACTION: AN APPROACH TO COGNITIVE ENGINEERING, North-Holland, New York, 1986.

Sycara, K., and Lewis, C., "Cooperation of Heterogeneous Agents through the Formation of Shared Mental Models," AAAI Workshop on Cooperation Among Heterogeneous Intelligent Agents, Anaheim, CA, July 1991.

Woods, D., Potter, S., Johannesen, L., and Holloway, M., "Human Interaction with Intelligent Systems: Volume I -- Trends, Problems, New Directions," CSEL REPORT 1991-001, Cognitive Systems Engineering Lab, Ohio State University, Columbus, OH, March 1991.