NUREG/CR-6635
BNL-NUREG-52565

Soft Controls:
Technical Basis and
Human Factors Review Guidance

Prepared by
William F. Stubler and John M. O'Hara / BNL
Joel Kramer / NRC

Brookhaven National Laboratory

Prepared for
U.S. Nuclear Regulatory Commission

NUREG/CR-6635
BNL-NUREG-52565

Soft Controls: Technical Basis and Human Factors Review Guidance

Manuscript Completed:
Date Published:

Prepared by
William F. Stubler and John M. O'Hara / BNL
Joel Kramer / NRC

Systems Engineering and Safety Analysis Group
Brookhaven National Laboratory
Upton, NY 11973-5000

Prepared for
Office of Nuclear Regulatory Research
U.S. Nuclear Regulatory Commission
Washington, DC 20555-0001
NRC JCN J-6012

ABSTRACT

In conventional control rooms, the predominant means for providing control input is via hard-wired, spatially dedicated control devices that have fixed functions. However, in human-system interfaces featuring computer-based technologies, the operator may interact via “soft” controls – devices having connections with control and display systems that are mediated by software rather than direct physical connections. Soft controls can have functions that are variable and context dependent rather than statically defined. For example, a particular action may produce different results based on the currently active mode of the control device. Also, device locations may be virtual rather than spatially dedicated. That is, personnel may be able to access a particular soft control from multiple locations within a display system. These characteristics provide new opportunities for operator errors and may affect operator response during time-critical tasks. The objective of this study was to develop human factors review guidance for soft control systems. A methodology for developing technically valid guidance was used. To support this objective, we developed a characterization framework for describing key design characteristics of soft control systems including: display devices, input devices, and methods of interaction. Then, we examined research in the following, areas (1) human error in soft control use, (2) general design approaches for error tolerance, and (3) human performance considerations associated with specific control actions. This research provided the technical basis upon which design review guidelines were developed for the following: display devices, input devices, information displays, and interaction methods. There were aspects of soft controls for which the technical basis was insufficient to support development of the guidance. These were identified as issues to be addressed in future research.

CONTENTS

Page

PART 1: Development and Technical Basis

CONTENTS (Continued)

Page

CONTENTS (Continued)

Page

PART 2: Review Guidance for Soft Controls

LIST OF FIGURES

Page

LIST OF TABLES

EXECUTIVE SUMMARY

The Human-System Interface Design Review Guideline, NUREG-0700, Rev. 1 (O'Hara, et al., 1996), was developed to provide guidance on human factors engineering (HFE) for the U.S. Nuclear Regulatory Commission (NRC). The NRC staff uses NUREG-0700 for (1) reviewing submittals of human-system interface (HSI) designs prepared by licensees or applicants for a license or design certification of a commercial nuclear power plant (NPP), and (2) undertaking HSI reviews that could be included in an inspection or other types of regulatory review of HSI designs, or incidents involving human performance. It describes those aspects of the HSI design review process that are important to identifying and resolving human engineering discrepancies that could adversely affect plant safety. NUREG-0700 also has detailed HFE guidelines for assessing the implementation of HSI designs.

In generating NUREG-0700, Rev. 1, several topics were identified as “gaps” because there was an insufficient technical basis upon which to formulate guidance. One such topic is the integration of advanced HSI technology into conventional NPPs. The NRC is currently sponsoring research at Brookhaven National Laboratory (BNL) to (1) better define the effects of changes in HSIs brought about by incorporating digital technology on personnel performance and plant safety, and (2) develop HFE guidance to support safety reviews, should a review of plant modifications or HSIs be necessary.

Based upon the literature, interviews, and site visits, O'Hara, Stubler, and Higgins (1996) identified changes in HSI technology and their potential effects on personnel performance. The topics were then evaluated for their potential safety significance (Stubler, Higgins, and O'Hara, 1996); soft controls was one HSI technology that was found to be potentially safety significant.

Soft controls are prominent features in the user interfaces of many digital systems in existing NPPs. Further, many plant systems that are upgraded with digital technologies are likely to have soft controls. However, soft controls have characteristics that provide new opportunities for operators to make errors, and may affect their response during time-critical tasks.

The objective of this study was to develop HFE review guidance for soft control systems based on a technically valid guidance development methodology. To support this objective, the following tasks were undertaken:

- Development of a characterization framework for describing key design characteristics of soft control systems

- Development of a technical basis using research and analyses of human performance that are relevant to soft control systems

- Development of HFE review guidelines for soft control systems in a format that is consistent with NUREG-0700, Rev. 1, and other NRC review guidance

- Identification of remaining soft control system issues for which research results were insufficient to support developing NRC review guidance

The status of each will be briefly addressed below.

Soft Control Characterization Framework

In conventional control rooms (CRs), the predominant means for providing control input is via hard-wired, spatially dedicated devices that exhibit a single function – single control philosophy (e.g., hand switches for pumps or valves). Each control typically has a single dedicated location in a control panel, and the control function it provides is always the same.

By contrast, in HSIs with computer-based technologies, the operator may interact with the plant via "soft" controls. Soft controls are input interfaces connected with control and display systems that are mediated by software, rather than by direct physical connections. Their functions may be variable and context dependent rather than statically defined. For example, a particular control action may produce different results based on the mode of the soft control. Also, devices may be located virtually rather than spatially dedicated. That is, personnel may be able to access a particular soft control from multiple places within a display system. Soft controls include the following:

- Devices activated from display devices, such as buttons and sliders on touch screens

- Multi-function control devices, such as knobs, buttons, keyboard keys, and switches that have different functions depending upon the current condition of the plant, the control system, or the HSI

- Devices operated via voice recognition

The characteristics of soft controls important to operator performance extend beyond the physical aspects of the input device. They include the characteristics of associated display devices, the presentation of information, and the methods of interaction between the operator and system under control. Soft controls were characterized along the following dimensions: general characteristics, display devices, input devices, and interaction methods.

Development of the Technical Basis

We consulted a broad range of sources in reviewing human performance concerns associated with soft controls, including general HFE literature on human-computer interactions. Also, we reconsidered general HFE literature on the control of complex human-machine systems, such as in NPPs, other process control industries, aviation, and medical devices. Another source was reports of incidents that resulted from human performance concerns associated with soft controls. These reports came from a variety of industries, especially NPPs, chemical manufacturing, and aviation. Yet another source of information was our interactions with industry personnel, including designers, operators, and trainers that took the form of interviews and walk-through exercises using the actual HSI or a high-fidelity training simulator. A variety of problems were identified from these sources ranging from accessing the wrong information, to making control inputs that were too big or small, to operating the wrong equipment. These problems were generally related to a lack of adequate feedback in the user interfaces of soft controls so that incorrect actions and their consequences were not always apparent to the operator. The more detailed analysis that was conducted in developing this guidance confirmed that these are important HFE issues that need to be addressed in the design of HSIs containing soft controls.

There was much general HFE information and documented industry experience to draw upon. First, theories and studies of human performance in complex human-machine systems that address such topics as HSI design, situation awareness, and human error were consulted. Second, general theory related to human-computer interaction was consulted; this literature described errors, especially slips, that occur with computer-based interfaces. Many of the descriptions of problems reported in process control industries lacked a structured treatment of human error. Reviewing these descriptions within the general framework of human error clarified the relationships between the soft control's characteristics, human performance effects, and system consequences. A third source of information was empirical studies of human-computer interaction; they tended to focus on less complex, user-paced activities, such as text processing. Their findings were considered when the HSI's characteristics and user tasks were relevant to process control and safety.

The review of industrial practices based on literature, interviews, and site visits indicated that the solutions implemented to resolve the problems of soft control varied among organizations, which may reflect the fact that there are few formal HFE guidelines or standards for soft controls. Different industries, manufacturing facilities, and HSI vendors have different approaches to similar problems. Many of the HSIs we observed had a variety of error-prevention measures, although they were not always implemented consistently within the same HSI. This suggested that the human performance problems associated with soft controls have not been adequately solved by industry.

Interestingly, there were relatively few empirical studies of the use of soft controls in process control settings. This was noted by others (Hoecker and Roth, 1996) and was reflected in the comments from HSI experts we interviewed. Further research may be warranted to confirm any guidance that reflects accepted design practices for human-computer interfaces but that may not have a strong empirical basis. In addition, further review and analysis may be warranted to support the development of guidance for topics that have not been, and are not likely to be, addressed by the more general field of human-computer interaction, and to expand on topics for which available technical information was inadequate to generate review guidance.

HFE Review Guidelines

Using the technical basis, guidelines for the review of soft controls were developed. These guidelines were designed in the standard format adopted in NUREG-0700, Rev. 1, The guidelines were organized into the following sections:

General Characteristics – The review guidelines in this section address design characteristics such as operator feedback, coordination of soft controls among operators, and operation with protective clothing.

Display Devices – The review guidelines in this section address design characteristics such as the provision of adequate display space.

Input Devices – The review guidelines in this section address design characteristics such as activation force and lift-off logic for pointing devices.

Display Design – The review guidelines in this section address design characteristics such as selection displays, input fields, and input formats.

Interaction Methods – The review guidelines in this section address design characteristics such as sequential actions; verification and confirmation steps; interlocks, lockouts, and lockins; error detection and correction; selecting plant variables or components; control inputs; handling stored data; and system response.

Soft Control Issues

As noted above, several human performance issues associated with soft controls were identified. They represent topics for which research is necessary before developing guidance. From a regulatory review perspective, many of them can be dealt with on a case-by-case basis during the design process review. Briefly, the issues included the following:

Time Delays and Control Stability – With the potential time delays in digital systems and the sequential nature of soft control actions, research is needed to better understand the relationship between time delays and stability of performance, especially in emergencies. Where delays affect performance, methods to support operators' performance should be identified.

Input and Feedback Methods for Continuous-Variable Inputs – Industry experience showed that entering numerical values is error prone, especially when using a keyboard or key pad. However, the popularity of the keyboard as an input device suggests that it may have some advantages (such as speed) compared to other methods, such as arrow keys and soft sliders. Feedback about the magnitude of entered values can support the detection and correction of input errors; two common methods are digital readouts and barcharts. More information is needed on the relative advantages of combinations of input and feedback methods.

Confirmation and Warning Messages – Both confirmation and error messages are prone to problems associated with the level of specification of operators' actions. For example, operators may confirm that the desired action is correct but not realize that the goal (e.g., the object being acted upon) may be wrong. Similarly, when receiving an error or warning message, users often cannot interpret the true cause of the problem.

Sequential Plant Control and Interface Management Tasks – Many plant control tasks are sequential, and different tasks can have similar but different sequences. For example, some pumps require closing the downstream valve before starting the pumps. Other pumps require that it be opened. In addition, sequential operations are often involved in the use of soft controls (e.g., the operator must access a selection display, select a component, open an input field, and then enter the input value). Industry experience suggests that the sequential constraints of soft control access can interact with the sequential nature of control tasks and increase the likelihood of capture errors (i.e., starting one task sequence and finishing with another) and misordered action sequences (i.e., performing actions in the wrong sequence).

Access to One Versus Multiple Input Fields at One Time – More research may be needed on the potential benefits and costs of providing access to one input field at a time versus multiple input fields. Some alternatives may include having displays giving access to groups of controls, tools for managing multiple open input fields, and methods for performing serial access more quickly and accurately.

Intelligent Agents – These are computer programs that perform information processing tasks for the operator somewhat autonomously. They are being developed to perform information management tasks in chemical plants with a user-initiated notification concept. Intelligent agents can help the operator manage suspended tasks. However, the potential benefits must be weighed against the operator's burdens in supervising these agents, and any potential problems that may result from their inappropriate application.

Interaction of Soft Controls with Automation – Increases in automation of computer-based systems pose greater cognitive demands on operators, especially for understanding and maintaining awareness of the status and behavior of these systems. Soft controls play an important role in conveying status information to operators and allowing them to interact with the systems. However, automation may also affect the appearance and behavior of controls and displays. Human factors review guidance is needed to address the interaction of soft controls with automation.

Soft Controls and Display Space – The amount and type of display space provided through the HSI is important for supporting control and monitoring tasks. For example, assigning controls to dedicated display devices can improve access time by reducing the need to navigate through displays. Increasing the number of display devices can reduce conflicts between demands for short-term control actions and long-term monitoring actions. Also, having additional display devices allows the operator to more easily keep track of temporarily suspended tasks. Human factors review guidance is needed to look at the minimum amount of display space needed to support soft control use, and also the tradeoffs between providing dedicated display devices and general-purpose ones.

Keyboards Versus Incremental Input Devices – Many soft controls used in process control applications provide the operator with the choice of changing control values via arrow buttons or via a keyboard. Keyboard entry may offer some performance benefits; however, industry experience suggests that entry via keyboard is more error prone. For example, large errors may result from typing mistakes. Further research is needed to examine the error rates associated with data entry via keyboards versus incremental input devices, especially when used in conjunction with features used for error prevention, detection, correction, and recovery.

Consistency of Soft Controls in Hybrid HSIs – A hybrid HSI may contain a variety of soft controls, especially if they are installed as a series of independent modifications, rather than in an integrated effort. In such a hybrid HSI, operators are expected to make frequent switches between different tasks with different interfaces. Studies of computer-based systems have produced some conflicting results on the effects of consistency. Thus, the goal of trying to maximize consistency between user interfaces may be counterproductive if the wrong type of consistency is achieved. Further research is needed to understand the dimensions of consistency that are important for reducing errors and ensuring effective operator performance across a variety of soft controls in a hybrid HSI.

PREFACE

This report was prepared by Brookhaven National Laboratory for the Division of Systems Technology of the U. S. Nuclear Regulatory Commission (NRC) Office of Nuclear Regulatory Research. It is submitted as part of the requirements for the project, “Human Factors Topics Associated with Hybrid Human System Interfaces” (NRC JCN J-6012), specifically as part of Task 3, “Develop Review Approaches.” The NRC Project Manager is Joel Kramer and the BNL Principal Investigator is John O'Hara.

ACKNOWLEDGMENTS

The authors wish to express their sincere gratitude to our colleagues Lew Hanes; Mike Fineberg and his review team at the Crew Systems Ergonomics Information Analysis Center (CSERIAC); and Bill Ruland, Greg Galletti, Jim Bongarra, Clare Goodman, Jim Stewart, and Rich Correia of the U.S. NRC for their review of the reports. These reviewers provided insightful comments and perspectives on the issues addressed in the report and their knowledge and understanding significantly contributed to the study.

We also wish to thank Barbara Roland, Mary Anne Corwin, and Avril Woodhead for their preparation and careful technical editing of the report.

ACRONYMS

ABB-CE	ASEA-Brown Boveri - Combustion Engineering
BNL	Brookhaven National Laboratory
BWR	Boiling water reactor
CR	Control room
CRT	Cathode ray tube
EPRI	Electric Power Research Institute
HFE	Human factors engineering
HSI	Human-system interface
NPP	Nuclear power plant
NRC	Nuclear Regulatory Commission
PMS	Plant Protection and Safety Monitor System
PWR	Pressurized water reactor
QDPS	Qualified Data Processing System
SPDS	Safety parameter display systems
VDU	Visual display unit

PART 1

Development and Technical Basis

1 INTRODUCTION

1.1 Background

The goal of this project is to develop human factors engineering (HFE) guidance to support safety reviews of hybrid human-system interfaces (HSIs) conducted by the U.S. Nuclear Regulatory Commission (NRC). In Task 1 of this project, Brookhaven National Laboratory (BNL) identified 14 human factors topics associated with hybrid HSIs (O'Hara, Stubler, and Higgins, 1996). In Task 2, they were consolidated into a set of ten, evaluated for their potential safety significance, and then prioritized. The methodology and results from Task 2 were documented in a technical report (Stubler, Higgins, and O'Hara, 1996). Based on this analysis, the NRC requested that human factors engineering (HFE) guidance be developed for several topics. This report documents the development of guidance for soft controls.

The results of this project will contribute to satisfying the NRC's goals of (1) maintaining safety, (2) increasing public confidence, (3) increasing regulatory efficiency and effectiveness, and (4) reducing unnecessary burden.

In conventional control rooms (CRs), the predominant means for providing control input is via hard-wired, spatially dedicated devices that have fixed functions (e.g., hand switches for pumps or valves). By contrast, in HSIs with computer-based technologies, the operator may interact with the plant via "soft" controls. These are control devices having connections with control and display systems that are mediated by software rather than direct physical connections. Consequently, their functions may be variable and context dependent rather than statically defined. For example, a particular control action may produce different results based on the mode of the soft control. Also, devices may be located virtually rather than spatially dedicated. That is, personnel may be able to access a particular soft control from multiple places within a display system. Soft controls include the following:

- Devices activated from display devices, such as buttons and sliders on touch screens

- Multi-function control devices, such as knobs, buttons, keyboard keys, and switches that have different functions depending upon the current condition of the plant, the control system, or the HSI

- Devices operated via voice recognition

Soft controls are prominent features in the user interfaces of many digital systems in existing nuclear power plants (NPPs), including the following:

- Plant control systems, such as moisture separator reheater systems in pressurized water reactor (PWR) plants (Meter and Olsen, 1996), and feedwater control systems in boiling water reactor (BWR) and PWR plants

- Plant protection systems, such as the ABB-CE Core Protection Calculator and the Westinghouse Eagle-21

- General Electric (GE) NUMAC digital upgrades for safety systems, including a wide range of radiation and neutron monitoring and control systems for BWRs and PWRs (Bhatt, 1992)

- GE FANUC digital upgrades for non-safety control systems, including makeup water treatment, radwaste control, and recirculation control (Bhatt, 1992)

- Cathode ray tube (CRT)-based monitoring systems, such as the Westinghouse Plant Protection and Safety Monitor System (PMS), Qualified Data Processing System (QDPS), and safety parameter display systems (SPDS)

Many plant systems that are upgraded with digital technologies are likely to have soft controls. However, soft controls have characteristics that provide new opportunities for operators to make errors, and may affect their response during time-critical tasks.

1.2 Organization of the Report

The report is divided into two parts. Part 1 describes the methodology used for developing guidance, and its technical basis. The objective of the study is described in Section 2 and the methodology in Section 3. Section 4 characterizes soft controls. Section 5 discusses the literature and information that was the technical basis for developing the review guidance. The way in which the technical information was used is discussed in Section 6. Section 7 summarizes the development process by describing the types of guidance developed – and the topics for which there are gaps – and makes recommendations for developing further guidance. The references cited are given in Section 8. Part 2 contains the specific HFE guidance for safety reviews of soft controls.

2 OBJECTIVE

The objective of this study was to develop HFE review guidance for soft control systems based on a technically valid guidance development methodology. To support this objective, the following tasks were undertaken:

- Development of a characterization framework for describing key design characteristics of soft control systems

- Development of a technical basis using research and analyses of human performance that are relevant to soft control systems

- Development of HFE review guidelines for soft control systems in a format that is consistent with NUREG-0700, Rev. 1, and other NRC review guidance

- Identification of remaining soft control system issues for which research results were insufficient to support developing NRC review guidance

3 METHODOLOGY

3.1 Overview

Figure 3.1 shows the overall methodology for developing guidance for NUREG-0700. The process is discussed in detail elsewhere (O'Hara, Brown, and Nasta, 1996; Stubler and O'Hara, 1996a). The portion of the methodology applicable to this report and project is boxed in the figure. This section of the report describes the general rationale behind guidance development.

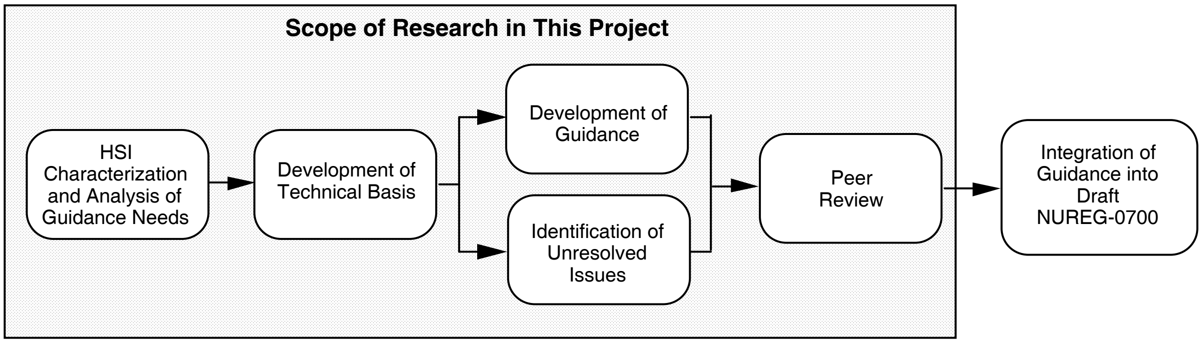

Scope of Research in This Project
HSI Characterization and Analysis of Guidance Needs
Development of Technical Basis
Development of Guidance
Identification of Unresolved Issues
Peer Review
Integration of Guidance into Draft NUREG-0700

Figure 3.1 Major Steps in Developing NUREG-0700 Guidance

The methodology for guidance development was directed by the following objectives:

- Establish a process that will result in valid, technically defensible, review criteria.

- Establish a process that can be generalized for any aspect of HSI technology needing review guidance.

- Establish a process that optimally uses available resources; i.e., a cost-effective methodology.

The methodology places a high priority on establishing the validity of the guidelines. Validity has both internal and external dimensions. Internal validity is the degree to which the individual guidelines are established and justified based on auditable technical information. The technical bases vary for individual guidelines. Some may be based on technical conclusions from a preponderance of empirical research, some on a consensus of existing standards, while others are based on judgement that a guideline represents good practices. Maintaining an audit trail from each guideline to its technical basis serves several purposes by enabling

- the technical merit of the guideline to be evaluated by others,

- a more informed application of the guideline since its basis is available to users, and

- deviations or exceptions to the guideline to be evaluated.

External validity is the degree to which the guidelines undergo independent peer review. Peer review is a good method of screening guidelines for their conformance to accepted HFE practices, and for comparing them to practical operational experience of HSIs in real systems.

For individual guidelines, these forms of validity can be inherited from the source documents in their technical basis. Some HFE standards and guidance documents, for example, already have good internal and external validity. However, if validity is not inherited, it should be established as part of the guidance development process. The NUREG-0700 methodology for guidance development was established to provide validity, both inherited from its technical basis, and through guidance development and evaluation.

Figure 3.2 depicts the process used to develop the technical basis and guidance. It emphases those information sources with the highest degree of internal and external validity for developing the technical basis. Thus, primary and secondary source documents were sought as sources of guidance first, followed by tertiary source documents, basic literature, industry experience, and others. From these sources, we identified design principles and lessons from industry experience. The guidance was developed using this technical basis as a foundation. For specific aspects of the topic, in which there was an inadequate technical basis to develop guidance, unresolved research issues were defined. Thus, both guidance and issues were established. The resulting documentation includes HFE guidelines, the technical basis, the development methodology, and unresolved issues.

Each step of this work – topic characterization, technical basis development, guidance development and documentation, issue identification, and peer review – is discussed in greater detail in the sections that follow.

3.2 Characterization of Soft Controls

The first step in the guidance development process was to identify areas needing guidance. Existing soft control systems were reviewed to identify the dimensions and characteristics along which to define them, and to highlight features important to personnel's performance. The characterization was important because it provided a structure within which the reviewer could request information about a system and structure the design review guidance. Section 4 describes the characterization of soft controls.

3.3 Development of Technical Basis

The development of detailed review guidelines began by collecting technical information upon which they would be based (see Figure 3.2). The process was designed to develop valid guidance in the most cost-effective manner. First, primary source documents were sought; these were HFE standards and guidance documents having both internal and external validity. The primary source documents generally had their own research bases, and the authors had used their knowledge and expertise in considering this research and operational experience to develop HFE guidelines. They often had been extensively peer reviewed, and they added tremendous value to individual research reports. They were developed by experts who consider the applicability and generalizability of research to real systems, include knowledge and expertise gained through operational experience and application of guidance, and modify the guidance based on extensive peer review. Such documents gave us a valuable starting place. However, many aspects of soft controls extended beyond the technology and human performance considerations covered by primary documents. In addition, conducting original research was outside the scope of this current project. Therefore, much emphasis was placed on relevant secondary sources, tertiary sources, basic literature, and industry experience (Figure 3.2).

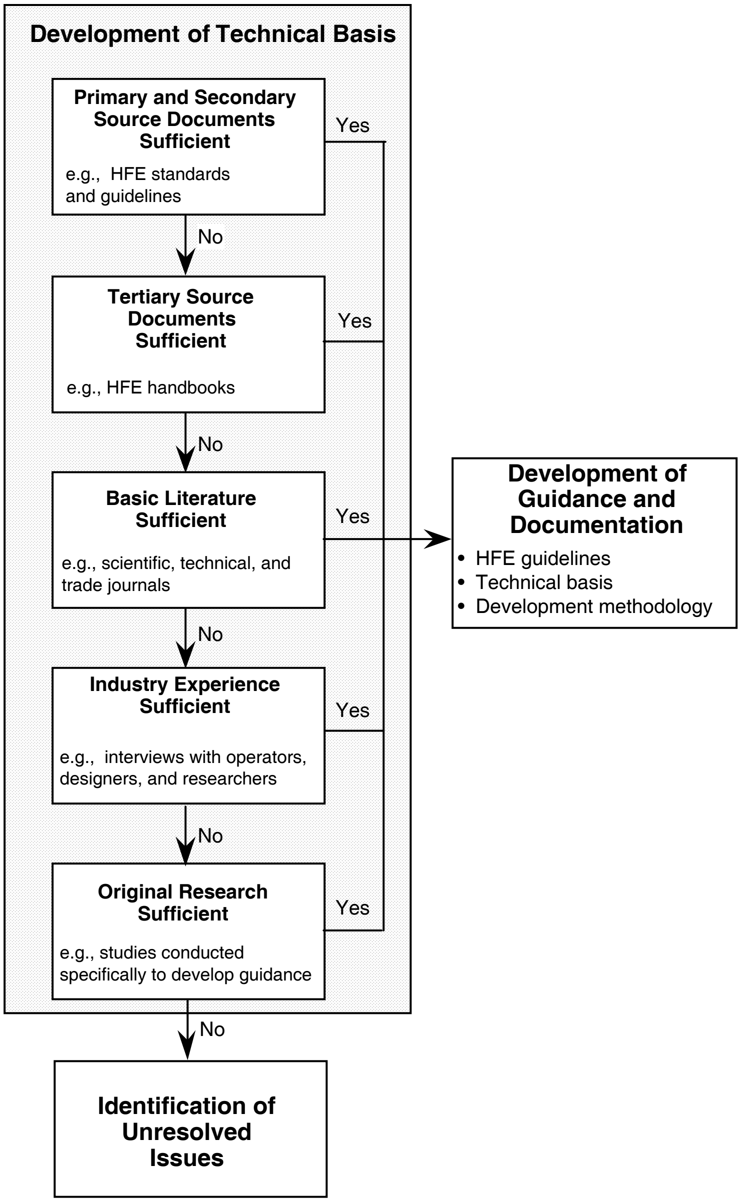

Development of Technical Basis
Primary and Secondary Source Documents Sufficient
e.g., HFE standards and guidelines
Yes
No
Tertiary Source Documents Sufficient
e.g., HFE handbooks
Yes
No
Basic Literature Sufficient
e.g., scientific, technical, and trade journals
Yes
No
Industry Experience Sufficient
e.g., interviews with operators, designers, and researchers
Yes
No
Original Research Sufficient
e.g., studies conducted specifically to develop guidance
Yes
No
Development of Guidance and Documentation
• HFE guidelines
• Technical basis
• Development methodology
Identification of Unresolved Issues

Figure 3.2 Technical Basis and Guidance Development Process

Secondary sources were documents for which either internal or external validity was established. They included human factors guidelines and standards developed for complex human-machine systems that were not classified as primary sources. Documents having neither internal nor external validity were considered tertiary sources; they included several human factors handbooks and texts. The basic literature consisted mainly of papers from research journals and technical conferences. The tertiary sources and basic literature gave a theoretical basis for understanding human performance related to complex human-machine systems, and a general theory for human-computer interactions involving soft controls, addressing user interface design, human error, and usability. Empirical studies of human-computer interactions in the basic literature covered a broad range of technologies and user tasks. For these, their applicability to NPP operations required engineering judgement because individual experiments tended to have unique constraints limiting their generalizability (such as their unique participants, types of tasks performed, and types of equipment used). Empirical studies were considered for the technical bases if they discussed the characteristics of user interface and tasks relevant to process control and safety applications. Industry experience included reports of incidents from a variety of industries.

In addition, much information about industrial problems and practices was obtained from interviews with industrial operators, designers, and trainers, and walk-through exercises[1] using the actual HSI or a high-fidelity training simulator. The following were contacted via either site visits or telephone interviews:

- ASEA-Brown Boveri - Combustion Engineering (HFE personnel)
- Atomic Energy of Canada Limited (HFE and HSI design personnel)
- Westinghouse Electric Corporation (HFE and HSI design personnel)
- Two foreign NPPs with computer-based HSIs (operators, trainers, and HSI design personnel)
- One domestic NPP upgraded with digital control systems (operators and design personnel)
- Five chemical plants with computer-based HSIs (operators, supervisors, and HSI design personnel)
- Two fossil power plants with computer-based HSIs (operators and HSI design personnel)

Industry practices included design approaches that have evolved through experience with computer-based HSIs. They were incorporated into the technical bases as practical examples of the HSI design strategies.

3.4 Development and Documentation of Guidance

Once the technical information was assembled, a set of guidelines was developed, and organized in a standard format (discussed in Section 6). The guidelines are presented in Part 2 of this document.

3.5 Identification of Issues

[1] Site visits conducted during this project included interviews and demonstrations of important features of the HSI. Some sites were revisited in related research sponsored by the U.S. NRC (Effects of User Interface Management on Crew Workload and Performance, JCN W 6546-6). Walk-through exercises at that time provided valuable insights into many human factors aspects of computer-based HSIs, including those associated with using soft controls.

Where there was insufficient information to provide a technical basis for developing valid design review guidance, an issue was defined. These issues are described in Section 7 of this report.

Issues reflect aspects of soft control design and use that require additional research to resolve. From a design review standpoint, they reflect aspects of soft control design and use that will have to be addressed case-by-case. For example, an issue can be addressed as part of design-specific tests and evaluations.

3.6 Peer Review

The resulting document containing the technical basis and guidance was submitted for review by knowledgeable individuals, including NRC personnel with expertise in human factors engineering and engineering fields directly related to the topic. In addition, they were reviewed by human factors specialists external to the NRC who have expertise in human performance in complex systems, such as NPPs and aviation. These external reviews included evaluations of the topic characterization along the following criteria: clarity, accuracy, and completeness, and the review guidance along the following criteria: organization, necessity, sufficiency, resolution, and basis. Comments from the peer reviews were incorporated into the present version of this document.

4 CHARACTERIZATION OF SOFT CONTROLS

The HSI provides the medium through which operators execute control actions and receive feedback on their effects. In conventional CRs, the predominant means for providing control input is through controls designed with a single function – single control philosophy. Each control typically has a single dedicated location in a control panel, and the control function it provides is always the same. Soft controls are input interfaces connected with control and display systems that are mediated by software, rather than by direct physical connections. Their functions may be variable and context dependent rather than statically defined. The characteristics of soft controls important to operator performance extend beyond the physical aspects of the input device. They include the characteristics of associated display devices, the presentation of information, and the methods of interaction between the operator and system under control.

This section gives a framework for describing the general design characteristics of soft controls that affect operator performance. It is arranged in four sections: general characteristics, display devices, input devices, and interaction methods. For clarity, we illustrate these characteristics by contrasting them with those of conventional controls.

4.1 General Characteristics

The following describes the unique general characteristics of soft controls compared to conventional ones.

Multiple Locations for Access – A conventional control typically has a unique location in the CR, and is used to control a specific plant variable (i.e., a quantity representing the status of a plant system or process). However, when plant variables are stored in computers, they may be accessed from computer-based devices in the HSI. A soft control for a particular variable may have many locations in the HSI. It may be accessed from more than one display device, and from multiple display pages within a display. Thus, soft controls lack the static spatial dedication characteristic of conventional controls.

Parallel Versus Serial Access – Conventional controls provide parallel access; all are visible at the same time. Thus, operators can visually scan controls to observe their status. Computer-based HSI components usually contain more displays and controls than can be viewed at one time via the display devices; because only part of the total set of displays can be viewed at once, the interface may create a keyhole effect. That is, the operator is forced to view the displays in small sequential views, similar to looking into a room through a keyhole. By limiting the number of soft controls that can be viewed or used at once, the keyhole effect forces serial rather than parallel access.

Present Versus Available – Conventional controls are spatially dedicated, and therefore continuously present in the CR. Soft controls may be designed to be continuously present, or to be retrieved from a display system when needed. In the latter case, they may be considered to be available but not present. In addition, the availability of soft controls may be restricted to specific conditions. For example, some soft controls, such as those used for configuring digital control systems, may have protective features (e.g., password protection) that limit their availability to specific personnel or situations.

Physical Decoupling of Input and Display Interfaces – Conventional controls typically have closely coupled input and display interfaces; that is, operators perform the input actions and monitor feedback at the same location. For example, the operator turns a rotary dial, observes its motion, and reads the new setting (the feedback) from its perimeter (i.e., where a pointer lines up with a value). However, with soft controls the location of control action may not be closely coupled with the presentation of feedback; the operator may take a control action in one place and read the setting elsewhere. For example, to use a pointing interface to perform a control action, the operator may manipulate a mouse on the surface of a console, causing a cursor to move across a nearby display screen and perform an action, such as selecting an icon. The results of this action may be displayed in yet another location, such as a window, indicating that a piece of plant equipment has been turned off or on. Here, the operator must monitor three locations to complete the control action: the mouse, the icon, and the window. This physical decoupling of the input device and the feedback displays may generate different monitoring demands to those for conventional controls.

Plant Control Versus Interface Management Control – In computer-based HSIs in which the keyhole effect obscures controls and displays, operators typically navigate displays and carry out retrieval actions to access them. Thus, actions controlling the HSI (i.e., causing displays to be presented) can be distinguished from actions that control the plant. These two types of actions may be undertaken with the same or different input and display devices. For example, an operator may use a mouse and CRT to access a display that contains a pump, and then use them to send a signal to operate the pump; in this case, the mouse and CRT operate both the HSI and the plant. Conversely, an operator may use a mouse and CRT to access the display and then use a keyboard with the CRT to operate the pump; i.e., the mouse operates the HSI, and the keyboard operates the plant.

Multiple Modes – While a conventional control typically performs a single control function, a soft control may perform a range of them each representing a different mode of a soft control device (e.g., mode 1 for performing function A, and mode 2 for performing function B). The software defines behavior of these functions. Options for control actions are usually communicated to the operator via displays. When the operator takes a control action, the software converts the results into a signal for the control system. Hence, a specific action, such as pressing a button, can produce different results depending on such factors as the particular display page currently accessed, the status of the control system, and the status of the plant.

Software-Defined Functions – Because operators' actions are interpreted by software, many operations may be initiated via a single action using a soft control. For example, a sequence of operations required for starting plant equipment may be linked to a single "Start" command. While conventional control systems also have this capability to some degree (e.g., via relays), software-defined functions can allow systems designers to develop more complexly linked operations.

Interface Flexibility – Computer-based technology can allow the user interface of soft controls to be adapted to changing needs or conditions of use. For example, the operator may be able to arrange the presentation of the control and its associated information for a current need or personal preference. Alternatively, the control and information may be automatically arranged, based on the current situation.

4.2 Display Devices

Soft controls are usually associated with display devices showing the variable that is being controlled, its current value or state, and the input provided by the operator. A distinction can be made between devices that are functionally dedicated and general purpose ones. The former is used for a specific function or set of functions. For example, a dedicated display device may be used only to interact with a particular system, such as feedwater control. A general purpose display device may be used to interact with a broad range of systems.

Soft controls may be implemented on a variety of visual display unit (VDU) hardware. Two common display technologies are CRTs and flat panels (e.g., light-emitting diode panels, plasma panels, thin film electroluminescent panels, electrochromics, electrophoretics panels, and liquid-crystal panels).

4.3 Input Devices

Many types of input devices may be used with soft controls; the following describes some common ones. Because the pace of innovation for computer-based technologies is rapid, this should not be considered an exhaustive listing. NUREG-0700, Rev. 1 describes human factors considerations associated with these devices.

Pointing devices allow operators to perform input actions upon displayed information by direct manipulation (Hutchins et al., 1986; Shneiderman, 1982). Pointing interfaces include the following:

Touch Screen – A control device that allows the user to communicate with the computer by touching a screen.

Light Pen – A pencil- or pen-like control device that interacts with the computer system through the display screen either by emitting or sensing light.

Mouse – A control device whose movements across a flat surface are converted into analogous movements of the cursor across the screen.

Trackball – A control device with which the user can control cursor movement in any direction by rotating a ball.

Joystick – A stick-type control device that can continuously control the cursor in any direction on a display screen.

Graphics Tablet – A device used to convert an image into digital code by drawing or tracing with a pen-like or puck-like instrument. The instrument is moved across the tablet generating a series of X-Y coordinates (also called a digitizing tablet).

The touch screen and light pen allow the operator to point directly to items on the display screen. The other four input devices have a less direct coupling between the input and display devices; the input device is used to position a cursor, which points to items on the display screen.

Other input devices may be used to interact with the soft controls via symbols and commands, rather than through direct manipulation; these include the following:

Alphanumeric Keyboard – A key pad used for typing letters or numbers into the computer. It can provide a variety of inputs, such as commands and numerical values.

Function Key – A key whose activation will affect a control entry. Detection of the signal usually causes the system to perform some predefined function.

This description assumes that the keyboard or function key is a separate physical device. If either one is presented on a display screen, it may be operated by a pointing interface.

Multi-function input devices may also be used; these include physical input devices, such as buttons, switches, and dials, used to access multiple plant variables. For example, a physical button located next to a display screen may perform different functions depending upon the information presented on the screen. Finally, soft controls may also be operated via voice input devices.

4.4 Methods of Interaction

4.4.1 Selecting Plant Variables or Components

Operators use controls to manipulate plant components, such as pumps, valves, and breakers, and thereby affect plant processes. These changes are reflected in plant variables. For example, a control action performed on a pump will change the variable that indicates whether the pump is "On" or "Off," and the variable giving the flow rate downstream of the pump. Thus, the control action may be described as controlling the component or the variable. The result is presented by the user interface of the control.

If a soft control is dedicated to a specific variable, then no separate step is required to specify the variable or the component that is to be manipulated. Selection is implicit in the act of selecting the device. However, many soft controls can control more than one plant variable or component. Therefore, a separate selection step is required; there are many methods to do this. Typically, the operator is required to identify a choice from memory, or to select one from a set of options.

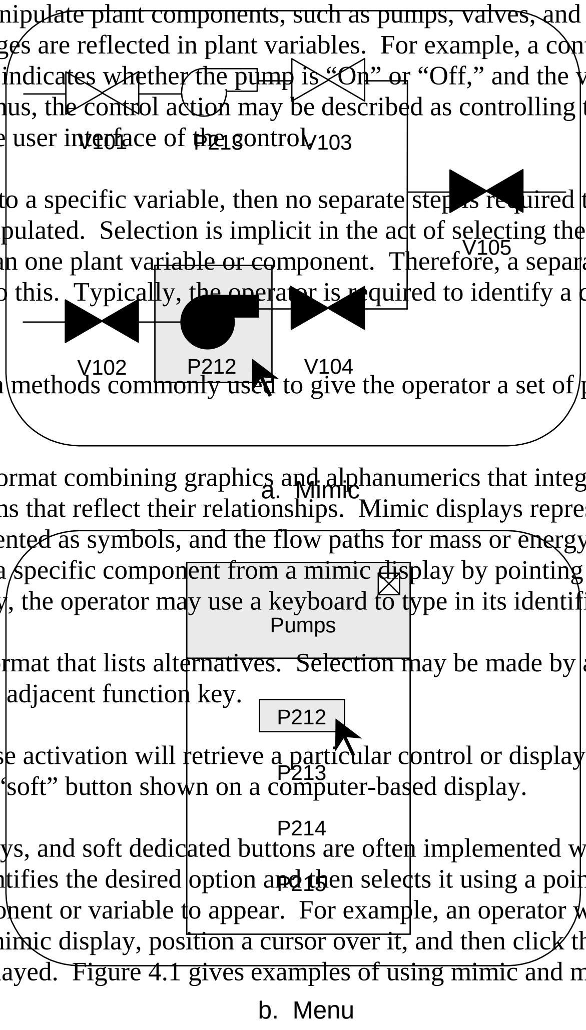
The following are interaction methods commonly used to give the operator a set of possible options via the HSI's user interface.

Mimic Display – A display format combining graphics and alphanumerics that integrates system components into functionally oriented diagrams that reflect their relationships. Mimic displays represent plant systems schematically. Plant components are represented as symbols, and the flow paths for mass or energy typically are represented with lines. Operators may select a specific component from a mimic display by pointing to it using a cursor or touch-screen interface; alternatively, the operator may use a keyboard to type in its identification code.

Menu Display – A display format that lists alternatives. Selection may be made by actions such as pointing and clicking, or by depressing an adjacent function key.

Dedicated Key – A key whose activation will retrieve a particular control or display. A dedicated key may be a physical "hard" button, or a "soft" button shown on a computer-based display.

Mimic displays, menu displays, and soft dedicated buttons are often implemented with direct manipulation interfaces. The operator identifies the desired option and then selects it using a pointing device that causes a display containing the desired component or variable to appear. For example, an operator wishing to close a valve may identify the valve icon in a mimic display, position a cursor over it, and then click the mouse button, causing a special input field to be displayed. Figure 4.1 gives examples of using mimic and menu displays for selecting components or variables.



a. Mimic

b. Menu

Figure 4.1 Two Typical Displays for Selecting Variables or Components (With On-Screen Cursor)

The following forms of interaction generally require the operator to choose from memory.

Command Language – A type of dialogue in which a user composes entries, possibly with minimal prompting by the computer.

Natural Language – A type of dialogue in which users compose control entries in a restricted subset of their natural language, e.g., English.

Query Language – A type of dialogue in which users compose questions using a special-purpose language to retrieve information.

Question and Answer – A type of dialogue in which a computer displays questions, one at a time, for a user to answer.

These methods of interaction may be augmented with online forms and other operator aids to support the operator in composing entries. Input is often made via alphanumeric keyboards, but other mediums, such as voice, may be used.

4.4.2 Providing Control Inputs

Providing control inputs often requires at least two steps: accessing the input field, and providing the control inputs.

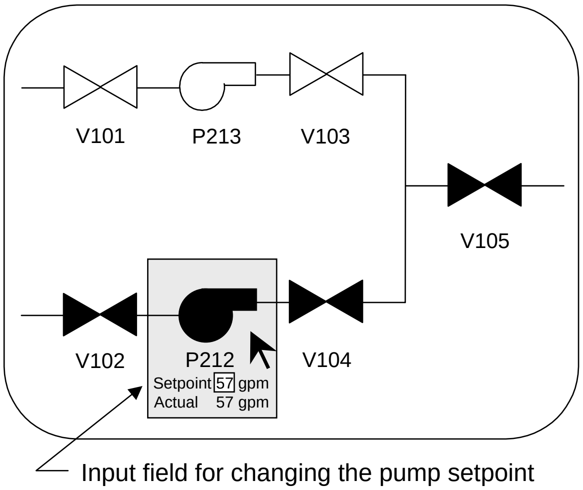
Input Fields

Input fields are areas of the display that are used by operators to enter input values to the control system. A soft control typically presents the input field in one of the following configurations:

Integral with the display – The operator provides input directly on the same display used to select the control device. For example, an operator may open or close a valve by clicking on its icon in a process display. Adjustment of the display screen may not be necessary because no new input window is introduced. Figure 4.2 gives an example. [Also see Degani et al., (1992).]

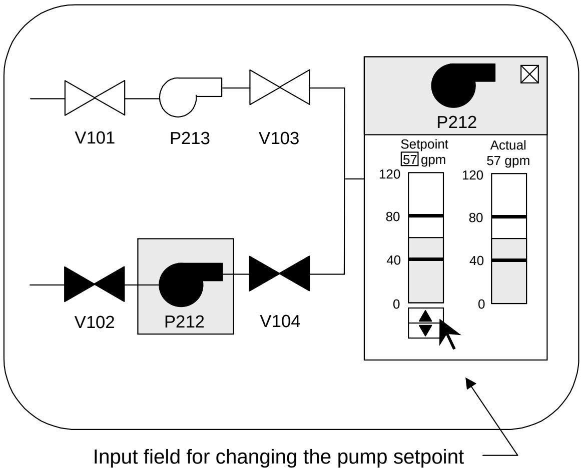
As a window within the display – A window to accommodate input values appears on the screen that was used to select the control. For example, an operator may select a component to control from a mimic display by clicking on it with a mouse; this causes an input window to be positioned within the display. This window may have a dedicated space in the display, or it may overlap and obstruct part of the existing display. Figure 4.3 gives an example. [Also see Hoecker and Roth (1996).]

As a display in a separate screen – After a soft control has been selected, an input display appears on a separate display device, usually located near the first one. An example is shown in Figure 4.4. [Also see Orendi (1996).]



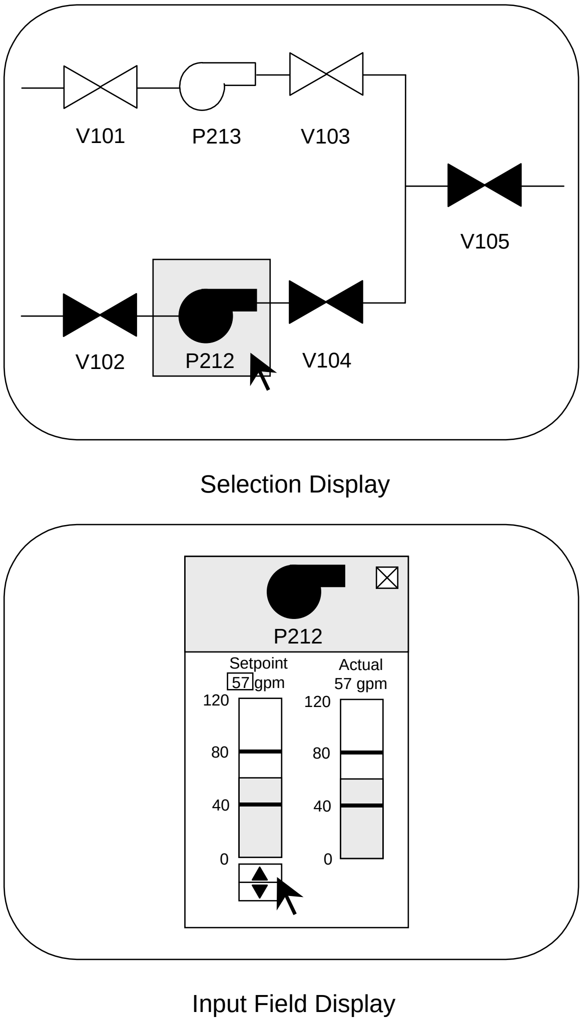
Input field for changing the pump setpoint (Setpoint is changed by keyboard after component is selected via cursor)

Figure 4.2 Soft Control Input Field Is Integral with Selection Display



Input field for changing the pump setpoint (Setpoint is changed by keyboard or by arrow buttons using a cursor)

Figure 4.3 Soft Control Input Field Is a Window Within the Selection Display



Selection Display

Input Field Display

Figure 4.4 Soft Control Input Field and Selection Display Are on Separate Devices

The input field configurations depicted in Figures 4.3 and 4.4 are more commonly used in process control applications than the integral configuration in Figure 4.2 because they provide more space for displaying setpoints and related values.

<u>Control Inputs</u>

Once an input field has been accessed, several different types of inputs can be entered to change the state of the plant. Three general categories are described below.

Command – A command is an instruction to a computer or system requesting it to perform an action, usually associated with plant information. For example, commands may be given to obtain, transfer, process, store, retrieve, delete, or display information about the plant status. Commands may also be used to control the plant (e.g., as an instruction to an automatic control system to perform a function, such as shutting down a piece of equipment).

Discrete-Variable Inputs – Many control actions involve selecting from a discrete set of states; breakers and valves may be changed from open to closed. Automatic controllers have discrete control modes (e.g., manual, automatic, and cascade). In addition, controls used for interface management may have discrete settings. For example, buttons may be pushed to access particular displays. A broad range of methods allow selection of discrete states. Input formats used for entering discrete-variable inputs are referred to as discrete-adjustment interfaces. They have individual settings that usually can be accessed using gross movements. Their operation is similar to physical controls that provide discrete-adjustment (Chapanis and Kinkade, 1972), such as push buttons and switches.

Continuous-Variable Inputs – Many control actions involve providing a value from a continuous range. One important example for process control is inputting a control setpoint. When operators change these setpoints, they typically select a value from a range which causes the setpoint to increase or decrease relative to its current value. Many controls in NPPs are unidimensional; they control only one variable at a time. However, there are controls that operate multiple variables. For example, a high-level controller for pressurizer pressure may control multiple variables, such as sprayer flow and heater temperature. In this case, the user gives input to one variable (pressurizer pressure) which, in turn, affects the other variables (flow and temperature).

For physical control devices, continuous variables are often set with continuous-adjustment control, such as rotary dials and slider switches in which values are accessed by some type of gross sluing motion followed by a fine adjustment (Chapanis and Kinkade, 1972). With soft controls, continuous variables may be adjusted in several ways. The following are three common means:

Incremental input devices that require motion – These are continuous-adjustment interfaces in which the position of the device corresponds to the magnitude of the input value. They are similar to continuous-adjustment control devices, such as dials, levers, and sliders. The magnitude of input provided by a dial corresponds to the degree to which it is rotated. A large change in a value requires a large rotation of the dial from its current position. An example of such a soft control is the soft slider (shown in a computer-based display) that resembles a barchart with a pointer directed toward the current value. It is used when the range of possible values and the ratio of a value to that range need to be displayed (NASA, 1992). Input is provided by sliding the pointer via a mouse or touch screen interface along the barchart scale to the desired value.

Incremental input devices with discrete inputs – These are input devices in which each actuation of the input device changes the variable by a specific amount. One example is a set of arrow buttons, a pair of buttons pointing in opposite directions either on a display screen or keyboard, used to increase and decrease a value sequentially (Apple Computer, Inc., 1996). Soft buttons typically are shown on the display screen and operated via a pointing interface device, such as a mouse or touch screen. Hard buttons may be physical keys mounted on a keyboard or panel and used in conjunction with a display screen. With each press of the increase button, the variable increases by a specific amount. If the button is held down, the variable will increase in proportion to the length of time that the button is depressed.

> A common design practice in computer-based HSIs is to have the input value change by the smallest unit of precision presented by the soft control device for each press of the arrow button. For example, if the soft control shows a variable to one decimal place, then pressing the button once changes the value by one tenth

(e.g., from 10.1 to 10.2). If the soft control presents a variable in integer values, then one press will change the current integer to the next (e.g., from 11 to 12). If a variable has a wide range, many presses may be needed to execute a large change or the button may have to be held down for a long time. Some soft controls feature a second set of arrow buttons that can change the input value by a larger amount for each press. One vendor of computer-based HSI systems for process plants has single arrow buttons [>] for small changes and double arrow [>>] buttons for large ones. The control system engineer can configure the size of the increment provided by the double arrow buttons. The vendor's standard values are 2%, 3%, 5%, or 10% of the range of the instrument.

In other computer-based control systems the incremental size changes as a function of plant or system state (e.g., a single press produces a large change during plant startup, but a small one when the plant is in its normal operating range).

Keyboards and number pads – These are input devices that represent values in digital form by actuation of a set of keys. For example, the value "100.7" requires five key presses: four for the numerical digits, and one for the decimal point.

Some computer-based control systems used for process control combine all three formats in a single soft control. Many systems allow the operator to choose between entering values via arrow buttons or the keyboard.

Control Feedback

When the operator enters an input to a soft control, the interface should provide feedback so that the operator can verify that the proper input was made. For example, the soft control may display the command, discrete-variable input, or continuous-variable input entered by the operator. This feedback may have many forms; visual feedback may include text and graphic formats, and sound (e.g., voice or audible tones) may also be used.

For a discrete-variable input, such as when an operator requests that a breaker be closed, feedback can be shown as text. The interface may give a message acknowledging that a "Close Breaker 100" input had been received. Feedback may also be presented graphically through a mimic display in which breaker 100 is represented in the closed state.

For a continuous-variable input, such as when an operator requests that a control setpoint be increased from 95 to 100 units, the feedback may also be shown in text or in graphical formats. Thus, the new setpoint might be displayed digitally (i.e., as "100"), or graphically as a barchart, a format commonly used in process control. The bar is usually depicted against a reference scale with its length or height corresponding to the magnitude (e.g., 100 units) of the input value. The two may be combined, with the interface showing the input value in both digital and barchart formats.

4.4.3 Response of the System

The total response time of the system may be described as the time between the submission of an item to a computer and the return of the results. Four response-time factors that affect the operators' ability to control the plant via the HSI are described below.

Display Retrieval Time – This is the time required for the HSI to present a new display after the onset of a command. For soft controls, this includes the time required to retrieve (1) a selection display from which a component or variable is to be selected, and (2) the input field in which operators provide commands. A slow response time for retrieving displays can delay the operators' access to important information.

Display Update Time – This is the interval at which displayed plant variables are updated with new data. If the update rate is slow relative to the rate of change of process variables, the displays may not represent the current state. For example, if a plant variable is rapidly increasing but the display is only updated once per minute, then during the later portion of that interval, the value displayed will be lower than the actual one.

Sampling Rate and Interval for Inputs – Computer-based display systems typically scan the input fields for new input provided by the operator. The sampling rate is the number of scans of an input field during a unit of time, and the sampling interval is the amount of time between samples. If the sampling interval is large, then there could be a long delay between when an input is entered and when it is received by the control system. Thus, if the sampling interval for inputs is 500 milliseconds and the operator completes an input action at the end of this interval, then the control system will not begin to respond to the operator’s input until one-half second has passed.

Plant’s Response to Inputs – This is the interval between the time at which an input is received by the control system and the plant reaching the desired state. It may have two components: (1) the time required for equipment to respond (i.e., an electrical breaker to close), and (2) the time required for the process to respond (i.e., a target temperature is reached). If the response is slow then the operator may have difficulty determining whether an input value was too high or too low, and the process value may overshoot or undershoot the target value. If the response time is fast, the operator may lack sufficient time to recognize and respond to input errors.

These response times are combined with the operator’s response time to determine the overall response of the human-machine system. The total time required to access a particular display is the sum of the time required for the operator to select the display and the HSI to respond (display retrieval time). The total time required to achieve a change in plant state is equal to the sum of the time required for the operator to enter the input value, the HSI to sample it, the plant to respond to the input, and the HSI to represent the change in the display.

5 TECHNICAL BASIS DEVELOPMENT FOR SOFT CONTROLS

This section discusses the human performance effects related to using soft controls based on our review of research and industry experience. The purpose of this review is two-fold: (1) to establish a technical basis on which to develop review guidance for soft controls, and (2) to identify human factors considerations that may have significant safety consequences but for which there is insufficient research and industry experience to form a basis for developing review guidance.

Section 5.1 has a general discussion of human error as it relates to soft controls. Section 5.2 describes general design approaches for making soft controls more tolerant of operator errors, and Section 5.3 discusses human performance considerations for specific operator actions involving the use of soft controls.

5.1 Human Error and Use of Soft Controls

There are many theoretical analyses of human error, with varying classifications of the types of errors. One widely accepted scheme divides errors into two major categories: mistakes and slips (Lewis and Norman, 1986; Norman, 1983 and 1981; Reason, 1990). This distinction is based on consideration of intention, a high-level specification that starts a chain of information processing which normally results in the accomplishment of that intention (Norman, 1983). An error in forming an intention, such as one that is not appropriate, is called a mistake. Mistakes are related to incorrectly assessing the situation or inadequately planning a response. Displays and controls are the main sources of information operators use for situation assessment and response planning. An error in carrying out an intention is called a slip.[1] Our discussion of errors focuses on slips, rather than mistakes, because controls are a primary means of carrying out intentions in process control settings. These intentions may be related to primary tasks, such as providing control inputs to plant systems, or secondary tasks, such as manipulating the user interface to access information or controls. [O'Hara, Stubler, and Higgins (1996) give a more extensive discussion of primary and secondary tasks.] Interface management tasks are referred to as secondary tasks because they are concerned with controlling the HSI rather than the plant. Slips involving primary tasks may result in the execution of inappropriate control actions. Slips involving secondary tasks are likely to cause delays in accessing controls and displays, or to disorient the operator within the display system.

A schema is a sequence of linked behaviors that, through repeated performance or deliberate training, becomes automatic (i.e., can be performed without a great amount of focused attention). To produce an action, a schema must be first activated in memory and then triggered. A schema controls behavior whenever its activation value and the goodness of match of its trigger condition reach their threshold levels (Norman, 1983). Slips result from "automatic" human behavior, when schema (i.e., subconscious actions intended to accomplish the intention) get waylaid en route to execution. Thus, while one action is intended, another is undertaken. Many slips occur with skilled users rather than with beginners learning new activities. The highly practiced behavior of an expert generates a lack of focused attention increasing the likelihood of some forms of slips. Lewis and Norman (1986) state that, on the whole, people can consciously attend to only one primary thing at a time. They can do many things at once only if most of the actions are automatic (subconscious) with little or no need for conscious attention. Thus, conscious attention is often focused at high levels, while low-level physical movements are controlled subconsciously. This inattention may incorrectly activate and trigger schemas and, therefore, produce slips.

Nagel (1988) noted that slips associated with discrete actions, such as entering data with a keystroke, are assuming increasing importance in aircraft as computer-based user interfaces and higher levels of automation are introduced. Review of research and industry experience indicates that slips are a widespread problem.

[1] Some researchers (e.g., Reason, 1990) distinguish two types of errors of execution – slips and lapses. For simplicity, this report uses a more general definition of slip (i.e., Norman, 1983), which encompasses both of these concepts.

5 SOFT CONTROL TECHNICAL BASIS DEVELOPMENT

Soft controls have characteristics that increase the likelihood of slips, compared to conventional HSI technologies. They are especially prone to the following types: unintentional activation, description errors, mode errors, misordering of components of an action sequence, capture errors, and loss-of-activation errors. Sections 5.1.1 through 5.1.6 describe these errors and the HSI characteristics that can influence their occurrence. Section 5.1.7 discusses factors contributing to the difficulty of diagnosing the cause of a slip.

5.1.1 Unintentional Activation

This type of slip occurs when a schema is not part of a current action sequence but becomes activated and triggered for extraneous reasons (Norman, 1983; Lewis and Norman, 1986). In a soft control, this can lead to the unintended actuation of an input device. Section 5.1.3, General Design Approaches for Error Tolerance describes a range of HSI design strategies for preventing unintentional activation errors.

5.1.2 Description Errors

Slips occur when the information that an operator uses to activate a particular schema is either ambiguous or undetected. Such slips are called description errors, since they occur when the appropriate action required to carry out the intention is not “described” adequately by the HSI (Norman, 1983; Lewis and Norman, 1986). The resultant ambiguity leads to an erroneous act, often closely related to the desired act. Description errors can be expected whenever the user interface is designed so that at a quick glance the distinctions among objects are not apparent, as where there are similarities in surface characteristics (e.g., general appearance). Physical proximity can increase the likelihood of description errors. For example, CRs with traditional hardwired instrumentation often contain banks of identical display or control devices. This invites the type of description error in which the correct operation is performed on the wrong control. Accordingly, conventional NPP HSIs use dedicated spatial locations, demarcations, labeling, and, where possible, uneven spacing of control and display devices to support the operators’ discrimination and proper identification.

Soft controls may increase the similarity of objects and decrease the separation between them, especially those represented via graphical user interfaces. For example, a display may contain many objects represented by similar graphical symbols or icons. In addition, soft controls often lack the spatial, tactile, and kinesthetic feedback of traditional hardwired controls that support rapid, correct identification. For example, the shape, texture, movement, and resistance of physical control devices may help operators identify them correctly. Holding a control that does not feel familiar may alert an operator that it is the wrong one. Many soft controls lack these physical characteristics that provide tactile feedback, so introducing new opportunities for description errors.

Norman (1983) described three design strategies for preventing description errors with physical control devices: arranging instruments and controls in functional patterns, perhaps in the form of a flowchart of the system; using shape coding to make the controls and instruments look and feel different from one another; and making it difficult to perform actions that can have serious implications and that are not reversible. These strategies support correct identification of controls, and increase the conscious effort needed to take control actions that have potentially serious consequences. Norman (1983) adapted these strategies for computer interfaces. We presented them below in slightly modified form more applicable to soft controls in process control and safety applications:

Organize options to provide context – The arrangement of control options should provide context to support identification of the correct device. One method is by having a functional organization scheme that associates the purpose of an option with the representation of the option. For example, the position of a control within a mimic or flowchart display shows its relationship to other components in the system, provides information about its function, and may help operators discriminate that control from similar looking ones. Norman also recommends functional organization for menus of controls.

Make options visually distinct – Design the command language or menu options distinctly, so they are not easily confused, either in perception or in the action required. This strategy may be extended to other items in display systems, such as icons used for selecting controls and symbols for control actions (e.g., open, close, increase, and decrease).

Require mental or physical effort to execute options that can have serious consequences – This strategy is used for actions that have important implications for operator performance or plant safety. For NPP applications, this strategy need not be restricted to irreversible actions, as suggested by Norman (1983). Its goal is to require greater conscious effort to carry out the control action. In focusing closely on the task, the user may recognize that the attempted action does not match the intention. A variety of approaches may be applied before, during, and after the action to make it difficult to complete (see Section 5.2, General Design Approaches for Error Tolerance).

Labeling and demarcation are other strategies used successfully in NPP CRs. Labeling can give unique identities to controls and displays; along with demarcation, it can make functional grouping of controls and displays more apparent.

An additional preventative strategy is to increase the distance between options in either physical space or the virtual space of the display system. For example, if two different controls closely located in the display system might be confused, the likelihood of a description error may be reduced by having them on different display pages or in different display devices.

Implementing soft controls on different input devices also may be a way of providing tactile feedback. Tactile feedback in the form of size and shape coding has long been used to prevent description errors with physical control devices. Different types of physical control devices help operators discriminate between them both before and during their use. Thus, an operator may release a control that does not feel like the right one. Soft controls often lack much unique physical feedback because the same input devices are often used to access and operate multiple controls. For example, the same mouse and keyboard may be used for all of the soft control devices of a computer-based display system. Then, a degree of unique physical feedback may be introduced by accessing some soft controls from different input devices. For example, special controls may be implemented on dedicated devices with unique tactile features, such as a specially shaped mouse. In addition, the use of dynamic tactile feedback on input devices, such as the mouse, is being explored. For example, when the cursor is moved over certain options on the display, the mouse vibrates, a raised surface emerges from the top of a button, or rolling resistance is increased (Rizzo et al., 1995; Gobel et al., 1995; Akamatsu and MacKenzie, 1996).

5.1.3 Mode Errors

Mode errors are defined as performing the operation that is appropriate for one mode when the device is in another one (Norman, 1983; Lewis and Norman, 1986). They comprise a large class of errors covering many types of human-machine systems, including computer-based devices. Mode errors occur most frequently in systems and devices with inadequate feedback on their mode or the state of the system. Depending upon specific characteristics, the consequences of mode errors can range from having no effect to an extremely serious one.

Modes are created when a control or display device is used for more than one function as, for example, when a single operator’s workstation accesses more than one soft control. (Note that control modes should not be confused with plant operating modes, such as startup, standby, and shutdown.) Mode errors occur when there is inadequate awareness of the device’s current mode (i.e., the user believes the device is in one mode when it is in another) and, as a result, performs an inappropriate input action. Mode errors associated with computer-based control systems are receiving growing attention because (1) computer-based technologies are being used in more and more human-machine systems, (2) computer-based control and display devices may contain more modes than traditional analog instrumentation (i.e., a single device may give access to many displays and control interfaces), and (3) the digital systems using computer-based technologies often are more advanced than their analog counterparts. Digital control systems for new plants or upgrades of existing ones may have additional capabilities for controlling plant processes or for testing the logic of the digital control system. These additional capabilities may be accessed through additional modes. Thus, the increase in modes and capabilities accessed through those modes may increase both the likelihood and consequences of mode errors. The following are two examples of mode errors that occurred in computer-based HSIs: one from a NPP, and one from a commercial aircraft.

A microprocessor-based overhead annunciator system locked up at a NPP after a mode error and typing error (Galletti, 1996; NRC, 1993a; NRC, 1993b). An operator discovered the lockup when he received an alarm on an auxiliary alarm printer and noticed that the corresponding window of the overhead annunciator system did not alarm as expected, and that the clock on the overhead annunciator system CRT was not updating. It was subsequently revealed that the overhead annunciator system could be locked up if a specific typed input was performed from a configuration workstation outside the main CR while the system was connected to the wrong computer port via an incorrectly positioned switch on a system panel.

The inspection showed that an operator had used this workstation to obtain historical alarm data just before the event. Computer records indicate that a panel switch was in the wrong position, which placed the system’s event-sequence recorder in the data-transfer mode rather than the operating mode. (This constitutes a mode error.) In this mode, the system allowed the operator to access password-protected software without issuing a warning. Commands entered from the workstation were routed to a high-priority link on the sequence-event recorder. The normal procedure for obtaining printout of alarm information when in the operating mode was to press the ALT and L keys together (ALT L). However, the ALT L command can be easily confused with CTRL L command (i.e., a data input error). In the data-transfer mode, CTRL L is a valid command that requires additional data input. The sequence event recorder processed the command and suspended communications to other data links, including the overhead annunciators, while it waited for additional input over the high-priority link. During this period, the overhead annunciator system was effectively locked up (NRC, 1993a; NRC, 1993b). Thus, the annunciator system lockup was initiated by a combination of a mode error and an input error. The errors were not detected when they occurred because the workstation provided little or no feedback about the effect of the commands on the annunciator system. The main indications of the lockup, the failures of the clock to update and annunciator tiles to respond properly, were not salient and, thus, were not noticed immediately.

Mode errors also have occurred with computer-based flight control systems on commercial aircraft. In 1990, pilots of an Airbus A30 aircraft entered inputs for the *angle* of descent when the controller was in a mode that accepted *rate* of descent; they were not aware that the flight control system was in the wrong mode. The pilots entered “33” intending to bring down the aircraft at an angle of 3.3 degrees. Instead the input instructed the control system to allow the aircraft to descend at the very rapid rate of 3300 feet per minute. Both the angle and rate modes accepted two-digit entries as valid input. The error was caught, but the aircraft descended well below the minimum descent altitude (FAA, 1996; Kletz et al., 1995). The FAA (1996) stated that this same error occurred in another Airbus A30 in 1993, resulting in a fatal crash.

Mode errors in automated systems are an increasing phenomenon (Woods et al., 1994). In some systems, such as flight control systems and medical devices, the number and variety of modes is rising. A single mode may be accessed through numerous paths. In addition, some automated systems change modes both by operator action and automatically when a preprogrammed target is reached. In NPPs, control modes may change automatically in response to changes in the plant or in the control system, such as an interruption of electrical power. For example, in one BWR, there was a momentary loss of a 24-volt AC power supply to the control logic of the feedwater pump speed control that caused the reactor feed pump controls to automatically switch from the automatic to the manual mode. As a result, the reactor water level increased, resulting in an alarm for high level (NRC, 1995). Mode transitions may not be obvious to the operator when they occur and their effects may not appear until much later. The increased capabilities and the increased autonomy of these new systems pose new demands on operators to be aware of modes (Sarter and Woods, 1995; Aviation Week, 1995a, b). Four design strategies for preventing mode errors are described next: eliminating modes, making modes distinct, providing different inputs for different modes, and coordinating inputs across modes.

Eliminating Modes – Norman's (1983) rationale for this strategy is simple: mode errors cannot occur if there is only one mode. However, multiple modes are normally eliminated by having additional dedicated control and display devices. This is not always possible in CRs where there may be insufficient space. Also, adding more devices to the HSI may increase the likelihood of description errors (e.g., choosing the wrong control). Thus, there is a tradeoff between mode errors and description errors (Norman, 1983). Therefore, this strategy should only be used when the HSI can accommodate additional control and display devices, and there are means to prevent description errors.

Making Modes Distinct – The goal of the second strategy is to ensure that the user is aware of the currently active mode by providing distinct, salient indications of mode state. This was apparently a major shortcoming in the user interface designs of both the annunciator workstation and the flight control systems described above. In the former case, the mode was indicated by a switch position which was either unknown to the operator, or not noticed.

Sellen, Kurtenbach, and Buxton (1990) studied the use of feedback for preventing mode errors. They describe four dimensions for characterizing the delivery of feedback on mode status:

- Modality of delivery (e.g., the sensory modality, such as visual, auditory, and kinesthetic, through which the information is received)

- Action-contingent versus action-independent delivery (e.g., whether the feedback depends upon an action being executed)

- Transient versus sustained delivery (e.g., the length of time over which the feedback is presented)

- Demanding versus avoidable feedback (e.g., whether the user can choose not to monitor the feedback)

Sellen et al. (1990) acknowledge that these characteristics may not be completely independent. For example, visual feedback is often avoidable; one may look away from it. In their study, Sellen et al. examined mode errors that occurred when users switched between the navigate and text-insert modes of a word-processing system. Two types of mode errors were studied: trying to enter text when in the navigate mode, and trying to navigate when in the text-insert mode. In one condition, feedback on mode status was visual – the screen switched to a pink background when in the insert mode. In another condition, feedback was kinesthetic – the operator had to hold down a foot pedal to access and stay in the insert mode. Experts made mode errors more frequently than novices, a finding consistent with Norman's (1983) description of slips as being the result of the subconscious execution of highly practiced actions. Both visual and kinesthetic feedback were effective in reducing mode errors. However, for expert users, the visual feedback was redundant with the kinesthetic feedback; the pink background did not improve performance further when the foot pedal was used.

From subsequent analyses, Sellen et al. (1990) concluded that determining the mode status was less demanding with kinesthetic feedback than visual feedback. Three possible explanations were given: (1) visual feedback was avoidable while kinesthetic feedback was not, (2) the visual feedback may have been less salient than the kinesthetic feedback, and (3) visual feedback competes with the visual editing task while the kinesthetic feedback does not. They concluded that the choice of the sensory feedback channel can be an important design consideration for reducing mode errors. This point also is made by others. For example, Rizzo et al. (1995) recommend using multi-sensory feedback (e.g., proprioceptive, tactile, acoustic) to reduce mode errors. One potential solution is a mouse that provides tactile feedback based on mode or position of the cursor on the display (Rizzo et al., 1995; Gobel et al., 1995; Akamatsu and MacKenzie, 1996).

Sellen et al. (1990) drew other conclusions: (1) mode errors can be reduced for both novice and expert users by appropriately designing the user interface, and (2) designing to reduce errors can also engender other improvements in the system's usability, including faster performance times and lower cognitive load on the user.

Coordinating Inputs Across Modes – The consequences of mode errors can be reduced by insuring that a command does not have very different meanings in different modes.

One approach is to require completely different commands for each mode based on the rationale that a command entered while the soft control is in the wrong mode will not be accepted by the system (Norman, 1983). There are three potential limitations in applying this approach to complex systems, such as the HSI of a NPP. The first is illustrated by the incident of the NPP annunciator system lockup. The commands *were* different for the two modes of this system. However, the command that caused the lockup in the data transfer mode (CTRL L) was very similar to a valid command for the operating mode (ALT L). It was entered by a typing error or the operators' failure to remember the intended command. Thus, the strategy proposed by Norman should be extended. A command that may have serious consequences in one mode should not be assigned a name similar to or easily confused with one that produces a benign action in that mode.

A second limitation resulting from having completely different commands for each mode is that it does not *prevent* the user from providing an inappropriate input. Instead, it increases the likelihood that inappropriate input will not be accepted by the system. This strategy should be coupled with error detection features to help the user recognize that the input has not been accepted and understand why the device rejects it (see Section 5.2.2, Error Detection).

A third limitation is that by requiring operators to use completely different commands for each mode it increases the complexity of the user interface. It may increase their mental workload because they cannot apply their understanding of how the soft control works in one mode to the other modes. Norman (1983) states that when people lack knowledge about the proper operation of some aspect of a device, they are likely to derive an understanding through analogy with similar aspects of the device. This may occur unconsciously and may influence behavior without the user realizing it. If the methods of interaction are not consistent across the device, then the user may "derive" incorrect input actions.

Operator performance can be enhanced by using the same commands in different modes – if they are used to perform similar functions (i.e., they are mapped to similar but related outcomes). Consistency in assigning commands to functions reduces the operators' mental workload because they can apply their understanding of how the command works in one mode to other modes. An example of a more compatible mapping of commands between modes is to have the function key "F2" list a directory when in mode 1 and list a file when in mode 2. A less compatible mapping is to have the function key "F2" list a directory in mode 1 and change the windows in mode 2. A key consideration is that commands that cause a benign action in one mode should not produce a different action with serious negative consequences in another mode. Commands that are destructive (e.g., delete file) or have serious safety consequences should have names that are unique or reserved for that special purpose and used consistently across the HSI. For example, the function key "F2" should not cause a benign action, such as listing a directory, in one mode and a destructive action, such as deleting a file, in another mode.

Some input actions involve entering data such as a control setpoint. In this case, the control action is defined by the control variable selected (e.g., angle of descent and rate of descent in the flight control system example). Mode errors associated with data entry can be more easily detected by requiring different input formats for different modes, such as a different number of digits, or a different position for the decimal point. If the operator is in the wrong mode, the system may reject an input whose format is not compatible with the current mode. However, the effectiveness of using different input formats to prevent mode errors may be limited by input errors. For example, if one mode requires a three-digit input and the operator fails to type one digit, the input may be accepted by a mode that requires a two-digit input. In the example of the flight control system, both modes accepted two-digit numerical inputs. The input entered for the angle of descent was an extreme, but valid, value for rate of descent. Accordingly, it was accepted by the flight control system and the pilot did not immediately detect the incorrect mode setting. Subsequently, the manufacturer corrected this aspect of the flight control system, now requiring a four-digit input for the descent rate mode (Kletz et al., 1995).

A special mode error consideration relates to systems that change modes automatically. Automated systems should be designed to inform the operator of their current operating mode, mode transition points, limits on operator actions, and circumstances under which the operators need to assume control. In addition, the operator must be aware of indications from the automated system or other means, of how to assume control without "fighting" the system or causing unnecessary transients. Accidents involving aircraft with automated management systems have been traced to a lack of the pilot's awareness of the operating mode of the automated system, and incorrect mental models of how control actions by the pilot or automated system would affect the aircraft. Inadequate feedback from the automated system was an important contributor to these errors (National Academy of Sciences, 1995; Sarter and Woods, 1995, 1992).

5.1.4 Misordering the Components of an Action Sequence

These slips include skipped, reversed, and repeated steps (Norman, 1981, 1983). Soft controls may be more prone to this type of slip than conventional controls because they introduce additional operations for accessing controls and displays, and providing inputs. These operations often must be executed sequentially.

Often, operators must perform a set of control operations in a specific order. For example, when configuring a fluid system, it may be necessary to establish the flow path, control mode, and setpoint of a flow controller in a specific sequence (e.g., A, B, C, D, and E). Misordered action sequences occur when operators skip operations, repeat them, or perform them out of sequence. One form of this error occurs when an operator skips a step thinking that it was completed. For example, an operator may carry out operations A, B, and C, and after some delay, may perform operation E thinking that D was completed. A factor in this type of error is the repetition of the task. If an operator has undertaken a set of operations repeatedly on several identical controllers, the memory of carrying out a particular action on the other units may increase the likelihood of the operator incorrectly concluding that it was completed for the present unit (Shaw, 1993). Thus, the sequential characteristics of soft controls can interact with repetitive, sequential tasks to increase the likelihood of errors involving misordering the components of the action sequence.

Operators may be less likely to make this type of error in a conventional CR with spatially dedicated controls because they can see the controller and associated instrumentation without navigating any displays; hence, it may be easier to determine which steps were completed. Also, the location of the controller in the control panel may provide useful cues. For example, the operator may recall that the last time a particular operation was performed, it was from a different location in the CR, and so conclude that the current controller had not been operated. Computer-based control systems tend to lack spatial dedication, particularly when multiple controllers are manipulated from the same display device. Displays should be designed to support operators in identifying tasks that are in progress. Ideally, operators should be able to check at a glance the status of related operations (e.g., A, B, C, D, and E) on a single display.

5.1.5 Capture Errors

A capture error may occur when an *infrequent* action requires a sequence of operations that overlaps with the sequence required for a *frequently* performed action. In attempting the infrequent action, the frequent one is performed instead. For example, an operator intends to perform task 1, composed of operations A, B, C, and D, but instead executes the more frequently performed task 2, consisting of operations A, B, C, and E. If the more frequent action was carried out recently, a capture error is even more likely (Norman, 1981). Capture errors may occur with soft controls that have similar interaction steps (i.e., where similar navigation actions and interaction dialogues may be required for different operations). For example, many soft controls have very similar interfaces for entering control inputs and similar paths for accessing them.

Lewis and Norman (1986) suggest two strategies for addressing capture errors: the first strategy is to minimize the overlapping sequences; the second is to improve the detection of errors. Capture errors occur at the point of divergence of the frequently and infrequently performed sequences. The HSI design may be directed at bringing the operator's attention to that critical point. For example, if the control system knows the intention of the operator (e.g., by requiring the operator to indicate the overall intention), it could highlight the proper path at the choice point, or initiate a warning if the wrong one is taken. Other approaches that do not require the control system to understand the operator's overall intention include designing the HSI such that important choice points are salient and the operational significance of the alternative paths are apparent. After passing the critical point, such features may support the operator's understanding of which path was taken.

5.1.6 Loss-of-Activation Errors

Loss-of-activation errors are one of the more common types of errors and are referred to as forgetting to do something. Loss-of-activation means that schemas that have been activated become deactivated due to decay and interference processes that often occur in human memory.[2] Memory can fail when events intercede between the preparation and execution of an intention, so that the intention may partially or completely decay. When this occurs, the action (schema) is no longer available for execution. A special case of loss-of-activation error involves

[2] Some researchers define a loss-of-activation error as a lapse, i.e., a failure to perform an intended action due to a memory storage failure.

forgetting part of an intended act while remembering the rest (e.g., retrieving a display while being unable to remember why it is needed).

One cause of loss-of-activation errors is the keyhole effect in computer-based HSIs (see description of parallel versus serial access in Section 3.1). Displays serve as reminders of tasks that must be completed, but the HSI may only have space for a few displays at one time. When operators respond to interruptions, they may suspend an ongoing task, and that display may be removed. If the operator then retrieves another display when responding to an unrelated inquiry, a needed reminder of the suspended task may be lost, and the suspended task may be forgotten. Interviews with operators in computer-based CRs, conducted as part of this project, confirmed that this was a problem.

Another cause of loss-of-activation is a long elapsed time between the operator starting an action and the plant system responding. For example, an operator may enter a control setpoint and a corresponding automatic ramp rate that is very slow. Several hours may pass until the plant variable reaches setpoint value. Meanwhile, the operator may forget that this change is underway because attention is focused elsewhere. In other cases, the plant system may respond quickly to an operator's input but the plant process reacts slowly. For example, the path to a heat sink may be established but the temperature may not show a discernable decrease for a long time. Thus, operators may forget that they were supposed to take action when the final condition was reached.

Memory aids are essential to preventing loss-of-actuation. Norman (1983) states:

> In many ways the old saying, "Out of sight, out of mind" is apt; if a set of operations is interrupted with other activities so that no reminder of them remains visible, the action sequence is apt to be forgotten. A good system design will not let this happen, but will redisplay uncompleted sequences (or unanswered questions) whenever there is a chance that they are no longer visible to the user. (p. 257)

The design of the HSI should support the operator in resuming interrupted or suspended tasks. The Electric Power Research Institute (EPRI) states that when an operator has accessed a control for an action sequence, the operator should be able to suspend it temporarily and then return to continue it (EPRI, 1993a). EPRI recommends that the HSI provide a reminder of the suspension, possibly as an on-screen message. Another approach is to have more display screens or implement a window-based display system (Norman, 1983). This allows tasks that are in progress to remain visible, as they are in spatially dedicated conventional CRs. In the case of a window, the entire display may not be not visible while other displays are being accessed, but the window serves as a reminder of something that the operator should open and look at.

Another approach is to automatically remind the operator that a suspended task exists and retrieve the display containing the task. One way to accomplish this is via intelligent agents, computer programs that perform information processing tasks somewhat autonomously. This approach is being developed for chemical plants to augment the operators' monitoring capabilities. The goal of the user-initiated notification is to allow operators to assign information processing tasks to agents according to task requirements or their personal preferences (Guerlain and Bullemer, 1996). One application is task scheduling, in which the HSI reminds operators of pending tasks, such as equipment tests, that have not been completed (Soken et al., 1993). Such agents could track suspended tasks and retrieve displays for the operator. However, as operators become more dependent upon these agents additional burdens may be associated with supervising them. Additional research is needed to weigh the potential benefits of intelligent agents for preventing loss-of-activation errors against the demands of supervising them, and any problems that may result from applying them inappropriately.

5.1.7 Diagnosing Slips: The Problem of Level

Slips can be identified by discrepancies between goals and the outcomes. However, even when an error is detected, its cause may not be clear. The recognition that *something* is wrong may not lead to the immediate discovery of *what* is wrong. The proper form of feedback is essential for proper diagnosis. Lewis and Norman (1986) state: “A major problem in the discovery of slips is the problem of levels: the level at which actions take place in the world differ from the level at which the intention is formed” (p. 419).

Actions can be specified at many different levels. For example, the action of driving one’s car to the bank may be described at the following levels, all of which are accurate at the same time (Norman, 1988):

- Driving to the bank
- Turning into the parking lot
- Making a right turn
- Rotating the wheel clockwise
- Moving my left hand upward and to the right
- Increasing the tension on the sternocostal portion of the pectoralis major muscle (p.111)

In a PWR NPP, the task of switching from the auxiliary feedwater system to the main feedwater system during a plant startup may be described as:

- Transfer from auxiliary/startup feedwater to main feedwater system
- Close the main feedwater bypass valve
- Operate controls for closing the valve
- Enter a valve position setting via keypad
- Type the number “9”
- Press with the index finger

Norman (1988) calls the most global description, at the top of the list, the high-level specification and the more detailed descriptions, in the bottom portion of the list, the low-level specifications. Errors may occur at any level.

In many situations, it is possible to detect that an action did not produce the intended results, but not know the level of specification at which the error occurred. Difficulties in determining the appropriate level of specification can thwart the correction of an error; often after detecting a problem, people attempt to correct it at the wrong level. Norman (1988) provides the following example of a car key that does not work:

> The first response is to try again, perhaps holding the key more level or straight. Then the key is reversed, tried upside down. When that fails, the key is examined and perhaps another tried in its stead. Then the door is wiggled, shaken, hit. Finally, the person decides that the lock has broken, and walks around the car to try the other door, at which point it is suddenly clear that this is the wrong car. (p. 112)

Norman states that this example is typical of the slips he has observed. The error-correction mechanism in humans seems to start at the lowest possible level of specification and slowly proceed to the higher levels. This tendency has implications for designing features for preventing slips and detecting and correcting them when they occur.

5.2 General Design Approaches for Error Tolerance

The HSI should be designed to protect against input errors. Design strategies for error tolerance may be considered in three categories. The first, error prevention, includes techniques intended to support the selection of appropriate responses and reduce the likelihood of choosing incorrect ones. The second category, error detection, includes techniques that increase the likelihood that the computer system will detect an incorrect input and bring it to the operator's attention or attempt to correct it, rather than proceed with a control action based on the erroneous input. The third category is error mitigation, which includes a set of techniques for reducing the consequences of an incorrect input once it has been entered by the operator and accepted by the system. Table 5.1 lists the specific design approaches for these three categories. Each approach is discussed next.

Table 5.1 General Design Approaches for Error Tolerance

Error Prevention • Elimination of undesirable actions • Distinct options • Arrangements that provide context • Feedback on input value (incremental input, magnitude representation, and significance representation) • Effort for input actions (activation logic, verification and confirmation steps, and activation force) • Inspection and transfer steps
Error Detection • Gag (lockouts, interlocks, and lockins) • Warn • Do Nothing • Self Correct • Let's Talk About It • Teach Me

Error Mitigation

- Undo command
- Recovery time (deferred system response and reduced system response rate)
- Automatic intervention

5.2.1 Error Prevention

The following describes user interface design techniques for reducing the likelihood that operators will enter improper inputs into computer-based systems; they are based on Norman's (1983) recommendations.

Elimination of Undesirable Actions – Actions that may produce negative consequences are not given as choices. This is one of the most straightforward strategies and has a simple rationale: if the slip-producing action cannot be performed, then the slip cannot not occur. For example, if a system will not function properly for input values greater than 10, then the range of options it provides should be values of 10 or less. As another example, if a control option will be destructive at a particular stage in a process, then the operator should not be able to choose it during that stage.

Distinct Options – Options are designed to be distinct (i.e., easy to identify and discriminate). Description errors can occur when the user interface does not have sufficient clear cues to allow the operator to select the correct response. For example, a display may contain many options represented by similar graphical icons so that an operator wishing to select option A may choose option B instead. Options may be made visually distinct by coding (e.g., size, shape, and color), labeling to provide a unique identity, and spacing. The distance between options may be increased in either physical space or the virtual space of the display system. Thus, if two different controls located near each other may be confused, they may be moved farther apart on the display page, put on different display pages, or on different display devices.

Arrangements that Provide Context – Controls and displays are arranged to provide a context for correct identification, and options within a display are arranged to support correct identification. One way to show context is by functional grouping (i.e., clustering controls or options that have related functions). Such function-based organization supports identification by helping the operator associate the location of the control or option with its function. For example, controls may be arranged in a flow pattern representing the stages of the process which they affect, and options on a menu can be arranged according to the functions they perform (e.g., all editing options are located in one menu). Other arrangements include grouping by sequence of use and by importance.

Feedback on Input Value – Slips involving numerical data can result in the wrong value being entered and may be prevented by providing salient cues when the value is entered. Rather than allow the operator to select and enter values unaided, the HSI can be designed to indicate the significance of particular values to plant operation and safety. These design features can support the operator in understanding the magnitude of the value and determining whether it is correct. Specific approaches include using incremental input interfaces, magnitude representation, and significance representation. Each is described below.

- *Incremental Input* – This approach entails replacing the keypad with an input device that requires values to be entered as a series of steps or increments, each of which provides feedback on the size of the change in the input value. For computer-based user interfaces, these include sliders and arrow buttons. With computer-based sliders, the magnitude of the change made to the variable is proportional to the distance that the slider is moved. With arrow buttons, the magnitude of the change is proportional to the number of times an increase or decrease arrow button is pressed or to the amount of time the button is held down.

- *Magnitude Representation* – This approach provides feedback on the size of the input value. In addition to displaying the entered digits numerically, the magnitude may be represented in other ways, commonly as a barchart. As the magnitude increases, the bar grows longer, and as it decreases, the bar shortens. Errors involving entering incorrect setpoint values may be prevented by user interfaces that draw attention to the magnitude of the value that is about to be entered. For example, if the actual value is shown on one barchart and the setpoint value on another, then the differences in the lengths of the bars will indicate the magnitude of the change. Such interfaces can enable the operator to detect large changes when small ones are intended.

- *Significance Representation* – The two approaches described above may be augmented with design features that convey information about the meaning of the input values. Various coding techniques can give qualitative information about the entered value, such as whether it is high, low, or within the normal operating range. For example, a barchart may be labeled with a scale indicating important values and the limits of ranges. The latter may be further enhanced with color codes or other schemes.

Section 5.3.3.3 discusses further the application of these design approaches to soft controls used for adjusting control setpoints.

Effort for Input Actions – If actions that may produce negative consequences cannot be eliminated, then they may be set up in ways that reduce the likelihood of unintentional actuation. Norman (1983) states, "Actions that can lead to difficulty should be difficult to do" (p. 257). Thus, their execution may require additional effort, such as in a series of deliberate steps. Such actions may be less likely to occur as the result of an operator's random movement and also may require greater attention. By having to focus on the action, the operator is less likely to revert to the type of "automatic" activity that could produce a slip. The following are strategies for preventing slips by designing the input action to require greater physical or mental effort.

- *Activation Logic for Interfaces* – Activation logic refers to the way the user interface responds to the operator's inputs. EPRI (1993a) states that the HSI should give the operator feedback showing the action that was selected before executing it. The goal of this recommendation is to avoid inadvertent actuation of plant equipment or inadvertent changes of displays. This type of feedback is important because a broad range of control actions may be accessed through the limited display area of a given soft control device, including manipulation of various plant components and of the user interface. The close proximity and similarity of input options within the display area may cause operators to select the wrong option. If an operator determines that executing an action would be undesirable, it should be possible to cancel or change it before the system runs it.

  A special case is the activation logic for pointing interfaces, such as those used with touch screens and "point-and-click" cursor interfaces that usually have one of the following types of activation logic: activation upon first touch or upon lift-off (Sears et al., 1992). With first-touch logic, a target is selected as soon as the cursor or finger makes contact. To select a target with the lift-off touch logic, the cursor or finger must enter the target and then be removed without touching the area surrounding it. Laboratory studies and operating experience (Degani et al., 1992; Hoecker and Roth, 1996; Sears et al., 1992) have shown that the first-touch logic is prone to accidental activation. Because targets are immediately activated, the operator does not have time to make corrections when the wrong target is contacted. The activation upon lift-off logic was found to be more forgiving of input errors. For example, if contact is made with the wrong target, this logic allows the operators to avoid actuation by moving out of the target area before release. It also allows the target to include additional forms of feedback. For example, when the target is contacted, it can change color or emit a sound to notify the operator that a particular target area has been entered. Therefore, the activation upon lift-off logic should be used rather than first-touch logic.

- *Verification and Confirmation Steps* – Verification steps are usually steps that are added to the input action. For example, the user selects an option and then presses the Enter key to verify the selection. Confirmation steps require the user to respond to a warning or advisory message; the user may respond to the question, “Are you sure you want to do this?” by pressing “Yes” or “No.” Both verification and confirmation steps attempt to reduce input errors by increasing the effort (i.e., the number of steps) and drawing a user’s attention to the input operation. However, at least two factors may limit their effectiveness.

  First, despite the designer’s intention of increasing the needed effort and attention, these steps often become automatic. For example, some verification steps are designed such that they can be easily “chunked” with the input action and become part of it. Then, the verification step loses its efficacy because it does not require thoughtful action. Kletz et al. (1995) provide the following example from a chemical plant:

  > To reduce the chance that operators will enter incorrect data or instructions, computers are sometimes programmed so that after someone has entered information and pressed the 'Enter’ button, the data or instructions are displayed for checking. Then 'Enter’ is pressed a second time. After a while, new operators are told [or learn] to enter information and then press 'Enter’ twice. (p. 28)

  Verification steps should be designed to require conscious effort on the part of the operator, thus breaking the train of automatic processing. Kletz recommends distinguishing the input step from the “Enter” step to prevent operators from combining them. For example, after entering a numerical value, the operator may be required to move the cursor before pressing the “Enter” button.

  The second factor relates to the level of specification. Confirmation steps should draw the operator’s attention to the goal of the action, not just to the action. Confirmation steps are usually ill-timed; they come just after the operator has initiated the action and is still fully content with the choice. Norman (1988) describes a typical interaction:

  > User: Remove file 'My-most-important-work.’
  > Computer: Are you certain you wish to remove the file 'My-most-important-work’?
  > User: Yes.
  > Computer: Are you certain?
  > User: Yes, of course.
  > Computer: The file 'My-most-important-work’ has been removed.
  > User: Oops, damn. (p. 113)

  If the user requests deletion of a wrong file, the computer’s request for confirmation is not likely to help the user detect the error because at this point, the user is apt to focus on confirming the *action* (e.g., deletion) rather than the *object* (e.g., the file 'My-most-important-work’). Because the effectiveness of verification steps may be limited for catching slips when the wrong object is specified, Norman (1988) recommends, instead, eliminating irreversible actions (See Undo Command and Deferring System Response in Section 5.2.3).

Some errors that occur in process control environments involve correctly specified objects and incorrectly specified actions. In these cases, making the action reversible may not be a desirable solution because it may further upset the plant. Thus, preventing incorrect entries is desirable. Confirmation messages may be useful for drawing attention to the incorrect action and, thereby, preventing the incorrect input from being transmitted to the system. In the following example, incorrectly entered input rapidly changed reactor power by 15 percent in 40 seconds (NRC, 1996).

A BWR plant was at 68 percent power when a newly installed digital adjustable-speed drive modification to the reactor's recirculation pumps was tested. Before the event, the licensee was preparing to increase the reactor's recirculation flow from 51 to 53 percent. It was intended to type all the instructions into the computer system, except the "Enter" command that would return the flow to 51 percent. In that way, if electrical harmonics were experienced in the adjustable-speed drive system, the "Enter" key could be pressed and the flow setpoint would be returned to 51 percent. Instead, an incorrect setpoint was typed (transposed digits) and then executed when the "Enter" key was accidentally pressed. These actions caused the reactor recirculation flow and the reactor power to drop. The error was immediately recognized and the correct value was input, thereby increasing reactor power.

In this incident, the object, the adjustable-speed drive system, was specified correctly, but the action was specified incorrectly. The intention was to enter, but not execute, a *small* (2 percentage point) decrease in the flow setpoint. Instead, a value that represented a *large* decrease was entered and immediately executed. Two slips were involved in this incident: a misordered action sequence (i.e., transposed digits) and the unintentional actuation of the "Enter" step. This incident might have been prevented if confirmation had been required. First, the confirmation step would have prevented the control system from immediately acting on the new setpoint after the "Enter" key was accidentally pressed. Second, the confirmation message could draw attention to the entered value which differed from the intended value. Third, the confirmation message could be designed to draw attention to the magnitude of the setpoint change, as we described in discussing feedback on the input value. For example, the message could state that the new value represents a large decrease from the current value and, consequently, may upset the plant system.

Thus, confirmation steps should be designed to address the operator's input at multiple levels, including the object that is to be acted upon, the action, the magnitude of the action (e.g., how many or how much), and its potential consequences. Also, the potential benefits of confirmation steps should be weighed by comparing their effects on operators' response time (e.g., potential delays) to the potential consequences of the errors that are being prevented.

- *Activation Force* – Many computer-based user interfaces use input devices that require very little force to activate. However, activation force may be used as a form of feedback for preventing errors when inputs are provided through special buttons, keys, and mice. For example, critical control actions may be designed to be undertaken using input devices that require higher activation forces than other controls; this sensory feedback may allow the operator to discriminate these controls from others. Also, the additional actuation force may reduce the likelihood of an accidental actuation. For example, a button requiring a higher force or a longer travel distance when depressing it may be used for critical input.

- *Inspection and Transfer Steps* – Inspection and transfer steps should be inserted into the transaction if the execution of particular actions may have serious consequences. An inspection step holds the user's inputs and allows the user (or someone else) to review them before their execution by the system. Special formats may be used in these steps to help users detect errors. A transfer step is a separate operation that the operator must take to move the input to a location where it can be executed by the system (e.g., data are moved from one file to another); transfer steps require more action than a mere verification.

  The following example of inspection and transfer steps is from a process control plant; however, similar approaches may be used in any system in which error prevention is important. Every month, operators enter updated information about the condition of the plant into a safety protection system. Because the system uses this information to determine the specific conditions under which it should automatically shut down the plant, it is critical that the values are entered correctly. In one of the steps, the operator enters the new values in a column of a table and then inspects them for accuracy. Adjacent columns show reference values for each variable, including the minimum- and maximum-allowable values and the value currently being used by the safety system. This format allows the operator to compare each new value to these reference values. Any large deviation from the value currently used by the system may be suspected of being an erroneous input. Additional error-prevention features are used, such as verification and confirmation steps.

  After all values have been entered and inspected for accuracy, the operator accesses a separate display page and executes a series of steps to transfer the data to the safety protection system. Each step requires a deliberate action and provides an opportunity for the operator to halt or cancel the transfer of the new data. These steps reduce the likelihood that inappropriate data will be transferred to the protection system as the result of a slip.

5.2.2 Error Detection

Input errors may also be avoided by having the computer system detect incorrect inputs before they are executed. Lewis and Norman (1986) describe six ways that systems can respond to incorrect errors: Gag, Warn, "Do Nothing," Self-Correct, "Let's Talk About It," and "Teach Me." Each is discussed below using examples taken from computer applications that are applicable to a broad range of human-machine systems, including process control and safety systems.

Gag – A "gag" is described as a function that prevents users from expressing unrealizable intentions. The constraints it places on users' actions aid in the discovery of erroneous behavior. As an example, a tutorial language system may process commands one character at a time as they are typed. If a user types a character that does not have a continuation into a legal command, it is not accepted. As a result, the user simply cannot enter anything but a legal command. Norman (1988) describes three types of gags: lockouts, interlocks, and lockins. The common usage of these terms in various industries may differ. For example, in some process industries interlock is used as a general term to encompass the full range of characteristics described below.

- *Lockouts* – Lockouts are devices that prevent the operator from entering a dangerous condition or an undesirable event from occurring. A simple example is a control that only accepts inputs within a specified range. With some conventional physical control devices, input values are constrained (locked out) by its design (e.g., a dial that cannot be rotated beyond the normal range of settings). Some computer-based user interfaces are programmed to only accept input values that are within a specified range. Kletz et al. (1995) call this type of lockout *input validation*. They describe two types: statically defined ranges and context-sensitive validation. A statically defined range does not change; values inside the range are accepted, and those outside are rejected. An example of context-sensitive validation is an interface that restricts input values based on the last value entered (e.g., large deviations from the previous value are not permitted). Other context-sensitive validation methods may restrict input values based on such factors as the condition of the system or the system's understanding of the operator's intentions.

- *Interlocks* – Interlocks require actions to take place in a proper sequence. For example, action C will be blocked unless conditions A and B have been satisfied. Operators' inputs that violate these constraints are blocked.

- *Lockins* – Lockins keep an ongoing operation active by preventing human action from prematurely terminating it. For example, if interrupting an operation may damage equipment, a lockin may be implemented so that operator inputs are ignored until the operation is completed. In NPPs, the safety injection system may have a lockin feature to prevent premature termination. Lockins can also occur by default, due to the failure to provide override capabilities, such as cancel and abort features. In such cases, the lockin may give the operator no option other than to wait until the operation has been completed.

A problem associated with all three types of gags is they do not make allowances for situations in which the gag is overly restrictive, including cases in which it may be detrimental. Sometimes a normally undesirable tactic may be the only thing an operator can do to solve a problem. Senders and Moray (1991) provide the following example from aviation:

> If a pilot accidentally lowered the undercarriage of an aircraft at high speed, the landing gear could be torn off. It is easy to imagine the passage of safety legislation requiring a mechanical interlock that would prevent the landing gear being lowered except during the landing sequence... Consider, however, the following case: a few years ago a 727 hit clear air turbulence and dropped thousands of feet. The pilot, operating contrary to standard procedure, lowered the landing gear. In this situation, the undercarriage acted as an airbrake and allowed him to regain control. (p. 115)

One of the guidelines for human-centered control automation (Billings, 1991) states, "Do not foreclose pilot authority to override normal aircraft operating limits when required for safe mission completion, without truly compelling reasons for doing so" (p. 86). This guideline states that if limitations are placed on pilot authority, they may be unable to fulfill their responsibilities for safety in abnormal conditions. The term "hard limits" refers to limits on control variables established by designers to restrict the operation of systems to specific, predefined ranges. In contrast, "soft limits" allow pilots to exceed the normal operating limits with, in some cases, some degradation in system performance. The guideline suggests that soft limits be used as a way to avoid limiting pilot authority while enhancing flight safety. This guideline is already being applied in aviation: "The softer approach has been taken by the MD-11 [aircraft], which permits pilots to override automatic protection mechanisms by application of additional control forces" (Billings, 1991, p. 29).

A second problem with gags is the lack of visibility. That is, if the user interface is not properly designed, operators may not understand what the gag is doing and why. In some cases, they may fail to recognize that an action has been blocked. In other cases, the operator may misinterpret the gag as an equipment malfunction and try to circumvent it. EPRI (1993a) states that when an operator action is blocked, the HSI should inform the operator unambiguously, and that in the future, software-based systems may provide specific information about the blocked action and why it was prevented. EPRI recommends automatically recording the blocked action.

Thus, a gag should be designed so the operator can understand which action(s) is being blocked and what conditions activated the gag. For a lockout, operators should be able to determine why an input has been blocked and what inputs are acceptable, especially for context-sensitive validation, which may use complicated rules for determining the acceptability of inputs. An interlock should inform the operator of the condition(s) that activated it and the conditions that must be satisfied to release it.

A third potential problem with gags is their automatic release. If operation B was blocked because condition A had not been satisfied, the gag should not automatically start operation B upon condition A being met. When a gag automatically releases, the operator may be surprised by the execution of a command that was entered long before and had since been forgotten. Such initiations may bring about unintended consequences. Accordingly, a separate operator action should be required before the blocked action can resume, such as acknowledging the change in conditions or re-entering the input that was previously blocked.

Warn – While gags block anything but appropriate responses, warnings merely inform operators of potentially undesirable situations. Operators then decide what action to take. A lockin that can be overridden by the operator is a type of warning, since it informs the operator that an action has potentially negative consequences, or an input value is not within the normally accepted range. However, it allows the operation to proceed if the operator desires.

Do Nothing – With this approach, the system simply fails to respond to an illegal input. No warning or messages are provided. The user is left to infer from the lack of response that the attempted action was not legal and does not work. The “Do Nothing” method has been effective in direct manipulation interfaces where actions and effects are readily apparent. Direct manipulation interfaces have the following properties that may be particularly well suited to the “Do Nothing” method (Shneiderman, 1982): “Continuous representation of the object of interest, physical actions or labeled button presses instead of complex syntax, and rapid incremental reversible operations whose impacts on the object of interest is immediately visible” (quoted by Hutchins et al., 1986, p. 91).

As an example, a software package for drawing may contain a brush icon that the user moves to paint objects. If there are objects that users are not allowed to color, a “Do Nothing” method would allow the user to go through all of the operations for applying color, but the color of the object would remain unchanged. The “Do Nothing” response differs from an error message in that an explicit message is not given to the user. Instead, the user infers the error from the system’s lack of response. This method relies on the visibility of the action and the ease with which a lack of response can be detected to convey the nature of the error.

When used properly, “Do Nothing” can be beneficial. First, it can focus the user’s attention on the task at hand rather than diverting attention to error messages. Second, this method avoids potentially complex warning messages, e.g., if a display contains much graphic detail or many potential targets for manipulation, the warning message may be more confusing than revealing. Because users learn by taking actions and observing their effects, this method is *not* appropriate for systems in which input actions and their effects are not readily observable, such as where inputs have complex interactions and responses are slow or not easily predicted. Lewis and Norman (1986) describe the remote copy command of a UNIX-based system. The command requires at least two logical arguments. If only one argument is given, the command fails but provides no explicit indication. If two arguments are given but neither refers to a valid file, the command appears to work but does not. Because the UNIX system does not provide adequate feedback, such as automatically displaying the state of its file system, the “Do Nothing” method is inappropriate.

The “Do Nothing” method should not be used for primary tasks because the plant’s response to control inputs can be complex. Operators should have clear feedback on whether they have provided the input properly, whether the system has received the input and is responding to it, and whether the system is progressing toward the desired state. However, the “Do Nothing” method may be appropriate for some interface management tasks that use direct manipulation, especially for those tasks in which the relationships between the input action and the response are obvious and additional feedback would be distracting. For example, an incorrect attempt to change the size of a display window could entail no response.

Self Correct – Once an error is detected, the system tries to guess the legal action that corresponds to the user’s current intentions. A spelling-correction feature is an example of this approach. “DWIM: Do What I Mean” (Teitelmann and Masinter, 1981) is a corrector used for writing computer programs on the Interlisp system. Teitelmann, the designer of DWIM, gave the following rationale: “If you have made an error, you are going to have to correct it anyway. So I might as well have DWIM try to correct it. In the best case, it gets it right. In the worse case, it gets it wrong and you have to undo it: but you would have had to make a correction anyway, so DWIM can’t make it worse” (quoted by Lewis and Norman, 1986, p. 423). While this approach may be appropriate for writing computer programs and some interface management or information-handling tasks, it may not be appropriate for operations that are directly related to controlling the plant because it could result in the wrong control action being taken.

Lewis and Norman state that automated self-correct features, such as DWIM, are only acceptable if they include good “Undo” facilities, so that inappropriate changes made by the system can be altered. Even with “Undo” facilities, automated self-correct systems can interfere with users’ activities if their facilities for detecting errors are overgeneralized. For example, a reported shortcoming of DWIM was that it sometimes corrected things that should not have been corrected. This places an additional mental burden on the user to learn, remember, and anticipate the types of correct inputs that these systems interpret as errors.

Let’s Talk About It – This method responds to user errors by starting a dialog. Lewis and Norman (1986) state that many Lisp systems have this type of error-detection response. When these systems detect a problem, they send the user a message describing it as best as can be determined. Then, they switch to a Lisp debugger mode, which allows the user to interact directly with the system to locate the error; thus, the responsibility for exploring the problem and finding a solution is shared between the user and the system. Lewis and Norman cite the “Let’s Talk About It” method as a possible solution to the level of specification problem and the error-message problem. For example, a message could be initially presented at the highest level, indicating that there is a problem and its seriousness. The user then could employ “Let’s Talk About It” to explore the problem to whatever depth desired. The user could be allowed to “trace down the levels to see where the original mismatch occurred, how that level was reached, and the state of the system at each level” (Lewis and Norman, 1986, p. 428).

Teach Me – With this method, the system queries the user when it detects a phrase or command that it does not understand, in effect, asking the user to teach it. Lewis and Norman describe a natural language inquiry system, called Clout, which asks for the definition of words or phrases found in user inquiries that it does not understand. If the user’s response also contains unknown words or phrases, Clout asks the user to define these. This questioning continues until all inputs are understood. The new words or phrases are stored by the system and are accepted without question in future interactions.

5.2.3 Error Mitigation

This section describes design techniques for minimizing the consequences of operators’ incorrect inputs once they have been accepted by the computer system. Techniques for error mitigation should not be considered substitutes for error prevention and detection; in many cases, it is more desirable to prevent errors than to mitigate their consequences.

Undo Command – An “Undo” feature should be used when it is possible to reverse an operation and restore the system to its previous state. “Undo” commands are generally not appropriate for tasks involving control of the plant because a reversal may upset the plant. However, the “Undo” command may be appropriate for tasks associated with managing the interface management or manipulating stored information. For example, an operator may select the “Undo” command to restore a data file after manipulating its format or performing mathematical manipulations.

Error Recovery Time – Design features that increase the amount of available time for error recovery should be considered for situations in which operations cannot be easily reversed. Some systems respond so quickly that they can process an input before the operator realizes that an incorrect input was entered or has time to change it; this can be problematic in any domain that is sensitive to errors. In these cases, it may be desirable to allow the operator longer to respond to the error by either deferring the system’s response or reducing the speed at which the system responds to the incorrect input. These approaches are described below.

- *Reduced System Response Rate* – Digital control systems can respond much more quickly to operators’ inputs than traditional analog control systems, and this can be a problem if incorrect values are entered. The system may respond before the operator has time to recognize the incorrect input and change it. The following are three examples from the chemical industry of accidents that occurred because digital control systems responded so quickly to inputs from soft controls that the operators did not have time to take countermeasures:

  > “An operator was attempting an emergency shutdown, and he stopped the flow of recycled hydrocarbon by setting the flow controller to 0. The 12-inch valve slammed closed, and the resulting hydraulic hammer ripped the line off a distillation column.” (Lorenzo, 1990, p. 18)

  > “An operator wanted to reduce the temperature on a catalytic cracker from 982° F to 980° F. Unfortunately he pressed the keys in the wrong order (908) and immediately pressed the Enter key. The PES [programmable electronic system] responded with impressive speed, slamming slide valves shut and causing a flow reversal along the riser.” (Kletz, 1993, p. 260)

  > “An operator was changing the feed rate from 75 to 100 gpm [gallons per minute]. She inadvertently typed 1,000 gpm into the computer-based controller, which responded by fully opening the feed valve. The excessive feed caused a rapid pressure rise that was relieved to the flare.” (Lorenzo, 1990, p. 18)

  Lorenzo states that if incorrect inputs from operators can result in damage to plant equipment then the system should be designed to limit the rate at which it responds to them. He suggests programmed limits in the control software, such as maximum ramp rates, and physical limits in the plant’s equipment, such as orifices and dampers, to reduce the rate at which processes can change. By limiting the system’s response rates, operators have more opportunity to detect and correct the erroneous input.

- *Deferred System Response* – This strategy causes the system to accept the operator’s input but not act upon it until later. One way of making actions reversible is to have a command act as if it were executed when, in fact, it has been deferred (Norman, 1983). This gives the operator the opportunity to reconsider and reverse the action.

> This strategy may be more appropriate for information handling tasks than for plant control tasks. An example is the command to delete files on a computer which used to be executed immediately. However, many newer computers place the files into storage where they may be deleted automatically later or remain indefinitely until the operator issues a separate command. Such reversible delete features may be useful in NPPs for recovering important data, such as deleted trend information or data on the safety shutdown system described in the discussion of inspection and transfer steps. Norman (1988) cautions that as users incorporate such features into their work they come to rely on them, assuming that they always will be available and operate correctly. Hence, failures of error-tolerant features may have worse consequences than if they had not been provided in the first place. For example, operators may be more willing to delete files if they think they can always recover them.

Automatic Intervention – This means of error mitigation involves actively preventing the propagation of consequences by blocking the evolution of an undesirable state, and may include automatically starting compensatory actions (Rouse, 1990). In NPPs, automatic systems intervene when conditions deviate from normal limits and approach undesirable states. Examples include automatic shutdown features which can safely stop the system, and override features which drive variables back into acceptable ranges.

5.3 Human Performance Considerations for Specific Control Actions

The preceding section discussed general categories of human errors associated with soft control systems and general design approaches for making them more tolerant of operator errors. The current section considers human performance related to specific actions taken when operators use soft controls. These actions include monitoring the system and process status, selecting and retrieving a control, providing control input, monitoring the system and process response, taking multiple control actions, using modifiable characteristics of soft controls, and coping with inconsistencies across the HSI. These discussions include specific instances of the categories of human errors described in Section 5.1. In addition, differences in the actions required for using conventional and soft controls are described because they may impose demands on operators that affect their performance and, ultimately, plant safety.

5.3.1 Monitoring the System and Process Status

Operators scan controls and displays to maintain awareness of the status of the plant and individual control systems. Specific controls may be checked for availability (e.g., in- or out-of-service), control mode (e.g., automatic, manual, cascade), and deviations between actual values and setpoints. This monitoring allows operators to identify the need for control actions and to determine the controls available for use.

In conventional hardwired CRs, each control typically is uniquely located on a control panel or board. This layout facilitates the overall assessment of the plant's status because many or all controls and displays are visible at the same time; navigating between various controls and displays is fairly simple. Operators can rapidly look for anomalies by scanning a control board from one end to the other. For example, at the start of a workshift, operators often scan the entire main control board to assess overall plant status. Also, monitoring of a particular area may be facilitated by the functional organization of a control panel, since functionally related controls and displays are usually grouped.

Computer-based HSIs have characteristics that may constrain monitoring and the use of controls. Potential problems in human performance are described below.

Fragmented View of Control Status

Unlike conventional hardwired CRs, computer-based HSIs often do not provide simultaneous access to all controls and displays. Because of the limited size of the displays, only a portion of the total set of the controls and displays can be viewed at once; this is described as the keyhole effect. Thus, monitoring the overall status of the control system and plant depends on the operator's ability to locate the proper displays. The keyhole effect results in serial rather than parallel access to displays. Once a display is retrieved, the operators' access to the controls contained in the display may be additionally constricted. For example, in some computer-based HSIs, detailed information about controls, such as control setpoint, alarm setpoints, controller output level, and component identification number, cannot be seen within plant process displays. Instead, it must be accessed serially through a set of separate detailed displays.

When this information can only be accessed for one component at a time, high mental workload demands are placed on operators as they rapidly navigate between the displays. Operators must view components separately, memorize status information, and then make comparisons. As a result of this fragmented view of controller status, operators may lose their understanding of the status of the individual controls and their relationship to the overall control system (Ranson and Woods, 1994). For example, some systems have displays giving detailed information about groups of controls, but accessing them removes or partially obscures the plant process display. When multiple display screens are not available, this can prevent the operators from seeing the status of the individual controls and the control system's status within the context of the plant process so that the operators may not have an adequate understanding of any of them. Thus, determining the overall status of a plant system and detecting control-setting anomalies may be limited by the operators' inability to quickly scan the settings of all control devices at one time via a computer-based HSI.

When soft controls are used, the operators' ability to assess the status of the control system and select control actions depends, in part, on the adequacy of the display system and the availability of display devices. The display system should be organized so that operators can rapidly find the displays with the controls of interest and obtain detailed information. The displays' content should allow operators to view groups of controls together to examine relationships and identify incorrect configurations. For example, they should be able to determine whether individual components, such as valves and pumps in a fluid system, are properly aligned and in proper control modes. Dependencies between components should be visible. Thus, when a component is in a cascade control mode (i.e., its setpoint is provided by another controller) , the operator should be able to observe the status of that component's controller and the higher level controller that provides its input.

The structure of the display system and display-retrieval mechanisms should allow displays to be identified and retrieved promptly, minimize the number of transitions between them, and allow detailed and overview information to be used together effectively. Some display systems use overlays that present detailed information on components, such as their setpoints and identification numbers, on the process displays. Operators then can use the overlays to simultaneously view this detailed information on multiple components; afterwards, it can be removed to reduce visual clutter.

Finally, displays are used by operators for a variety of short-term needs, such as retrieving controls and taking control actions, and long-term needs, such as monitoring the status of important variables. There should be adequate display space to allow such usage. This may be accomplished by providing spatially dedicated display devices for important or frequently used displays, and general-purpose ones for the variety of monitoring and control tasks.

5.3.2 Selecting and Retrieving a Control

Before undertaking a control action, the operator must select the desired control. In a conventional CR, selecting a hardwired control entails finding its location on a control panel, usually by looking and walking around the CR. Hardwired controls generally have a single location, often within a functionally related group of controls. With human factors upgrades made as part of post-TMI detailed CR design reviews (e.g., labels and demarcations), finding a specific control is reasonably easy, although some errors may still occur.

In contrast, soft controls usually do not have the same degree of spatial dedication. The CR may have more than one display device that accesses the control. Each of these display devices may contain multiple displays with the desired control. Complex interface management tasks, such as display selection and manipulation, may be required to choose and retrieve a soft control, and this complexity may interfere with the operators' ability to promptly access needed controls (IEC, 1993). To address these concerns, EPRI (1993a) recommended that the access to a particular soft control be flexible and multiple; providing multiple means can decrease the likelihood that an operator will not be able to access an essential control. If there are multiple methods and the operator has difficulty with one, then another way can be used.

Typical interface management tasks involved in selecting and retrieving controls are described in this section. Accessing a soft control may entail four steps: (1) selecting a display device, (2) accessing a selection display and choosing a variable to be controlled, (3) accessing the input field, and (4) coordinating the input fields with the selection displays. Each step may impose demands on human performance that result in errors or delay the operators' response.

Selecting a Display Device

When an HSI contains computer-based display devices, selecting a display device entails considering the controls and displays that it can access, as well as taking into account the proximity of the display device to other task demands in the CR. Having multi-function display devices in the CR provides operators with more options for selecting display devices. But having to consider the location of the display device may interfere with scrutinizing the content of the controls and displays.

In placing display devices in the CR, thought should be given to the fact that multiple devices may contain controls and displays that appear similar, but differ in content. Operators may accidentally choose the wrong control or display by choosing one more conveniently located near task demands. Similar looking controls and displays should be adequately identified.

Accessing a Selection Display and Selecting a Variable to be Controlled

Unless a soft control is dedicated to a single variable, the operator must pick the variable that is to be controlled. This may entail two steps. First, the operator must reach the correct display from which the variable will be selected; this is called a selection display because the operator selects components or variables from it. Display systems contain a variety of selection displays, such as menu and mimic displays. The second step is to access the correct component or variable from within the selection display. Soft controls provide a variety of means to do this including dedicated keys, commands, and direct manipulation (e.g., pointing to the object that is to be accessed). These interaction methods and some associated errors are described below.

Selection via Command versus via Direct Manipulation – As a part of this research project, BNL conducted interviews and walk-through exercises with operators at two chemical plants that gave them two methods for accessing soft controls for taking control actions. The first method was via commands. Operators had to recall a three-digit code and then either press a dedicated button labeled with it, or type it on a keyboard. This produced a display with eight control interfaces. The operator then could access the desired control and manipulate it. If the wrong display was accessed, the operator had to scan the controls in the display, determine that the wrong display had been accessed, and then try a different three-digit code. Operators were required to remember 20 to 40 of these three-digit codes, depending upon their assigned console.

The second method was direct manipulation. The operator could use a pointing device to indicate the desired component in the mimic display, which caused the control interface display to appear. All the operators used direct manipulation because it allowed them to make their selection within the context of the mimic display. There was a close coupling between the action of monitoring the process and the action of selecting a component to control. They avoided the list display because it had higher mental workload (i.e., remembering the three-digit identification code) and a less direct connection between monitoring and selecting. The process of recalling codes (and recovering from any associated errors) introduced delays and uncertainties into the selection process.

Misordering the Components of an Action Sequence – When a soft control is used to manipulate multiple variables, the operator must select one variable (i.e., a plant component), perform the control action, and then deselect it before moving to the next. Errors involving misordering the components of an action sequence (Norman, 1981, 1983) may occur if the operator does not undertake these operations in the proper order. If the operator fails to deselect the last previously controlled variable, the control action may be performed on the wrong plant component (Shaw, 1993). To avoid misordered action-sequence errors involved in selecting variables, soft controls may minimize either the number of steps or the sequential constraints on those steps.

Description Errors – Selecting the wrong component can also be caused by description errors (Norman, 1981, 1983, 1988; Lewis and Norman, 1986) resulting from the similarity of display pages and the similarity of symbols and labels used to designate selectable objects. These similarities may increase the likelihood of operators selecting the wrong component or failing to detect that the wrong component was chosen (Shaw, 1993). For example, the following incident occurred at a chemical plant. "An operator should have moved three tons of water into reactor A. He misread the display and moved three tons into reactor B which was already full. There was a large spillage of cyanide material" (Kletz et al., 1995, p. 32).

The following are specific examples of description errors in selecting plant components or variables.

- *Text Displays* – Displays with text formats used for selecting plant components may be especially prone to description errors. For example, Kletz et al. (1995, p. 27) states, "On one plant an operator could call up a list of valves, select one of them by entering a two-digit number and then operate it. Inadvertently, the operator called up the list for the wrong section of the plant, did not notice what had happened – as all the lists looked alike with similar numbering – and opened the wrong valve. Many tons of chemical were lost to drain."

- *Mimic Displays* – Description errors may result from the similarity of mimic displays. Excessive reuse of layouts and display elements in displays may cause them to look alike, causing operators to mistake one display page for another. For example, Kletz et al. (1995, p. 27) states, "The pages of a computer display sometimes look alike. This saves development time, and thus cost, but can cause confusion during an emergency. An operator may turn to the wrong display page and not realize he has done so." Thus, consistency in display design has its limits. Using a standard set of symbols and layout conventions is important for reducing the mental workload associated with finding and interpreting the information in the displays. However, the practice of creating displays by copying an existing one and then minimally modifying it for another application should be avoided because it may result in a set of displays that are so similar that an operator may easily confuse them.

Mimic displays are often used for selecting plant components. For example, an operator may access a component by pointing to the symbol in a mimic display that represents the plant component that is to be controlled. However, the wrong component may be picked if another component has a symbol similar to the one desired, or if the symbol is not clearly labeled. Symbols used on mimic displays should be visually distinct and components adequately marked to identify them correctly. Mimic displays should provide adequate context to support the correct identification of components. For example, when mimic displays have multiple components that appear similar, their relationships to other components, such as direction of flow and hierarchical relationships, should be clearly depicted.

- *Multiple-Loop Controller* – The multiple-loop, programmable, digital controller is one type of soft control prone to description errors. These devices are considered for upgrading analog control panels because they can fit in the same panel space as a hardwired analog controller or display. They have multiple channels, each capable of acting as a separate control device. For example, a single controller may be able to control 10 variables, each on a separate control loop. The operators access these loops through the device's user interface. Each channel is typically accessed from a separate display page. However, due to the limited display space of the device, operators may fail to correctly identify the loop they have accessed. Operators may make "wrong loop" errors by changing control setpoints on the wrong loop (wrong plant variable) (Shaw, 1988).

Accessing the Input Field

Input fields are areas of the display that are used by operators to enter values into the control system. Soft controls show input fields in a variety of ways. Three typical configurations were described in Section 4.4.2: integral with the display, as a window within the display, and as a separate screen. With the latter two configurations, errors can occur when operators associate the input field with the wrong plant component because the display system does not have adequate cues to help the operator associate the input window with the correct component in the selection display. For example, if operators are interrupted while selecting a component, or have multiple input displays open at one time, they may forget which component is associated with the input display. Operators may perform a control action in the input field, thinking that a different component is being controlled (Ranson and Woods, 1994); this is considered a type of mode error.

The design of the HSI should support the operator's understanding of which plant component is being controlled. First, it should provide clear feedback indicating which component was selected; an operator looking at the selection display should be able to quickly determine this. Second, the HSI should provide cues linking the input field to the plant component or variable so that, in looking at the input field, it should be apparent which plant component is being controlled. Starting at the input field, the operator should be able to quickly trace the component or variable back to its representation in the selection display and to other displays depicting the plant process. The following are strategies for designing the HSI to achieve this.

Graphical Coding – When a component is selected it should be visually distinct from other components in the display. Graphical codes, such as shape and color coding or symbols might be used. A similar code may be applied to the input field to strengthen its association with the component. For example, when a component is selected from a mimic display a colored border may appear around it along with a similar border around the input field to visually associate the two.

Labeling – The input field should be labeled with sufficient information to uniquely identify the component that is to be controlled. The label should include a singular identification code for the component that matches its representation in the selection display. Other information may describe the component (e.g., valve, pump, breaker) and identify and describe the components that immediately precede and follow it in the system. This particular labeling scheme had been adopted for one chemical plant visited during this study.

Landmarks – Woods (1984) describes landmarks as one of several supporting strategies for easing transitions between successive views of a display system. Landmarks are prominent display features that appear in multiple displays and can be seen at a glance. They are familiar reference points that orient the user when moving from one display to another. For example, if a component is selected from a mimic display, the input field could contain a copy of the component's icon. In addition, the input field could contain a small representation of a part of the mimic display. The input field could depict the component that is to be controlled, as well as the preceding and following components. [This approach was shown in a presentation describing I&C upgrades for the Temelin Nuclear Plant (Orendi, 1996).] These methods may support the operator's understanding of which component is being controlled and its relationship to the rest of the plant system.

Animation – Animation is a visually salient technique for demonstrating relationships between display objects. It can strengthen the association between a component and its input field. Animation may be applied when a component is first selected; the input field may be made to appear as if it were "popping out" of the selected option. When the input field is closed, it could appear to go back into the option, similar to the way a closed file goes back into a folder in the user interface of a Macintosh computer.

Many soft controls were observed during our site visits. The systems varied considerably in the use of the above approaches. This diversity indicates that there is no industry consensus on the best way to represent associations between the input fields and the selection displays. Important trade-offs exist between the need for descriptive detail and the need to reduce visual clutter. Evaluations should also consider the likelihood of confusing the components. For example, if a soft control manipulates a set of components arranged in parallel, then each component would have the same components preceding and following it. Thus, other descriptive information may be needed to identify the components.

<u>Coordinating the Input Fields with Selection Displays</u>

When the input field appears as a window within the selection display, window management techniques may be needed to coordinate the input field with the selection displays. The following are common approaches.

Dedicated Space for Input Field – The displays of some soft controls have a space dedicated to the input field that is normally blank. When a component or variable is selected, the input field appears in that space. This approach limits the amount of space that can be used for the display because the dedicated space must be reserved.

Overlapping Window for Input Field – For other display systems, the input field appears as a window that overlaps and obscures a part of the selection display. Either the operator or the display system must position the window so it does not interfere with the operator's actions. Assigning this task to the operator creates yet another interface management task, detracting from the primary role of controlling the plant. Some systems automatically place it at the end of the display farthest from the selected component, so that the operator can view the portions of the display closest to the selected component. However, the rest of the display can only be seen by closing the input field window. While this approach relieves the operator of the window management tasks, it also limits use of the selection display.

Another way to prevent confusion between input fields and components is to restrict the types of operations that can be carried out with them. Misidentification of an open input field can be exacerbated by an interface in which multiple input fields can be opened at the same time with few restrictions on their access. Many computer-based display systems restrict access to input fields, such that only one can be opened at once; this reduces demands on the operator's memory since only one control must be remembered. Other systems with cursor interfaces cause the input field to disappear as soon as the cursor is moved out of it. For example, clicking on a component in the selection display causes its input field to appear with the cursor automatically appearing within it. If the operator moves the cursor outside the input field, it closes and the component must be selected again if the operator wishes to perform subsequent control actions upon it. This arrangement forces the operator to access one control at a time, which lessens the cognitive demands of keeping track of open input fields. However, it may increase operator response time and lead to errors related to misordered action sequences. (See Section 5.1, Human Error and Soft Control Use, and Section 5.2.5, Performing Multiple Control Actions, for discussions of human performance associated with sequential constraints on control access.)

Some computer-based display systems have displays that give access to as many as eight controls at one time. However, these group displays typically require an entire display screen. Separate devices are needed to view the corresponding plant process and selection displays at the same time. More research may be needed on the potential benefits and costs associated with restricting access to one input field at a time.

5.3.3 Providing Control Input

To provide a control input, such as starting or stopping a pump or changing a control setpoint, operators must take actions using the control's user interface. This often includes accessing an input device, such as a keyboard or mouse, providing input (e.g., On/Off or numerical setpoint value), verifying the input, and verifying that the input was accepted by the control system.

The design characteristics of soft controls can impose demands for entering inputs that differ from those of conventional analog controls. The following describes these human performance concerns in greater detail.

5.3.3.1 Interface Management Tasks and Control Input Tasks

Some input actions control plant equipment while others control the HSI (e.g., cause display screens to appear). The design of the user interface should prevent the operator from confusing interface management tasks and control input tasks and possibly operating plant equipment inadvertently; the HSI should clearly distinguish between them (EPRI, 1993a). These actions should look different to the operator, with different interfaces and, possibly, different input devices. For example, if a mimic display has a direct manipulation interface, the operator should not perform the same action on the same type of display objects (e.g., clicking on icons) to access additional information and to actuate plant equipment. Because these actions and interfaces are so similar, description errors may result (e.g., an operator may click on an icon to obtain information but instead operate a piece of plant equipment). Description errors may be prevented by making these operations appear and behave differently. For example, clicking on the equipment icon could retrieve an input field rather than operating the plant component in a single step. Clicking on an icon to obtain information could present the requested information, or retrieve a different type of input field. Also, the control actions and interface management actions could require different confirmation steps.

5.3.3.2 Providing Discrete Input

Many control actions used for operating the plant or manipulating the display systems involve switching between discrete settings. For example, breakers and valves may be changed from open to closed. Automatic controllers may be changed from one discrete control mode (e.g., manual, automatic, and cascade) to another. Interface management tasks include selecting displays and opening and closing windows. In discussing physical control devices, Chapanis and Kinkade (1972) describe discrete-adjustment controls as devices with individual settings that can be selected with a gross movement (e.g., they snap into place), and contrast them with continuous-adjustment controls, which require a slewing motion and fine adjustment and should be used for precise adjustments along a continuum. They state that discrete-adjustment controls should be used rather than continuous-adjustment controls "...when the controlled object is to be adjusted for discrete positions or values only. Discrete-adjustment controls are preferred when a limited number of settings is required, or when precision requirements are such that a limited number of settings can represent the entire continuum" (p. 346).

These same considerations can be applied to soft controls. If a task requires an operator to select a setting from a set of individual ones, then a discrete-adjustment control should be provided. Often, computer interfaces have a continuous-adjustment control, such as a slider or scroll bar, for viewing a group of individual options. Because selecting a specific setting with a continuous-adjustment control can be awkward, there should also be a discrete-adjustment control, such as a set of arrow buttons.

Discrete-adjustment controls can have momentary or continuous operation. The former produce an effect only while the user is providing an input, such as a button that sounds a buzzer for as long as it is depressed. Controls with continuous operation produce an effect until the user gives the next input or until a predefined action sequence is terminated by some criterion. An example is a light switch with a turn-on, turn-off pattern of operation.

For soft controls, discrete-adjustment interfaces should also provide feedback about their operating state (e.g., On/Off) after activation. If it is of the continuous operation type, there should be continuous feedback on its current state. For example, the Macintosh computer user interface uses checkboxes for options that have two states (Off and On) and remain in that state after selection. When the option is On, an "X" appears in the box. When the option is Off, the box is empty; the checkbox indication remains until the state of the control is changed. If an input interface has multiple settings, feedback should indicate which setting was selected.

Discrete-adjustment interfaces for soft controls may be located on-screen or off-screen. The most common on-screen discrete input interfaces are buttons and interactive menus, usually activated by pointing devices, such as touch screens and cursors. Multiple discrete-adjustment interfaces can be combined to work in a coordinated fashion. For example, a group of buttons may be arranged in an array such that the output of only one button is active at one time. Apple Computer, Inc. (1996) calls these radio buttons because they resemble the arrangement of buttons on a car radio (i.e., each button is dedicated to one radio station). Other on-screen interfaces exist, such as rotary selector switches activated by gesture input devices (Greenstein and Arnaunt, 1988), but are not commonly used in process control.

The function key is most common off-screen discrete-adjustment interface, causing the system to perform some predefined function for the user when it is activated. Function keys are usually located on a keyboard or key pad, but may also be located along the VDU. For example, some multifunction displays in aircraft cockpits consist of a CRT with physical push buttons around its perimeter. Labels appearing on the CRT indicate the current function of each button. Off-screen, discrete-adjustment interfaces may include other physical controls. The same set of physical controls that provide discrete inputs in conventional hardwired control systems (e.g., rotary selector switches and toggle switches) may be used similarly for soft controls.

5.3.3.3 Changing Setpoints and Other Continuous Variable Inputs

Many control actions involve entering a value from a continuous range, for which Chapanis and Kinkade (1972) recommend continuous-adjustment controls. One important example for process control is entering a setpoint. When operators change these setpoints, they typically select a value from a range of acceptable ones, causing the setpoint to increase or decrease relative to its current value. The control system then tries to keep the plant variable within an acceptable range around the control setpoint.

In conventional CRs, spatially dedicated, hardwired control devices are typically used. Often, a single control device is dedicated to a single variable. To provide an input, physical movement of the device is often required. In many cases, the position of the controller corresponds to the value of the input such that a large change in the setpoint value would require a large movement of the control from its current position. For example, the flow rate of a fluid system may be manipulated by rotating a dial to a desired setting. The value may be increased or decreased by rotating it in opposite directions.

By contrast, soft controls may not be spatially dedicated; a single input device may be used to manipulate more than one variable. Also, operating a soft control may not require physical movement of the input device. Human performance considerations are described below for a variety of input devices.

<u>Keyboards and Number Pads</u>

Many soft controls feature keyboards and number pads for entering input values. When entering multiple-digit values, the operator's key strokes are prone to misordered action sequence slips due to transposing, adding, or omitting digits. This is because the movement of the keys does not correspond to incremental changes in the variable, as in the example of the physical dial where a small misadjustment normally results in a small error in the input value; in contrast, a small typing error at a keyboard may cause either a large or small input error, depending upon the digits entered. Numerous events resulting from errors in typed input have been reported in NPPs, chemical plants, commercial aviation, and other computer-based systems. Our interviews with operators at nuclear power, fossil power, and chemical plants indicated that this is a widespread problem. This also is a problem in medical devices; patients undergoing afterloading brachytherapy treatment have received incorrect dosages of radiation when numerical values specifying parameters of their treatment (e.g., exposure time) were typed incorrectly into the computer that controls the radiation source (Callan et al., 1995). The following are four example events reported in the literature or event reports. The last two events were also described in Section 5.2.3, Error Mitigation:

- "Incorrect keypad entry caused a Group 6 isolation and standby gas treatment start [in a NPP]" (Lee, 1994, Table 2).

- "A pilot set the heading in a plane's inertial navigation system as 270° instead of 027°. The plane ran out of fuel and had to land in the Brazilian jungle. Twelve people were killed" (Kletz et al., 1995, p. 31).

- "An operator wanted to reduce the temperature on a catalytic cracker from 982° F to 980° F. Unfortunately he pressed the keys in the wrong order (908) and immediately pressed the Enter key. The PES [programmable electronic system] responded with impressive speed, slamming slide valves shut and causing a flow reversal along the riser" (Kletz, 1993, p. 260).

- “An operator was changing the feed rate from 75 to 100 gpm [gallons per minute]. She inadvertently typed 1,000 gpm into the computer-based controller, which responded by fully opening the feed valve. The excessive feed caused a rapid pressure rise that was relieved to the flare” (Lorenzo, 1990, p. 18).

The HSI should include features to reduce the likelihood of typing errors, particularly for variables in which errors may threaten plant safety (Kletz, 1993; Kletz et al., 1995; Shaw, 1988). Approaches for prevention should provide feedback about the input value, preferably in both numerical text and graphical formats. A digital readout should indicate the magnitude of the input value, and reference values should be provided to aid operators in evaluating the correctness of input values. As an example, for control setpoints the reference values should include the actual value of the process variable, the current setpoint value, and the alarm limits. Graphical feedback may include a barchart with an analog representation of the entered value (e.g., the length of the bar corresponds to the magnitude of the value). Error detection approaches may include confirmation steps, gags (e.g., input validation checks for unacceptable values), and warnings. Error mitigation measures, such as delaying or deferring the system’s response, may create additional opportunities for the operator to change an incorrect value.

Incremental Input Devices

As opposed to alphanumeric keyboards, incremental input devices change the magnitude of input values sequentially. Using them to produce a large-magnitude change in a value usually requires a large input action or numerous small ones. These actions are a form of feedback about the magnitude of the change that may reduce the likelihood of producing large errors, or increase the likelihood of detecting them. The following describes human performance concerns associated with these devices.

Nulling Problem – This concern occurs when a device that requires physical movement is used to control more than one variable, and the absolute value of an output value is proportional to the position of the device (Buxton, 1986). The problem is illustrated with the following example. Two variables, A and B, having ranges of zero to 100, are controlled by the same input device. Initially, variable A reads zero, variable B reads 10, and the control is initially set to manipulate variable A. Variable A is raised to 100 by moving the control from its minimum to its maximum position (e.g., a slider is moved from the bottom to the top position, or a dial is rotated from its beginning to its end). Next, the operator wishes to increase variable B to 100. However, the input device is already at its maximum position. The operator must move the input device to the 10 position (to match variable B) and then reset it before beginning to manipulate variable B. This operation takes time to learn, time to carry out, and is a source of error.

Buxton (1986) states that the nulling problem results from the designer’s choice of a device directly linking the variable values to its absolute position. An alternative is to use an input device that links the value of the variable to a *relative* position, such as a set of arrow keys that increases or decreases the value with each press. Another example is a rotary dial that increases the input value when turned in one direction and decreases it when turned the other direction, but does not stop turning at its minimum or maximum end points (e.g., the dial is free to turn but the value stops changing at the maximum or minimum). Another alternative is an automatic reset feature. When a variable is selected, the device automatically aligns itself with the current value of the new variable. For example, if a slider interface were used, it would automatically move from the 100 to the 10 position when variable B is selected. Such an automatic reset feature may be easy to implement on a computer-based input device, but more difficult on a physical device, such as a mechanical slider.

Soft Sliders – A soft slider is an input format used to directly manipulate a variable. A soft slider resembles a barchart with a pointer directed toward the current value. The pointer can be slid along the length of the barchart scale to the desired value, usually via a pointing interface, such as a touch screen or mouse. Soft sliders are used when the range of possible values and the ratio of a value to that range need to be displayed (NASA, 1992).

Hoecker and Roth (1996) compared soft sliders to soft arrow buttons (i.e., buttons that increase or decrease a variable by a given amount when they are pressed). A prototype user interface was used in a simulated feedwater control task for a PWR NPP. There were no statistically significant differences in task completion times for the two input methods. However, the participants, who were experienced in NPP operations, clearly preferred arrow keys to the sliders. This surprised the investigators because early studies of human-computer interactions suggested that well-designed drag-and-drop direct manipulation, as represented by the slider, would yield superior performance as well as user acceptance compared to the arrow button, which more closely resembles conventional "hard" controls. While both input methods were considered direct manipulation interfaces, the slider was considered more direct because it allowed the operator to move a pointer to the position of the desired setpoint. In contrast, the arrow buttons represented an intermediate operation; operators pressed the buttons, which caused the slider to move. (The slider was both an input device and the indicator for the setpoint value.)

The operators' preference for the arrow buttons was attributed, in part, to differences in the types of input actions required. The arrow buttons were stationary and required pressing, while the slider moved vertically on the display screen and required the user to move the mouse across a mouse pad. The characteristics of both devices resulted in undesirable actions; however, the effects were more severe for the slider. First, it was possible for the operator to move the cursor off the target area of the slider or arrow button, which interfered with the control action. This problem occurred more often with the soft slider because the user had to move the mouse in a straight line while avoiding side to side motions, and it had no constraints, like those of a physical slider, to prevent these motions. Moving the cursor off the target area was less of a problem for the stationary arrow buttons because the mouse was only moved to position the cursor over a button. That is, a value is changed via an arrow button by clicking rather than moving the mouse. Second, operators undershot or overshot the intended setting more often with the slider. Because it allowed continuous motion in the vertical direction, fine adjustments were needed to achieve the desired setting. By contrast, the arrow buttons changed the setting by a discrete amount with each press so operators undershot or overshot the intended target fewer times. Accordingly, operators were able to time share when operating the arrow buttons – they could look at other process displays while using the arrow buttons. Operators could not do this when using the slider because of the greater attention required to position it.

Operators commented that they had to work much harder, mentally, to operate the soft slider than they did to operate the arrow buttons. They also recognized that "in a more realistic task situation, say a fast-paced event with higher premiums for speed and against error, and involving multiple controls, displays, and operators, this seemingly small difference between interaction modes could have multiplicative interfering effects on their ability to perform under pressure" (Hoecker and Roth, 1996, p. 215).

Many computer-based HSIs, such as those for Space Station Freedom (NASA, 1992) and process industries, feature a dual interaction method that combines arrow buttons with a slider. Typically, one arrow button is located at each end of the slider, allowing the operator to use either the slider or the arrow buttons. Discussions with human factors designers for a foreign vendor of computer-based NPPs indicated that some operators use soft sliders for coarse adjustments of input values, and then the arrow buttons to finally adjust the input value.

NASA (1992) gave the following guidance on the physical characteristics of soft sliders:

- The range of values should be indicated on horizontal sliders with the low value being on the left and the high value on the right, and on vertical sliders with the low value on the bottom and the high value on the top.

- The digital value should be displayed.

- The length of the slider depends on the range of values depicted and the increment between individual values. (For a minimum length, they recommend that the slider subtend at least 5.7 degrees of visual angle, determined from the expected viewing distance. For example, for a viewing distance of 20 inches, the minimum recommended length is 2 inches. Their recommendation for a maximum length of a slider presented on a CRT is the width of the CRT for horizontal sliders, or its height for vertical sliders, minus ½ inch of clearance space at each end.)

- When the bar in the slider depicts a range of values in which part of the range represents critical information, the appropriate code should be used for that critical range.

The physical dimensions of the soft slider should be sufficiently large to allow the operator to quickly read the current and target positions with the required degree of precision and also allow the slider to be positioned with the required accuracy. Accuracy may be affected by the characteristics of the input device (e.g., mouse devices may allow more accurate positioning than a touch interface due to the relatively large size and irregular shape of the finger). In addition, a very long slider may produce slow response times due to the long distance that must be traveled, and the need to keep the pointing device on the linear path of the slider.

The Macintosh Human Interface Guidelines developed by Apple Computer, Inc. (1996) are consistent with the NASA guidance. In addition, Apple suggests labeling the slider to indicate the values within its range. For example, a slider used for controlling a speaker's volume may be marked in increments from 0 to 7. Apple suggests providing additional labels or graphical codes to describe the individual values. For example, labels describing their relative volume may be applied to each of the intervals (0 to 7). Alternatively, Apple suggests applying graphical coding to the bar. For example, a slider used for controlling a screen's brightness may have a color code on its bar which varies from a dark shade for the low brightness values to a light shade for the high brightness values.

Arrow Buttons – The Macintosh Human Interface Guidelines describe arrow buttons as a pair of buttons pointing in opposite directions used to increase or decrease a value sequentially (Apple Computer, Inc., 1996). In process control applications, the operation of the arrow keys often is individually set for each soft control (i.e., for each plant variable). For example, for variables that have a relatively narrow operating range, a single press of an arrow key may cause a small change in the input value. For wide-ranging variables, pressing the arrow key once may result in a larger change in setpoint. Several soft controls with arrow-button interfaces were examined during our site visits. Operators were not concerned that different controllers responded differently to presses of the arrow keys. Instead, they stated that these differences allowed them to adjust each controller more quickly and accurately.

Some soft controls feature two sets of arrow buttons, one for making large changes and one for making small changes in the input value. One vendor of a computer-based HSI system for process plants provides one set of arrow buttons (designated with single arrows) for making small changes, often set at the smallest unit of precision presented by the soft control. There is a second set of arrow buttons (designated with double arrows) for making large changes; the size of their increment may be set by a control system engineer to equal 2% to 10% of the instrument's range. Human factors literature has little guidance on using separate sets of arrow buttons for large and small changes in input values, nor is there guidance on the relative size of inputs provided by the different sets of arrow buttons.

Some vendors of digital control systems install an adaptable gain feature for incremental input buttons. This feature allows the amount of change produced by the input button to vary as a function of some variable, such as the plant's power level. For example, a controller may be configured such that a single press of an arrow key increases its setpoint by a small amount when the plant is at high power, and by a large amount when the plant is at a low power level. While this property may make the controller more responsive to plant state, it may also put additional burdens on operators for anticipating its operation. They may require additional information from the control system to remind them that the controller will respond differently for particular conditions. Also, they may need feedback when providing input. For example, operators may think that a controller is faulty if it does not behave according to their expectations. Input verification steps may become increasingly important for providing the ability to check and change inputs before they are executed by the control system. Further study is needed of the potential effects of adaptable gain features on operators' performance.

Apple Computer, Inc. (1996) recommends that a set of arrow buttons have the following characteristics:

- A label that specifies what is being controlled (i.e., identifies the variable)

- A box indicating the current value either as numbers or in words

- A value that changes by a given amount for each button press

- A value that changes continuously as long as the button is depressed (i.e., until the user releases the mouse button)

- Color codes, such as highlighting, to provide feedback on their actuation state (e.g., while the user holds down the mouse button, the arrow button remains highlighted)

In some cases, it may not be apparent which way the indicator will change when an arrow button is pressed. For example, when arrow buttons are used to change a date display, it may be unclear whether actuating a button will incrementally change the days (changing the months when the last day is reached) or whether the months and days are changed separately after being selected by the user. The arrow buttons should be labeled to indicate their effects (Apple Computer, Inc., 1996).

Input Feedback

When data are entered for continuous variables, the HSI should provide feedback about the magnitude of the input. Two common methods in process control applications include digital readout and graphical presentations. Digital readouts simply show the input value numerically. Graphical presentations can depict it in many other ways, often as a barchart in which the height or length of the bar is proportional to the magnitude of the input value (Section 5.2.1). Digital and graphical feedback formats are often combined. For example, the magnitude of the input value may be depicted by a barchart that includes a digital readout.

Galletti (1996) gave an example of inadequate feedback for inputs provided via incremental input buttons. An NPP operator assumed manual control of a new full-range digital feedwater control system during power ascension. He tried to "bump" open the feedwater valve using a series of short intermittent key presses to increase the controller's output signal. However, the operator was unaware that each press corresponded to only about 0.1% demand so that the series translated into negligible changes in valve position demand. As a result, the plant tripped on low steam generator level. A review of this event determined that one contributing factor was that the feedback from the new digital controller to the incremental manual manipulations was not as clear as the floating-needle indications of the former analog system. Thus, the operator could not adequately detect the magnitude of the input or that it was too small for plant conditions.

Error detection may be aided by providing reference values that allow the operator to judge the appropriateness of the input value. Reference values commonly used in process control applications include the variable's range, alarm limits, and the current value, presented in digital or graphical formats.

Keyboards Versus Incremental Input Devices

Many soft controls used in process control applications provide the operator with the choice of changing control values by arrow buttons or a keyboard. Keyboard entry may provide some benefits in performance. For example, a large change in a setpoint value may be made faster via keyboard entry, compared to arrow buttons, because the keyboard allows the user to change the value in a single step by entering the new number, while incremental input devices require the change to be made as a series of steps (e.g., multiple presses of an increase or decrease button). However, industry experience suggests that entry via keyboard is more prone to error. For example, at one plant we visited in conjunction with this project, the HSI allowed the operators to choose between the two methods. Both provided feedback via a digital readout and a barchart. The operators stated that they almost always used the arrow key because they wanted to avoid errors in input and that they only use the keyboard for systems with low safety significance. However, there are few empirical studies that compare these data entry methods.

5.3.4 Monitoring System and Process Response

After making an input, operators monitor the effect on the plant's components and systems to determine whether equipment is responding as the operator intended (e.g., valves open or close, pumps start or stop). Plant process variables are monitored (e.g., temperatures, pressures, and flows) to determine whether processes are progressing toward the desired goal.

Feedback and Time Delays

Monitoring feedback from plant processes is specially challenging for operators because process dynamics exhibit time delays and fluctuations, and have other dynamic characteristics. Some potential problems in using computer-based controls relate to the delays in obtaining feedback. Time delays in processing the control input and receiving feedback on the control action can destabilize the system; in general, system lags are harmful to performance (Wickens, 1986). In complex systems, such as NPPs, there are numerous sources of time lag, including the response characteristics of controls and displays of the HSI, data transmission lines (e.g., data highways), and the plant process itself, all of which can make it difficult for operators to evaluate the results of their actions. Care must be taken to insure that time delays do not interfere with the ability of operators to control the plant. The response time of the HSI should be consistent with the rate at which the process changes, if the operator is to manipulate the control system in response to these changes.

Operators require prompt, clear feedback that their control actions are producing the desired response in the plant system and process. Lorenzo (1990) states that when there is no such feedback, operators tend to overreact. He cites the following example of chemical plant operators overreacting to slow feedback:

> A computer-based control system was so overloaded by a process upset that it ceased to update the video terminals in the CR. Unaware that the displayed information was inaccurate, operators unknowingly moved valves to their fully open or closed limits while waiting for the display to show some response. The mis-positioned valves worsened the upset, eventually causing an emergency shutdown of the unit when some relief valves lifted. (p. 15)

Carruth and Sotos (1996) give the following example of NPP operators having trouble responding to feedback that was not consistent with their expectations based on previous experience:

> The more modern upgrade technology required the operator to adapt to solid state displays and push button controls where they previously had edgewise panel meters and slide controls. The step-wise nature of the digital replacement's operation, in contrast with the continuous nature of the old analog equipment's performance, presented some serious conceptual challenges for experienced plant technical staff. The instrument technicians were accustomed to observing stable signals at all points in the old analog driven loops. The digital equipment in high gain applications, produced output signals that appeared to jump around. Technical experience with analog instrumentation made them somewhat uncomfortable and led them to believe something was wrong with the signals observed. (p. 1901)

This comment indicates that the behavior of digital equipment can challenge experienced personnel and should be addressed through proper design, implementation, and personnel training.

Coordination of Soft Controls with Process Displays

Soft controls should be coordinated with process displays so that operators can readily verify that the control actions have had the intended effect on plant systems and processes. Inadequate coordination between them can make such verification troublesome (Ranson and Woods, 1994).

5.3.5 Performing Multiple Control Actions

The following describes three problems in undertaking multiple control actions via soft controls: performing control actions in a rapid succession, suspending and resuming tasks, and coordinating the use of soft controls among multiple operators.

Performing Control Actions in Rapid Succession

Operators must sometimes perform multiple control actions rapidly one after the other, or in a particular sequence. For example, a set of components may be functionally related such that changing one affects the operation of the others. Thus, operators may have to make a series of adjustments to the set of components. However, some characteristics of soft controls may not be compatible with rapid or sequential operation. Computer-based control systems may impose additional sequential constraints on control actions, interfering with the operator's ability to transit rapidly between control devices, or to quickly check the status of multiple control devices. For example, to operate a soft control, the operator must first choose the component to be controlled, retrieve the display containing it, retrieve the control input field for that component, and then carry out the control action. In a typical implementation of soft controls, only one control can be accessed and operated at once. Although the time needed for the sequence of actions required to access and operate a control may not be long, the overall effect can be disruptive when there are demands to operate multiple controls in rapid succession. Also, this type of interaction may prevent the operator from checking the status of more than one control at one time. These effects can be aggravated when there are relatively long delay times in calling up display pages. The cumulative effect of these sequential actions may delay or interfere with an operator's response during a transient.

A multiple-loop programmable controller is a digital controller that can control multiple variables via independent channels, one per control loop. These controllers may be particularly prone to sequential constraints on control access. For example, Shaw (1993) states:

> The manipulation of one loop does not usually take much time. However, in an upset condition the operator must often operate several, or many, loops at almost one time. If he must operate the loops in a serial fashion, often the case in shared display, the time to gain control of a loop and make the desired manipulation adds up. For example, if the operator has to place each of twelve controllers in manual and adjust the output to specific values, the time can easily be excessive. (p. 266)

Concern over the ability of operators to respond quickly in rapidly changing situations, such as upset conditions, is reflected in EPRI's guidance for licensing digital upgrades: "The primary concern is that some digital systems do not allow manual manipulation of multiple controlled devices simultaneously" (EPRI, 1993b, pp. 5-9). The problems with this type of control sequence extend beyond simple timing concerns and include disruption of the operator's concentration on the control action because of the need to shift attention when navigating from one control to the next (Ranson and Woods, 1994).

The design of the HSI should facilitate control actions that must be performed in a rapid succession by minimizing the sequential constraints associated with the user interface. The following describes five approaches for doing this.

Minimize the Number of Displays – The number of displays that must be accessed to retrieve components that are normally controlled together should be minimized. Displays used for selecting plant components, such as menus and mimics, should include components that are monitored and controlled together.

Minimize the Number of Retrieval Steps – The number of steps that must be performed to retrieve related groups of controls should be minimized. Ideally, the operator should be able to reach each component from a selection display in a single, simple input action.

Minimize Delays – The amount of time required to select displays and to select plant components from the displays should be minimized. The display's response time should be almost instantaneous.

Increase the Number of Display Devices – An alternative approach for reducing sequential constraints on control actions is to increase the number of display devices available for control actions. Thus, dedicated display devices may be used (e.g., displays that are only used for particular control actions) or the number of general purpose displays increased (i.e., additional displays through which operators can access controls). In a visit to a fossil power plant that had a computer-based HSI, an operator described another strategy whereby two controls that were to be used together were placed on adjacent VDU screens so they could be operated simultaneously. This strategy was possible because sufficient general-purpose VDUs were available for these controllers and other display needs. Sequential access to plant components can be facilitated by advanced soft controls that control multiple variables together (Ranson and Woods, 1994). An example may be a controller for primary coolant subcooling margin that allows the operator to simultaneously change the control setpoints for coolant temperature and pressure by selecting a point in a plot of pressure versus temperature.

Reduce the Need to Operate Controls – Another approach to addressing sequential constraints on control access is to reduce the need to access controls to make adjustments. Digital control systems are often more stable than conventional analog systems (i.e., they may exhibit less oscillation and drift). Therefore, operators may not have to make control adjustments as frequently as with analog control systems. Also, designers incorporate features, such as more advanced levels of automation, to reduce the need for some control activities. These factors tend to reduce the consequences of sequential constraints on control actions.

Suspending and Resuming Tasks

A second problem related to multiple control actions is the difficulty that operators encounter when suspending and then resuming tasks. Often, they must interrupt a sequence of operations to undertake other tasks. One example is a transaction sequence, a series of steps performed by the operator to accomplish a larger task. The task of changing a control setpoint may involve multiple steps in selecting the variable and entering the new value. An important concern is the extent to which the operator's earlier entries in the sequence will be saved so that these steps do not have to be taken again. Wagner et al. (1996) describe various types of interruptions to transaction sequences and provide guidelines for interruption and resumption, given in Section 9.

Another task that may be interrupted is a control operation comprised of many individual actions. For example, aligning a fluid system may involve operating a set of pumps and valves in a particular order. When resuming a sequential control operation, the operator must recall the operation and the particular steps that were not completed. This operation may be difficult when the operator has difficulty remembering which task was suspended (see the discussion of Loss-of-Activation Errors in Section 5.1.6) and the HSI's design does not make retrieval of suspended tasks easy. The operators we interviewed stated that they sometimes know that a task has been suspended but cannot remember what it was. Their difficulty in remembering is exacerbated by the fact that the display containing the task had been removed from the display screen and could not be retrieved via the "Previous Display" button, which could only recapture the most recently accessed display. EPRI (1993a) states that operators should be able to return to and continue a suspended control action sequence with a minimum number of actions. They should not be required to restart an action sequence from the beginning when it has been temporarily suspended.

The following approaches support the operators in finding a display that contains a suspended task. They assume that the operator will remember that a task has been suspended, and that once the display with the task is found and seen, the operator will remember what needs to be done. The first approach is to include a "previous display" feature that goes back further than one display. A second approach is to provide an interaction history that lists previously accessed displays and opens them. A third approach is to provide a "bookmark" feature allowing operators to designate displays containing tasks in progress. Operating a bookmark can rapidly access the predesignated display. A fourth approach is to have enough display devices so that the suspended task does not have to be removed from view to perform other tasks (e.g., the task remains visible in a display screen until the operator resumes activity on it).

Coordinating Soft Control Use Among Operators

If a soft control can be accessed from more than one display device in a CR, then there should be some means for coordinating its use among operators to prevent them from interfering with each other's actions. One practice used in the process control industry is to assign control capability for each soft control to particular workstations. Thus, operators at any workstation can access it and observe its control settings. However, if they wish to change the control setting, they must ask the operator at the assigned workstation to do it. In this way, only one operator is responsible for each control but can maintain awareness of the needs of other operators. An alternative approach may be to restrict access to the control to only one operator at a time, preventing several people from trying to use the same control simultaneously. A group-view display allowing operators to more easily observe each other's use of the shared soft control also might suffice (Stubler and O'Hara, 1996b).

5.3.6 Using Modifiable Characteristics of Soft Controls

An important characteristic of computer-based user interfaces is that they can provide more flexibility in their configuration and operation than do traditional analog technologies. For example, some computer-based HSIs have displays that operators can modify for particular tasks or personal preferences. In plotting a trend, the operator may include or exclude plant variables, define coding for displayed items, and define axes and scales (O'Hara, Stubler, and Higgins, 1996). Another form of flexibility relates to display devices. Many computer-based HSIs in process plants enable operators to access displays from, or move display pages between, multiple display devices. Moray (1992) states that the philosophy of allowing each operator to reconfigure the user interface according to personal preference is gaining momentum in the design community. He believes that such flexibility can create new opportunities for error.

Moray (1992) describes a control console that was designed for an engine CR of a twin-engine ship. Controls for the port engine were on the left side of the console, and controls for the starboard engine were on the right. Above these hardwired controls were three CRTs (i.e., left, middle, and right), each capable of displaying information for either engine. This flexibility uses the display screens efficiently but creates the opportunity for violations of stimulus-response stereotypes. For example, if the middle CRT were occupied and the sailor opened two displays for the port engine, the second display for the port engine would appear above the starboard engine's controls. As a result, the sailor could unconsciously associate the starboard engine controls with the port engine display because they were located together. This potential problem was identified in a human factors evaluation made before installing the design and subsequently in tests with a dynamic simulator. While troubleshooting a problem with one engine, the sailor placed a display on the opposite CRT, examined it for a few moments, and then used the controls (the wrong ones) located below the screen to shut down the engine. After trying unsuccessfully for nearly half a minute to shutdown the engine, the sailor suggested that the simulator was faulty. Had this event happened at sea, one engine would have been throttled back hard while the other continued to run at full throttle. The ship would have made a full power turn at high speed, which could have severe consequences, such as a collision.

A similar problem can be envisioned in a hybrid HSI of a NPP. For example, an HSI consisting largely of hardwired controls and displays could be upgraded with a set of VDUs providing supplemental computer-based displays. A display for one train of a control system might be shown on a VDU located near the controls for a similar, but different train. The confusion resulting would be like that in Moray's example. Operators may unconsciously associate the wrong sets of hardwired controls and displays with the computer-based displays, and, consequently, operate the wrong control.

Moray further states that if several crew members share a control console and each can reconfigure portions of it according to personal preferences, there is enormous opportunity for introducing incompatibilities from moment to moment. One crew member's preference may cause serious problems for another. For example, if the engine console in Moray's example had been staffed by one crew member for each of the two engines, the preference of one sailor to position a display on the opposite CRT might create a stimulus-response incompatibility for the second person. Also, if an operator leaves a part of the console and later returns, the operator may be uncertain of the extent to which it was changed by another person. In addition, operators may have difficulty in remembering changes they made previously.

Moray offers two guidelines on stimulus-response stereotypes and the flexibility in computer-based HSIs. First, "The system should provide clear feedback to the user if stimulus-response stereotypes are violated" (Moray, 1992, p. 62). For his example of the engine control console, he suggests that when a display for one engine is placed over the controls for the other, the frame of the display should flash repeatedly. Second, "Where possible, automatic reconfiguring to preserve stimulus-response stereotypes should be implemented" (Moray, 1992, p. 62). He suggests for the same example, that when a new display is selected, the system should arrange the displays to achieve the most compatible arrangement possible (e.g., port displays on the left, starboard on the right, and additional port or starboard displays on the middle CRT).

Moray also suggests the following general principle for using flexibility in computer-based HSIs. "As far as possible, make the software responsible for preventing the human from [producing] a configuration that violates good human factors principles, and if the latter must be violated, minimize the violation and give very strong feedback as long as the violation remains" (Moray, 1992, p. 63). This principle can be applied to any reconfigurable feature of an HSI. For example, flexibility for positioning displays on multiple VDUs may violate population stereotypes. Operators who are accustomed to seeing mimic displays in an arrangement that reflects a left-to-right convention for the process flow may be confused if they are presented in a different order (e.g., displays representing later stages of the process flow are located to the left of displays representing earlier stages). As another example, some displays allow operators to select parameters and the symbols and coding schemes used to present them. Their presentation may be inconsistent with symbols and coding schemes used elsewhere in the HSI.

5.3.7 Coping with Consistency Across the HSI

This section addresses the integration of soft controls into an HSI, especially the consistency of soft control user interfaces. A hybrid HSI may contain a variety of soft controls, especially if the soft controls are installed as independent modifications, rather than in an integrated effort. In such a hybrid HSI, operators might be expected to make frequent switches between different tasks with different interfaces.

Consistency in human-machine interfaces is usually considered in a transfer paradigm in which the higher the similarity between two tasks, the higher the transfer of skills and training, and the higher the consistency. Negative transfer can generate human errors and potential safety concerns. However, Tanaka et al. (1991) suggest that human performance may suffer most when users frequently switch between tasks with slightly different, rather than very different, user interfaces. For example, they note that there is a high degree of consistency between the UNIX operating system and the Microsoft operating system MS-DOS, and less consistency between either UNIX or MS-DOS and the Macintosh operating system. Thus, the greater degree of consistency between UNIX and MS-DOS should provide superior transfer. However, users were more confused if they switched between UNIX and MS-DOS than if they switched between the UNIX and Macintosh operating systems. This finding suggests that conventional HFE principles of consistency and standardization for the more traditional user interfaces may need to be applied differently to computer-based systems.

A possible implication of this study may be that differences existing between computer-based and traditional analog interfaces in hybrid HSIs may not seriously burden operators' performance, but that differences between slightly different computer-based interfaces may impose high burdens, causing errors. EPRI (1993, 10.4-48) states: "The sequence of operations by which an operator takes a control action should be standardized to the maximum degree practical. These standard operating sequences and the method by which they are presented to the operator shall be established and validated through active simulation. These standardized practices shall be documented." EPRI's rationale is that standardized operating sequences should minimize training and reduce the potential for errors. Thus, the goal of trying to maximize consistency between user interfaces, as suggested by EPRI, may be counterproductive if it is the wrong type of consistency. Further research is needed to understand the dimensions of consistency important for reducing errors and ensuring effective performance across a variety of soft controls in a hybrid HSI.

6 GUIDANCE DEVELOPMENT

A set of guidelines was developed to address the human performance considerations identified in Section 5, using the source materials discussed in Section 3. The high-level design review principles from NUREG-0700, Rev. 1, supported this development. These principles were previously established from a review of research and industry experience on integrating personnel into complex systems. These principles reflect the important design goals of (1) maximizing personnel's primary task performance (i.e., process monitoring, decision making, and control), (2) minimizing secondary task demands unrelated to the primary task (e.g., the distracting effects of tasks, such as configuring a workstation), and (3) minimizing human error and making systems more tolerant to such errors when they occur.

These guidelines were developed in the standard format adopted in NUREG-0700, Rev. 1. An example is presented below:

> **9.4.4-6 Appropriate Use of Soft Sliders**
> A soft slider should be considered as an input device when the range of possible values and the ratio of a value to that range need to be displayed.
> ADDITIONAL INFORMATION: A soft slider (also called a slider bar or a scroll bar) is an input format used to directly manipulate a variable over a set range of values. Soft sliders are typically maneuvered via pointing interfaces, such as a touch screen or mouse. They may require careful hand-eye coordination to ensure that the pointing device does not leave the linear path of the slider nor overshoot or undershoot the intended target. If the operator's tasks do not permit careful hand-eye coordination, then other interfaces, such as arrow keys, should be used. The slider format sometimes is combined with arrow buttons.
> *Discussion*: This guideline was derived from NASA (1992) and Hoecker and Roth (1996).

Each of the guidelines is composed of the following components:

- *Guideline Number* – Within each section, individual guidelines are numbered consecutively. Each has a number which reflects its section and subsection location, followed by a dash and then its unique number.

- *Guideline Title* – Each guideline has a brief, unique, descriptive title.

- *Review Criterion* – Each guideline contains a statement of an HSI characteristic with which the reviewer may judge the HSI's acceptability. The review criterion is not a requirement and discrepant characteristics may be judged acceptable as per the procedures in the review process and considerations described in NUREG-0700, Rev.1. The word "should" is used to denote a recommendation. The word "may" is used to denote permission; it applies to a characteristic that is acceptable but not necessarily recommended (e.g., a preferable alternative may exist).

- *Additional Information* – For many guidelines, additional information is provided which may include clarifications, examples, exceptions, details on measurements, figures, and tables. This information is intended to support the reviewer's interpretation or application of the guideline.

- *Discussion* – This section summarizes the technical basis on which the guideline was developed. It may consist of identifying the primary source documents, the technical literature such as journal articles, or the general principle from which the guideline was derived. This section will be removed when the guidance is integrated into NUREG-0700, Rev. 2.

In place of the Discussion section will be a Source field:

- *Source* – The source field identifies the NUREG or NUREG/CR (or other document) that contains the technical basis and development methodology used for the guideline. As is standard practice, the source field will cite this document (as it will appear in its final form).

The guidelines, contained in Section 9, were organized into the following sections:

- General Characteristics
- Display Devices
- Input Devices
- Display Design
- Interaction Methods

7 SUMMARY

We consulted a broad range of sources in reviewing human performance concerns associated with soft controls, including general HFE literature on human-computer interactions. Also, we reconsidered general HFE literature on the control of complex human-machine systems, such as in NPPs, other process control industries, aviation, and medical devices. Another source was reports of incidents that resulted from human performance concerns associated with soft controls. These reports came from a variety of industries, especially NPPs, chemical manufacturing, and aviation. Yet another source of information was our interactions with industry personnel, including designers, operators, and trainers that took the form of interviews and walk-through exercises using the actual HSI or a high-fidelity training simulator. A variety of problems were identified from these sources ranging from accessing the wrong information, to making control inputs that were too big or small, to operating the wrong equipment. These problems were generally related to a lack of adequate feedback in the user interfaces of soft controls so that incorrect actions and their consequences were not always apparent to the operator. The more detailed analysis that was conducted in developing this guidance confirmed that these are important HFE issues that need to be addressed in the design of HSIs containing soft controls.

There was much general HFE information and documented industry experience to draw upon. First, theories and studies of human performance in complex human-machine systems that address such topics as HSI design, situation awareness, and human error were consulted. Second, general theory related to human-computer interaction was consulted; this literature described errors, especially slips, that occur with computer-based interfaces. Many of the descriptions of problems reported in process control industries lacked a structured treatment of human error. Reviewing these descriptions within the general framework of human error clarified the relationships between the soft control's characteristics, human performance effects, and system consequences. A third source of information was empirical studies of human-computer interaction; they tended to focus on less complex, user-paced activities, such as text processing. Their findings were considered when the HSI's characteristics and user tasks were relevant to process control and safety. The guidelines that resulted from this work are presented in Section 9. This guidance has been peer reviewed and is available for staff review of soft controls and for integration into a future revision of NUREG-0700.

The review of industrial practices based on literature, interviews, and site visits indicated that the solutions implemented to resolve the problems of soft control varied among organizations, which may reflect the fact that there are few formal HFE guidelines or standards for soft controls. Different industries, manufacturing facilities, and HSI vendors have different approaches to similar problems. Many of the HSIs we observed had a variety of error-prevention measures, although they were not always implemented consistently within the same HSI. This suggested that the human performance problems associated with soft controls have not been adequately solved by industry.

Interestingly, there were relatively few empirical studies of the use of soft controls in process control settings. This was noted by others (Hoecker and Roth, 1996) and was reflected in the comments from HSI experts we interviewed. Further research may be warranted to confirm any guidance that reflects accepted design practices for human-computer interfaces but that may not have a strong empirical basis. In addition, further review and analysis may be warranted to support the development of guidance for topics that have not been, and are not likely to be, addressed by the more general field of human-computer interaction, and to expand on topics for which available technical information was inadequate to generate review guidance.

The following topics were not fully addressed by the sources reviewed.

Time Delays and Control Stability

With the potential time delays in digital systems and the sequential nature of soft control actions, research is needed to better understand the relationship between time delays and stability of performance, especially in emergencies. Where delays affect performance, methods to support operators' performance should be identified.

7 SUMMARY

Input and Feedback Methods for Continuous-Variable Inputs

Industry experience showed that entering numerical values is error prone, especially when using a keyboard or key pad. However, the popularity of the keyboard as an input device suggests that it may have some advantages (such as speed) compared to other methods, such as arrow keys and soft sliders. Feedback about the magnitude of entered values can support the detection and correction of input errors; two common methods are digital readouts and barcharts. More information is needed on the relative advantages of combinations of input and feedback methods. Questions include the following: What are the relative error rates for inputs provided via keyboard, arrow keys, and sliders when they are paired with feedback from digital readouts and barcharts? What are the tradeoffs in speed versus accuracy between these methods? For example, does a keyboard and barchart combination yield superior performance in terms of both time and errors? Do interfaces that combine these features support or inhibit performance (e.g., sliders that incorporate arrow keys)? For arrow buttons, how is an operator's performance affected by using separate sets of arrow buttons for large and small changes in input values, or by adaptive gain features that vary the change produced by a button press as a function of another variable? Such information is needed to support the development of guidelines covering the appropriate use of these input and feedback formats. (These input and feedback formats are discussed, respectively, in Sections 5.2.1 and 5.3.3.3.)

Confirmation and Warning Messages

Both confirmation and error messages are prone to problems associated with the level of specification of operators' actions. For example, operators may confirm that the desired action is correct but not realize that the goal (e.g., the object being acted upon) may be wrong. Similarly, when receiving an error or warning message, users often cannot interpret the true cause of the problem. (The discussions of confirmation steps and warnings are discussed, respectively, in Sections 5.2.1 and 5.2.2.)

Sequential Plant Control and Interface Management Tasks

Many plant control tasks are sequential, and different tasks can have similar but different sequences. For example, some pumps require closing the downstream valve before starting the pumps. Other pumps require that it be opened. In addition, sequential operations are often involved in the use of soft controls (e.g., the operator must access a selection display, select a component, open an input field, and then enter the input value). Industry experience suggests that the sequential constraints of soft control access can interact with the sequential nature of control tasks and increase the likelihood of capture errors (i.e., starting one task sequence and finishing with another) and misordered action sequences (i.e., performing actions in the wrong sequence). (See discussions in Sections 5.1.4 and 5.1.5.)

Access to One Versus Multiple Input Fields at One Time

More research may be needed on the potential benefits and costs of providing access to one input field at a time versus multiple input fields. (See the subsection titled "Coordinating the Input Fields with Selection Displays" in Section 5.3.2.) Some alternatives may include having displays giving access to groups of controls, tools for managing multiple open input fields, and methods for performing serial access more quickly and accurately.

Intelligent Agents

These are computer programs that perform information processing tasks for the operator somewhat autonomously. They are being developed to perform information management tasks in chemical plants with a user-initiated notification concept. Intelligent agents can help the operator manage suspended tasks. However, the potential benefits must be weighed against the operator's burdens in supervising these agents, and any potential problems that may result from their inappropriate application. (See the discussion of Loss-of-Activation Errors in Section 5.1.6.)

Interaction of Soft Controls with Automation

Increases in automation of computer-based systems pose greater cognitive demands on operators, especially for understanding and maintaining awareness of the status and behavior of these systems. Soft controls play an important role in conveying status information to operators and allowing them to interact with the systems. However, automation may also affect the appearance and behavior of controls and displays. Human factors review guidance is needed to address the interaction of soft controls with automation. (See the discussion of Mode Errors in Section 5.1.3 as one example.)

Soft Controls and Display Space

The amount and type of display space provided through the HSI is important for supporting control and monitoring tasks. For example, assigning controls to dedicated display devices can improve access time by reducing the need to navigate through displays. Increasing the number of display devices can reduce conflicts between demands for short-term control actions and long-term monitoring actions. Also, having additional display devices allows the operator to more easily keep track of temporarily suspended tasks. Human factors review guidance is needed to look at the minimum amount of display space needed to support soft control use, and also the tradeoffs between providing dedicated display devices and general-purpose ones. (See discussions in Sections 5.3.1 and 5.3.5.)

Keyboards Versus Incremental Input Devices

Many soft controls used in process control applications provide the operator with the choice of changing control values via arrow buttons or via a keyboard. Keyboard entry may offer some performance benefits; however, industry experience suggests that entry via keyboard is more error prone. For example, large errors may result from typing mistakes. Further research is needed to examine the error rates associated with data entry via keyboards versus incremental input devices, especially when used in conjunction with features used for error prevention, detection, correction, and recovery.

Consistency of Soft Controls in Hybrid HSIs

A hybrid HSI may contain a variety of soft controls, especially if they are installed as a series of independent modifications, rather than in an integrated effort. In such a hybrid HSI, operators are expected to make frequent switches between different tasks with different interfaces. Studies of computer-based systems have produced some conflicting results on the effects of consistency. Thus, the goal of trying to maximize consistency between user interfaces may be counterproductive if the wrong type of consistency is achieved. Further research is needed to understand the dimensions of consistency that are important for reducing errors and ensuring effective operator performance across a variety of soft controls in a hybrid HSI. (See discussions in 5.3.7, Coping with Consistency Across the HSI.)

8 REFERENCES

Akamatsu, M. and MacKenzie, I. (1996). Movement characteristics using a mouse with tactile and force feedback. *International Journal of Human-Computer Interaction, 45*, 483-493.

Apple Computer, Inc. (1996). *Macintosh human interface guidelines*. Cupertino, CA: Apple Computer, Inc.

Aviation Week (1995a). Automated cockpits special report, Part 1, 142 (5), 52-65.

Aviation Week (1995b). Automated cockpits special report, Part 2, 142 (6), 48-57.

Bhatt, S. (1992). Retrofits to BWR safety and non safety systems using digital technology. In *Proceedings of Advanced Digital Computers, Controls, and Automation Technologies for Power Plants* (EPRI TR-100804). Palo Alto, CA: Electric Power Research Institute.

Billings, C. (1991). *Human-centered aircraft automation: A concept and guidelines*. NASA Tech. Memo No. 103885.

Buxton, W. (1986). There's more to interaction than meets the eye: Some issues in manual input. In D.A. Norman and S.W. Draper (Eds.), *User-centered system design*. Hillsdale, NJ: Erlbaum.

Callan, J., Kelly, R., Quinn, M., Gwynne, J., Moore, R., Muckler, F., Kasumovic, J., Saunders, W., Lepage, R., Chin, E., Schoenfeld, I., and Serig, D. (1995). *Human factors evaluation of remote afterloading brachytherapy* (NUREG/CR-6125). Washington, DC: U.S. Nuclear Regulatory Commission.

Carruth, R. and Sotos, W. (1996). Design concepts for the reactor protection and control process instrumentation digital upgrade project at the Donald C. Cook nuclear plant units 1 and 2. *IEEE Transactions on Nuclear Science, 43*, 1899-1902.

Chapanis, A. and Kinkade, R. (1972). Design of controls. In H. P. Van Cott and R. G. Kinkade (Eds.), *Human engineering guide to equipment design*. Washington, DC: American Institutes for Research.

Degani, A., Palmer, E., and Bauersfeld, K. (1992). 'Soft' controls for hard displays: Still a challenge. In *Proceedings of the Human Factors and Ergonomics Society 36th Annual Meeting.* Santa Monica, CA: Human Factors Society.

EPRI (1993a). Advanced light water reactor utility requirements document, Volume II, ALWR evolutionary plant, Chapter 10, *Man-machine interface systems* (revisions 5 & 6). Palo Alto, CA: Electric Power Research Institute.

EPRI (1993b). *Guideline on licensing digital upgrades* (EPRI TR-102348). Palo Alto, CA: Electric Power Research Institute.

FAA (1996). *The interfaces between flightcrews and modern flight deck systems.* Washington, DC: Federal Aviation Administration.

Galletti, G. (1996). Human factors issues in digital system design and implementation. In *Proceedings of the 1996 American Nuclear Society International Topical Meeting on Nuclear Plant Instrumentation, Control, and Human-Machine Interface Technologies*. La Grange Park, IL: American Nuclear Society.

Gobel, M., Luczak, H., Springer, J., Hedicke, V., and Rotting, M. (1995). Tactile feedback applied to computer mice. *International Journal of Human-Computer Interaction*, *7*, 1-24.

8 REFERENCES

Greenstein, J. and Arnaunt, L. (1988). Input devices. In M. Helander (Ed.), *Handbook of human-computer interaction*. New York: Elsevier.

Guerlain, S. and Bullemer, P. (1996). User-initiated notification: a concept for aiding the monitoring activities of process control operators. In *Proceedings of the Human Factors and Ergonomics Society 40th Annual Meeting.* Santa Monica, CA: Human Factors and Ergonomics Society.

Hoecker, D. and Roth, E. (1996). Operators' use of alternative soft control prototypes in a simulated control room task. In *Proceedings of the 1996 American Nuclear Society International Topical Meeting on Nuclear Plant Instrumentation, Control, and Human-Machine Interface Technologies*. La Grange, IL: American Nuclear Society.

Hutchins, E., Hollan, J., and Norman, D. (1986). Direct manipulation interfaces. In D.A. Norman and S.W. Draper (Eds.), *User-centered system design*. Hillsdale, NJ: Erlbaum.

IEC (1993). *Nuclear power plants - control rooms - operator controls* (IEC 1227). Geneva, Switzerland: International Electrotechnical Commission.

Kletz, T. (1993). Computer control - living with human error. *Reliability Engineering and System Safety*, *39*, 257-261.

Kletz, T., Chung, P., Broomfield, E., and Shen-Orr, C. (1995). *Computer control and human error*. Houston, TX: Gulf Publishing Co.

Lee, E.J. (1994). *Computer-based digital system failures* (Tech. review report AEOD/T94-03). Washington, DC: U.S. Nuclear Regulatory Commission.

Lewis, C. and Norman, D. (1986). Designing for error. In D. Norman and S. Draper (Eds.), *User-centered system design*. Hillsdale, NJ: Erlbaum.

Lorenzo, D. (1990). *A manager's guide to reducing human errors: Improving human performance in the chemical industry*. Washington, DC: Chemical Manufacturers Association.

Meter, L. and Olsen, G. (1996). Millstone nuclear unit 3 control system digital upgrade. In *Proceedings of the 1996 American Nuclear Society International Topical Meeting on Nuclear Plant Instrumentation, Control, and Human-machine Interface Technologies*. La Grange, IL: American Nuclear Society.

Moray, N. (1992). Flexible interfaces can promote operator error. In H. Kragt (Ed.), *Enhancing industrial performance: Experiences of integrating the human factor*. Washington, DC: Taylor and Francis.

Nagel, D. (1988). Human error in aviation operations. In E. Wiener and D. Nagel (Eds.), *Human factors in aviation*. New York: Academic Press.

NASA (1992). *Space station freedom program (SSFP) flight human-computer interface standards*. Washington, DC: National Aeronautics and Space Administration.

National Academy of Sciences (1995). *Digital instrumentation and control systems in nuclear power plants: Safety and reliability*. Washington DC: National Academy Press.

Norman, D. (1988). *The psychology of everyday things*. New York: Basic Books.

Norman, D. (1983). Design rules based on analyses of human error. *Communications of the ACM, 26,* 254-258.

Norman, D. (1981). Categorization of action slips. *Psychological Review, 88,* 1-15.

NRC (1996). *NRC information notice 96-56: Problems associated with testing, tuning, or resetting of digital control systems while at power*. Washington, DC: U.S. Nuclear Regulatory Commission.

NRC (1995). *Inspection report no. 95-12*. Washington, DC: U. S. Nuclear Regulatory Commission.

NRC (1993a). *NRC augmented inspection team (AIT) report nos. 50-272/92-81 and 50-311/92-81*. Washington, DC: U.S. Nuclear Regulatory Commission.

NRC (1993b). *NRC information notice 93-47: Unrecognized loss of control room annunciators*. Washington, DC: U.S. Nuclear Regulatory Commission.

O'Hara, J., Brown, W., Stubler, W., Wachtel, J., and Persensky, J. (1996). *Human-system interface design review guideline* (NUREG-0700, Rev. 1). Washington, DC: U.S. Nuclear Regulatory Commission.

O'Hara, J., Stubler, W., and Higgins, J. (1996). *Hybrid human-system interfaces: Human factors considerations* (Draft BNL technical report J6012-T1-4/96). Upton, New York: Brookhaven National Laboratory.

O'Hara, J., Stubler, W., Higgins, J., and Brown, W. (1996). *Integrated system validation: Methodology and review criteria* (NUREG/CR-6393). Washington, DC: U.S. Nuclear Regulatory Commission.

O'Hara, J., Brown, W., and Nasta, K. (1996). *Development of the human-system interface design review guideline, NUREG-0700, Revision 1* (BNL technical report L-1317-2-12/96). Upton, NY: Brookhaven National Laboratory.

Orendi, R. (1996). Control room I&C upgrades, innovations, and HMI considerations for the Temelin nuclear plant. In *Proceedings of the 1996 American Nuclear Society International Topical Meeting on Nuclear Plant Instrumentation, Control, and Human-machine Interface Technologies*. La Grange, IL: American Nuclear Society.

Ranson, D. and Woods, D. (1994). *Controlling what's important with soft controls: Problems, opportunities, and benefits* (CSEL report no. 1994-01, Rev 1). The Ohio State University: Cognitive Systems Engineering Laboratory.

Reason, J. (1990). *Human error*. New York: Press Syndicate of the University of Cambridge.

Rizzo, A., Ferrente, D., and Bagnara, S. (1995). Handling human error. In J. Hoc, P. Cacciabue, and E. Hollnagel (Eds.), *Expertise and technology: Cognition and human-computer cooperation*. Hillsdale, NJ: Erlbaum.

Sarter, N. and Woods, D. (1992). Mode error in supervisory control of automated systems. In *Proceedings of the Human Factors Society 36th Annual Meeting*. Santa Monica, CA: Human Factors Society.

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

Sears, A., Plaisant, C., and Shneiderman, B. (1992). A new era for touch-screen applications: High precision, dragging icons, and refined feedback. In R. Hartson and D. Hix (Eds.), *Advances in human-computer interaction,* Vol 3. New York: Wiley.

Sellen, A., Kurtenbach, G., and Buxton, W. (1990). The role of visual and kinesthetic feedback in the prevention of mode errors. In D. Diaper et al. (Eds.), *Human-Computer Interaction - INTERACT '90*. New York: Elsevier.

Senders, J. and Moray, N. (1991). *Human error: Cause, prediction, and reduction*. Hillsdale, NJ: Erlbaum.

Shaw, J. (1993). Distributed control systems: cause or cure of operator errors. *Reliability Engineering & System Safety*, *39*, 263-271.

Shaw, J. (1988). Reducing operator error in distributed control systems. *Intech*, (March), 45-50.

Shneiderman, B. (1982). The future of interactive systems and the emergence of direct manipulation. *Behavior and Information Technology, 1,* 237-256.

Soken, N., Bullemer, P., Ramanathan, P., and Reinhart, W. (1994). Human-computer interaction requirements for abnormal situation management in industrial processes. In *Petroleum Division Symposium on Computers and Engineering*. Houston, TX: American Society for Mechanical Engineers.

Stubler, W., Higgins, J., and O'Hara, J. (1996). *Evaluation of the potential safety significance of hybrid human-system interface topics* (BNL Technical Report J6012-T2-6/96). Upton, NY: Brookhaven National Laboratory.

Stubler, W. and O'Hara, J. (1996a). *Proposed guidance development for hybrid human-system interface issues* (BNL technical report J6012-T3-10/96). Upton, NY: Brookhaven National Laboratory.

Stubler, W. and O'Hara, J. (1996b). *Group-view displays: Functional characteristics and review criteria* (BNL technical report E2090-T4-12/96). Upton, NY: Brookhaven National Laboratory.

Teitelmann, W. and Masinter, L. (1981). The Interlisp programming environment. *Computer*, *14 (4)*, 25-33.

Wagner, D., Birt, J., Snyder, M., and Duncanson, J. (1996). *Human factors design guide (HFDG): For acquisition of commercial off-the-shelf subsystem, non-developmental items, and developmental systems* (DOT/FAA/CT-96/1). Springfield, VA: National Technical Information Service.

Wickens, C. (1986). The effects of control dynamics on performance. In K. Boff, L. Kaufman, and J. Thomas (Eds.), *Handbook of perception and human performance*. New York: Wiley.

Woods, D. (1984). Visual momentum: a concept to improve the cognitive coupling of a person and computer. *International Journal of Man-Machine Studies*, *21*, 229-244.

Woods, D., Johannesen, L., Cook, R., and Sarter, N. (1994). *Behind human error: Cognitive systems, computers, and hindsight* (CSERIAC SOAR 94-01), Wright Patterson Air Force Base, Ohio: Crew Systems Ergonomics Information Analysis Center.

PART 2:

Review Guidance for Soft Controls

9 SOFT CONTROL HFE DESIGN REVIEW GUIDELINES

The guidelines presented in this section follow the characterization of soft control systems discussed in Section 4. The guidelines reflect the findings from our literature review on the effects of soft controls on personnel performance, specifically the human performance considerations identified in Section 5. As described in the HSI design review procedure in Part 1 of NUREG-0700, Rev. 1, the first step is to select the subset of guidelines relevant to the unique aspects of the particular design. It is recognized that a wide range of soft control designs exists and that some may not include all of the characteristics and functions addressed in these guidelines. It is anticipated that for individual reviews the reviewer will use the soft control characterization to determine which features should be evaluated.

As indicated in Section 6, guidelines were developed from the findings and source materials reviewed in Section 5. They were constructed in the standard format adopted in NUREG-0700, Rev. 1, and are organized into the following sections:

- General Characteristics
- Display Devices
- Input Devices
- Display Design
- Interaction Methods.

These new guidelines will be integrated with the design review guidance in NUREG-0700, Rev. 1.

9.1 General Characteristics

9.1-1 Avoiding Violations of Human Factors Principles Produced by Reconfigurable Features
Reconfigurable features that provide flexibility in the presentation and use of soft controls should be designed to avoid violations of human factors principles.
ADDITIONAL INFORMATION: Reconfigurable features include characteristics that operators can manipulate to deal with changing plant conditions, operator tasks, or personnel preferences. For example, operators may be able to alter the position of displays and controls, or select particular variables and coding to appear in operator-configurable displays. The HSI should be designed to minimize, and, preferably, eliminate configurations that may violate human factors principles and may have serious consequences to plant safety (e.g., result in improper control actions). Some such human factors principles include compatibility with stimulus-response stereotypes, control-display compatibility, and consistency of information coding. Where all such configurations cannot be eliminated, the HSI's features should reduce the severity of these violations. For example, an automated system may contain rules for evaluating the degree of violation and either manage the presentation of configurations to minimize violations, or suggest alternative arrangements when the operator requests a less than ideal one.
Discussion: Guidance on violations of human factors principles is based on Moray's recommendations (1992), which were derived from evaluations of a reconfigurable control and display system. This guidance is consistent with the High-Level Human-System Interface Design Review Principles of Controls/Displays Compatibility in Appendix A.2, and Flexibility in Appendix A.4 of NUREG-0700 (NRC, 1996a).

9.1-2 Operator Feedback for Deviations from Human Factors Principles

If the use of a reconfigurable feature may result in soft control configurations that deviate from human factors principles and may increase the likelihood of errors, then the HSI should provide feedback alerting the operator to this condition and information for avoiding errors.

ADDITIONAL INFORMATION: It may not always be possible or practical to identify and eliminate all configurations that may cause deviations from human factors principles. When the resulting error may have serious consequences to plant safety (e.g., an improper control action), the system should provide feedback indicating the presence of the deviation, and, if possible, direct the operator toward the appropriate response. This feedback should persist for the duration of the deviation.

Discussion: This guideline is based on Moray's recommendations (1992), which were derived from evaluations of a reconfigurable control and display system.

9.1-3 Coordinating Soft Control Use Among Operators

If a soft control can be accessed from more than one location in the HSI, protective measures should ensure its coordinated use among multiple operators.

ADDITIONAL INFORMATION: The HSI should be designed to allow operators to maintain awareness of each other's use of the soft control so their actions do not interfere. For example, two operators should not be able to operate the same soft control simultaneously from different places without being aware of each other's actions. Coordination problems may be minimized by assigning the control capability for a soft control to a particular individual or workstation (e.g., while the settings of a soft control can be viewed from multiple display devices, it can only be operated from one device). Alternatively, coordination may be supported by features that restrict access to soft controls one user at a time, and group-view displays that allow operators to observe each other's actions.

Discussion: Assigning the control capability for a soft control to a particular individual and workstation is a common practice in process control industries. Employing group-view displays to give a shared view of the control and to coordinate its use is consistent with the High-Level Human-System Interface Design Review Principles of Situation Awareness in Appendix A.2 of NUREG-0700 (NRC, 1996a). Stubler and O'Hara (1996b) describe additional technical basis considerations for group-view displays.

9.1-4 Operation with Protective Clothing

Soft controls should be designed to accommodate any protective clothing that operators may be required to wear.

ADDITIONAL INFORMATION: In some plant locations, environmental conditions necessitate wearing protective clothing that can limit the ability of personnel to manipulate soft controls. For example, gloves may reduce manual dexterity and tactile sensitivity, degrading the ability of personnel to operate soft controls quickly and accurately. As another example, eye protection, such as goggles, may become foggy or distort vision and, thus, interfere personnel's ability to view computer-based display devices.

Discussion: This guideline is an application of the High-Level Human-System Interface Design Review Principles of Physiological Compatibility and Personnel Safety in Appendix A.1 of NUREG-0700 (NRC, 1996a).

9.2 Display Devices

9.2-1 Adequate Display Space

Adequate display space should be provided so that short-term monitoring and control tasks do not interfere with longer-term tasks.
ADDITIONAL INFORMATION: Making control actions available via a general-purpose display device may require other plant information to be removed from the operator's view. Sufficient general-purpose display devices should be provided so that short-term control actions can be undertaken without interfering with long-term ones (e.g., they can be performed on separate devices). Alternatively, control actions can be supported by dedicated special devices.
Discussion: A problem operators reported during site visits was that the limited number of general-purpose display devices resulted in disruption of on-going tasks and led to loss-of-activation errors (Norman, 1983, 1986), in which suspended activities were forgotten because the pertinent displays were no longer visible.

9.3 Input Devices

9.3-1 Activation Force

Where practical, activation force may be used as a form of feedback for preventing input errors.
ADDITIONAL INFORMATION: High activation forces within the acceptable range for controls can reduce the likelihood of accidental actuation from stray motions of the operators. High activation forces can also draw their attention to the action and, thereby reduce the likelihood of some types of errors. For example, critical control actions may require using input devices that require higher activation forces than other controls.
Discussion: Norman (1983) states that one way of reducing the likelihood of execution errors is to make the actions required for executing the task difficult.

9.3-2 Lift-Off Logic for Pointing Interfaces

Pointing interfaces should activate on lift-off, such as when the finger leaves the target area of a touch screen or when the button is released on a mouse.
ADDITIONAL INFORMATION: Pointing interfaces are operated via cursors or touch screens. With the lift-off touch logic, the cursor or finger must enter the target and then be removed without touching the area surrounding the target. By contrast, first-touch logic selects target as soon as the cursor or finger makes contact, and is more prone to activation errors.
Discussion: Studies and operating experience showed that first-touch logic is prone to problems of accidental activation (Sears et al., 1992; Hoecker and Roth, 1996). Because targets are activated immediately upon contact, the user does not have time to make corrections if the wrong target is contacted. The lift-off touch logic is more forgiving of input errors than first-touch logic. For example, if contact is made with the wrong target, the lift-off touch logic allows the operators to avoid actuation by moving out of the target area before release. The lift-off logic also allows additional forms of feedback to be added to the targets. For example, when the target is contacted, it can notify the user by changing color or making a sound.

9.3-3 No Activation When Display Is Inoperable
Operators should not be able to activate a soft control if its display is not working.
ADDITIONAL INFORMATION: A reported problem with touch screens is that sometimes their buttons may remain active even though the video image is not visible. Thus, an operator could touch a blank screen and provide a valid input. Such problems may be avoided by requiring multiple actions, such as separate selection and activation steps, for inputs that may have serious consequences (e.g., affect the operation of plant equipment).
Discussion: Personnel reported this as a potential problem at a site visit. This guideline is an application of the High-Level Human-System Interface Design Review Principle of Feedback in Appendix A.2 of NUREG-0700 (NRC, 1996a).

9.4 Display Design

9.4.1 General

9.4.1-1 Representing Relationships Between Control System Components
The display capabilities of soft controls should allow operators to quickly assess the status of individual components of a control system and their relationships with other components.
ADDITIONAL INFORMATION: Due to the limited size of the display devices used with soft controls, not all components of a control system may be visible to the operator at once. However, they should allow the operator to rapidly view relationships between functionally related components. For example, if a controller is part of a hierarchical control system, the operator should be able to see higher-level controllers that provide control inputs and lower-level ones that receive inputs. Rapid assessment of the control system's status should be supported by such features as displays that depict these relationships, and retrieval mechanisms that give rapid access to detailed information on individual control system components.
Discussion: The limited size of display devices can lessen the operator's ability to rapidly assess control system status and determine necessary control actions (Ranson and Woods, 1994). This guideline is an application of the High-Level Human-System Interface Design Review Principle of Feedback in Appendix A.2 of NUREG-0700 (NRC, 1996a).

9.4.1-2 Making Options Distinct
The user interface should be designed so that operators can, at a glance, distinguish options by such characteristics as context, visually distinct formats, and separation.
ADDITIONAL INFORMATION: Slips involve errors in performing well-practiced, unconscious actions. Description errors, a type of slip, involve performing the wrong set of well-practiced actions for the situation. They occur when the information that activates or triggers the action is either ambiguous or undetected. Many control input actions involve the selection of options, such as choosing between alternative commands or selecting a plant component to perform a control action upon it. Description errors that result in selecting a similar but incorrect option may be prevented by organizing options to supply context (such as by functional organization), making options visually distinct, and separating options that operators may confuse. Options may be separated by placing them on different display pages or different display devices.
Discussion: This guideline is derived from Norman's (1983) recommendations for reducing description errors.

9.4.2 Selection Displays

9.4.2-1 Visually Distinct Selection Displays
Displays used for selecting components and variables should be visually distinct to support choice of the correct display.
ADDITIONAL INFORMATION: A selection display shows a set of components or variables that may be chosen by the operator for a control action. One common format is the mimic, in which components are arranged as a schematic diagram. Excessive reuse of layouts and display elements in mimic displays may cause them to look alike and so may contribute to operators searching the wrong selection display for the component that they wish to manipulate. Selection displays should be laid out and labeled so operators readily recognize and distinguish them.
Discussion: Industry experience suggests that the similarity of mimic displays, caused by excessive reuse of layout and display elements, has contributed to operators' errors in selecting components and variables (Kletz et al., 1995).

9.4.2-2 Visually Distinct Components
The representation of components and variables within selection displays should be visually distinct to support their correct selection.
ADDITIONAL INFORMATION: Using a standard set of symbols and layout conventions in displays is important in reducing the mental workload associated with finding and interpreting information. However, these factors may also cause components to look alike and may contribute to operators selecting the wrong component. The symbols and graphical icons used to represent different types of components should be designed to be readily recognized and distinguished. In addition, they should be clearly labeled for correct identification.
Discussion: Industry experience indicates that the similarity of symbols and graphical icons within displays has contributed to operator errors in selecting components (Shaw, 1993).

9.4.2-3 Identification of Loops on Multiple-Loop Controllers
The loops of multiple-loop controls should be distinctly marked to prevent the selection or use of the wrong loop.
ADDITIONAL INFORMATION: A multiple-loop controller is a digital controller that can control multiple variables via independent channels, one per control loop. Each channel acts as a separate control device. For example, a single controller may be capable of controlling 10 different variables, each on a separate control loop. Operators access these loops through the user interface of the controller device. However, because there may be few cues to identify the loops, operators may fail to correctly recognize the loop accessed and may control the wrong variable.
Discussion: Experience in industry shows that the lack of cues for identifying individual control loops contributed to errors in which the operator selected and manipulated the wrong loop (Shaw, 1988).

9.4.3 Input Fields

9.4.3-1 Cues for Matching Input Fields to Selection Displays
An operator looking at the input field for providing a control input should be able to determine which plant component or variable is being controlled.

ADDITIONAL INFORMATION: The design of a soft control should provide a salient link between the input field and the corresponding variable or component. Starting at the input field, the operator should be able to quickly trace the component or variable back to its representation in the display that was used to select it. Three methods that might be used are graphic coding, landmarks, and animation. Graphic codes, such as borders, symbols, and colors, may be applied to both the representation of the component in the display from which it was selected and to the input field, making a strong visual association between them. For example, if the selection display has a mimic format, the input field may contain the symbol for the selected component. It also may contain symbols for the components that precede and follow it in the flow path. Animation may be used when an input field is opened and closed. The input field could appear as if it were "popping out" of an option selected from a display, and "go back" into the option when the field is closed.
Discussion: Industry experience indicates that poor coordination between the presentation of the input field and the selection display can contribute to errors in which the wrong plant component or variable is operated (Shaw, 1988, 1993). Suggestions for coordinating them are based on the concept of visual momentum (Woods, 1984).

9.4.3-2 Labeling of Input Fields
The input field should be labeled with sufficient information to uniquely identify its corresponding component.
ADDITIONAL INFORMATION: Labeling should include a unique identification code for the component, matching its representation in the selection display. It may also describe the component (e.g., valve, pump, breaker) and identify those components that immediately precede and follow it in the system.
Discussion: Inadequately labeled input fields may result in description errors (Lewis and Norman, 1986; Norman, 1983) in which the control operation is performed on the wrong component.

9.4.3-3 Coordination of Soft Controls with Process Displays
Displays should be readily accessible from the input field so the operator can readily verify that the control actions have had the intended effect on plant systems and processes.
ADDITIONAL INFORMATION: Inadequate coordination of input fields with plant process displays can make it difficult for operators to verify that control actions have had the desired effects on plant systems and processes.
Discussion: Prompt access to plant displays can support operators in verifying that the intended control actions were performed (Ranson and Woods, 1994).

9.4.4 Input Formats

9.4.4-1 Appropriate Use of Discrete-Adjustment Interfaces
Discrete-adjustment interfaces should be used for selecting among a set of individual settings or values.
ADDITIONAL INFORMATION: Discrete-adjustment interfaces are computer-based formats with individual settings that can be accessed by fairly gross movements; their operation is similar to discrete-adjustment controls, such as push buttons. By contrast, continuous-adjustment interfaces are computer-based formats that have continuous ranges usually accessed using some type of slewing motion, requiring a gross movement followed by a fine adjustment; their operation is similar to that of continuous-adjustment controls, such as rotary dials or sliders. Discrete-adjustment interfaces are preferred when the operator must select one option from a limited number of choices, or when precision requirements are such that a

limited number of settings can represent the entire continuum of values. The most common discrete-adjustment interfaces used with soft controls are individual buttons and radio buttons (a group of buttons representing a set of related options). However, other formats also are possible, such as rotary selector dials operated via cursor or gestural interfaces. Some computer interfaces have a continuous-adjustment control, such as a slider or scroll bar, for looking at a group of individual options. Because choosing a specific setting with a continuous-adjustment control can be awkward, there should also be a discrete-adjustment control, such as a set of arrow buttons.
Discussion: This guideline extends the guidance on discrete adjustment originally developed for physical control devices (Chapanis and Kinkade, 1972).

9.4.4-2 Labeling Selection Options in Discrete-Adjustment Interfaces
The selection options in discrete input formats should be clearly labeled.
Discussion: This guideline is an application of the High-Level Human-System Interface Design Review Principle of Task Compatibility in Appendix A.2 of NUREG-0700 (NRC, 1996a).

9.4.4-3 Feedback for Discrete-Adjustment Interface with Multiple Settings
Discrete-adjustment interfaces should indicate which setting was selected.
Discussion: This guideline is an application of the High-Level Human-System Interface Design Review Principle of Feedback in Appendix A.2 of NUREG-0700 (NRC, 1996a).

9.4.4-4 Feedback for Discrete-Adjustment Interface with Continuous Operation
If a discrete-adjustment interface has continuous operation, it should provide continuous feedback on the current state.
ADDITIONAL INFORMATION: A continuous-operation control continues to produce an effect until the user provides the next input, or until a predefined action sequence is stopped by a termination criterion. An example is a button that changes to the activated state when pressed and remains in that state until it is pressed again. An example of continuous feedback in a soft control is a checkbox format in which an “X” appears in the box to indicate that an option has been selected, and disappears only after the option is de-selected.
Discussion: This guideline is an application of the High-Level Human-System Interface Design Review Principle of Feedback in Appendix A.2 of NUREG-0700 (NRC, 1996a).

9.4.4-5 Appropriate Use of Continuous-Adjustment Interfaces
Continuous-adjustment interfaces should be used when precise adjustments along a continuum are needed or when many discrete settings are present.
ADDITIONAL INFORMATION: Continuous-adjustment interfaces, such as soft sliders, provide continuous adjustment and are, therefore, suited to selecting a setting from a continuum. Because these interfaces often require a gross slewing movement followed by fine adjustment, setting them correctly may require more time and attention than discrete input formats. Therefore, they should not be used in place of a discrete-adjustment interface for selecting from a small set of options. Continuous-adjustment interfaces are recommended when there are more than 24 discrete settings.
Discussion: This guideline extends guidance on discrete adjustment originally developed for physical control devices (Chapanis and Kinkade, 1972).

9.4.4-6 Appropriate Use of Soft Sliders

A soft slider should be considered as an input device when the range of possible values and the ratio of a value to that range need to be displayed.
ADDITIONAL INFORMATION: A soft slider (also called a slider bar or a scroll bar) is an input format used to directly manipulate a variable over a set range of values. Soft sliders are typically maneuvered via pointing interfaces, such as a touch screen or mouse. They may require careful hand-eye coordination to ensure that the pointing device does not leave the linear path of the slider nor overshoot or undershoot the intended target. If the operator's tasks do not permit careful hand-eye coordination, then other interfaces, such as arrow keys, should be used. The slider sometimes is combined with arrow buttons.
Discussion: This guideline was derived from NASA (1992) and Hoecker and Roth (1996).

9.4.4-7 Indicating the Range of Values on Soft Sliders

The range of values should be indicated on horizontal sliders with the low value on the left and the high value on the right, and on vertical sliders with the low value on the bottom and the high value on the top.
Discussion: This guideline was derived from NASA (1992).

9.4.4-8 Displaying the Digital Value on Soft Sliders

The numerical value to which a soft slider is set should be presented in digits on the soft slider.
Discussion: This guideline was derived from NASA (1992).

9.4.4-9 Dimensions of Soft Sliders

The physical dimensions of the soft slider should allow the operator to read the current and target positions and position the slider with the required precision, accuracy, and response time.
ADDITIONAL INFORMATION: The length of the slider is determined, in part, by the range of values depicted, the increments between individual values, the degree of precision required for reading the slider's position, and the user's expected viewing distance. The accuracy with which the slider may be positioned may be affected by characteristics of the input device (e.g., mouse devices may allow more accurate positioning than a touch interface due to the size and irregular shape of the finger). A very short slider may be difficult to read or position precisely. A very long slider may produce slow response times due to the long distance that must be traveled and the need to keep the pointing device on its linear path.
Discussion: This guideline was derived from NASA (1992).

9.4.4-10 Depicting Critical Ranges on Soft Sliders

When part of the range of values depicted by a soft slider represents critical information, such as alarm limits, those values should be coded to facilitate recognition.
ADDITIONAL INFORMATION: Graphical codes may be applied to distinguish the normal operating range, alarm limits, and other abnormal operating ranges.
Discussion: This guideline was derived from NASA (1992).

9.4.4-11 Appropriate Use of Arrow Buttons

A set of arrow buttons should be considered as the input device when it is desirable to incrementally increase or decrease a variable from its previous value.
ADDITIONAL INFORMATION: Arrow buttons change values sequentially as each increase or decrease button is pressed. In addition, values may change continuously if a button is held down. These inputs provide feedback about the magnitude of the change (i.e., the magnitude increases with the number of presses or the time that a button is held down). Such feedback may reduce the likelihood of producing

large errors or increase the likelihood of detecting them. Some soft controls have two sets of arrow buttons, one for small and one for large incremental changes. Arrow buttons are sometimes combined with a slider in a soft control.
Discussion: This guideline was derived from Apple Computer, Inc. (1996). The specific examples described above were identified through a review of industry experience. Using arrow buttons for feedback on the magnitude of the change is an example of the High-Level Human-System Interface Design Review Principle of Feedback in Appendix A.2 of NUREG-0700 (NRC, 1996a).

9.4.4-12 Indicating Current Value for Arrow Buttons
Arrow buttons should have a display indicating the current value of the variable being controlled.
ADDITIONAL INFORMATION: The current value should be shown in a format consistent with the type of variable being controlled. Numerical values should be presented as digits, and textual values (e.g., Low, Medium, and High) as words.
Discussion: This guideline was derived from Apple Computer, Inc. (1996).

9.4.4-13 Uniform Changes in Values Via Arrow Buttons
Each press of an arrow button should change the current value uniformly.
Discussion: This guideline was derived from Apple Computer, Inc. (1996).

9.4.4-14 Feedback Regarding Arrow Button Actuation
Arrow buttons should provide salient feedback when they are actuated.
ADDITIONAL INFORMATION: Feedback should be sustained when the button is held down and momentary when the button is momentarily pressed.
Discussion: This guideline was derived from Apple Computer, Inc. (1996).

9.4.4-15 Apparent Operation of Arrow Buttons
Labeling and other coding should be used when the operation of the arrow buttons is not apparent.
ADDITIONAL INFORMATION: For example, when arrow buttons are used to change a date display, it may be unclear whether actuating a button will incrementally change the days (and change the month when the last day is reached), or whether the month and day values are changed separately after being selected by the user. The arrow buttons should be labeled or coded to indicate their effects.
Discussion: This guideline was derived from Apple Computer, Inc. (1996).

9.4.4-16 Reference Values For Continuous Variable Inputs
Reference values should be provided to help the operator judge the appropriateness of values when entering continuous variable inputs.
ADDITIONAL INFORMATION: Reference values commonly used in process control applications include the variable’s range, alarm limits, and the current value. Reference values may be presented as digits or graphs.
Discussion: This guideline is an application of the High-Level Human-System Interface Design Review Principles of Feedback and User Model Compatibility in Appendix A.2 of NUREG-0700 (NRC, 1996a).

9.5 Interaction Methods

9.5.1 General

9.5.1-1 Minimizing Soft Control Modes
The excessive use of modes in soft controls should be avoided.
ADDITIONAL INFORMATION: Modes occur in soft controls when a display or input device is designed for more than one function. For example, a soft control that is used for manipulating multiple variables may have a separate mode for each one (e.g., individual modes for variables A, B, and C). In addition, there may be multiple modes for a single variable, each allowing it to be controlled in a different way (e.g., variable A may have separate modes for manual control, automatic control, and testing). Mode errors occur when the user believes the device is in one mode when it is in another and, as a result, performs an inappropriate input action. The likelihood of mode errors can be lessened by reducing the number of modes; if multiple modes do not exist, then mode errors cannot occur.
Discussion: Norman (1983, 1988) recommends minimizing the number of modes as a way of reducing the likelihood of mode errors.

9.5.1-2 Distinctive Indication of Soft Control Modes
When multiple modes exist, they should be distinctively marked so the operator can determine the current mode at a glance.
ADDITIONAL INFORMATION: Distinct labels may be used to indicate the currently active mode.
Discussion: Norman (1983, 1988) recommends distinctively indicating modes to prevent mode errors. This guideline is also an application of the High-Level Human-System Interface Design Review Principle of Feedback in Appendix A.2 of NUREG-0700 (NRC, 1996a).

9.5.1-3 Coordination of Destructive and Safety-Significant Commands Across Modes
A command that produces a benign action in one mode should not cause a different action with serious negative consequences in another mode.
ADDITIONAL INFORMATION: A command is an instruction provided by an operator requesting a computer system to perform a particular action. Actions that are destructive (e.g., delete file) or have serious safety consequences should have unique commands. For example, the function key “F2” should not have a benign action, such as listing a directory, in one mode but a destructive action, such as deleting a file or operating important plant equipment, in another mode.
Discussion: Norman (1983, 1988) states that mode errors can be avoided by ensuring that commands are not valid in more than one mode.

9.5.1-4 Unique Commands for Destructive and Safety-Significant Commands
Unique commands associated with actions that have important consequences should not be easily confused with other commands used in the same or different modes.
ADDITIONAL INFORMATION: Reserving special commands for special actions can prevent mode errors because, if the command is entered while the device is in the wrong mode, it will not be accepted by the system. A unique or reserved command should not be so similar to other commands that a valid entry may result from incorrectly entering another command. For example, if the command “CNTL X” is reserved for a special action, then similar commands, such as “ALT X” and “Shift X,” should not be valid, even in other modes. The combination of a mode error and the incorrect entry of the command may execute an unintended action.
Discussion: Norman (1983, 1988) states that mode errors can be avoided by ensuring that commands are not valid in more than one mode. However, industry experience showed that equipment can fail from the

combination of a mode error and an incorrectly entered command (Galletti, 1996; NRC, 1993a; NRC, 1993b).

9.5.1-5 Discrimination of Interface Management Actions and Process Control Actions

The design of the user interface should clearly distinguish between interface management actions and process control actions.
ADDITIONAL INFORMATION: Actions required for interface management tasks and plant control tasks should look different. This may be accomplished by providing different interfaces, different coding for interfaces, and, possibly, different input devices.
Discussion: This guideline was derived from EPRI, (1993a).

9.5.1-6 Reducing the Likelihood of Unintended Actuation

For actions that can have significant negative consequences, the user interface should be designed to reduce the likelihood of unintended actuation by requiring deliberate action for their execution.
ADDITIONAL INFORMATION: Deliberate actions should be required for inputs having serious potential consequences. Actions that require physical effort in the form of multiple steps or higher actuation forces may be less likely to occur accidentally as the result of a random motion of the operator. Also, actions that require greater attention, such as multiple steps and checks, may reduce the likelihood that the operator will revert to the type of “automatic” activity that could cause a slip. However, control actions that require multiple steps also should be designed to reduce the likelihood of other errors (i.e., the failure to complete a set of steps in the correct order).
Discussion: Norman (1983) recommends preventing unintended actuation by requiring added mental or physical effort for actions that may have significant negative consequences. This guideline also is an application of the High-Level Human-System Interface Design Review Principle of Error Tolerance and Control in Appendix A.4 of NUREG-0700 (NRC, 1996a).

9.5.1-7 Feedback For Selected Actions Before Execution

The HSI should give the operator feedback indicating the action that was selected and allow the action to be canceled before it is executed.
ADDITIONAL INFORMATION: The goal of this recommendation is to avoid unintended manipulation of plant equipment or unintended interface management actions. Feedback about the selected option is important because a broad range of actions may be accessed through a soft control device, including manipulation of various plant components and of the user interface. The close proximity and similarity of input options within the display area may result in operators selecting the wrong ones. Operators should be able to cancel or modify an action if they determine that its execution would be undesirable.
Discussion: This guideline was derived from EPRI (1993b). This guideline is also an application of the High-Level Human-System Interface Design Review Principle of Error Tolerance and Control in Appendix A.4 of NUREG-0700 (NRC, 1996a).

9.5.1-8 Use of Error-Mitigation Approaches

Error-mitigation approaches should not be the sole means for achieving error tolerance, but should be used in conjunction with other means for error prevention and system-assisted error detection.
ADDITIONAL INFORMATION: Error-mitigation mechanisms limit the effects of incorrect inputs after they have been entered into the control system. Two strategies include reducing the rate of the system’s response and deferring it. Both are intended to provide time for detecting and correcting input errors and

for reversing them. Error mitigation should not be considered a substitute for error prevention and detection.
Discussion: This guideline was derived from a broad body of literature, including Norman (1988, 1983, 1981) and Lewis and Norman (1986), which describe many approaches for reducing error. This guideline is also an application of the High-Level Human-System Interface Design Review Principle of Error Tolerance and Control in Appendix A.4 of NUREG-0700 (NRC, 1996a).

9.5.1-9 Undo Features

If undo features are provided they should be consistently available.
ADDITIONAL INFORMATION: Undo features minimize the effects of operators' errors by allowing them to undo or reverse previous actions. Users tend to rely upon undo features and incorporate them into their work. Failures of undo features may have worse consequences than if they were not provided in the first place. For example, operators may be more willing to delete files if they think they can recover them.
Discussion: This guideline is based on a recommendation by Norman (1983, 1988) on undo features.

9.5.2 Sequential Actions

9.5.2-1 Indicating the Status of Sequential Actions

Computer-based HSIs should support operators in rapidly assessing the status of sequential actions in progress.
ADDITIONAL INFORMATION: An action sequence is a set of operations that must be performed in a specific order. Errors involving misordering the components of an action sequence include skipped, reversed, and repeated steps. Soft controls may be more prone to this type of slip than conventional controls because they introduce additional operations for accessing controls and displays and providing inputs that also often have sequential constraints on their execution. In addition, many control operations must be performed in particular sequences. For example, when configuring a fluid system, it may be necessary to establish the flow path, control mode, and setpoint of a flow controller in a specific sequence of operations (e.g., A, B, C, D, and E). One form of error occurs when an operator skips a step thinking that it was completed. For example, an operator may perform operations A, B, and C and after some delay or interruption, may perform operation E thinking that D already was finished. The repetitiveness of the task is a factor in this type of error. If an operator has performed a set of operations repeatedly on several identical controllers, the memory of performing a particular operation on the other controllers may increase the likelihood of the operator incorrectly concluding that the operation was completed on the present controller. Thus, the sequentiality of soft controls can interact with repetitive, sequential tasks to increase the probability of errors involving misordering the components of the action sequence. The display design of computer-based HSIs should support operators in identifying tasks that are in progress; ideally, they should be designed so that the status of related operations (e.g., A, B, C, D, and E) can be checked at a glance from a single display.
Discussion: Shaw (1988, 1993) reported operator errors associated with repetitive control actions using soft controls. Norman (1981, 1983) gave a general discussion of errors involving sequential actions. The ability to rapidly assess the status of sequential actions that are in progress is consistent with the High-Level Human-System Interface Design Review Principle of Situation Awareness in Appendix A.2 of NUREG-0700 (NRC, 1996a).

9.5.2-2 Drawing Attention to Points Where Similar Sequences Diverge

The design of the HSI should draw the operator's attention to points where operational sequences that have multiple steps in common begin to diverge from each other.
ADDITIONAL INFORMATION: A capture error occurs when an infrequently performed action requires a sequence of operations that overlaps with the sequence required for a frequently performed action. In attempting the infrequent action, the frequent one is performed instead. For example, an operator intends to perform task 1, consisting of operations A, B, C, and D, but instead executes the more frequently performed task 2, (composed of operations A, B, C, and E). Capture errors often occur at the point of divergence of the frequently and infrequently performed sequences. HSI design efforts may be directed at that critical point to bring it to the operator's attention. For example, if the control system knows the operator's intention (e.g., by requiring an indication of the overall intention), it could highlight the proper path at the choice point, or initiate a warning if the wrong one is taken. Another approach is to draw the operator's attention to important choice points (i.e., points where the sequence of operations differs from the sequences of similar tasks) by coding, labeling, and caution messages. Yet another way is to incorporate features drawing attention to the operational significance of alternative paths and supporting an understanding of which path has been taken.
Discussion: This guideline is derived from Lewis and Norman (1986).

9.5.2-3 Operator Interruption of Transaction Sequences

The HSI should allow the operator to interrupt or terminate a current transaction sequence.
ADDITIONAL INFORMATION: A transaction sequence is a series of steps the operator undertakes to accomplish a larger task. For example, the task of changing a control setpoint may involve multiple steps for selecting the variable and entering the new value. If different types of interruptions or terminations exist, then each should have a separate control option and a distinct name. Table 9.1 lists interruption and termination types.
Discussion: This guideline is derived from Wagner et al. (1996).

Table 9.1 Different Types of Interruptions or Terminations for Transaction Sequences

Backup or Go Back	A nondestructive option that returns the display to the last previous transaction.
Cancel	An option that erases changes just made by the user and restores the current display to its previous state.
End, Exit, or Stop	An option that concludes a repetitive transaction sequence.
Pause and Continue	Options that interrupt and later resume a transaction sequence without any changes to either the data entries or the logic of the interrupted transaction.
Restart or Revert	An option that cancels entries made in a transaction sequence and returns the user to the beginning. If a restart will result in the loss of data or changes, a confirming action is required of the user.

Review	A nondestructive option that returns to the first display in a transaction sequence, permitting the user to review a sequence of entries and make necessary changes.
Suspend	An option that permits the user to preserve the current state of a transaction while leaving the system and permits resumption of the transaction later.

9.5.2-3 Interrupted Sequence Prompt
The HSI should support the operator in maintaining awareness or recalling tasks that were interrupted or suspended by giving a reminder.
ADDITIONAL INFORMATION: A loss-of-activation error occurs when an intended action is not carried out due to a failure of memory (i.e., the intention has partially or completely decayed from memory). One way of preventing loss of activation is to have an on-screen message reminding the operator of the suspended task. If necessary, the system should prompt the operator with information on how to resume it. A second approach is to provide more display screens or implement a window-based display system to keep tasks that are in progress visible, as they would be in spatially dedicated conventional CRs.
Discussion: This guideline is derived from EPRI (1993a), Norman (1981, 1983), and Wagner et al. (1996).

9.5.2-4 Resumption of Interrupted Sequences
A minimum number of actions should be required for the operator to resume a control-action sequence that was temporarily suspended.
ADDITIONAL INFORMATION: When an operator has interrupted a sequence of operations, a minimum number of actions should be required to resume it. The operator should not be required to restart the sequence from the beginning. One way of supporting the operator in finding a display containing a suspended task is to have a “previous display” feature that accesses a sequence of previous displays. A second approach is an interaction history feature that lists previously accessed displays and provides access to them. A third method is to include a “bookmark” feature allowing operators to designate displays containing tasks that are in progress. Thereafter, few actions or none should be required to resume the task.
Discussion: This guideline is derived from EPRI (1993a) and Norman (1981 and 1983).

9.5.3 Verification and Confirmation Steps

9.5.3-1 Separate Action For Verification Steps
Verification steps should be separate from input actions.
ADDITIONAL INFORMATION: Verification steps are usually steps added to the input action. For example, the user selects an option and then presses the Enter key to verify it. Verification steps reduce the likelihood of input errors by increasing the effort (i.e., the number of steps) and drawing users’ attention to the input operation. However, they can lose their effectiveness if operators can perform them unconsciously as part of the input action.
Discussion: Industry experience showed that when attention is not focused on verification steps, they can lose their effectiveness for preventing errors (Kletz et al., 1995).

9.5.3-2 Confirmation of Goals

When feasible, confirmation steps should draw operator attention to the goal of the action, not just to the action.
ADDITIONAL INFORMATION: Confirmation steps require the user to respond to a warning or advisory message. For example, the user may respond to the question, “Are you sure you want to do this?” by pressing “Yes” or “No.” Like verification steps, confirmation steps attempt to reduce input errors by increasing the effort (i.e., the number of steps) and drawing users’ attention to the input operation. A problem with confirmation steps is that they are often ill-timed, occurring just after the operator initiated the action and is still fully content with the choice. If the user requests an action but specifies the wrong object to be acted upon (e.g., operator requests a file deletion but specifies the wrong file), the system’s request for confirmation is not likely to help the operator detect the error. At this point, the operator is apt to focus on confirming the *action* (e.g., deletion) rather than the *object* (e.g., which file). The potential benefits of confirmation steps should be weighed by comparing their effects on the operator’s response time (e.g., potential delays) to the potential consequences associated with the errors that are being guarded against.
Discussion: This guideline is derived from Norman (1988).

9.5.4 Interlocks, Lockouts, and Lockins

9.5.4-1 Use of Interlocks, Lockouts, and Lockins

Interlocks, lockouts, and lockins should be provided to restrict personnel actions that may affect plant safety.
ADDITIONAL INFORMATION: An interlock is a feature that requires operator actions to proceed in a specific sequence. A lockout prevents personnel from providing input that may generate a negative effect. Statically defined lockouts may restrict operator inputs to a specific, predefined range or set of values. Context-sensitive lockouts may restrict input values based on the current situation. A lockin keeps an ongoing operation active by preventing personnel from terminating it prematurely. Personnel actions that may affect plant safety include control actions and manipulating stored data important to safe plant operation.
Discussion: This guideline is derived from Norman (1988).

9.5.4-2 Operator Override of Interlocks, Lockouts, and Lockins

The design of interlocks, lockouts, and lockins should not limit the operators’ authority unless there is a clear safety reason.
ADDITIONAL INFORMATION: A limitation of error-prevention measures (e.g., interlocks, lockouts, and lockins) that cannot be overridden by the operator is their inability to make allowances for situations in which they are overly restrictive and possibly detrimental to safety. Sometimes a normally undesirable tactic may be the only thing an operator can do to solve a problem.
Discussion: This guideline is derived from Norman (1988), Senders and Moray (1991), and Billings (1991).

9.5.4-3 Visibility of Interlocks, Lockouts, and Lockins

Interlocks, lockouts, and lockins should be designed to indicate which actions are being blocked and what conditions activated the block.

ADDITIONAL INFORMATION: A lockout blocks operator inputs that it considers unacceptable or not achievable. When this occurs, the operator should be able to determine why an input was blocked and what inputs are acceptable, especially for context-sensitive validation in which complicated rules may be used for assessing the acceptability of an input value. An interlock should inform the operator of the condition(s) that activated it and the conditions that must be satisfied to release it. Lockin features should show the operator what action is being "locked in" (i.e., the action that is being caused to operate without interruptions) and how it can be canceled.
Discussion: This guideline is derived from EPRI (1993a) and Norman (1988).

9.5.4-4 Automatic Logging of the Activation of Interlocks, Lockouts, and Lockins
The activation of an interlock, lockout, or lockin should be automatically logged.
Discussion: This guideline is derived from EPRI (1993a).

9.5.4-5 No Automatic Actuation of Blocked Actions
An interlock, lockout, or lockin should not initiate an action that was previously blocked merely because the status of the triggering condition has changed.
ADDITIONAL INFORMATION: If operation B was blocked because condition A was not satisfied, the system should not automatically start operation B when condition A is met. Instead, a separate action should be required (e.g., the operator should be required to take a specific action to allow operation B to resume).
Discussion: Interviews with operations personnel from process control facilities indicate that, in some systems, an automatic actuation can occur without operator notification or action after a trigger condition has been satisfied. This guideline is an application of the High-Level Human-System Interface Design Review Principles of Situation Awareness and Feedback in Appendix A.2 of NUREG-0700 (NRC, 1996a).

9.5.5 Error Detection and Correction

9.5.5-1 Warning Message Content
Warning messages should draw operators' attention to the goal of the action, not just to the action.
ADDITIONAL INFORMATION: Actions may be described in many levels of detail. Often error messages are not effective because they are directed toward the wrong level of detail, so that the description of what is wrong may not match the operator's understanding of what was done. An alternative is to allow the operator to interrogate the warning. For example, the initial warning could be given at a very high level, corresponding to the system's understanding of the operator's intent but then could allow the operator to obtain information at lower, more detailed levels, such as describing how the action was performed and why it was inappropriate for the goal.
Discussion: This guideline is derived from Lewis and Norman (1986). It also is an application of the High-Level Human-System Interface Design Review Principle of Feedback in Appendix A.2, NUREG-0700 (NRC, 1996a).

9.5.5-2 Automatic, Self-Correct Features for Interface Management Action

Automatic, self-correcting features should only be used for interface management actions, such as retrieving displays.

ADDITIONAL INFORMATION: Automatic, self-correcting features detect and automatically correct errors that users make when providing inputs; for example, a "Delete" command that is incorrectly entered as "DLE" will be automatically changed to its correct form "DEL" and then executed. These systems can interfere with user's activities if their error-detection facilities are overgeneralized (i.e., they interpret correct entries as being errors), since the system may substitute an incorrect response for the correct one provided by the user, thereby affecting plant operation and safety. Additional mental burdens may be imposed on the user to learn, remember, and anticipate the types of correct inputs that these systems will interpret as errors. Therefore, automated, self-correcting features should not be employed for plant-control actions. Instead, other approaches should be used, such as warnings and confirmation steps.

Discussion: This guideline is an application of the High-Level Human-System Interface Design Review Principle of Error Tolerance and Control in Appendix A.4 of NUREG-0700 (NRC, 1996a).

9.5.5-3 Undo Capabilities for Self-Correct Features

Automatic, self-correcting features should only be used if they include good "Undo" capabilities, so that inappropriate changes made by the system can be reversed by the user.

Discussion: Lewis and Norman (1986) recommend providing automatic, self-correcting features only if there is an "Undo" capability. This guideline is also an application of the High-Level Human-System Interface Design Review Principle of Error Tolerance and Control in Appendix A.4 of NUREG-0700 (NRC, 1996a).

9.5.5-4 Use of Inspection and Transfer Steps

Inspection and transfer steps should be considered if inputs are complex, or if incorrect inputs can seriously affect safety.

ADDITIONAL INFORMATION: Inspection and transfer steps are intermediate steps included in a sequence of operations to create additional opportunities for detecting and correcting faulty inputs. Rather than entering data directly into the control system, the data may be sent to a holding file for review and approval. Thereafter, a command may be entered to transfer the data from the holding file into the active portion of the control system.

Discussion: This guideline is an application of the High-Level Human-System Interface Design Review Principle of Error Tolerance and Control in Appendix A.4 of NUREG-0700 (NRC, 1996a).

9.5.6 Selecting Plant Variables or Components

9.5.6-1 Identification of Plant Variables and Components

The HSI should support the identification of plant variables and components based on recognition rather than relying strictly upon recall.

ADDITIONAL INFORMATION: The HSI should present the options available to operators for selecting plant variables and components. For example, they may be shown via menus or mimic displays to facilitate recognition. Where there are multiple variables, their selection should not be based strictly upon the ability of operators to recall components' identification codes.

Discussion: This guideline is an application of the High-Level Human-System Interface Design Review Principle of Cognitive Workload in Appendix A.3 of NUREG-0700 (NRC, 1996a).

9.5.6-2 Simple Input Actions for Selection
The operator should be able to select a component or variable from a display by using simple input actions.
ADDITIONAL INFORMATION: Multi-step or complex input operations, such as transcribing identification codes, should be avoided. The demands of making a selection should be minimized so as not to compete with cognitive resources needed for assessing plant conditions and planning responses. However, in some cases, such as for controls that are very important to plant safety, more complex actions may be required to reduce the likelihood of accidental actuation.
Discussion: This guideline is an application of the High-Level Human-System Interface Design Review Principles of Cognitive Workload and Response Workload in Appendix A.4 of NUREG-0700 (NRC, 1996a).

9.5.6-3 Minimize Action-Sequence Errors for Selecting Plant Variables
If a sequence of actions is required to select a component or variable, the HSI should be designed to prevent misordered action-sequence errors.
ADDITIONAL INFORMATION: When a soft control is used to manipulate multiple plant components or variables, the operator may need to select one, perform the control action, and then deselect it before controlling the next. Errors involving misordering the components of an action sequence may occur. If the operator fails to deselect the last component or variable (i.e., the one that was previously controlled), the control action may be performed on the wrong one. The HSI may minimize the likelihood of misordered action-sequence errors by minimizing the number of selection steps, reducing sequential constraints on selection steps, and providing feedback for identifying out-of-sequence steps.
Discussion: Industry experience shows that misordered action-sequence errors can affect plant operations when soft controls are used (Shaw, 1993). This guideline is an application of the High-Level Human-System Interface Design Review Principles of Simplicity of Design in Appendix A.1 and Feedback in Appendix A.2 of NUREG-0700 (NRC, 1996a).

9.5.6-4 Minimize the Number of Retrieval Steps for Controls that are Used Together
When a group of controls must be used together, their retrieval should require a minimal number of actions.
ADDITIONAL INFORMATION: Excessive selection steps can prevent prompt access to controls and can cause misordered action-sequence errors. One approach to reducing the number of selection actions is to present, on the same display, controls that are used together.
Discussion: Industry experience showed that when a task requires multiple plant components to be operated together and the components are presented on different display pages, excessive navigation may be required to access the components and monitor plant status (Ranson and Woods, 1994). This guideline is an application of the High-Level Human-System Interface Design Review Principle of Task Compatibility in Appendix A.2 of NUREG-0700 (NRC, 1996a).

9.5.7 Control Inputs

9.5.7-1 Automatic Reset of Multi-Variable Controls
If an input device controls more than one variable, the operator should not have to reset the device to match the value of the new variable before executing a control action.
ADDITIONAL INFORMATION: When switching between variables, the control should automatically display the current value of that variable and position the input device consistent with that value. The

operator should not be required to adjust the input device to match the current value of a new variable. For example, if variable A is currently set at a value of 100 and variable B at 10, when selecting the latter, the operator should not be required to adjust the input device to the 10 position before executing a control action.
Discussion: The nulling problem (Buxton, 1986) occurs when an input device controls multiple variables and the control settings correspond to the physical position or orientation of the device. If the input device does not automatically reset when a variable is selected, then the operator must reset it manually. This operation takes time to learn, time to carry out, and is a source of error.

9.5.7-2 Numerical Input Values

The HSI should provide feedback to support the operator in verifying the correctness of numerical values entered.
ADDITIONAL INFORMATION: At a minimum, the value should be depicted as digital readout. However, additional feedback can further aid operators in detecting input errors. For example, for control setpoints, reference values can convey the implications of the new value for plant operations and, thus, support the operator in identifying a value that is too large or too small. Reference values include the actual value of the process variable, the current setpoint value, the normal operating limits, and the alarm limits. Graphical feedback might include a barchart depicting the input value (i.e., the bar's length corresponds to the magnitude of the entered value). The reference values and the graphical representation may be combined.
Discussion: Industry experience revealed that entering numerical values, such as control setpoints, is prone to errors, especially when done on a keyboard (Kletz et al., 1995; Kletz, 1993; Lorenzo, 1990; Shaw, 1988). The ability of operators to detect errors can be enhanced when the HSI indicates the significance of the input to the operator's goals (Lewis and Norman, 1986).

9.5.8 Handling Stored Data

9.5.8-1 Minimize the Use of Irreversible Actions

The design of the HSI should minimize the use of irreversible actions for handling stored data.
ADDITIONAL INFORMATION: The design of HSI should seek to eliminate irreversible actions in handling stored data. The operator should be able to reverse an action with an "Undo" capability. If an action cannot be designed to be reversible, the user interface should be designed to reduce the likelihood of unintended actuation.
Discussion: This guideline is derived from recommendations by Norman (1988).

9.5.8-2 Deferring Execution of Operations that are Destructive to Stored Information

Whenever practical, irreversible operations that destroy stored information should be deferred and require a separate action for their execution rather than being carried out immediately.
ADDITIONAL INFORMATION: Operations that are destructive to stored information include modification and deletion of files. One way of making actions reversible is to defer their execution, giving the operator an opportunity to reconsider and reverse the action. An example is the command to delete a file. Many computers place the files in a storage location where, depending upon the computer, it may be deleted automatically in the future, or remain indefinitely until the operator issues a separate command. This feature allows the user to easily recover the file. Such reversible delete features may be beneficial in NPPs for recovering trend information or other data important for the safe operation of the plant.

Discussion: Norman (1983) recommends avoiding having irreversible operations. Having an action appear to be carried out when, in fact, it has only been deferred is one way of making the action reversible and allowing operators to recover from errors.

9.5.9 System Response

9.5.9-1 Actuation Feedback

Soft controls should provide feedback about their operating state after activation.
ADDITIONAL INFORMATION: Momentary controls, which operate only during actuation (e.g., while a button is pressed) should provide feedback during operation. Continuous-operation controls, which remain operating after actuation, should provide continuous feedback.
Discussion: This guideline is derived from Apple Computer, Inc. (1996) and Chapanis and Kinkade (1972).

9.5.9-2 Operator Notification of Automatic Mode Changes

Systems that can change mode automatically should provide feedback to make the operator aware of the current mode.
ADDITIONAL INFORMATION: The HSI should inform the operator of the current operating mode, mode-transition points, limits on operator actions, and circumstances in which operators must assume control. This feedback should support the operator in assuming control without unnecessary actions and without unnecessarily disrupting plant systems and processes.
Discussion: This guideline is derived from Sarter and Woods (1995, 1992).

9.5.9-3 Delaying System Response

Where appropriate, systems that are sensitive to incorrect inputs should be designed to limit the rate at which these inputs can affect the process.
ADDITIONAL INFORMATION: Limiting the rate at which a system responds to an operator’s inputs can provide opportunities for the operator to detect and correct erroneous material. Methods for delaying system response include programmed limits in the control software, such as maximum ramp rates, and physical limits in plant equipment, such as orifices and dampers, to limit the rate at which processes can respond to inputs. These methods may be used when the system’s slower response will not degrade plant operation or safety. These methods should be used with other methods that prevent errors and detect them.
Discussion: Industry experience showed that digital control systems can respond faster than older analog systems (Kletz et al., 1995; Lorenzo, 1990). When the operator inputs incorrect data, these systems can respond so quickly that the operator does not have enough time to take corrective actions. Lorenzo (1990) recommends methods for delaying system response, such as programmed limits in the control software, and physical limits in plant equipment.

GLOSSARY

Action sequence: A set of operations that must be performed sequentially to carry out a control action.

Alphanumeric keyboard: A keyboard used for typing letters or numbers into the computer.

Arrow buttons: A pair of buttons used to change a value by increments each time they are pressed. Often, the button that produces an increase is marked with an upward arrow and the button that produces a decrease is marked with a downward arrow.

Automatic mode: A mode in which processing proceeds without human intervention (as contrasted with interactive and manual modes).

Automatic, self-correcting features: Features that detect and automatically correct errors that users make when providing inputs. For example, a "Delete" command that is incorrectly entered as "DLE" may be automatically changed to its correct form, "DEL", and then executed.

Barchart: A graphic figure in which numeric quantities are represented by the linear extent of parallel lines (or bars). The length of the line (or bar) is proportional to the numbers represented. Barcharts are useful for comparing separate entities or showing a variable sampled at intervals.

Buffer: A file or device that temporarily stores data.

Button: A type of hardware control device or a defined control region on the display screen which, when selected, causes an action.

Capture error: An error of execution (slip) that occurs when an *infrequently* performed action requires a sequence of operations, some of which are the same as or similar to those of a *frequently* performed action. In attempting the infrequent action, the more frequent action is performed instead. For example, an operator intends to perform task 1, composed of operations A, B, C, and D, but instead executes the more frequently performed task 2, composed of operations A, B, C, and E.

Cascade control mode: An automatic control mode in which a controller receives its control setpoint from a higher-level one.

Closed window: A window which is not visible and which requires some action by the user to gain perceptual and functional access. For example, a user may select and open an icon that represents a window or, in contrast, might input a command to open a specific window. (See also active and inactive windows.)

Coding: Use of a system of symbols, shapes, colors, or other variable sensory stimuli to represent specific information. Coding may be used (a) for highlighting (i.e., to attract a user's attention to part of a display), (b) as a perceptual indicator of a data group, or (c) to symbolize a state or attribute of an object (e.g., to show a temperature level or a warning).

Command: (1) The act of instructing the computer or system to perform an action. (2) An entry provided by a user, which instructs the computer system to perform an action.

Command language: A type of dialogue in which a user composes entries, possibly with minimal prompting by the computer.

Confirmation step: A step in a transaction sequence that requires the user to respond to a warning or advisory message. For example, the user may respond to the question, “Are you sure you want to do this?” by pressing “Yes” or “No.”

Continuous-adjustment interfaces: Computer-based formats that have continuous ranges usually accessed with some type of slewing motion requiring a gross movement followed by a fine adjustment. Their operation is similar to that of continuous-adjustment controls, such as rotary dials or slider switches.

Control: A mechanism used to regulate or guide the operation of a component, equipment, subsystem, or system.

Cursor: A display graphic used to indicate the position of the user's operation on the display (such as an arrow or flashing bar).

Data validation: Functional capabilities that check data entry items for correct content or format, as defined by software logic.

Default: A value or setting that is used if no alternative is specified. The system assumes the default value unless it is specifically overridden. Defaults represent predetermined, frequently used values for data or control entries intended to reduce entry actions required from the user.

Description error: An error of execution (slip) that involves performing the wrong set of well-practiced actions for the situation. Description errors occur when the information that activates or triggers the action is either ambiguous or undetected.

Direct manipulation: The user manipulates symbols in the display by directly interacting with the symbol using a display structure, such as a pointer, and a cursor-control device, such as a mouse.

Discrete: Consisting of distinct or unconnected elements.

Discrete-adjustment interfaces: Computer-based formats with individual settings that usually can be accessed using fairly gross movements. Their operation is similar to discrete-adjustment controls, such as push buttons.

Display: A specific integrated, organized set of information. A display can include several display formats (such as a system mimic including barcharts, trend graphs, and data fields).

Display device: The hardware used to present the display to users such as video display units and speakers for messages.

Enter: An explicit user action that affects computer processing of user entries. For example, after typing a series of numbers, a user might press an Enter key that will add them to a database, subject to data validation.

Enter key: A key used to indicate completion of data entry for the current field or record.

Entry: (1) The act of inputting information to the system. (2) Something which has been entered, such as data or a command.

Error-tolerant features: Characteristics of the HSI that minimize the effects of operators' errors.

Feedback: System or component response (e.g., visual or aural) which indicates the extent to which the user's desired effect was accomplished. Feedback can be either intrinsic or extrinsic. The former is that which the individual senses directly from operating the control devices (e.g., clicks, resistance, control displacement). The latter is that which is sensed from an external source that indicates the consequences of the control action (e.g., indicator lights, display changes, aural tones).

Field: An area of the display screen reserved for displaying data, or for the user to enter data. In a database, a specified area used for a particular category of data, for example, equipment operational status.

Flowchart: A diagram that illustrates sequential relations among elements or events. Flowcharts are often shown as boxes connected by arrows.

Function key: A key whose activation will affect a control entry. On detecting the signal, the system usually performs some predefined function for the user.

Graphics tablet: (Digitizing tablet) Device used to convert an image into digital code by drawing or tracing with a pen-like or puck-like instrument. The instrument is moved across the tablet, generating a series of X-Y coordinates.

Human factors engineering (HFE): The application of knowledge about human capabilities and limitations to the design of a plant, system, and equipment. HFE ensures that designs, human tasks, and the work environment are compatible with the sensory, perceptual, cognitive, and physical attributes of the personnel who operate, maintain, and support them. (See human factors.)

Human factors: A body of scientific facts about human characteristics. The term covers all biomedical, psychological, and psychosocial considerations; it includes, but is not limited to, principles and applications in the areas of human factors engineering, personnel selection, training, job performance aids, and human performance evaluation (See human factors engineering).

Human-system interface (HSI): The means through which personnel interact with the plant, including the alarms, displays, controls, and job-performance aids. Generically, this also includes maintenance, test, and inspection interfaces.

Icon: Pictorial, pictographic, or other nonverbal representation of objects or actions. (See Field.)

Inspection and transfer steps: Intermediate steps that are included in a sequence of operations providing opportunities for detecting incorrect inputs. They are an additional level of defense against input errors. Rather than entering data directly into the control system, the inputted data may be sent to a holding file which one or more people must review and approve. Thereafter, a command may be entered to transfer the data from the holding file into the active portion of the control system.

Input field: The area in a display that is used to enter input. For example, a soft control may have an area in which operators can enter numerical data to adjust control setpoints or commands to execute actions.

Interface management: Actions performed by the operator to control the HSI rather than the plant, including finding and retrieving displays and adjusting display windows. Operators typically navigate through displays and retrieve needed controls and displays.

Interlock: A feature that requires operator actions to proceed in a specific sequence. For example, action B must be performed after action A, and action C after action B.

Joystick: A stick-type control device that can provide continuous cursor control in any direction on a display screen.

Light pen: A pencil- or pen-like control device that interacts with the computer system through the display device screen either by emitting or sensing light.

Lockin: A feature that keeps an ongoing operation active by preventing personnel actions from terminating it prematurely.

Lockout: A feature that prevents personnel from providing input that may have negative effects. Statically defined lockouts may restrict operators' inputs to a specific, predefined range or set of values. Context-sensitive lockouts may restrict input values based on the current situation.

Loss-of-activation error: An intended action is not carried out due to a failure of memory (i.e., the intention has partially or completely decayed from memory). A special case of loss-of-activation errors involves forgetting part of an intended act while remembering the rest (e.g., retrieving a display while not being able to remember why it is needed).

Manual mode: A processing mode in which the user is assumed to provide all inputs (as contrasted with interactive and automatic modes).

Menu: A type of dialogue in which a user chooses one item out of a list of displayed alternatives. Selection may be made by actions such as pointing and clicking and by depressing an adjacent function key.

Mimic: A display format combining graphics and alphanumerics used to integrate system components into functionally oriented diagrams that reflect the components' relationships.

Misordered components of an action sequence: A slip involving skipped, reversed, or repeated steps. Soft controls may be prone to this type of slip because they require sequential operations for accessing and using controls and displays.

Mistake: An error in intention formation, such as forming one that is not appropriate to the situation. Mistakes are related to incorrectly assessing the situation or inadequately planning a response.

Mode error: Performing an operation that is appropriate for one mode when the device is in another mode. Mode errors occur when the user believes the device is in one mode when it is in another one.

Mouse: A control device whose movements across a flat surface are converted into analogous movements of the cursor across the screen.

Multiple-loop programmable controller: A digital controller that can control multiple variables via independent channels, one per control loop.

Natural language: A type of dialogue in which users compose control entries in a restricted subset of their natural language, e.g., English.

Output: The data that are the product of an information handling operation or series of operations; the data emitted from a storage device; the data being transferred from primary storage (central processing unit) to secondary storage (tape, floppy disk); electrical pulses; reports produced by a printer or typewriter unit; a general term for output media, such as cards and tape. Contrasts with Input.

Pointing interface: A computer-based user interface operated via cursor or touch screen.

Plant variable: A variable that represents the status of a plant system or process. For example, the variable reactor coolant system pressure represents the pressure inside the piping of the reactor coolant system. (See variable.)

Query language: A type of dialogue in which users compose questions using a special-purpose language to retrieve information.

Question and answer: A type of dialogue in which a computer displays questions, one at a time, for a user to answer.

Schema: A sequence of linked behaviors that, through repeated performance or deliberate training, becomes somewhat automatic to the individual. A schema may be executed when the type and strength of the stimulus matches the trigger for the schema.

Selection display: Any display from which the operator may make a selection, such as choosing a plant variable, plant component, or a command. Two formats commonly used for selecting plant components and variables are the menu and mimic.

Should and **may**: The word "should" is used to denote a recommendation; "may" is used to denote permission and applies to a characteristic that is acceptable but not necessarily recommended (e.g., an equally acceptable alternative may exist).

Slip: An error in carrying out an intention. Slips result from "automatic" human behavior, when schemas, in the form of subconscious actions that are intended to accomplish the intention, get waylaid en route to execution. Thus, while one action is intended, another is accomplished. An expert's highly practiced behavior leads to the lack of focused attention that increases the likelihood of some forms of slips.

Soft control: A control device that has connections with the control or display system that are mediated by software rather than direct physical connections. As a result, the functions of a soft control may be variable and context dependent rather than statically defined. Also, the location of a soft control may be virtual (e.g., within the display system structure) rather than spatially dedicated. Soft controls include devices activated from display devices (e.g., buttons and sliders on touch screens), multi-function control devices (e.g., knobs, buttons, keyboard keys, and switches that perform different functions depending upon the current condition of the plant, the control system, or the HSI), and devices activated via voice input.

Soft slider: An input format used to directly manipulate a variable over a set range of values (also called a slider bar or a scroll bar). A soft slider resembles a barchart with a pointer directed toward the current value. They are typically manipulated via pointing interfaces, such as a touch screen or mouse. Input is provided by sliding the pointer along the length of the barchart scale to the desired value. It is used when the range of possible values and the ratio of a value to that range must be displayed.

System: An integrated collection of plant components and control elements that operate together, and possibly in conjunction with other systems, to perform a function.

System response time: The elapsed time between the initiation of a command and the notification to the user that the command was completed.

Text: The primary display for word processing, consists of alphanumeric character strings in linear arrays, making up words, sentences, and paragraphs. The main body of printed or written matter on a page or in a message.

Touch screen: A control device that allows the user to communicate with the computer by touching a screen.

Trackball: A control device with which the user can control cursor movement in any direction by rotating a ball.

Undo: A capability that reverses the effect of the previous operation.

Unintentional Activation: A slip that occurs when a set of actions (schema) that is not part of a current action sequence becomes activated and then triggered for extraneous reasons. It can lead to the unintended actuation of an input device.

Value: Specified data for a particular parameter or variable.

Variable: A quantity that can assume any of the given set of values.

Verification step: A step in a transaction sequence that requires the user to verify an intention to perform a particular action. For example, the user selects an option and then presses the Enter key to verify the selection.

Visual angle: A measure, in degrees, of the size of the retinal image subtended by a viewed object. It represents the apparent size of an object based on the relationship between an object's distance from the viewer and its size (perpendicular to the viewer's line of sight). An object of constant size will subtend a smaller visual angle as it is moved farther from the viewer. Visual angle typically is defined in terms of minutes of visual arc.

Workload: The physical and cognitive demands placed on plant personnel.

Workstation: The physical console at which a user works.