Helping the Operator in the Loop: Practical Human Machine Interface Principles for Safe Computer Controlled Systems

Andrew Rae

System Safety and Quality Engineering
11 Doris St Hill End Queensland 4101

ajrae@ssqe.com.au

Abstract

This paper addresses a need for practical human machine interface (HMI) design principles for safety-related computer controlled systems. Most HMI design principles and guidelines focus on aspects of usability. Whilst usability is important for reducing the incidence of human error, more is required of a safe human machine interface. A safe HMI design must reduce incidence of human error but equally importantly, it must reduce the consequences of both operator and computer error by helping the operator to mitigate errors. By this, we do not suggest that it is always necessary, or even advisable, to keep an operator in the decision-making path. However, where a human acts as a system component, they should be used as effectively as possible. Good HMI principles and guidelines can help with this.

In this paper we firstly examine existing sets of principles and guidelines for HMI design. We then establish a simple model of a computer controlled system with an operator in the loop, and, through hazard analysis, identify those sequences of actions which involve the operator as either cause of, or mitigation for, a hazardous system state. From these sequences, we isolate the operator actions which should be discouraged or facilitated by the HMI, and derive a set of general safe HMI principles.

We argue that this approach provides a sound argument for both the necessity and the completeness of our principles. The paper goes on to discuss how the approach may be extended to derived detailed guidelines from the safe HMI principles.

Keywords: safety, human factors, human-machine interface, principles, guidelines

1 Introduction

There is a large body of human-machine interface design literature (Perlman 2007), which has at its heart three main goals – effectiveness (helping operators achieve their intentions) efficiency (reducing time taken and the incidence of operator errors) and satisfaction (providing an enjoyable experience) (ISO 2006). Traditionally, human factors research for safety has focussed specifically on reducing the incidence of operator error (see, for example Redmill and Rajan (1997)).

However, no method of user interface design can completely mitigate the fact that operators make mistakes. They make, amongst others: slips, lapses, substitutions, sequence errors, post-completion errors, rule misapplications, and even errors of intention.

Automated systems also may have errors. They contain, amongst others: requirements errors, specification errors, design errors, and simple typographic bugs.

When operator mistakes and computer errors manifest, often the last line of defence is the operators themselves. Safe system design involves not just reducing the likelihood of hazardous failures, but mitigating their consequences as well. This requires HMI design that facilitates the operator acting to mitigate errors, whatever their source.

In this paper we document our investigation into existing HMI principles and guidelines, and discuss why we found them inadequate for our purpose. We then conduct a hazard analysis based on a model which incorporates both a fallible human and a fallible control system, and identify mitigations using the HMI.

Our work differs from previous similar studies in the following ways:

- we are concerned specifically with safety, not general usability;
- we are concerned not just with HMI mitigations for operator errors, but with facilitation of the operator mitigating their own or computer system errors; and
- we seek to construct a sound argument for both the necessity and completeness of our principles.

Note that throughout this paper, we will be using the term "usability" to refer to the ISO 9241-110 goals of effectiveness, efficiency and satisfaction (ISO 2006). Thus, safe HMI principles may accord with usability principles, but may also conflict with usability.

2 Motivation and Goals

Our motivation for this work comes from projects where a "human in the loop" has been identified as a mitigating factor in accident sequences. Where this reduces the required integrity of some software components, it creates

integrity requirements on the user interface and on the operator.

It is insufficient in such circumstances to merely specify a requirement that the user interface should provide facility for the operator to mitigate the hazard, as this facility could be compromised through interface design which prevents the operator from:

a) recognising the need for action;

b) choosing an appropriate action; and

c) successfully taking the appropriate action.

Whilst user interfaces vary markedly between application domains and operating environments, it is reasonable to expect that there may be a small core of generic safety principles which apply to any interface.

These safe HMI principles could be used to inform larger sets of application-specific usability requirements, or included directly in requirements specifications as a path to meeting the integrity requirement placed on the interface through the assumption of a human in the loop.

An alternate approach to specifying user-interface requirements would be to apply analysis to the tasks to be performed by the user, such as through User-centric Design (Katz-Haas 1998) or Cognitive Work Analysis (Rasmussen 1994). These approaches are predicated on the availability of a usability professional, and participation of a pool of users in either iterative evaluation of designs (User-centric Design) or evaluation of existing systems (Cognitive Work Analysis). Without questioning the value of such approaches, we see a role for principles and guidelines for organisations and projects who are unable to resource full usability studies using these techniques.

3 Existing Guidelines

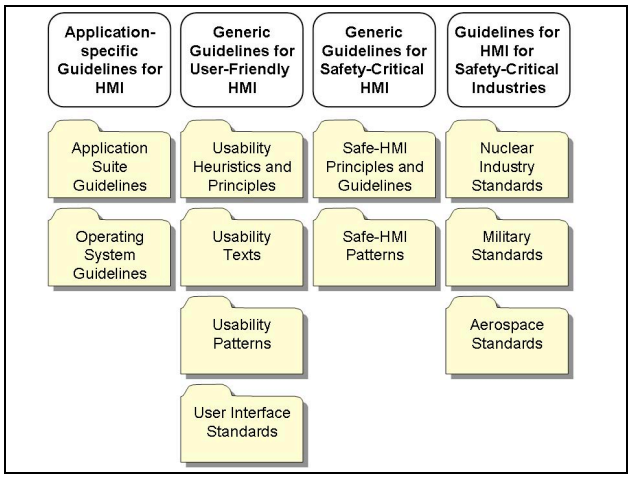

Figure 1- Categories of Existing HMI Guidelines

Existing guidelines, as shown in Figure 1, fall into four main categories:

1. application suite or operating system style guides;
2. generic guidelines for user-friendly interface design;
3. generic guidelines for safety critical HMI design; and
4. HMI guidelines for safety-critical industries.

3.1 Style Guides

Style guides (see for example Benson (2004)) provide a number of usability benefits which reduce the learning time for interfaces, and arguably the rate of operator error. They achieve this principally through providing interface elements which behave in a consistent way between applications.

General usability principles, typically extracted from more abstract guides such as those discussed in Section 3.2, are used to generate very specific, practical guidance.

For each element type that comprises the interface, instructions are given as to its appearance and its behaviour. For example, the following items are extracted from Benson's (2004) guidelines on alerts.

- An alert should have the same border type as a dialog.
- Alert windows should have no title.
- An alert should not appear in the panel window list, unless it is, or may be, the only window presented by an application.
- An alert should be raised above an application window when the application is selected from the window list.

3.2 Generic Guidelines

3.2.1 Usability Heuristics and Principles

Nielson (1994), Tognazzini (2003) and Schneiderman (1998) present short lists of user-friendly interface design principles, based primarily on observed problems with usability of websites and applications.

For example, Nielson's ten heuristics for interface evaluation are:

1. Visibility of system status

The HMI should provide appropriate, timely feedback.

2. Match between system and real world

The HMI should follow real-world conventions, rather than use system-oriented terms. Information should appear in a natural and logical order.

3. User control and freedom

The HMI should provide facility to leave unwanted states and correct user errors easily, through functions such as undo and redo.

4. Consistency and standards

The HMI should follow conventions for the particular platform being used.

5. Error prevention

The HMI should either not contain error-prone conditions, or should check for such conditions and ask the user to confirm before they commit to the action.

6. Recognition rather than recall

The user should not be required to remember information from one part of the dialog to another. Instructions should be visible or easily retrievable.

7. Flexibility and efficiency of use

Accelerators and tailoring should be provided for expert users, but should be invisible for novice users.

8. Aesthetic and minimalist design

Dialogs should contain the minimum amount of information necessary to perform their function.

9. Help users recognize, diagnose, and recover from errors

Instead of using error codes, errors should be presented in plain language, with clear instructions on what action is expected from the user to recover from the error.

10. Help and documentation

Documentation should be minimalist, searchable, and task-oriented.

3.2.2 Usability Texts

Whole books can be, and have been, written just containing, explaining and illustrating user interface guidelines. Examples of such works include Brown (1989), Mayhew (1992) and Smith-Mosier (1986).

As an example of the guidance provided in such works, Mayhew (1992) lists fifteen design goals for user interfaces, followed by detailed guidelines backed by experimental results. Mayhew's design goals include some principles important for safety (such as robustness and protection) some that are irrelevant for safety (such as "What You See is What You Get" for word-processing) and some which can be actively unsafe (such as invisible technology and use of defaults).

3.2.3 Usability Patterns

Usability patterns show how recurring problems in user interface design can be solved according to high level principles (Mahemoff 1998). Usability patterns often specify particular features provided by the interface, such as (Ferre 2003):

- Context-Sensitive Help
- History Logging
- Form/field Validation
- Shortcuts
- Undo
- Wizard
- Tour

3.2.4 User Interface Standards

International Standards (ISO 9241-110, ANSI/HFS 100) provide a consensus view on what constitutes good usability practice. ISO 9241-110 is based on seven principles:

1. Suitability for the task

"A dialogue is suitable for a task when it supports the user in the effective and efficient completion of the task. In a dialogue which is suitable for the task, the user is enabled to focus on the task itself rather than the technology chosen to perform that task."

2. Self-descriptiveness

"A dialogue is self-descriptive to the extent that at any time it is obvious to the users which dialogue they are in, where they are within the dialogue, which actions can be taken and how they can be performed."

3. Conformity with user expectations

"A dialogue conforms with user expectations if it corresponds to predictable contextual needs of the user and to commonly accepted conventions."

4. Suitability for learning

"A dialogue is suitable for learning when it supports and guides the user in learning to use the system."

5. Controllability

"A dialogue is controllable when the user is able to initiate and control the direction and pace of the interaction until the point at which the goal has been met."

6. Error tolerance

"A dialogue is error-tolerant if, despite evident errors in input, the intended result may be achieved with either no or minimal corrective action by the user. Error tolerance is achieved by means of damage control, error correction, or error management to cope with errors that occur."

7. Suitability for individualisation

"A dialogue is capable of individualization when users can modify interaction and presentation of information to suit their individual capabilities and needs."

3.3 Usability and Safety

So far, we have discussed guidelines and principles for usability. Whilst usability and safety can be complementary, they can also be in conflict. The core reason for this is that operators have operational goals as well as safety goals. For example, air traffic and railway controllers have as an operational goal the movement of people and cargo between locations. Power station and process plant operators have the goal of keeping the plant running. An interface can be highly effective in facilitating the achievement of operational goals, but deficient in discouraging unsafe actions, and facilitating recovery from unsafe states and actions.

A related concern is the conflict between efficiency of the user and safety. HMI features such as customization,

shortcuts, and even "smart" interfaces which anticipate the user's intentions contribute to efficiency, but may detract from safety. On the other hand, HMI features such as redundant input which contribute to safety can detract from usability.

As well as being insufficient for safety, usability can also be too broad a concern. For our purposes, we require a minimalist set of guidelines. Focusing on usability issues that are peripheral to safety can distract (and thus detract) from achieving as safe an interface as possible.

Section 3.4 summarises the work on safe user interface principles and guidelines.

3.4 Generic Guidelines for Safe HMI Design

3.4.1 Leveson's Safe HMI Guidelines

Leveson (1995) provides a list of guidelines specifically intended for safe user interface design. The list does not distinguish between process requirements, general principles, and specific guidelines. In total, there are sixty listed items, although many overlap or address the same objective.

A representative selection of the guidelines includes:

Process requirements:

2. Begin the design process by considering the operator and maintain that perspective throughout.
3. Involve the operators in design decisions and safety analysis throughout the development.

General principles:

1. Design the HMI to augment human abilities, not replace them.
13. Provide adequate feedback to keep operators in the loop.
16. Provide facilities for operators to experiment, to update their mental models, and to learn about the system. Design to enhance the operator's ability to make decisions and intervene when required in emergencies.

Specific guidelines:

15. Distinguish processing from failure. Provide real-time indication that the automated control system is functioning, along with information about its internal state (such as the status of sensors and actuators), its control actions, and its assumptions about the system state.
28. Flag rather than remove obsolete information from computer displays. Require the operator to clear it explicitly or implicitly.
49. Avoid displaying absolute values: Show changes and use analog instead of digital displays when they are more appropriate. Provide references for judgement.

Leveson does not give an explicit source for her guidelines. Whilst this in no way invalidates the guidelines, it makes it difficult to ensure that the guidelines are necessary or complete for safety.

3.4.2 Safe-HMI Patterns

Hussey (1999) provides "patterns" for designing safety-critical user interfaces. These patterns are derived through review of safety-critical interface literature. Each pattern includes a high level principle, and some detailed "solutions" or guidelines.

An example pattern with associated solutions is:

1. Operator Awareness Maintained

The operator should be involved in the routine operation of the system, to ensure that they have a current mental model of the system.

- Activities requiring passive actions should be minimized.
- Tasks should be varied and encourage operator experimentation.
- The HMI should provide feedback on the system state.
- The operator should be informed of any anomaly.
- Obsolete information should be flagged rather than removed.

The other patterns presented by Hussey are "Undo", "Error Prevention", "Slip Reduction", "Warn", "Clear Displays", "Independent Information / Interrogation" and "Knowledge-based Reasoning Supported".

Hussey's patterns provide motivation for each principle, and are a useful categorisation of the key principles for safe HMI design in the literature. As they are intended to be a summary of already existing good guidance, no argument is provided for their completeness.

Mahemoff (1999) provides an alternate taxonomy of patterns. Mahemoff groups fourteen patterns according to the aspects of human-user interaction.

Task Management

- The HMI should enable operators to reverse the tasks they have performed.
- Complex tasks should be split into sequences of simpler tasks.
- Operators should be required to perform tasks multiply.
- Related task steps should be grouped so that the effects are not realised until all task steps are completed.

Task Execution

- The HMI should be arranged so that affordances are provided which reduce the likelihood of mechanical user error.

- The HMI should disallow the user from performing actions which are erroneous according to the system state.

Information

- The HMI should provide a close mapping to reality.
- The HMI should allow the user to compare the current state to previous states.
- The HMI should be able to provide additional information on request.
- The HMI should provide ways to record the completion status of tasks.

Machine Control

- Interlocks should be used to detect and block hazards from occurring.
- Tedious or difficult tasks should be automated.
- Automatic shutdown should be used if shutdown is inexpensive and simple.
- Warnings should be given if safety-critical margins are approached.

Mahemoff's patterns lack the detailed guidance provided by Hussey (1999), and share the same difficulty in that there is no argument provided for their completeness.

3.5 Guidelines for HMI for Safety-Critical Industries

NUREG 0700 (O'Hara 2002) contains very detailed guidance on HMI evaluation for nuclear power plants. This guidance is based on the explicit twin goals of supporting the operators in monitoring and controlling the plant without imposing excessive workload, and facilitating the recognition, tolerance and recovery from human errors.

In support of these goals, the guidelines embody eighteen principles:

1. Personnel Safety
2. Cognitive Compatibility
3. Physiological Compatibility
4. Simplicity of Design
5. Consistency
6. Situation Awareness
7. Task Compatibility
8. User Model Compatibility
9. Organisation of HSI Elements
10. Logical/Explicit Structure
11. Timeliness
12. Controls/Displays Compatibility
13. Feedback
14. Cognitive Workload
15. Response Workload
16. Flexibility
17. User Guidance and Support
18. Error Tolerance and Control

No direct derivation of the principles from the goals is provided.

MIL-STD-1472F (1999), Section 5.14 provides the USA military standard for user-computer interfaces. Whilst it claims to provide only general principles, to allow for flexibility in usability design, it actually contains a list of very specific guidelines for the appearance of displays and functionality of controls. Example guidelines are:

> 5.14.3.3.2 Flash. Flash coding shall be employed to call the user's attention to mission critical events only. No more than 2 flash rates shall be used. Where one rate is used, the rate shall be 3 – 5 flashes per second. Where two rates are used, the second rate shall be not greater than 2 per second.
>
> 5.14.3.5.6.2 Vertical extension. Where lists extend over more than one display page, the last line of one page should be the first line on the succeeding page.

NASA-STD-3000 (1995) Section 9 provides the equivalent NASA guidance, and DOT/FAA/CT-01/08 (2001) provides the equivalent Federal Aviation Authority guidance.

4 Evaluating Existing HMI Guidelines for Application to Safety Critical Design

We seek to evaluate the reviewed guidelines against three criteria:

- Applicability to safety
- Completeness
- Necessity

Rather than attempting to establish objective measures for these criteria, we examine the arguments raised in support of the guidelines.

It is found that the guidelines stem from:

1. Direct observation of users attempting to complete tasks on real systems; e.g. Nielson (1994), and Tognazzini (2003)
2. Controlled experiments involving users attempting to complete simulated tasks; e.g. Mayhew (1992)
3. The judgement and experience of the guideline author; e.g. Leveson (1995)
4. Reference to literature and other guidelines; e.g. Benson (2004) and Hussey (1999)

Whilst 1 and 2 are sound arguments in support of necessity and completeness of the guidelines, they do not directly address the issue of applicability to safety.

Arguments 3 and 4 are essentially appeals to authority, and as such have limited value.

5 Hazard Analysis of an Operator-in-the-Loop System

5.1 System Model and Action Sequences

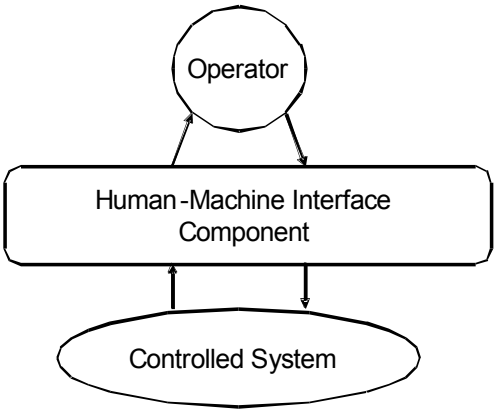

Figure 2 - An Operator-in-the-Loop Computer Controlled System

Figure 2 shows a typical computer-control system. For the purpose of this analysis we consider any automated functions to be part of the controlled system.

Table 1 - States of Components

Component	States
Operator	Aware of unsafe state, Not aware of unsafe state
Interface	Action pending, No action pending
System	Safe, Unsafe

Table 1 shows the states each of the components may be in. Analysis of this model is shown in Figure 6. This diagram shows all of the state transitions involving operator or HMI action. Note that in addition to these shown transitions, the controlled system can also spontaneously transition between the safe and unsafe states.

There are four sequences of actions in this diagram. The first sequence, where an operator takes a safe action, transmitted by the HMI, leading to a safe state, does not have an impact on safety.

Figure 3 indicates how an operator action, transmitted by the HMI, can lead to an unsafe state. This figure may be extended by any number of iterations where the operator detects the unsafe action and fails to undo it.

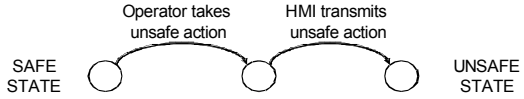

Figure 3 - Operator Initiated Unsafe State

Figure 4 shows how an operator can detect and undo an unsafe action, keeping the system in a safe state.

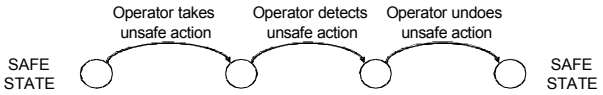

Figure 4 - Operator Detects and Undoes an Unsafe Action

Figure 5 shows how an operator can detect and correct an unsafe state of the controlled system, returning the system to a safe state.

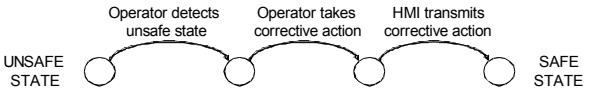

Figure 5 - Operator Detects and Corrects an Unsafe State

For each operator action, we define minimum principles to enable (in the case of actions leading towards a safe state) or discourage (in the case of actions leading towards an unsafe state) the action.

In addition to these principles, there is also a requirement for *fidelity* of the HMI – that is, it must faithfully transmit requested controls, only transmit requested controls, and accurately present information about the system.

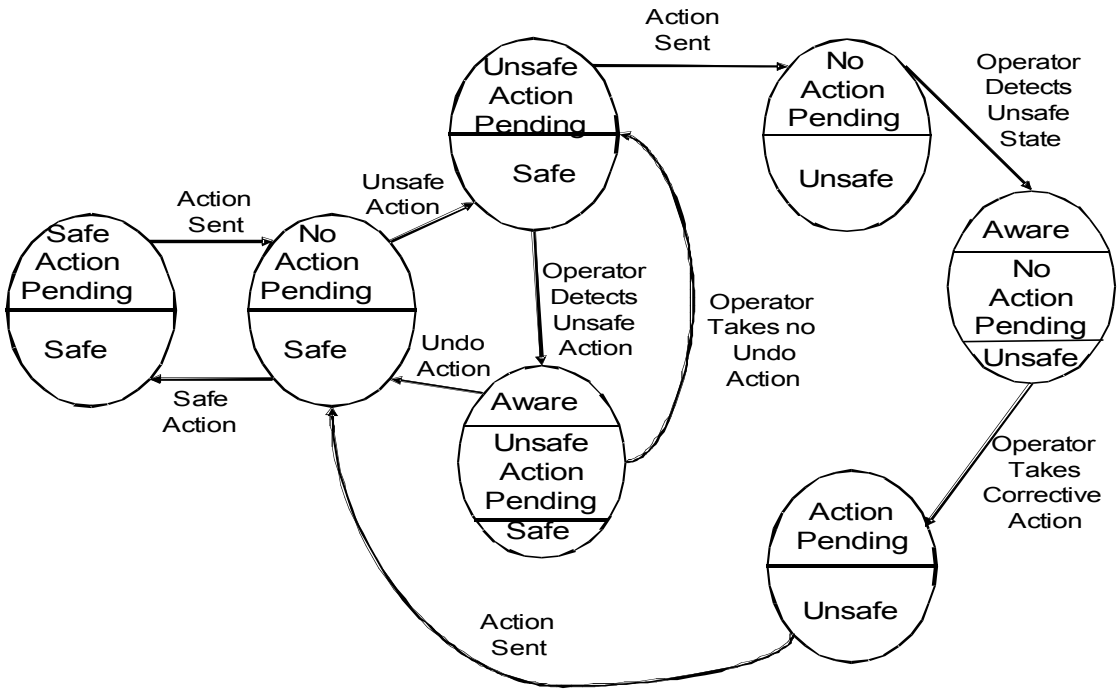

Figure 6 - State Diagram for a System with an Operator-In-The-Loop

5.2 Discouraging the Operator from Taking Unsafe Actions

Without considering detailed cognitive models for the operator, there are essentially three reasons an operator may take an unsafe action – malicious intent, errors of intent, and errors of skill. Malicious intent is considered a security issue and is beyond the scope of this paper. An error of intent is where the operator is unaware that the consequence of their action is an unsafe state of the system. An error of skill is where the operator takes an action that they did not intend to take.

To prevent an error of intent, the operator must be aware of the current state of the system, and have a mental model sufficient to allow prediction of the consequences of actions.

Principle 1: The HMI should make the operator aware of the current state of the controlled system, as it relates to safety

Principle 1 seems self evident, but is frequently violated. The worst form of violation is where directly misleading information is displayed. This was evident in the Three Mile Island nuclear accident, where there was no positive feedback on controls to some valves. Accordingly, the display showed the *requested* state of the valves, as opposed to the *actual* state of the valves (Kemeny 1979).

More subtle violations of principle 1 occur when the state of the system is truthfully displayed, but in a fashion which the operator cannot comprehend. For example, there might be too much information on a display, or the information may be displayed in a misleading fashion.

Principle 2: The HMI should assist the operator in establishing a mental model of the system and controls that is accurate with respect to the safety behaviour of the system

An accurate mental model of the system and controls is necessary to evaluate the consequences of actions. An example of this principle being violated is British Midland Flight 092, where after an engine fire the pilots shut down the wrong engine. One of the reasons for this mistake is that their mental model of the system did not take into account automation which reduced the fuel flow to both engines when one engine was shut down, leading them to believe that they had corrected the problem (AAIB 1990).

Errors of skill, on the other hand, arise directly out of features in the HMI design that facilitate such errors.

Principle 3: The HMI should remove or minimise features that induce errors of skill

Error-inducing features range from designs which place too much short-term memory load on the operator, to controls which appear similar but perform different functions. For example, in December 1999, an RAF Tornado jettisoned two fuel tanks near a road. The resulting inquiry postulated that the incident was partially caused by selection of the wrong button, and recommended review of the controls (BBC 2000).

5.3 Enabling the Operator to Detect Unsafe Actions

Detection of unsafe actions before they translate into unsafe states can only occur if there is a delay between action and consequence. Such a delay may involve a simple time delay, or addition of an extra state between action and consequence, such as a “select before execute” model of operation.

Principle 4: The HMI should provide a delay before execution of potentially unsafe actions

We note here that this principle may not be universally true, due to the existence of *ambiguous* actions, which, depending on the state of the system, are potentially either unsafe or corrective. As shall be noted in Principle 10, corrective actions should not be subject to undue delay.

Identification and suitable treatment of ambiguous actions is an open question which shall not be dealt with further here.

Principle 5: For potentially unsafe actions, the HMI should provide feedback to the operator indicating the selected action

Principle 6: The HMI should alert the operator to the consequences of potentially unsafe actions

The stereotypical example satisfying principles 4, 5 and 6 is a confirmation dialog. It may provide a delay – by requiring an additional user action such as pressing a button, or entering a password. It may also provide feedback on the user's selected action, and for known dangerous actions, a warning as to the potential consequences of the action.

5.4 Enabling the Operator to Undo an Unsafe Action

Providing a delay, feedback and a warning to the operator of the consequences of an action are useless if the operator has no opportunity to rectify the situation.

Principle 7: The HMI should provide a facility for the operator to rapidly reverse the consequences of potentially unsafe actions

As an example of this principle, on railways, it is often necessary to limit access to a section of track to allow maintenance work. This is known as "placing a block". Removing such a block is hazardous if performed inadvertently, since it may expose maintenance staff to railway traffic. Whilst initially placing a block may require several actions, if a block is removed inadvertently it should be able to be restored with a single action.

5.5 Enabling the Operator to Detect an Unsafe State

In addition to Principle 1, which makes the operator aware of the current state of the controlled system, a safe HMI should actively alert the operator when a state is unsafe.

Principle 8: The HMI should provide a facility to alert the operator to potentially unsafe states of the controlled system

As with principle 1, principle 8 may be subtly violated through provision of alerts that are not readily comprehensible by the operator. During the Three Mile Island reactor incident, more than 100 alarms were active, and the computer printer ran two hours behind events due to the volume of alerts (Kemeny 1979).

5.6 Enabling the Operator to Take Corrective Action

In order for the operator to *plan* a corrective action, they require an adequate mental model of the system and its controls. This is provided through Principle 2.

In order to actually *take* corrective action, the action must be possible, and should not be subject to undue delays.

Principle 9: The HMI should provide facility for the operator to issue any control at any time

This principle is somewhat controversial, as it seems intuitively at odds with the requirement to discourage the operator from taking unsafe actions. We contend that an HMI should discourage the operator from making errors of intent and skill, should delay potentially unsafe actions and provide opportunity to reverse unsafe actions, but should not ultimately override the operator's judgement as to what constitutes a safe action.

There are two reasons for this. Firstly, the operator has sources of information beyond the sensors provided by the system. Secondly, the operator has judgement and problem-solving ability superior to that of other system components. It is for these reasons that operators are included in the loop, and we should not limit the very abilities that make them valuable.

An example of a system which violates this principle is one where the allowed user actions are limited by the mode of the system. Note that here we disagree directly with Mahemoff, who indicates that the HMI should disallow actions which are erroneous according to the system state.

One way of obeying the principle is through provision of emergency controls such as "battle shorts" which are explicitly labelled and protected against accidental operation.

Note that it is insufficient to simply provide functionality. Facilities associated with restoring the system to a known safe state should be quick to select and operate.

Principle 10: HMI facilities associated with restoring the system to a safe state should not be subject to undue delays

6 Evaluation of the Principles

The contribution of this paper lies not just in presenting a set of principles for safe HMI, but in deriving them in such a way that there is a sound argument for their necessity and completeness.

We argue that our method has resulted in:

1. The basic requirements for safe interaction with a controlled system being completely specified; and
2. For each of these requirements, provision of principles sufficient and necessary to ensure achievement of the requirement.

One way of evaluating our principles is comparison with those discussed in Section 3. With respect to those sources which discuss usability and user-friendliness, it is to be expected that some principles would be common to safety and usability, some usability heuristics would be specific implementations of safety principles, some useability heuristics would not have direct safety implications, and that in some cases useability and safety would conflict.

With respect to sources which discuss safe human interfaces, we would hope to find general accord between their principles and guidelines and our safe HMI principles.

Space constraints prevent us from presenting the detailed results of this comparison here. The only notable point of difference between our principles and those stated elsewhere is the already indicated issue of principle 9.

7 From Principles to Guidelines

Whilst the principles derived in Section 4 have utility for developing and evaluating HMIs for safety critical control systems, it is our intention to extend the guidance through the development of specific design guidelines in support of the general principles.

To produce these guidelines, we will continue our analytical approach, tailoring the approach for each principle. We provide an indication of our intended technique for the first three principles below.

7.1 Principle 1

For principle 1, which relates to flow of information about the state of the system from the interface to the operator, we will apply Hazard and Operability Studies (IEC 2001). By considering how the keywords *too much*, *too little*, *none, too early, too late*, and *wrong type* apply to the information flow, we can derive detailed guidelines.

For example, consider the keyword *too much*. This can be interpreted as an excess of information inhibiting the operator's ability to comprehend the state of the system. To mitigate this possibility we include a guideline that the system should minimise the information provided to the operator. To mitigate the possibility of *too little* information, we expand the guideline to give the operator the capacity to request further information.

7.2 Principle 2

Principle 2 relates to the formation by the operator of an accurate mental model of the system and its controls. To derive guidelines for this principle, we draw on research on how people form accurate mental models, summarised by Mayhew (1992). Key factors are:

- having a user interface that draws maximally on real-world knowledge;
- allowing the operator to test and refine hypotheses about the system;
- consistent use of any metaphors or symbols within the application; and
- similarity of the application with other applications.

7.3 Principle 3

Principle 3 is that the HMI should reduce or minimise error-inducing features. To determine what an error-inducing feature is, we draw on Reason's (1990) Generic Error Modelling System (GEMS). As we are concerned here with mistakes of skill rather than of intention or knowledge, we are interested in what causes *slips* and *lapses*.

Slips and lapses include *capture errors*, where a familiar sequence of actions "takes over" a sequence of intended actions; *description errors*, where the objects of different actions are close together or visually similar; *data-driven errors,* where new information "overwrites" the current information the operator is using; *associative activation errors*, where a similar familiar action is substituted for an intended action; *loss of activation errors,* where the goal is forgotten mid-action sequence; and *mode errors*, where there is a mismatch between the expected and actual function of a control due to the mode of the interface.

Guidelines will be provided for the avoidance of each of these error types.

8 Use of the Guidelines

For an ideally resourced project, the HMI would be designed by a human factors expert, evaluated by an independent expert using a set of usability heuristics, undergo extensive user testing, and evolve through a succession of prototypes.

In practice, human factors experts are not always available, and usability testing by real users is only possible once the interface nears completion, requiring expensive rework if substantial changes are necessary.

Even for projects where extensive usability evaluation is included in the budget, an initial set of requirements is necessary for the HMI design. Hence, we provide these principles as a starting point for HMI design.

The principles can also be used as heuristics for evaluating or assessing HMI for safety critical evaluations.

9 Conclusion

In this paper, we have derived a set of principles for safe HMI design, based on the philosophy that an HMI for a safety critical system should not only reduce operator error, but allow the operator to mitigate both their own errors and dangerous systems states from other causes.

We have provided a completeness and necessity argument for our principles, and shown how the principles can be extended to provide detailed guidelines.

At this stage, our work is theoretical, not empirical. We can point to instances where design in contravention of our principles has led to accidents, but formulating an experiment to show that an interface following these principles could prevent an accident is problematic.

Our future plans are to trial the principles on industry projects to test their suitability and applicability as requirements – note that this is a separate issue to their efficacy in improving safety – and extend the approach to provide detailed guidelines in support of the principles.

10 Acknowledgement

I acknowledge the contribution of Dr Alena Griffiths, who reviewed the draft of this paper and provided some valuable insights.

11 References

AAIB (1990): *Report on the accident to Boeing 737-400 - G-OBME near Kegworth, Leicestershire on 8 January 1989,* Air Accidents Investigation Branch

ANSI/HFS 100 (1988): *American National Standard for Human Factors Engineering of Visual Display Terminal Workstations*

Bainbridge, L (1985): *Guide to Reducing Human Error in Process Operation*, Human Factors in Reliability Group

BBC (2000): *Bomber Crew Hit Wrong Button*, BBC News 6 December 2000, http://news.bbc.co.uk/2/hi/uk_news/scotland/1057485.stm

Benson et al. (2004): *Gnome Human Interface Guidelines 2.0*, Gnome Usability Project.

Brown, C.M. (1989): *Human-Computer Interface Design Guidelines,* Ablex Publishing, New Jersey

DOT/FAA/CT-01/08 (2001): *Computer-Human Interface Guidelines*

Ferre et al. (2003): A Software Architectural View of Usability Patterns, INTERACT

Hussey (1999): Patters for Safety and Usability in Human-Computer Interfaces, SAFECOMP

IEC (2001): *IEC 61882 Hazard and Operability Studies – Application Guide*

ISO 9241-110 (2006): *Ergonomics of human-system interaction -- Part 110: Dialogue principles*

Katz-Haas, R (1998): *Ten Principles for User-Centred Web Design*, Usability Interface, Vol. 5, No. 1

Kemeny (1979): *Report of the President's Commission on the Accident at Three Mile Island*

Leveson, N.G. (1995): *Safeware: System Safety and Computers,* Addison Wesley

Mahemoff, M and Hussey, A (1999): Patterns for Designing Safety-Critical Interactive Systems, Technical Report No. 99-23 Software Verification Research Centre

Mahemoff, M and Johnston, L (1998). Principles for a Usability-Oriented Pattern Language**,** *OZCHI '98 Proceedings*, Los Alamitos, CA.

Mayhew, D. J. (1992): *Software User Interface Design*, Prentice Hall, New Jersey

MIL-STD-1472F (1999): *Human Engineering*

Murphy, N. (1998): Safe Systems through Better User Interfaces, *Embedded Systems Programming*

NASA-STD-3000 (1995): *Man-Systems Integration Standards*

Nielsen, J. (1994): Heuristic evaluation. In Nielsen, J., and Mack, R.L. (Eds.), *Usability Inspection Methods*, John Wiley & Sons, New York, NY.

O'Hara et al. (2002): *NUREG-0700 Human-System Interface Design Review Guidelines*, Rev 2, US Nuclear Regulatory Commission

Perlman, G (2007): HCI Bibliography, http://www.hcibib.org/ Accessed 7 Aug 2007

Rasmussen, J. et al. (1994): *Cognitive Systems Engineering,* Wiley, New York.

Reason, J (1990): *Human Error*, Cambridge University Press, New York, NY.

Redmill, F and Rajan, J (1997): *Human Factors in Safety Critical Systems*, Butterworth Heinemann, Oxford

Schneiderman, B. (1998): *Designing the User Interface: Strategies for Effective Human-Computer Interaction,* Adison-Wesley, Reading, MA.

Smith, S.L. and Mosier, J.N. (1986): *Guidelines for Designing User Interface Software*, The MITRE Corporation

Tognazzini, B. (2003): First Principles of Interaction Design, http://www.asktog.com/basics/firstPrinciples.html Accessed 10 Feb 2007