Impact
and the
Design of the Human-Machine Interface

A.M. Dearden and M.D. Harrison
University of York

ABSTRACT

In this paper, we consider the concept of the *impact* of an action or human error. We begin from an informal definition of impact as:

- the effect that an action or sequence of actions has on the safe and successful operation of a system;
- and develop a quantitative measure of the impact of specified behaviours.

It is important that human-machine interface designers should understand the relationship between operator actions and the hazards associated with a system. We demonstrate how impact can be assessed prior to, or in parallel with, the design of the human-machine interface, and show how impact assessments could be used to allow risk analysts to inform designers about the relationship between operator actions and system hazards. To illustrate our approach we present a simple case study.

INTRODUCTION

Operator error has been blamed for many accidents and incidents in safety-critical systems. It is important that human-machine interface[1] designers should understand the relationship between operator actions and the hazards associated with a system. In this paper, we consider the concept of the *impact* of an action or human error, and show how this concept may be useful to interface designers. We begin from an informal definition of impact as:

- the effect that an action or sequence of actions has on the safe and successful operation of a system.

We show how a quantitative measure of impact could be generated prior to, or in parallel with, interface design. We argue that an analysis of impact: would complement existing design and analysis techniques by providing valuable additional information for the early stages of design; and that quantification of impact may be more soundly based than attempts to quantify human-error probabilities. We also make some observations on the relevance of impact analysis to the design of autonomous and semi-autonomous computer based control systems.

DESIGNING TO TAKE ACCOUNT OF HUMAN ERROR

A number of techniques are already available to support the design and analysis of interfaces for safety-critical systems. Current techniques can be divided into three major classes. We consider these three classes separately below.

An extended form of this paper can be found in *The Proceedings* of COMPASS '96.

Authors' Current Addresses:
Department of Computer Science, University of York, York, YO1 5DD, UK.

Based on a presentation at COMPASS '96.

[1] Throughout this paper, *interface* refers to the human-machine interface.

into three major classes. We consider these three classes separately below.

Quantifying Human Error Probabilities

The first set of techniques seek to quantify the probability that a human operator will commit a particular error when performing some procedure. This permits analysis of the risks associated with a given interface design. Perhaps the best exemplar of this class is the Technique for Human Error Rate Prediction [7]. Unfortunately, these techniques suffer from the obvious difficulty associated with any attempt to construct quantitative, predictive models of human behaviour. Also, they can only be applied after the design has been refined to quite a high level of detail. Villemeur [10 (chapter 6)] provides an overview and critique of this class of techniques.

Using Cognitive Accounts of Human Error

The second class of approaches begins by attempting to understand the cognitive behaviour of the operators of complex systems. On the basis of cognitive models, qualitative explanations of the mechanisms that give rise to human-error can be generated, and suggestions for design techniques that might avoid or compensate for these mechanisms are identified. For examples of this type of technique, see [9, 12, 8].

These approaches have the advantage that they can be used to guide the initial design of the interface. However, from the point of view of risk analysis, they are limited because:

- they do not take into account the relative severity of hazards that might be associated with the operation of the system;
- nor do they consider the relative reliability of the machine components (as opposed to human components) that contribute to the safety of the human-machine system.

Thus, whilst they are able to identify possible weaknesses in a design and may be used to suggest general improvements to the design, they are not able to support trade-off decisions between design alternatives.

Qualitative Analysis Using Multi-Disciplinary Teams

These methods involve systematic investigation of the possible consequences of human errors or component failures by multi-disciplined teams, and discussion of possible design responses. The teams include risk analysts, human factors engineers, and domain engineers (i.e., those responsible for the design of the machine elements of the system). Kirwan [6] provides a review of methods in this class. The major disadvantage of these techniques is that they are very resource intensive. Indeed, the process may be characterised as providing the interface designer with the full time services of the domain engineer and the risk analyst, whilst the design is refined, and supplying the risk analyst with the full time services of an interface designer and a domain engineer whilst the analysis is conducted. Alternatively, a team analysis could be conducted after the interface design has been developed in some detail, in which case, the early stages of design would have to proceed unsupported by risk analysis information.

The Place of Impact Analysis

We claim that analysis of impact complements these other techniques by providing the interface designer with a way of interpreting some of the information generated by preliminary risk analysis. We seek to complement existing approaches by providing a technique that:

- supports the early stages of design with quantitative information relating human-error to hazards;
- avoids the problems associated with probabilistic predictions of human behaviour;
- explicitly supports trade-off decisions within a design; and
- is not as resource intensive as a detailed team-based analysis.

QUANTIFYING IMPACT

To develop a quantitative measure of impact, we begin from the observation that risk (as used in probabilistic risk assessment) is a function of two variables: probability and severity. Rather than attempting to quantify the probability of a human-error occurring, our approach quantifies the severity that should be associated with erroneous actions or erroneous executions of procedures performed by an operator. Such a measure can be used to identify the most important areas of interaction design, and so enable interface designers to focus techniques such as those suggested in [9, 12, 8], on those areas of the design where human-error may be most significant.

Given the philosophy that no single point of failure in a system should cause a hazard, a single operator error should only ever result in an increase in the probability of a hazard. Consequently, our measure of impact is a measure of the change in risk associated with the system.

To compute the measure two assumptions are important.

Firstly, we assume that the interface design will always lag behind the design of the machine elements of the system. For example, in developing the design of a chemical plant, the conceptual layout of pipes, valves, tanks, boilers, heat exchangers, process vessels, etc., will be developed, before the interface that permits control of these components is designed. Figure 1, on next page, presents a possible ordering of activities in the early stages of a design process grouped into three categories: risk analysis activities; activities associated with the design of the machine elements of the system; and activities related

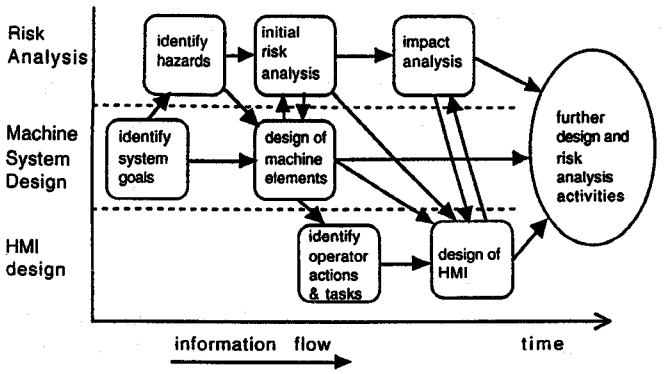


Fig. 1. Incorporating Impact Analysis into the Design Process

to the design of the interface. Figure 1 shows that impact analysis is dependent on the availability of an initial risk study of the machine elements of the human-machine system, and a specification of the tasks and actions that the operator may be able to perform.

Fault trees relating physical component failures to known hazards, output from the initial risk analysis, can then be used to measure impact. Note that these fault trees do not need to include information about human error. At this stage, the analysis should only deal with the machine components, and so should be based on (relatively) hard failure data. This contrasts with the imprecision that arises in attempts to quantity human-error probabilities.

This first assumption may not hold in some cases because of the desire to maintain consistency with previous generations of a design, which will influence operators' behaviour. However, we would argue that an analysis of impact could still be valuable to guide design modifications, particularly when new functionality is introduced.

The second assumption is that any state of a component that can be reached by normal control actions on the part of the operator, is also identified as a possible failure mode of the component.

If this second assumption is met, the state changes that the operator can effect by acting on the system can be linked to basic events of the fault trees. For instance, if the operator is able to open and close valves, and a basic event in some fault tree is the failure of some valve in the open state, then opening the valve should be associated with this basic event.

To quantity the impact of the operator's action, we consider how the probability of the root hazard of the fault tree is affected when the probability of the basic event changes from its normal value (i.e., the probability of spontaneous failure in a given time interval) to one. Impact is then a function of this probability change and the severity of the hazard. If the result of the action is a large increase in the probability of the hazard, this corresponds to a large impact. Conversely, actions that effect only as small change to the hazard probability will have a smaller impact.

In terms of risk analysis, the probabilistic element of our proposed impact measure is closely related to the Birnbaum basic event importance [3]. This value is defined by:

$$B = g(1_i, Q(t)) - g(0_i, Q(t))$$

where g(1;!(t)) is the probability of the top event of a fault tree given that event i is known to have occurred, [2] and $g(0_i, Q(t))$ is the probability of the top event given that event i is known not to have occurred. Our proposed impact measure is:

$$I = g(1_i, Q(t)) - Q(t)$$

Notice that the impact of an action is the same whether it is performed by a human operator or some other agent. This suggests that it may be useful for the designers of autonomous or semi-autonomous computer based control systems to consider the impact of the actions that the control system can initiate.

In the rest of this paper, we illustrate the computation of impact in a simple case study and discuss how impact might be used to inform interface design.

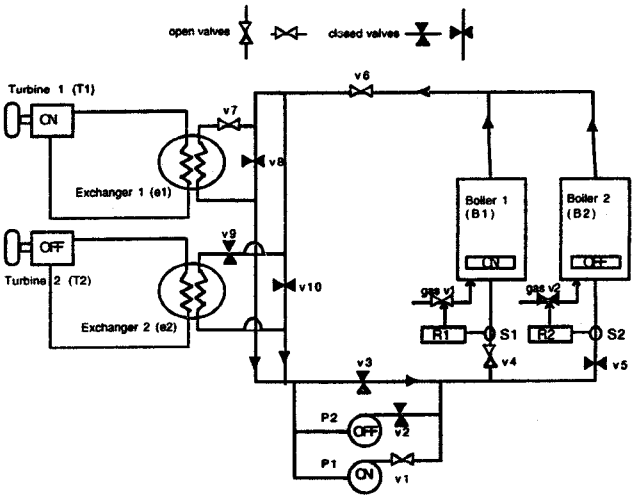


Fig. 2. A Simple Two Turbine Power Plant

AN EXAMPLE OF IMPACT ANALYSIS

Figure 2 shows a schematic representation of a power plant design. In the plant two pumps are arranged in parallel. These pumps circulate heating fluid through (one or two) gas fired boilers and then through (one or two) heat exchangers which transfer the heat to (one or

[2] Q(t) is the unavailability of a system at time t, i.e., the probability of the top event of the fault tree given no prior assumptions.

two) secondary circuits in which steam driven turbines generate electricity.

The plant has three major operating modes, namely:

- low power A — using boiler 1, pump 1, exchanger 1, and requiring valves v1, v4, v6 and v7 to be open with all other valves closed as in figure 2:
- low power B — using boiler 2, pump 2, exchanger 2 and requiring valves v2, v5, v6, v9 to be open with all others closed; and
- high power — using both pumps, both boilers, both exchangers and requiring valves v1, v2, v4, v5, v6, v7, v9 open and all others closed.

The major hazards associated with the plant are either overheating of the boilers or failure to generate electricity. To guard against overheating of the boilers, sensors monitor the flow of heating fluid into the boilers at points (s1, s2), and regulators (r1, r2) can be used to cut off the flow of gas into the boilers through the gas valves (gv1, gv2).

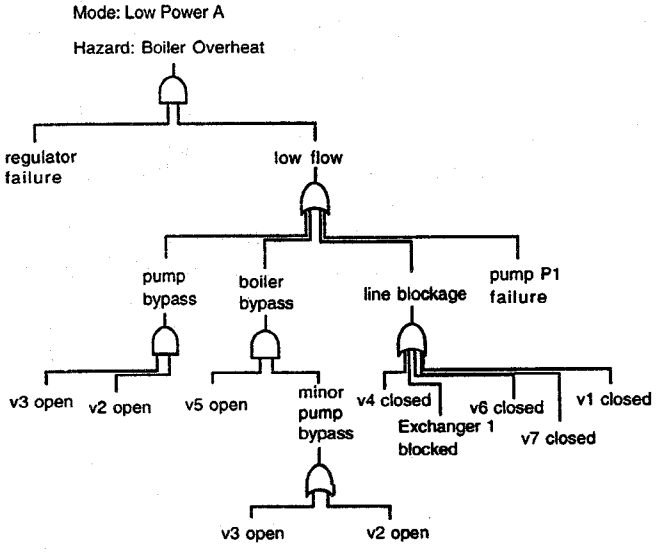


Fig. 3. A Fault Tree for the Hazard of Boiler 1 Overheating Whilst in Low Power A Mode

Impact of a Single Action

Figure 3 shows a fault tree constructed for the hazard of overheating the boiler 1 whilst the system is in the low power A mode. As we have already noted, the fault tree does not include failures that might arise as a result of human-error placing the plant in a non-standard state, merely the conditions that might give rise to boiler 1 overheating with the system in one operating mode that makes use of boiler 1. Also the tree does not analyse the different ways that the gas regulatory system for the boiler might fail, e.g., failure of the sensor, controller or gas inlet valve, but treats the regulatory system as a single component.

Analysis of the fault tree shows that the minimal cut sets[3] for the boiler overheating are:

- boiler 1 regulator failure and v6 closed;
- boiler 1 regulator failure and v7 closed;
- boiler 1 regulator failure and v1 closed;
- boiler 1 regulator failure and v4 closed;
- boiler 1 regulator failure and exchanger 1 blocked;
- boiler 1 regulator failure and v3 open and v2 open;
- boiler 1 regulator failure and v5 open and v3 open;
- boiler 1 regulator failure and v5 open and v2 open; and
- boiler 1 regulator failure and P1 failed off.

For the purposes of this example we assign (arbitrary) component failure probabilities as follows:

- Probability of failure for regulator as 0.01 failures per unit time;
- for any valve to fail open 0.01 per unit time;
- for any valve to fail closed 0.001 per unit time;
- for pump to fail off 0.02 per unit time; and
- for the exchanger to be blocked 0.0001 per unit time.

In a real analysis, these probabilities would be based on reliability statistics for similar components, collected over extended periods. This contrasts sharply with the poor quality of data available to estimate human-error probabilities.

Using these figures we can now consider questions such as "What is the impact of opening valve v5?" This action affects only two of the above minimal cut sets, namely sets (g) and (h). The probability of the hazard arising (spontaneously) from these cut sets per unit time is given by the expression:

P(b 1 regulator failure) × P(v5 fails open)
× P(v3 fails open v v2 fails open) = 0.00000199

If v5 is known to be open the appropriate expression is:

P(b1 regulator failure)
× P(v3 fails open v v2 fails open) = 0.000199

Giving an impact of 0.000197 to 3 significant figures.

- Similar computations give changes in hazard probability of:

a) switching off the pump Pl: 0.00980;

[3]Cut set analysis is a standard procedure for fault trees. A cut set of a fault tree is a set of events that is sufficient to cause the root event of the fault tree. A minimal cut set is a cut set for which any proper subset does not necessarily lead to the occurrence of the root event. Many computer packages for fault tree analysis provide for the automated identification of minimal cut sets.

b) closing any of the valves v4, v6, v7 or v1: 0.00999;
c) opening valve v2 or v3: 0.000197.

From the above calculations we may determine the relative impact of these operations with respect to the hazard of the boiler overheating.

Design Responses to Impact Assessment

Given some understanding of relative impacts an interface designer may have a number of possible responses. Here we suggest a few possible heuristics.

H1 Consider the use of guarding dialogues for high impact actions.

In our power plant, supposing that the interface is implemented using a VDU with the operator selecting components using a mouse, we might introduce an additional warning message, or require explicit confirmation if the operator attempted to close any of the valves v4, v6, v7 or v1 when boiler b1 was in the low power A mode.

H2 Make the effects of high impact actions easily perceivable.

An alternative approach might concentrate on ensuring that the interface made the effect of these high impact actions easily perceivable by the operator. This may be done by providing alarms, or by making particular pieces of information about the plant state more readily perceivable.

H3 Where high impact interactions are identified, use the internal nodes of fault trees to guide design techniques.

An important part of design techniques such as those suggested in [9, 12, 8], is the identification of the relationship between the functional purpose and abstract functions of a system and the physical actions that can be performed on the system. The generalised conditions describing the internal nodes of fault trees may provide useful information to interface designers attempting to understand these relationships.

Impact and Multiple Hazards

Many actions will have an effect on more than one hazard. In directing design resources, display resources, and deciding where guarding dialogues are appropriate, interface designers must rank the various actions and errors that might occur with respect to these multiple hazards. An action may result in a large probability increase with respect to one hazard, and a low, zero or possibly negative probability increase (making the hazard less likely to occur) with respect to some other hazard. If each action is associated with a vector of probability changes, indexed by the set of potential hazards, then the problem of impact analysis becomes one of comparing these multi-dimensional vectors.

A partial ordering of these vectors can be achieved by defining an action $A_i = (A_{i1}, A_{i2} ..., A_{in})$ to have greater negative impact than action $A_j = (A_{j1}, A_{j2} ..., A_{jn})$ (where A_{ik} is the increase in hazard probability for the k^{th} hazard when action A_i is performed) if and only each element of A_i is greater than the corresponding element of A_j i.e.:

$$A_i \sqsupseteq_{Impact} A_j \Leftrightarrow (\forall k : 1..n \bullet A_i(k) \geq A_j(k))$$

This type of ordering can be described as a "dominance" ordering. For a complex system it may be that this partial ordering becomes so flat (in the sense that few actions result in a higher probability increase with respect to all the hazards under consideration than other actions) that it provides little useful information to the designer.

Thus, we may expect that it will be necessary in most cases to investigate some parts of the ordering in more detail in order to guide the design.

One approach would be to elicit, from the risk analyst, exchange values that might indicate how much of an increase in probability in one hazard might be exchanged for a given reduction in the probability of another. Standard techniques to elicit this type of information have been developed within economic utility theory [5]. The simplest such analysis might seek to assign a numerical value for the "severity" of each hazard, and treat the impact of each action as a weighted sum of the probabilities, i.e.,

$$Impact(A_i) = \sum_{h \in Hazard} Severity(h) \times (g(1_i, Q(t)) - Q(t))$$

This simple analysis may represent an extreme position, since it forces the risk analyst to make judgments about relative severity of hazards that he or she may regard as incomparable. This can lead to ethical problems, since some hazards result in injury or loss of life, whilst others might result in financial losses. This would mean that the risk analyst's decisions would be open to interpretation as placing a monetary value on human life. Also, this approach may generate more information than is genuinely useful to the interface designer who needs to select a subset of actions and tasks for particular attention, rather than deriving a complete ordering over the actions.

Other approaches may be possible that refine the ordering of actions beyond the initial partial order; but do not necessarily generate exchange values and complete utility functions. For instance, we might start with the dominance ordering introduced above, and then ask the risk analyst to make pairwise comparisons between actions and their associated vectors. Where the analyst was prepared to make a definite comparison, the result of

these comparisons could be represented by propositions of the form:

$$(\text{action}_i, A_i) \succ_{\text{impact}} (\text{action}_j, A_j)$$

This would allow the analyst to record a definite preference between actions without necessarily leading to exchange values and a total ordering over the actions. By using this type of approach, we might avoid exacerbating the numbers game of risk analysis, at the same time as providing information to guide the design of the interface. A similar approach to event ranking is pursued by Johnson [4]. The problem of exploring such orderings falls into the general problem of multi-criteria decision making. For an introduction to this topic, the reader is referred to [11].

Impact of Errors in Procedures

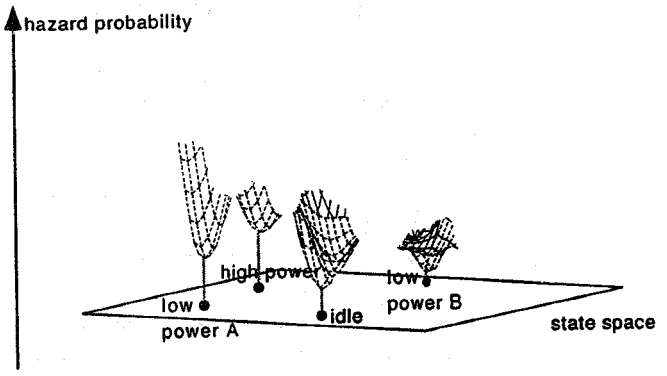


Fig. 4. Fault Trees Define a Partial Function from System States to Hazard Probabilities

By considering single actions being performed when the plant is in a particular state, we can obtain some information on the appropriateness of interlocking mechanisms and the assignment of display resources. Interface designers also need to consider the structure of tasks and procedures for operating the plant and the possible effects of errors in the performance of these procedures. In order to analyse the impact of errors in procedures, it is useful to develop our understanding of the information encapsulated in fault trees.

Fault trees provide a means by which a risk analyst can estimate, for each major operating mode of the plant, the probabilities of each known hazard occurring. Formally we can regard the set of fault trees for a system as defining a partial function from the state space of the system to the space of n-dimensional probability vectors, where n is the number of potential hazards identified for the system. If only the root probabilities of the fault trees are considered, then domain of the function is just the set of major operating modes. If, in addition, we consider variations that include faults appearing at the leaves of the fault tree, this expands the definition of the partial function to include points in the neighbourhood of the major operating modes. Figure 4 illustrates this interpretation with respect to the probabilities of a single hazard.

Procedures, when correctly implemented, should move the system from one operating mode (the starting mode) to another (the target mode). Errors in procedures, or previously introduced latent errors, will result in the system reaching a different state in the neighbourhood of the target mode. Therefore, the impact of an error in a procedure can be measured by the difference between the hazard probability vector for the target mode, and the hazard probability vector for the mode that is reached. We are currently investigating the use of formal software engineering models as design representations for the interface. If such models can be used to predict the state that will be reached as a result of a given error, then information about the impact of such errors could be made available to designers using the models. For a description of work using formal models of the interface to predict the effects of error, see [2, 1].

DISCUSSION AND FURTHER WORK

In this paper we have described an approach to human-machine interface development that seeks to provide the interface designer with information about the relative impacts associated with erroneous performance of actions or procedures by an operator. This information may be particularly significant where automated support enables operators to effect major changes on the system by means of a small number of actions.

However, the work presented is at an early stage of development, and a number of issues need to be addressed before such a technique could be taken out of the laboratory and used in an industrial setting. In particular:

- The technique assumes that the set of failure states used to construct the fault trees is a superset of the states reachable by operator control actions on the system. Exclusion of a state from a fault tree is interpreted by this technique as implying that the state has no effect (or a negligible effect) on the probability of the root hazard. If the operator can perform actions that have not been considered in constructing the fault trees, then the impact of these actions will be incorrectly assessed as zero. This assumption needs to be investigated further to discover whether it is violated in practice, and whether the technique could be adapted to deal with cases where it is violated.
- Techniques need to be developed to elicit judgments from risk analysts and safety engineers about the relative impacts of different actions in a form that is useful to interface designers. We hope to adapt techniques from decision theory such as those described in [11, 5].

- The technique, as it stands, does not address temporal issues that may affect the safety of systems. For instance, what the impact of some action being performed too early or too late might be. This is a major limitation for practical application to complex systems. We are currently investigating these issues.

ACKNOWLEDGMENTS

The authors wish to thank Paul Zanelli of the Department of Computer Science, University of York for his assistance in generating Figure 4. This work is funded by the UK Engineering and Physical Sciences Research Council, Grant Number GR/J07686.

REFERENCES

[1] A.M. Dearden and M.D. Harrison, 1996,
Risk Analysis, Impact and Interaction Modelling,
in F. Bodart and J. Vanderdonckt, editors,
Proceedings of DSVIS 96, Springer.

[2] R. Fields, P. Wright and M. Harrison, March 1995,
A Task Centered Approach to Analysing Human Error Tolerance Requirements,
in P. Zave, editor,
Proceedings, RE '95,
The Second IEEE International Symposium on Requirements Engineering,
York, UK,
pp. 18-26,
IEEE, New York.

[3] E.J. Henley and H. Kumamoto, 1981,
Reliability Engineering and Risk Assessment,
Prentice Hall.

[4] C.W. Johnson,
Documenting the Design of Safety-Critical User Interfaces,
Interacting with Computers,
to appear.

[5] R.L. Keeney and H. Raiffia, 1976,
Decision With Multiple Objectives,
Prentice Hall.

[6] B. Kirwan, 1992,
Human Error Identification in Human Reliability Assessment,
Part I: Overview of Approaches,
Applied Ergonomics, 23(5):299 - 318.

[7] A.D. Swain and H.E. Guttman, 1983,
Handbook of Human Reliability Analysis with Emphasis on Nuclear Power Plant Applications,
Final Report,
Technical Report NUREG/CR- 1278 SAND80-0200 RX, AN,
U.S. Nuclear Regulatory Commission.

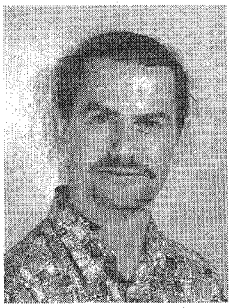
[8] D.A. Thurman and C.M. Mitchell, 1995,
A Design Methodology for Operator Displays of Highly Automated Supervisory Control Systems,
in *Proceedings of the 6th IFAC/IFIP/IFORS/SEA Symposiums on Analysis, Design and Evaluation of Man Machine Systems.*

[9] K.J Vicente and J. Rasmussen, 1992,
Ecological Interface Design: Theoretical Foundations,
IEEE Trans. on Systems, Man and Cybernetics, 22(4):589 - 605.

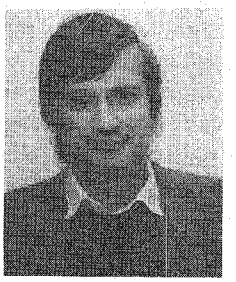
[10] A. Villemeul, 1992,
Reliability, Availability, Maintainability and Safety Assessment,
Volume 2,
John Wiley.

[11] P. Vincke, 1992,
Multicriteria Decision-Aid,
John Wiley and Sons.

[12] D.D. Woods and E. Hollnagel, 1987,
Mapping Cognitive Demands in Complex Problem Solving Worlds,
Int. Journal of Man-Machine Studies, 26:257-275.



Andrew Dearden obtained his bachelors degree in mathematics at Durham in 1980. After working as a teacher for some years, he obtained a masters in computer science at the University College London in 1990, and a doctorate at York in 1995, on the application of formal models in the design of interactive case memory systems. His current research objectives are to develop methods for integrating risk analysis activities with human-machine interface design in complex systems, and the design of interactive systems to support multi-criteria decision making.



After a doctorate at the Programming Research Group in Oxford, **Michael Harrison** spent about ten years in industry with a software consultancy and a semiconductor company: designing banking systems (Midland Bank), process monitoring systems (British Steel Corporation), communication systems (Reuters) and advanced programming environments (Inmos). Since moving to York in 1983, he has researched the specification and modelling of interactive systems with a particular emphasis on the extent to which design can draw upon an understanding of the user's view of the system. His current research objectives are to produce methods of designing and evaluating interactive systems based on mathematical techniques, and to develop analysis techniques and design methods that address issues of human dependability in complex systems.