CREWS Report 98-34

submitted to

Special Issue on Interface Issues and Designs for Safety-Critical Interactive Systems of the ACM Transactions on Computer - Human Interaction.

A Causal Model of Human Error for Safety Critical User Interface Design

by

Julia Galliers, Shailey Minocha and Alistair G. Sutcliffe

Centre for HCI Design
School of Informatics
City University
Northampton Square
London EC1V 0HB
United Kingdom

+44 (0)171 477 8469
jrg@csr.city.ac.uk
{s.minocha, a.g.sutcliffe}@city.ac.uk

A Causal Model of Human Error for Safety-Critical User Interface Design[1]

Julia Galliers, Shailey Minocha and Alistair Sutcliffe

Centre for HCI Design
School of Informatics
City University
Northampton Square
London EC1V 0HB
United Kingdom
E-mail: jrg@csr.city.ac.uk
Tel: +44-171-477-8469
Fax: +44-171-477-8859

ABSTRACT

This paper describes a method of assessing the implications for human error on user interface design of safety-critical software. In previous work we have proposed taxonomy of influencing factors that contribute to error. In this paper, components of the taxonomy are combined into a mathematical and causal model for error, represented as a Bayesian Belief Net (BBN). The BBN quantifies error influences arising from user knowledge, ability and the task environment, combined with factors describing the complexity of user action and user interface quality. The BBN model predicts probabilities of different types of error, slips and mistakes, for each component action of a task involving user-system interaction. We propose an Impact Analysis Method that involves running test scenarios against this causal model of error in order to determine those user actions that are prone to different types of error. Applying the proposed method will enable the designer to determine the combinations of influencing factors and their interactions that are most likely to influence human error. Finally we show how such scenario-based causal analysis can be useful as a means of focusing on specifically relevant guidelines for safe user interface (UI) design. In the paper the proposed method is demonstrated through a case study of an operator performing a task using the control system for a laser spectrophotometer.

1 INTRODUCTION

This paper introduces a method to assist with the identification of requirements for the design of software intensive safety critical systems. Of particular interest is the fact that most safety critical systems involve automated software control and human operation in a social context. Many spectacular system failures are caused by human and user interface design errors rather than failure in software functioning, e.g. the much publicised London Ambulance Service and Therac-25 accidents [Leveson &

[1] This research has been funded by the European Commission ESPRIT 21903 'CREWS' (Co-operative Requirements Engineering With Scenarios) long-term research project.

Turner 1993; Leveson 1995] were attributable to poor operator UI design as well as unreliable control software.

Risk assessment methods such as HAZOP, Failure Modes and Effects Analysis (FMEA), etc. have been adapted for a design orientation [Earthy, 1995], but these methods do not deal with complex causes of human error [Hollnagel, 1993]. Human factors are considered as influences on failure modes but the approach to predicting possible errors is still crude and takes little account of cognitive psychology or social causes of failure. Several researchers, notably Reason [1990] and Hollnagel [1993] have called for a more systematic and theoretically grounded approach to safety design for human operation. Design methods need to account for potential human error and its causes in poor human computer interface design, operator training, as well as the safety culture of the organisation.

In our previous work, [Sutcliffe & Minocha 1998, Sutcliffe et al., 1998a] we have proposed a taxonomy of influencing factors that might contribute to human error. This paper shows how components of the taxonomy can be combined into a causal model for error, represented as a Bayesian Belief Net (BBN) and used to model the error influences arising from user knowledge including safety awareness, ability, and the task environment. These are combined with factors describing the complexity of user action and UI quality in many different test scenarios of projected system usage. The BBN model predicts probabilities of errors as slips and mistakes for each individual action within the task.

Probabilistic assessment techniques for human error are not new. THERP (Technique for Human Error Rate Prediction, [Bell and Swain, 1985] has been widely used to determine the probabilities of paths and path segments through an event tree, as successful or not. Combinations of probabilities are simply arrived at by multiplying the individual probabilities together. More rigorous Bayesian methods for combining probabilities are currently becoming more common in software reliability research and safety-critical system design generally, but rarely with respect to determining predictions for human error rates. One exception to this is the work of Phillips and Humphreys [1990] in which influence diagrams (Bayesian networks augmented with decision variables and a utility function) are applied to a study of human reliability in the context of pressurised thermal shock events for two nuclear power stations in the United States.

The method described in this paper aims to go further than making predictions of error rates. Firstly, human errors are distinguished as mistake-errors and slip-errors [Reason 1990]. The predictions of these two error types are related, via our BBN model in conjunction with precise scripts of user tasks - both normal course and in the event of failure - to each particular action within a task. The method also makes error predictions for each of those particular actions according to different test scenarios in which operators with varied skills operate with, for example, high or low levels of stress and varying levels of motivation and time constraints. Finally, the method then directs the analyst or designer to *particular* guidelines for safe UI design based on the action-type, the nature of its safety-criticality e.g. whether the risk is of personal injury or some other type, and whether the requirement be to eliminate, reduce or control the risk of these human errors. These guidelines are a refinement of the guidelines for safe HMI design proposed by Leveson [1995].

The paper is organised in five sections. We start in section 2 with the background. Section 2.1 describes the different error types. Section 2.2 gives an outline of the previous work from which the generic influencing factors were identified, and finally section 2.3 offers a short description of the nature of BBNs. Section 3 then introduces the Impact Analysis Method for assessing the possible impact of human error on user interface design for safety-critical software. Section 3.1 provides an overview of the three phases of the method - causal, consequence and design analyses. Section 3.2 then describes the causal analysis phase for predicting probabilities of mistake-errors and slip-errors for individual actions within a task, given a variety of test scenarios; Section 3.3 follows with the consequence analysis, and section 3.4 describes the design analysis phase in which the framework of design guidelines is employed. Here Leveson's original guidelines for safe HMI design are constrained and associated with particular action-types. In section 4, we illustrate the use of the method with a case study. This employs data from a task and hazard analysis previously gathered and described in Sutcliffe, [1998] concerning the control system of a laser spectrophotometer. The paper concludes with a discussion in section 5.

2 RELATED WORK AND BACKGROUND

2.1 Types of Human Error

Following a large body of research on human error by Reason [1990], for the method described and illustrated in sections three and four of this paper, we distinguish between two different types of error: *slips* and *mistakes*.

Slips (and lapses) are skill-based errors that happen when an action is incorrectly performed, frequently during familiar work requiring little attention. In our view, tasks of physical complexity, such as complex manipulations involving precise movements and detailed co-ordination are more prone to slip-errors. Mistakes, on the other hand, are rule-based or knowledge-based errors and associated with problem solving. Rule-based mistakes can occur by the application of "bad" rules or the misapplication of "good" rules; knowledge-bound mistakes are rooted in bounded rationality, incomplete or inaccurate knowledge. In this paper, tasks of high cognitive complexity are considered more prone to mistake-errors.

2.2 Influencing Factors

In previous work, [Sutcliffe & Minocha 1998, Sutcliffe et al, 1998a, Sutcliffe et al, 1998b] a method is described known as CREWS1

[1 CREWS (Co-operative Requirements Engineering with Scenarios) is an ESPRIT 21903 Long Term Research Project being carried out at Centre for HCI Design, City University, London, UK.]

-SAVRE (Scenarios for Analysis and Validation of Requirements) which has a primary objective to analyse design requirements arising from the dependencies between a computer system and its environment. Secondly, it allows explicit analysis of design requirements arising from the consequences of failure and human error, and thirdly, it provides high-level guidance on reasons why errors might occur as well as suggesting generic requirements to deal with the causes and consequences of error. Generic requirements are reusable and can be used as an agenda of issues pointing towards areas for further requirements investigation, or requirements that can be refined with more domain-specific knowledge, or as high level design solutions that may be added directly to a requirements specification.

An important component of the CREWS-SAVRE method is a taxonomy of *influencing factors* that describe the necessary preconditions for errors to occur. These influencing factors are grouped into four categories:

- environmental conditions,
- management and organisational factors,
- task/domain factors and
- User/personnel qualities.

Each affect different human internal variables such as fatigue, stress, workload and motivation. These in turn, affect the probability of human errors, manifest as slips and mistakes.

In the Impact Analysis Method proposed in this paper, we deal with only three layers of influencing factors - environmental conditions, the task/domain and user/personnel. The reasons for this are to reduce the complexity of the BBN model for this initial analysis, which we anticipate will be extended and improved in further iterations.
Domain facts that describe the working environment, the people who will operate, control and manage the system are used to capture domain-specific influencing factors. Such 'domain scenarios' can either be taken from real-life by observations or by interviewing the users, or postulated to cover a variety of organisational and work situations that may occur in the domain. The implications of user factors are summarised in Table I. Some of these factors can be measured objectively by using psychological questionnaires. For instance general ability and accuracy/concentration can be measured by intelligence aptitude scales, decision making and judgement by locus of control scales, while domain and task knowledge can be measured by creating simple tests for a specific task/domain. The main implications of the personnel factors are for personnel selection and training while generic requirements indicate the need for computer based intelligent assistants, critics, and aide memoir information displays.
Table I User / Personnel Factors with problems for slip and mistake type errors and generic requirements to counteract these problems

Personnel Qualities	Skilled Task Problems	Decision / Problem Solving Tasks	Implications / Generic Requirements
General Ability	More slips especially complex tasks, longer learning time	Inability to deal with unexpected events	Personnel selection appropriate for task / job description
Knowledge of domain	More slips, longer learning time	Slow performance, failure in complex problems	Improve training on the job experience, scenario-based training
Skill / task knowledge training	Slow operation, more slips	Mistakes, slow performance	Improve training, help manuals, aid memoirs, mentors
Judgement / decision-making	-	poor initiative, more mistakes, wrong decisions	Select personnel appropriate for task
Concentration / accuracy	More slips, more ordered events	more mistakes, poor decisions, poor checking	Discipline and training to improve performance, select appropriate personnel
Motivation	More slips, slow operation, short cuts	Increase mistakes, slow performance, short cuts	Improve incentives, job satisfaction, select appropriate personnel

The impact of poor personnel factors will be worse for skilled performance and action-slip errors, as shown in column 2, while for tasks that require decisions and problem solving there will be worse performance and more mistakes. These are illustrated in column 3, with generic requirements and recommendations for training listed in column 4. In addition, user personnel factors are used with assessment of task complexity (see Table II), to match users' abilities to tasks of appropriate complexity, while also giving people sufficient challenge and responsibility in their job.

Table II Task / Domain factors with implications for errors and generic requirements

Task factor	Implications	Requirements / Training issues
High Volume	Delays, bottlenecks, performance degrades, fatigue	Buffer & smoothing workloads, automate if possible, design breakpoints, batch schedules
Complexity	User fatigue and stress, mistakes, problem solving failure	Match complexity to user abilities and experience, decompose task, simplify procedures
Repetitiveness	Boredom, poor attention, slips, missed events	Automate if possible, provide task variety, swap operators frequently
Interruptions	Attention slips, missing and mal-ordered events, capture errors	Provide aid memoirs, agendas, status / progress checks, screen-out unwanted events

Time pressure	Late events, slips, capture errors, omissions	Smooth event arrival, automate task, give user time to think, provide holding actions
Multitasking / task switching	Attention slips, missing and mal-ordered events, losing thread problems	Aid memoirs, agenda managers, clear mode / status indicators, schedule switching if possible

Task/domain factors are elicited by questions about the task complexity and environmental constraints on users. As with the other factors, the complete documentation of the method [Sutcliffe & Minocha 1998] provides a comprehensive list of questions and elicitation techniques. Implications for task/domain factors for user performance and errors are listed in column 2 of Table II, with generic requirements and user training recommendations in column 3. Increases in interruptions, and more task switching make slip type errors more likely, higher volumes and time pressures also lead to more slip-errors and delays. Complexity and repetitiveness, on the other hand, have implications for task allocation and increased mistakes if users are not given tasks that match with their abilities, or training to carry out their allocated tasks. It should be noted that many task performance problems also have 'knock-on' effects on users through increased levels of stress and fatigue.

Table III Environmental Factors

Factor	**Effect on User / Operator**	**Potential Errors - Users**	**System Failures**	**High level requirements**
Temperature too high / too low	Fatigue, discomfort, stress	Concentration fails, attention slips	Failure or degraded action	Air conditioning
Humidity excessive	Fatigue, discomfort, stress	Concentration slips, attention fails	Failure or degraded action	Air conditioning
Poor visibility	Eye strain, fatigue, stress	slips reading data, transcription errors	only applies to optical scanners	light screens, blinds, better VDU luminance
High noise levels	Audio communication difficult, stress	Misunderstandings, speech messages not heard, slips	audio output interference	sound insulation, noise protection for users
Excessive vibration	Visual communication impaired, stress	Misreading of visual displays, slips	Failure if not robust	Isolate and insulate from vibration

Dust and dirt	Discomfort, poor motivation	Concentration fails, slips	degraded action, mechanical failure	controlled environment, sealed under protective clothing

The implications for environmental factors such as adverse temperatures, humidity, dust levels etc. on human errors and system safety are summarised in Table III. Adverse environmental factors increase discomfort and stress for users, these in turn will lead to more skill-based slip errors when the operator's concentration fails. Moreover, when it is not a routine human operation, that is it involves decision-making or problem solving, adverse environmental factors will lead to more mistakes and degraded performance.

2.3 Bayesian Belief Network Analysis

In this section, we introduce BBNs as a means of combining the influencing factors described above in section 2.2 into a more formal and predictive model of human error.

BBNs are graphical networks that represent probabilistic relationships between variables. They offer decision support for probabilistic reasoning in the presence of uncertainty and combine the advantages of an intuitive representation with a sound mathematical basis in Bayesian probability [Pearl 1988]. BBNs are useful for inferring the probabilities of events which have not as yet, been observed, on the basis of observations or other evidence that have a causal relationship to the event in question. For example, a doctor might have observed a variety of symptoms in his patient. Using a BBN, s/he can determine the probabilities that these symptoms are caused by each of the several possible alternative diseases. From this the probabilities of then further complications can be predicted.

A BBN is made up of *nodes* and *arcs*. The nodes represent variables and the arcs represent (usually causal) relationships between variables. The example in Figure 1 is a fragment of the net described in more detail in section 3 of this paper. It shows the complexity of an action as a variable that is affected causally by two factors - the level of physical detail and the cognitive complexity of the task. Variables with either a finite or an infinite number of states are possible in a BBN, so the choice of measurement scale is left to the analyst's discretion. As demonstrated in the case study (see section 4), we generally assign these variables to one of the three possible states: *high, medium* or *low*.

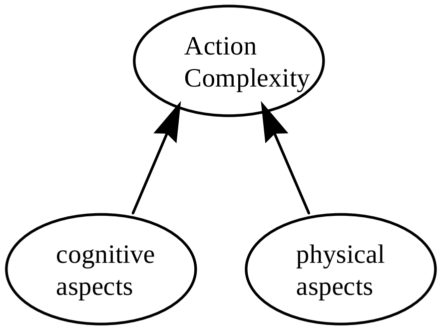


Figure 1 Simple BBN fragment, the two prior nodes have a causal influence on the posterior node of action complexity

In the above fragment, if we know that both the cognitive and physical aspects of a particular action are high, then the probability of the overall complexity being high is greater than if we know the action has a low level of physical detail and involves little cognitive ability. In the BBN we model this by filling in a node probability table (NPT). Table IV shows the NPT for the *action complexity* node.

Table IV **Node Probability Table (NPT) for the *action complexity node***

	Cognitive	high			medium			low		
	Physical	high	medium	low	high	medium	low	high	medium	low
action complexity	high	0.9	0.7	0.6	0.6	0.5	0.3	0.3	0.2	0.0
	medium	0.1	0.2	0.3	0.3	0.2	0.3	0.3	0.2	0.1
	low	0.0	0.1	0.1	0.1	0.3	0.4	0.4	0.6	0.9

Nodes with incoming arcs are associated with a table of conditional probabilities as shown in Table IV. Each arc and tables of conditional probabilities represent knowledge about one node that is useful for predictions about another node. Hence in Table IV column 1 the requirements engineer has asserted that if cognitive aspect of the action is high and the physical detail of the action is high then the probability of overall action complexity being high is 0.9, and medium 0.1 with a zero probability of being low. The table is configured by estimating the probabilities for the output variables by an exhaustive pairwise combination of the input variables. BBNs can accommodate both probabilities based on subjective judgements (elicited from domain experts) and, probabilities based on objective data.

When the net and NPTs have been completed, Bayes theorem is used to calculate the probability of each state of each node in the net. The result of this calculation is a probability distribution for the states of each node. Then, when evidence is available to determine the states of particular nodes from particular scenarios, the values entered are propagated through the network, updating the values of other nodes. These calculations are entirely automated by a BBN tool - *Hugin Explorer* (see Hugin A/S web site, *www.hugin.dk*). The result is a network from which predictions can be made regarding the probability of certain variable(s) being in particular state(s), given the combination(s) of evidence entered.

Section 3 presents an example BBN, which models a set of influencing factors. How these factors impact in combination upon the probabilities of mistake-errors and slip-errors is discussed.

3 THE IMPACT ANALYSIS METHOD

3.1 Overview of the Method

The Impact Analysis Method comprises the following stages:
A. Causal Analysis

- Use the results of the task analysis and risk assessment methods such as hazard analysis in conjunction with the set of generic influencing factors in order to select a set of influencing factors that may influence the occurrence of human error for the domain in question;
- Derive a mathematical model using BBNs which represents the probabilistic relationships between these influencing factors and their effect on predictions of human error rates, distinguishable as mistake-errors and slip-errors ;
- Run numerous test scenarios using the BBN model with the aid of a BBN tool, in order to make predictions of mistake-errors and slip-errors for each individual action comprising a particular task in the domain, given varying aspects of the user, the environment and the UI design;

B. Consequence Analysis

- Match each individual action of the user's task against a taxonomy of action-types ;
- Assess the nature of safety-criticality for each individual action of the task ;

B. Design Analysis

- Map each individual action against the proposed framework of guidelines for safe UI design in which action-types are matched against appropriate design guidelines for the domain categorised with respect to elimination, reduction or control measures ;
- Derive specific design requirements for each action in the task based on the results of the causal analysis for different test scenarios and for the nature of the action's safety-criticality assessed during consequence analysis.

The Causal, Consequence and Design Analyses comprising the Impact Analysis Method outlined above are described in more detail in the sections 3.2, 3.3 and 3.4 respectively. These analyses are illustrated in the case study in section 4.

3.2. Causal Analysis

A domain specific BBN model is created by selecting a sub-set of the influencing factors described in section 2.2 that apply to the domain and application under consideration. Many combinations of influencing factors are possible and each domain requires a particular model. In Sutcliffe et al [1989b] we have suggested a general modelling tool that can be instantiated with domain specific information. Figure 2 shows the BBN model of the influencing factors which were determined following a collection of domain facts gathered in domain scenarios, as well as task and hazard analyses for our case study domain, as described in section 4.

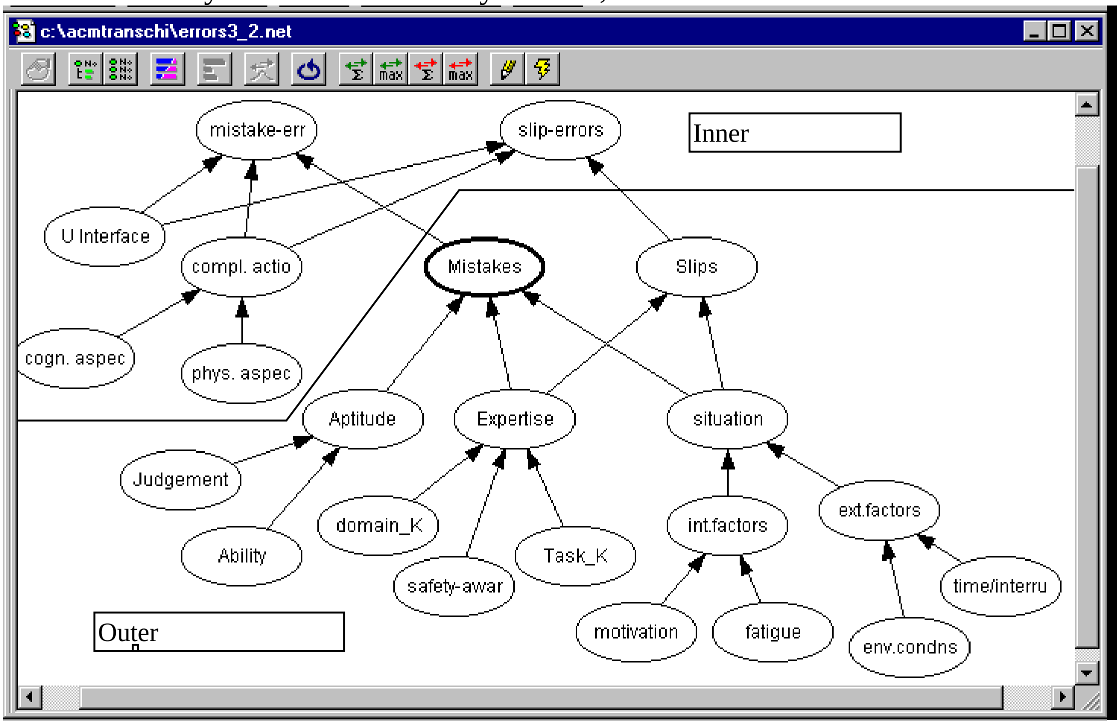


Figure 2 A BBN network for Error Analysis

The BBN is configured so that the "situation" comprising the relationships between external and internal factors on the operator contributes to slips more strongly than to mistakes. User knowledge, affected both by domain and task sub-sets as well as level of safety awareness, contributes to both types of errors but more strongly to mistakes. General ability, caused by combinations of aptitude and judgement, influences primarily mistakes. These aspects of the BBN that model general properties of the user and task context, make up the *outer net*. The outer net merges with the *inner net* that describes design qualities of the user interface, and complexity of each user action in the scenario. User actions are characterised according to their cognitive complexity and physical operational detail. This combination of inner and outer factors affect the probability of human errors as either mistakes or slips.

The BBN therefore models all the generic user factors listed in Table I of section 2.2 apart from concentration which is specific to individuals and tasks, so it is difficult to estimate. From the task/domain table (Table II) in section 2.2, multitasking has been merged with interruptions because both disrupt the normal flow of work; repetitiveness and volume have not been used as these were judged to be less important influences on error. Complexity is modelled at the action rather than task level. We have also limited the environmental factors listed in Table III to the most significant factors of temperature, humidity and dust levels (see Figure 5 in section 4.2).

The BBN model, once configured, is run against a task script of a particular user's behaviour with the system. In order to do this, each input node is set to one of its alternative states, in this case high, medium or low. For example, of the outer BBN variables, an able user will have the nodes - *judgement* and *ability* set to high. *Domain knowledge* may be low but *task knowledge* and *safety awareness* high in a user familiar with technical and safety procedures in his workplace, but unfamiliar with the technical background. *Motivation, fatigue* and level of *time/interrupts* can also be set to high, medium or low. *Environmental conditions* can be set to good, average or bad. For the inner BBN variables, the following estimates have to be made for *each* action comprising a particular task:

(a) The action has to be assessed as high/medium/low *cognitive complexity*. Any action requiring complex decision-making, problem solving or judgement is rated as high complexity. Conversely simple physical actions and computer I/O are rated low.

(b) *Physical complexity* is rated for each action-step. Complex manipulations involving precise movements and detailed co-ordination are rated high, whereas single action-step, discrete actions (e.g. stop/start machine) are rated low.

(c) If the action involves a human computer interface then its usability is assessed (*U interface* node). Measures for usability can be acquired from evaluation of users' observed problems. Alternatively, usability can be rated by answering the following questions, which also point to generic requirements for the user interface:

- Does a command or function exist for the user to achieve their goal? If not, then a missing requirement is indicated.
- Can the used find the command or action to achieve their goal? If not, then the menus or the user interface metaphor needs attention
- Is it clear how to carry out the action with the computer? If it is not, then the prompts, cues and user interface metaphor need to be improved.
- Can the user perceive and understand the effect of their action? If not, then feedback should be added or made clearer.

Usability evaluation is a complex subject and the above questions provide the minimum guidance, for more details the reader is referred to [Sutcliffe et al., in press] or [Monk et al. 1993]. If a prototype of the UI design does not exist the usability score is set to medium for all potential human computer actions, and low (i.e. no design problems) when actions are unlikely to involve human computer interaction.

As the outer and inner BBN variables are set, the BBN tool automatically calculates the propagation of probabilities throughout the network. In our example BBN in Figure 2, this gives a prediction of the probabilities of there being high, medium or low numbers of mistake-errors (and similarly for slip-errors) for each action and under the particular set of circumstances modelled under each test scenario.

Note that some of the influencing factors representing the user and task/domain can be set as constants in the application. For example, in our case study in section 4, the user being modelled is assumed to have little domain knowledge. The value of the *domain* *knowledge* node was therefore set to low for all test scenario runs. Other aspects of the user were varied however, as were all aspects of the environment. The quality of the user interface was set to medium. Mistake-error and slip-error probabilities for each action and for the different test scenarios were then compared.

3.3 Consequence Analysis

In safety-critical contexts, tasks can either be routine, that is, the day-to-day operations which would inevitably include aspects such as monitoring and controlling potentially hazardous situations, or they may involve some complex decision making for some abnormal course of events. The latter would include analysis and diagnosis of what might have gone wrong as well as some kind of recovery/repair procedures [Rasmussen et al, 1994]. We have devised taxonomy of the different action types of relevance to safety-critical tasks:

- Physical operation
- Detection
- Entering/editing data
- Situation analysis and diagnosis
- Goal evaluation and priority setting
- Activity planning
- Monitoring and verification of action
- Repair.

Furthermore, for the causal analysis above, these actions are distinguished according to the relative cognitive and physical complexity as discussed above. For example, ‘analysis and diagnosis’ involve problem solving and are highly cognitive in nature. Entering data on the other hand, is a physical activity involving, for example, keyboard skills, but editing data may also involve some cognitive aspects depending on the nature of the task and the data.

Each action analysed during causal analysis is characterised as one of the above types as a preliminary to the design analysis described in the next section. Next, for consequence analysis, the effects of human error on the performance of each of these types needs to be considered in terms of the nature of safety-criticality of that action type. For example, whether incorrect or non-performance may cause personal injury to the user or operator, or whether the risk is of injury to others, or whether the risk is to the accuracy of the activity (data) being carried out (which may then have safety-critical implications) and so on. This has implications for the design decisions made in the next phase.

3.4 Design Analysis

The aim of the design analysis stage is to propose UI design requirements to reduce the probabilities of human error predicted by the causal and consequence analyses. Casual and consequence analyses, as discussed above, are used to identify the high-risk actions and potentially safety-critical human errors. These analyses are not

enough to make a system safe; rather the information obtained in these analyses needs to be used in the design of the UI. Therefore, safety should be designed into the system. The goal of safe UI design is to eliminate the likelihood of human error or, if that is not possible, to reduce the associated risk to an acceptable level.
A design can be made safer by:

(i) Eliminating the likelihood of human error occurring;

(ii) reducing its likelihood;

(iii) Controlling human error and reducing the likelihood of it being the cause of an accident.

Thus the design principles or measures that we have proposed fall into three design strategies of elimination, reduction and control, as listed in Table V. Table V presents a framework of generic recommendations for identifying and incorporating human factors requirements into the UI design in such a way that it improves the overall safety of the system. It is too bold an objective to produce a comprehensive catalogue of UI design guidelines for safety-critical systems. Instead, as Table V indicates, our aim has been to structure, using Leveson's guidelines [1995], the guidelines within a framework of categories of principles (see Appendix I for a complete list of these principles and guidelines under their category headings). Here, it needs to be emphasised that usable UI's may not necessarily be safe. In fact, usability and safety frequently conflict. So the framework has been populated by design guidelines for safe systems which may not perhaps yield usable interfaces.

The proposed guidelines are 'generic' and context-free in nature, and are intended to be generally applicable across a wide number and variety of design problems. Also, note that these guidelines are not requirements (which are more specific in nature), but are used to *derive* design requirements in conjunction with the results of domain-specific causal and consequence analyses.

Table V Design Strategies

Elimination	**Reduction**	**Control**
1. <u>Task Conformance</u> a. Task Completeness b. Information Adequacy	1. <u>Observability</u> a. Operation Visibility b. Browsability c. Persistence	1. Containment
2. <u>Simplicity</u>	2. <u>Learnability</u> a. Familiarity b. Predictability	2. <u>Recoverability</u> a. Reachability b. Forward Error Recovery c. Backward Error Recovery
<u>3. Similarity and Dissimilarity</u>		<u>3. Feedback</u> a. Feedback on user actions b. Feedback on errors
<u>4. Location Compatibility</u>		4. <u>Responsiveness</u>
<u>5. Flexibility</u>		<u>5. Warning</u>
<u>6. Task Migrability</u>		<u>6. Robustness</u>
<u>7. Commensurate Effort</u>		

Design guidelines for human factors have been criticised for their limited use, as they do not capture the rich sensitivity of the context that is required for effective design [Rasmussen et al. 1994, Gould 1988]. To overcome this criticism and to increase the utility of the proposed guidelines in Table V and Appendix I, we have in Table VI mapped the different action types listed above in section 3.3 against the design guidelines, for the domain example illustrated in our case study. Depending upon the user task and the actions involved in performing that task, Table VI aims to guide the designers in identifying the corresponding design measures for elimination, reduction, or control of human error and its consequences. It is very important that these guidelines are not applied uncritically. The aim of this stage of the Impact Analysis Method is to highlight the interpretation of guidelines within the context of the system in question; their usage needs to be mediated by good judgement.

Table VI Task Characteristics and Design Measures

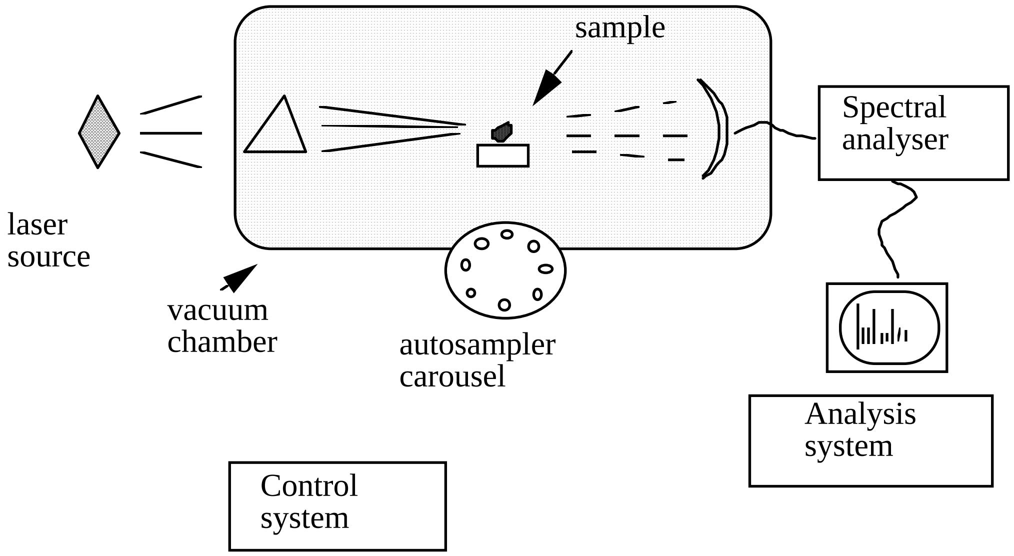
Action types	**Primary Design Measures**
Physical operation	location compatibility, simplicity, similarity and dissimilarity, commensurate effort, operation visibility, persistence, containment, task completeness
Detection (to detect and identify the occurrence of an off-normal condition, or an unsafe act)	operation visibility, persistence, warning, responsiveness, information adequacy, task completeness, browsability, predictability
Entering/editing data	feedback on user actions, operation visibility, information adequacy
Situation analysis and diagnosis	browsability, feedback on errors, operation visibility
Goal evaluation and priority setting	operation visibility, browsability, predictability, information adequacy
Activity planning	information adequacy, browsability, familiarity
Monitoring and verification of actions	operation visibility, persistence, feedback, warning
Repair	task migrability, reachability, forward error recovery, backward error recovery, robustness, information adequacy

4. CASE STUDY

4.1 Introduction

The case study system is a control system for a laser spectrophotometer. For this case study the task analysis, hazard analysis and gathering of domain facts and scenarios is reported in detail in [Sutcliffe 1998]. Here we describe how we have used this information to refine the generic influencing factors described in section 2.2 into a set of specific influencing factors from which the BBN shown in Figure 2 was created.

The laser spectrophotometer is a scientific instrument that analyses the visible spectra that are created by the laser ionisation of a chemical sample. In normal operation the laser should emit a high-energy light beam that strikes the chemical sample causing it to emit light energy that is detected by the sensor. The light spectrum is analysed and results displayed on the computer VDU. The spectra have a characteristic pattern for each chemical. The physical system model is illustrated in Figure 3. Each laser emission cycle is controlled by software. There are two sub-systems: one controls the operation of the laser, while the second detects and analyses emitted spectra. The controlled system operational sequence is shown in Figure 3 below the model.



Laser-spectrophotometer

System operations

install sample

set control parameters.... power, expose time

excite laser

detect emitted spectra

analyse spectra

remove sample

print/display result

Figure 3 Physical Model of the Spectrophotometer System.

Three types of system user were identified. First are expert users, usually research scientists who possess considerable domain expertise and become skilled in system operation i.e. task knowledge. They require the ability to conduct complex analyses, with full control of the system for different analyses, and presentation of complex results. The experts' safety awareness is variable as not all laboratories have good safety practice. The second group of experts is chief technicians who plan analyses in public laboratories and companies. Accuracy and reliability of the results are at a premium. These users need support for planning sessions and investigation of the results. Their safety awareness is generally good.

The third user group is skilled laboratory technicians whose knowledge of the domain and system operation will be average or low. From their viewpoint system operation is a routine job. In multiple runs they may make errors from lack of attention and boredom. Their safety awareness depends on training. In this case study we focus solely on the third user group, hereafter referred to as the operator, and the activity of planning a sample run which is shown in Figure 4.

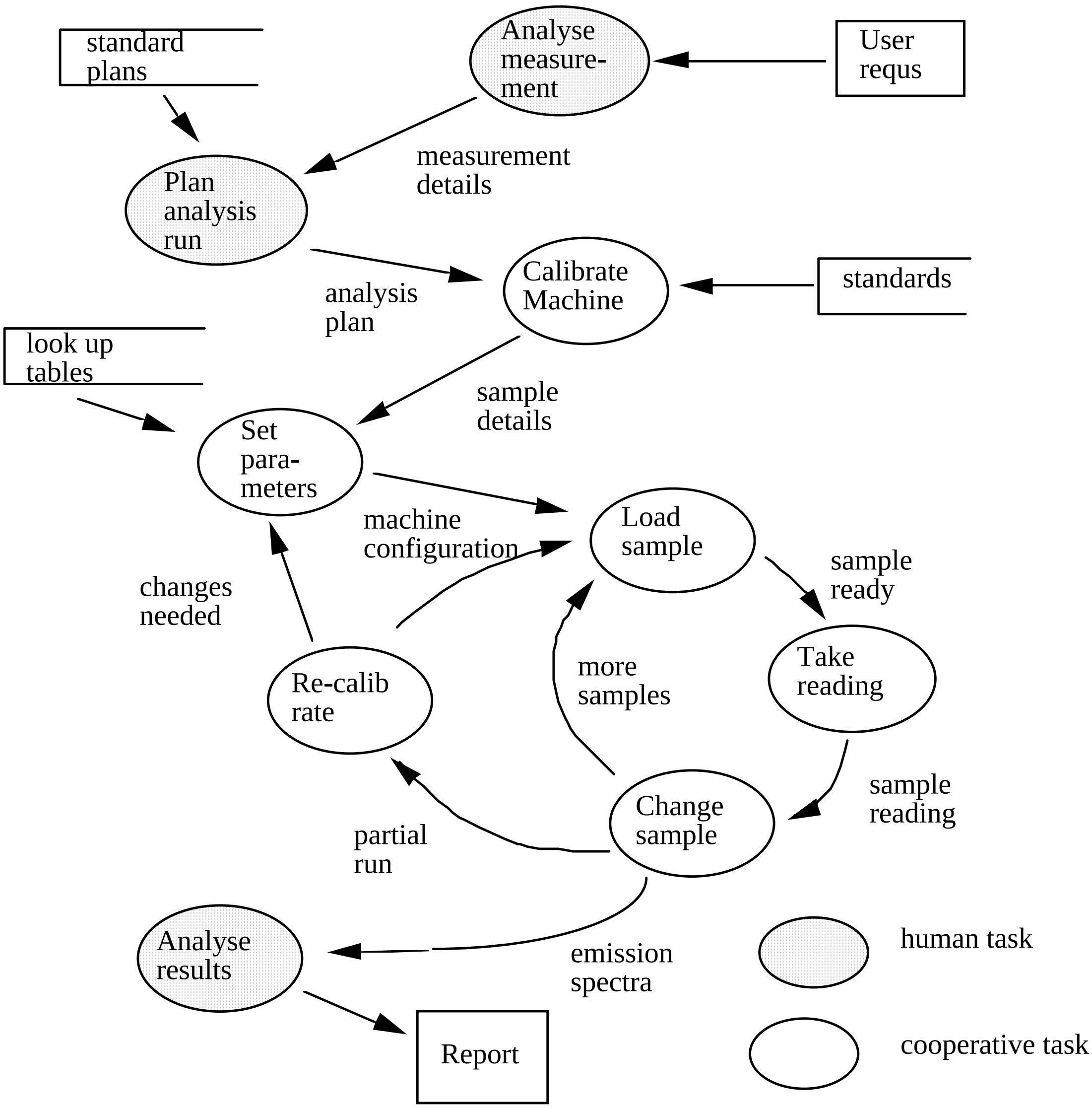


Figure 4 Task Model for the System Operator expressed in data flow diagram format

Task analysis identified the tasks involved in normal operation and also various fault scenarios, such as if there should be a power failure or failures in the emission detection apparatus [Sutcliffe, 1998]. Generation of domain scenarios revealed a sensitivity to temperature, humidity and dust levels, which is of particular relevance given these machines are intended not only for the UK market, but also for export around the globe. Excessive temperatures were found to lead to inaccurate readings and increased operator fatigue as calibrations then have to be carried out more frequently. High humidity increases condensation, leading to possible sample contamination and hence inaccurate readings. High levels of dirt or dust have the same effect on machine reliability and caused increased operator workload due to corrections and abandoned runs. Excessive vibration leads to inaccurate readings, failure in the autosampler mechanism and makes the operator's task again more difficult. Excessive noise has little effect on machine reliability but increases operator stress and therefore contributes to a work environment that could lead to increased errors.

Safety culture and level of operator training obviously also varies according to the place of operation. The worst case environment is a poorly maintained laboratory with excessive ranges of temperature, humidity and contaminants, and no maintenance being carried out.

4.2 The Causal Analysis

The BBN shown in Figure 2 models four groups of 'outer' BBN variables including:
- external factors such as the environmental conditions, and amount of time and interrupts,
- internal factors such as fatigue and motivation,
- operator knowledge of the domain and task as well as safety issues, and
- operator judgement and ability

These causally affect the 3 factors, situation, aptitude and expertise.

The 'inner' BBN variables are:
- task complexity defined according to the relative emphasis on physical and cognitive aspects, and
- quality of the user-interface

As mentioned above, from the hazard analysis of the laser spectrophotometer system, it was found that the system is particularly sensitive to temperature, humidity and dust. The environmental conditions node of the BBN can therefore be expanded as shown in Figure 5.

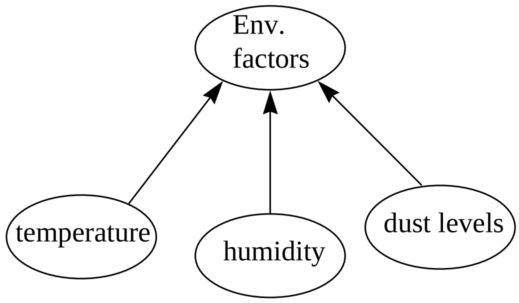


Figure 5 Environmental factors affecting the laser spectrophotometer

Of the three types of system user described in section 4.1 - research scientists, chief technicians and operators - we report a detailed impact analysis of the normal course of an operator task, *planning the sampling run*. In this, the operator has to calibrate the machine against a set of standard samples, then go through the cycle of loading samples for each measurement as shown in Figure 4.
For this case study we assumed the operator to be of medium ability and maintained the *aptitude* node as medium. We also assumed some knowledge of the domain but not a lot; the *domain knowledge* node was set therefore to medium. Finally, the quality of the user interface (*U interface* node) was similarly set as a constant medium value.

Thirty two test scenarios were then run in which there were different combinations of the following operator attributes:

- safety awareness (high or low),
- level of task knowledge (high or low)
- internal factors of motivation (high or low) and fatigue (high or low)
- environmental conditions (good or bad) which refer to temperature, humidity and dust levels, and
- amount of time/level of interrupts (high or low)

Each of the 32 scenarios was run for each of the different actions in the task. The actions are shown in Table VII below, according to their type (as listed in section 3.3), complexity and safety-criticality. The safety-criticality aspects are considered during the consequence analysis phase (discussed in section 4.3).

Action complexity is categorised according to the combination of physical and cognitive aspects of the action as follows:

- *simple*: low physical complexity and low cognitive complexity
- *physical*: high physical complexity and low cognitive complexity
- *cognitive*: low physical complexity and high cognitive complexity
- *complex*: high physical complexity and high cognitive complexity

Table VII Actions comprising the task: *planning the sampling run*

		Safety-criticality	
Action-type	**Action complexity**	**Risk: personal injury**	**Risk: data accuracy**
Physical operation	physical	eg.loading / changing samples	eg.loading/changing samples,

	simple		eg.printing, saving, displaying results, take reading
Detection	cognitive	e.g. mis-loaded sample	
Entering/editing data	complex		e.g. calibrating the machine, setting parameters
Analysis & diagnosis	cognitive		e.g. analysing results, analysing measurements
Goal evaluation	cognitive		eg. planning re-calibration
Activity planning	cognitive		eg. planning analysis run
Monitoring actions	complex		eg. checking parameters
Repair	complex		eg. re-calibration and resetting of parameters

4.2.1 Results of Causal Analysis

Figures 6 (a-e) illustrate the results for five of the 32 test scenarios:

a) Test scenario 1: an optimistic one with a safety aware operator, well motivated with plenty of task knowledge, lots of time, few interrupts, a clean and comfortable environment in terms of heat and humidity, and not suffering from fatigue.

b) Test scenario 2: again this operator is well motivated, safety aware and trained in the task. There is little fatigue. But, the environmental conditions are bad and time constraints and interrupts are high.

c) Test scenario 3: This operator is well trained in terms of task knowledge but not safety aware. S/he is tired and lacking in motivation with little time and too many interrupts, yet the environmental conditions are good.

Test scenario 4: Here both safety awareness and training in terms of task knowledge are low. Everything else however, is good. i.e. the operator is motivated and not tired, the environmental conditions are good and there are few time constraints or interrupts.Test scenario 5: This is a worst case scenario. Safety awareness and task knowledge are low; the operator is unmotivated and tired as well as the environmental conditions of heat, humidity and dust being bad. Time constraints and interrupts are also high.

The interpretation of the resulting probabilities for errors in the bands ‘high, medium, low’ has to be calibrated for the domain. In this example low was taken to be an absolute frequency which would not significantly impede normal system operation, i.e. <0.1% errors i.e. < 1 error per 1000 actions. Medium was estimated as a range from 0.1% to 1.0% errors, i.e. ranging from 1 error per 1000 actions to 1 error per 100 actions. High is >1.0% errors, i.e. >1 error per100 actions.

It may be noted that the results of the BBN causal analysis shown in Figures 6 a - e indicate quite high probabilities of error for both mistakes and slips. Further development of the BBN's NPTs is necessary for more realistic error predictions. Our aim in this paper however, was not to emphasise the significance of the actual probability values obtained with the BBN tool. Rather we are interested in the relative probabilities produced and the differences between the test scenarios in order to highlight the complex interactions between the various influencing factors and how they impact upon human error.

Test scenarios 1 and 2 allow a direct comparison of the effects of the external factors, i.e. environmental conditions (heat, humidity and dust) and level of time constraints/interrupts. In these two scenarios both the operators are safety aware and knowledgeable about the task. They are also motivated and not tired. In scenario 2 however, the environmental conditions are bad and there is little time and plenty of interruptions. The model predicts the probability of both mistake-errors and slip-errors will rise in scenario 2, but the amount of increase is much greater for slip-errors than mistake-errors. The probability of more than 1 slip in 100 operations occurring whilst performing a complex action-type such as re-calibration of the machine for example under test conditions 1, is 39%. In test scenario 2 this rises to 63% For simple actions such as printing the results, the probability of a high level of slip-errors also rises from 16% to 41%. In contrast the probability of high levels of mistake-errors only rises from 16% for a simple action in scenario 1, to 20% in scenario 2.

Test scenarios 4 and 5 similarly allow a comparison of the effects of the external factors but within the context of both representing operators with low safety awareness and little task knowledge. They also differ in that the operator in scenario 4 is motivated and not tired whereas these are not the case in scenario 5. This lack of motivation and increased fatigue raises the probability of high levels of slip-errors approximately 10% more than in the above comparison where only external factors were bad. Both mistake-errors and slip-errors are at a very high level for all action-types in these two scenarios, the probability of high levels of mistake-errors ranging from 60% for simple action-types to 80% for complex action-types. Slip-errors are much lower in scenario 4 and probabilities of high level range from 22% for simple action-types to 44% for complex action-types. This implies that the internal factors of motivation and lack of tiredness make a substantial difference to slip errors.

Test scenarios 1 and 4 only differ in terms of levels of operator safety awareness and task knowledge. In scenario 1 the operator is safety aware and has lots of task knowledge; in scenario 4 the opposite is the case. Otherwise, the environmental conditions and time/interrupts are favourable in both test scenarios and the internal factors of motivation and fatigue are also positive. The results show that the predicted probability of high levels of mistake-errors is much higher in scenario 4 than in scenario 1 - from 16% for simple action-types in scenario 1 to 60% in scenario 4. Similarly the probability of high levels of mistake-errors for complex action-type rises from 44% for a complex action-type in scenario 1 to 78% in scenario 4. In contrast, slip-errors are less affected. They only rise by a probability of about 5% in scenario 4 in comparison with scenario 1, for all action-types. This means that training in both task knowledge and also safety awareness is very important to reducing the potential for mistake-errors rather than slip-errors. Actions involving greater cognitive activity, such as analysing results, are more at risk when task training and safety awareness are reduced.

Test scenario 3 gives predictions of high levels of mistake and slip-errors in a context where the environmental conditions are good but motivation and fatigue are a problem, as well as there being a time constraints and many interruptions. Safety awareness is low, but task knowledge is high. The predictions of high levels of slip-errors are similar to those from scenario 2 where the environmental conditions are bad but motivation and fatigue are not a problem. The other difference is that in scenario 2 the level of safety awareness is high. This has an impact on the results by making the probability of high levels of mistake-errors about 10% less likely for all action-types in scenario 2 than in scenario 3. Safety-awareness is shown therefore to impact slightly more upon mistake-errors than slip-errors.

By examining the results of all 32 test scenarios, it was also found that of the external factors, time/interrupts has a relatively greater impact on particularly high levels of slip-errors, than environmental conditions. In general, the implications from these results are firstly, the importance of adequate training of operators in task knowledge *and* safety awareness. Secondly, good design of work environments and user interfaces should enhance motivation, reduce fatigue, encourage sufficient time for each task, and limit (if possible) any detrimental effects of environmental factors such as heat and humidity, particularly for actions of high physical complexity .

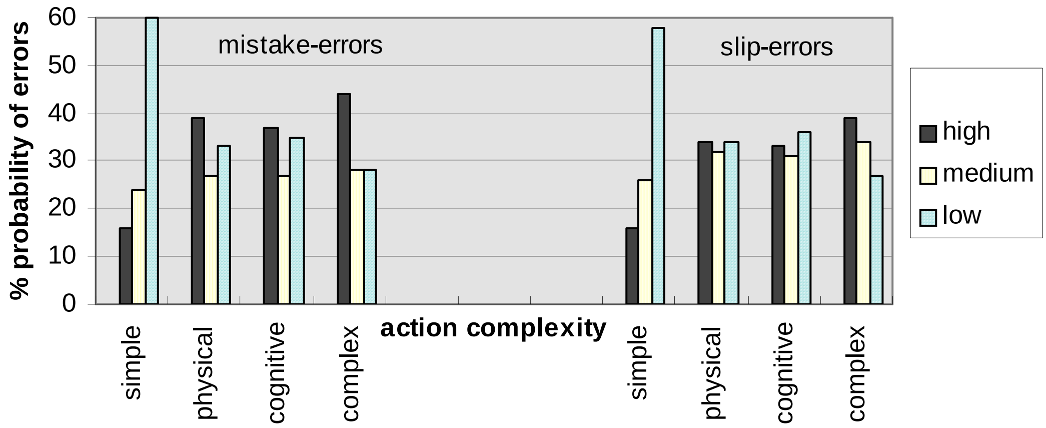


Figure 6a Test Scenario 1

Best case: operator is safety aware, has good task knowledge, internal factors are good, the environmental conditions are good and the level of time/interrupts is low

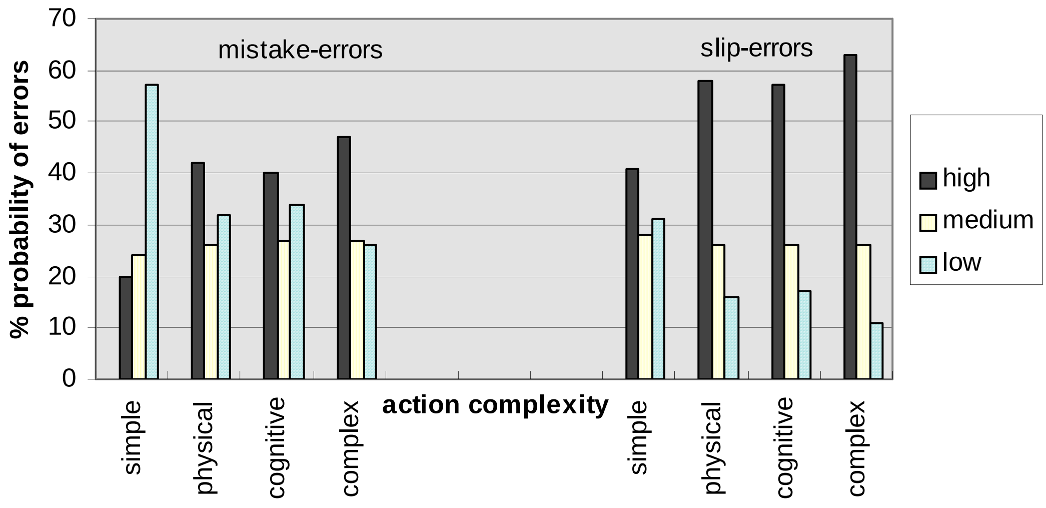


Figure 6b Test Scenario 2

Motivated operator is safety aware, has good task knowledge, internal factors are good, but the environmental conditions are bad and the level of time/interrupts is high

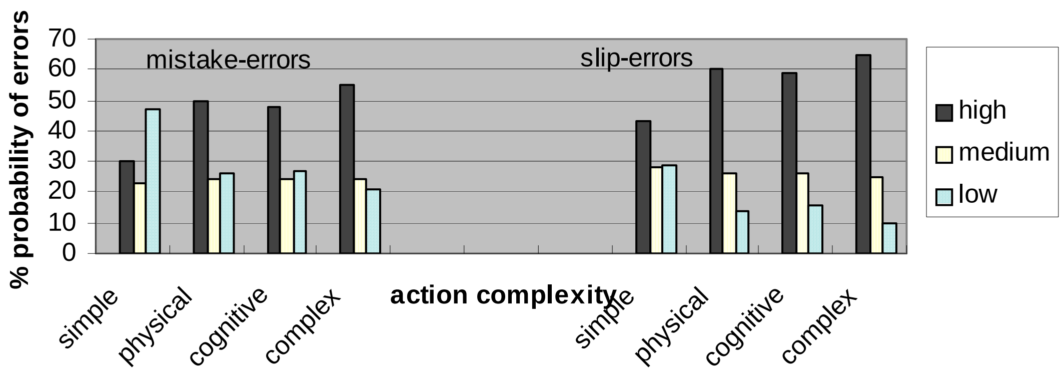


Figure 6c Test Scenario 3

Well-trained operator with task knowledge is not safety aware. The environmental conditions are good but internal factors such as motivation and fatigue are bad and the level of time/interrupts is high

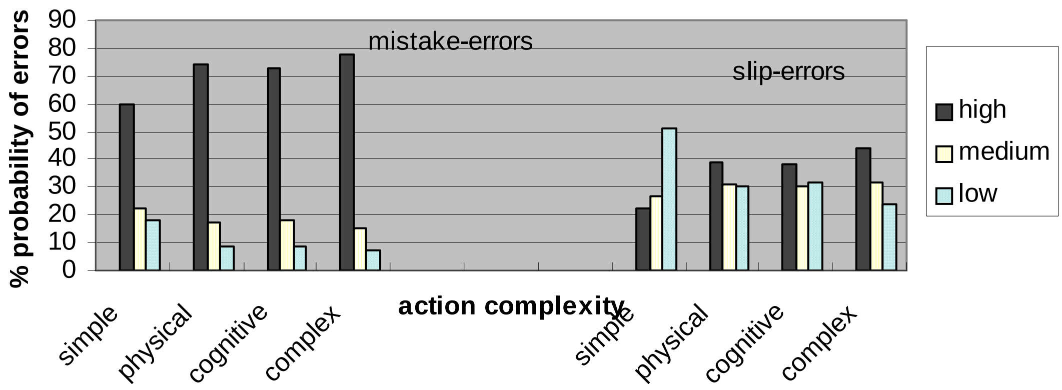


Figure 6d Test Scenario 4

Operator lacks safety awareness and task knowledge, but the environmental conditions and level of time/interrupts are both good, and the operator is motivated and not tired.

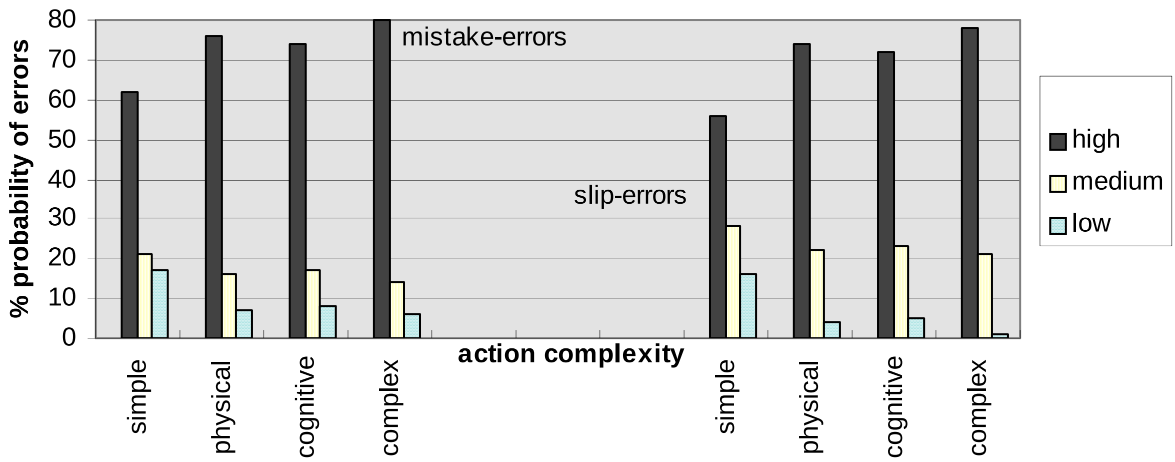


Figure 6e Test Scenario 5

Worst case: operator is not safety aware, has little task knowledge, internal factors are bad, the environmental conditions are bad and the level of time/interrupts is high

4.3 Consequence Analysis

For consequence analysis, we examine the safety-critical nature of each individual action. This is a preliminary to making appropriate design decisions for elimination, reduction or control of that risk.

In Table VII the actions that comprise the task *planning the sample run*, are distinguished according not only to their complexity, but also to the nature of their safety-criticality. Loading samples, for example, is shown to be a physical operation with more physical complexity than cognitive for the operator. It was also considered, from the results of the hazard analysis referred to in section 4.1 and described in detail in Sutcliffe [1998] as potentially dangerous to the operator if, for example, a poisonous chemical such as an arsenic compound is being assayed. Furthermore, the causal analysis indicates that ‘loading the sample’ action is particularly prone to slip errors when the environmental conditions are bad. There is risk therefore of personal injury from incorrect handling of the sample. Incorrectly detecting a mis-loaded sample is similarly dangerous for the same reasons. On the other hand, mistakes and/or slips occurring in the performance of some of the other tasks, such as calibrating the machine or analysing the results, are risky mainly in affecting the accuracy of the results data, upon which future safety critical decisions may be then made.

4.4 Design analysis

An exhaustive design analysis concerns an entire system. However, in this paper our aim has been to restrict our design analysis to safe user interfaces.

In this phase we look at each of the actions in the *planning the sample run* activity and relate them, via their action-type, to particular safety-related UI design guidelines as indicated in Table VI in section 3.4. Having thereby established the subset of appropriate guidelines for each action, we then incorporate knowledge from our earlier causal and consequence analysis, and make *specific* recommendations based upon the action-type, the nature of it’s safety-criticality, and the propensity towards mistake and/or slip errors. For example, the action ‘loading/ changing the sample’ is of the type, physical operation. According to Table VI, relevant design measures for such action types are:

- Location compatibility,
- Task completeness
- Simplicity i.e.
- Similarity and dissimilarity
- commensurate effort
- operation visibility
- persistence
- containment

We also know from Table VII and our consequence analysis, that the nature of the safety-criticality of this action concerns both personal risk as well as risk to the accuracy of the data. From our causal analysis we have established that as a physically complex action, requiring less cognitive capabilities than physical ones, high levels of slip-errors are more likely, especially in contexts where the environmental conditions are bad and the level of time/interrupts are high. Mistake-errors become more prevalent for such action-types when the level of training in task knowledge is low and also when there is little safety awareness. Our design guidelines can now therefore be made more directly applicable to this particular action of this particular task:

- Location compatibility. e.g. Location of the autosampler carousel should be such that it is easily accessible (appropriate height, orientation, etc.) to the operator for manual loading and removing of samples, without spillage. Also, samples should align easily into specially designed holders of an exact size.
- Task completeness eg. In contexts where the system may be used by relatively untrained operators, instructions for safe operation or aide memoirs should be clearly displayed on the autosampler itself.
- Simplicity: eg. The autosampler mechanism should be very simple and obvious in it's use.
- Similarity and dissimilarity This principle is not of particular relevance to this 'loading/changing samples' action.
- commensurate effort. eg. The autosampler should be designed so that it is impossible to load a sample in a container of the wrong size or type.
- operation visibility. eg. The control system should indicate to the operator both correct and incorrect loading of samples into the autosampler, in this case with visual and audio indicators.
- persistence. e.g. The visual and audio indicators of incorrect loading/changing of samples should persist for a sufficient duration to alert the operator.
- containment. e.g. It should not be possible to excite the laser for analysis until a incorrectly loaded sample has been reloaded correctly.

Another example is the 'analysing results' action. This is of the type: analysis and diagnosis and comes under the "cognitive" category in terms of complexity because it involves predominantly problem solving skills rather than physical manipulations. According to Table VI, the primary relevant design measures for such action types are:

- Browsability
- Feedback on errors
- Operation visibility

From the consequence analysis we note that the risk here is to the accuracy of data. From the causal analysis we know that the cognitive nature of the action makes it more prone to mistake-errors, particularly where the operator lacks relevant training and safety awareness skills.

- Browsability e.g. whilst examining the emission spectra via the spectral analyser, the operator should have available the initial set of parameters upon which the laser was calibrated for example, exposure time, power, and so on.
- Feedback on errors e.g. Because the 'analysing results' action is part of the normal course of the task *planning a sample run*, this design guideline is not relevant.
- Operation visibility e.g. The control system should display to the operator the current state of the sample run. In order to analyse results, for example, the sample run must be completed.

CONCLUSIONS / DISCUSSION

Although accident models in the safety engineering literature [e.g. Leveson 1995, Reason, 1990] describe accidents in terms of dysfunctional interactions between the socio-technical system variables (e.g. organisational processes, political and social climate of an organisation, management's safety policies etc.), the models do not relate the effects of these interactions on human error. On the other hand, the research on human error in cognitive psychology and human sciences [e.g. Reason, 1990, Norman, 1990] has typically focused on the work of individual users only from a cognitive perspective. Even when some of the human error models in cognitive science consider that causes of human error can lie in the environment, the effects of these contextual (influencing) factors has not been explicitly derived. In this paper we have made an attempt to fill the void between the various accident and human error models.

The Impact Analysis Method is an attempt towards performing a systematic and comprehensive analysis of influencing factors and their interactions. The BBN model and tool support enable the designer to perform an exhaustive analysis of influencing factors and their effect on the probabilities of occurrence of human error. The consequence analysis then involves determining the safety risks due to human error. The design analysis, as a final step in this method, informs improved UI design to eliminate, reduce or control human error. (Aspects not dealt with in this paper, such as improved ergonomic conditions, efficient personnel selection and training procedures, better task allocation and job design mechanisms etc. are dealt with in our earlier work [Sutcliffe & Minocha, 1998].).

To aid the design analysis, we have proposed a taxonomy of action types, types of risks involved due to human error, and the corresponding UI design measures to cater for these risks. The application of the UI design measures/guidelines has been demonstrated in the case study. Guidelines, proposed as checklists as in Leveson [1995], are difficult to interpret and apply. To overcome this limitation, we have not only categorised the UI design guidelines into three strategies, but also related them to the various action-types in a task. Domain knowledge and expertise of the designer, are, of course, required then to specifically apply the proposed guidelines for the domain and application in question. Also, if a prototype of the UI design is available, heuristic/expert evaluation of the prototype [Sutcliffe et al., in press] can be performed using these guidelines to assess the safety, effectiveness and robustness of design. If safety risks are identified in such evaluations, then the design can be modified to reduce the probability of human errors occurring.

One aspect of the Impact Analysis Method is that it comprises a marrying of a quantitative analysis via the BBN model with qualitative advice in the form of generic design requirements. As discussed in section 4.2.1, we have not concentrated on the actual quantities, i.e. the probabilities of error, but rather on a comparison between these under different test scenarios. When, on the other hand, such realistic data is available, it would then be possible to provide further guidance in the form of acceptable and unacceptable ranges of probabilities of error, and the corresponding generic design requirements/advice.

The Impact Analysis Method can also be developed further in terms of enhancing the BBN model by incorporating the remaining influencing factors of the taxonomy developed in the CREWS-SAVRE method [Sutcliffe et al, 1998a, 1998b]. Such a BBN model would then be able to predict the probabilities of occurrence of human error not only due to the factors in the current BBN model, (see Figure 2) but a host of other socio-technical variables such as work organisation (management's commitment to safety, work procedures, work schedule, task composition, task allocation, etc.) and social environment (communication, discipline, morale of the employees, information sharing etc.). This would expand the coverage of the safety risk analysis and, consequently, its role in determining requirements to counter these risks.

In conclusion, whilst acknowledging the limitations of the Impact Analysis Method in its current state, we believe it offers a significant step forward towards the goal of safe and error-tolerant UI design.

REFERENCES

BELL B.J. AND SWAIN A. D. 1985. Overview of a procedure for human reliability analysis. *Hazard Prevention*, 22-25.

EARTHY J.V. 1995. *Full Hazops of Programmable Electronic Systems.* Contesse Project, Report No. 7266-9-0-0.3, Lloyds Register, Croydon, U.K.

GOULD J. D. 1988. How to design Usable Systems, in M. Helander (ed.), *Handbook of Human Computer Interaction*, Amsterdam, North Holland, 757-789.

HOLLNAGEL E. 1993. *Human Reliability Analysis: Context and Control.* London: Academic Press.

LEVESON N. 1995. *Safeware: System Safety and Computers*. Reading MA: Addison Wesley.

LEVESON N. AND TURNER C.S. 1993. An investigation of the Therac-25 accidents. *IEEE Computer*, 18-41.

MONK A., WRIGHT P., HABER J. AND DAVENPORT L. 1993. *Improving your Human Computer Interface*, Prentice Hall.

NORMAN D. A. 1990. The 'problem' with automation: Inappropriate feedback and interaction, not 'over automation'. In D.E.Broadbent, J.Reason and A. Baddeley, eds. Human Factors in Hazardous Situations, Clarendon Press, Oxford, UK. pp137-145

PEARL J. 1988. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference,* Morgan Kaufman.

PERROW C. 1984. *Normal Accidents: Living with High Risk Technology*. New York: Basic Books.

PHILLIPS L. AND HUMPHREYS P. 1990. A Socio-Technical Approach to Assessing Human Reliability In Oliver, R.M. and Smith, J.Q. (eds) *Influence Diagrams, Belief Nets and Decision Analysis*, John Wiley and Sons.

REASON J. T. 1990. *Human Error*. Cambridge University Press.

RASMUSSEN J., PEJTERSEN A.M. AND GOODSTEIN L.P. 1994. *Cognitive Systems Engineering*, John Wiley and Sons.

SUTCLIFFE A.G. 1998. Using Scenarios for Safety Critical User Interface Design, *Centre Technical Report* CHCID/98/7.

SUTCLIFFE A. G. AND MINOCHA S. 1998. *CREWS-Scenarios for Acquisition and Validation of Requirements - The Method*, Tutorial, 1998.

SUTCLIFFE A. G., MAIDEN N., MINOCHA S. AND MANUEL D. 1998a. Supporting Scenario-based Requirements Engineering, to appear in *IEEE Transactions on Software Engineering*.
SUTCLIFFE A.G., GALLIERS J. AND MINOCHA S. 1998b. Human Error and Systems Requirements, submitted for possible presentation in the *fourth International Symposium on Requirements Engineering (RE'99)*, Limerick, Ireland.
SUTCLIFFE A.G., RYAN M., SPRINGETT M.V., AND DOUBLEDAY A. *in press.* Model Mismatch Analysis: Towards a Deeper Explanation of User's Usability Problems. *International Journal of Human Computer Studies* .
Acknowledgements

This research has been funded by the European Commission ESPRIT 21903 long term research project 'CREWS' - Co-operative Requirements Engineering With Scenarios. The project partners include RWTH-Aachen (project co-ordinator), City University, London, University of Paris I, France, FUNDP, University of Namur, Belgium.

Appendix I UI Design Strategies, Categories and Principles for Safety-critical Systems

Elimination Strategies

1 Task Conformance

Task conformance is the degree to which the system services all of the tasks the user wishes to perform and in the way that the user understands them.

a Task completeness

- Provide electronic / paper-based checklists for users.
- Integrate critical actions into the task.
- Make safety-critical operational steps incremental.

b Information Adequacy

- Provide independent means for operators to check safety-critical information.
- Provide alternative sources of critical information in case the computer display fails.
- Do not overwhelm the user with a large amount of marginally relevant or irrelevant information.
- Provide simple ways for the users to obtain additional information.
- Make all information needed for a single decision process visible at the same time (for e.g. put it on one display).

2. Simplicity

- Design a System that matches human capabilities.
- Design to aid the user, not take over; do not oversimplify the user's task.
- Interactions between components should be limited and straightforward (loose coupling between components).

3. Similarity and Dissimilarity

- Design the control panel to mimic the physical layout of the plant or system.
- Avoid similarity of critical controls.
- Avoid similar sequences of actions for different critical actions.

4. Location Compatibility

- Provide frequently used displays centrally and group displays of information used together.
- Avoid proximity, interference, or awkward location of critical controls.

5. Flexibility

Flexibility is the multiplicity of ways the user and system exchange information.

- Design flexible computer displays; this implies that for controlling dynamic systems, design for command-driven dialogues where the system can react to commands initiated by user. Prefer a mixed-initiative dialogue for human error prone situations where both the system and user can initiate prompts for each other.

6. Task Migrability

Task Migrability is the ability to pass control for the execution of a given task so that it becomes either internalised by user or system, or shared between them.

- Allow for transfer of control for execution of tasks between system and user. For example, on the flight deck of an aircraft, there are so many control tasks that must be performed that a pilot would be overwhelmed if she had to perform them all as well. Mundane control of the aircraft's position within its flight envelope is greatly automated. However, in the event of an emergency, it must be possible to transfer flying controls easily and seamlessly from the system to the pilot.

7. Commensurate Effort

- Make potentially dangerous actions difficult or impossible, that is, minimise the potential for inadvertent activation of a function, for example, an unsafe action should not be initiated by pushing a single key / button.
- Make fail-safe actions easy and natural.
- Provide interlocks to prevent inadvertent, potentially hazardous human actions.

Reduction Strategies

1. Observability

Observability is the ability of the user to evaluate the internal state of the system from its perceivable representation.

a Operation Visibility

- Continually update users on the current process state.

b Browsability

- Allow the user to explore the current internal state of the system via the limited view provided at the interface.
- Provide navigation mechanisms through the observable system states.

c Persistence

- Highlight the status of safety-critical components or variables visually as well as by audible sounds for a sufficient duration to alert the user.
- Do not permit overrides of potentially safety-critical failures or clearing of status data until all data has been displayed and perhaps not until the user as acknowledged seeing it.
- Flag rather than remove obsolete information from computer displays. Require the user to clear it explicitly or implicitly.

2. Learnability

Provide facilities for users to experiment, to update their mental models, and to learn about the system. It is also defined as the ease with which new users can begin effective interaction and achieve maximal performance.

a Familiarity

Familiarity is the extent to which a user's knowledge and experience in other real-world or computer-based domains can be applied while interacting with a new system.

- Design should reflect normal tendencies and expectations.
- Use icons with a standard interpretation. Choose icons that are meaningful to users, not necessarily to designers.
- Maintain manual involvement or ways to update mental models.

b Predictability

Predictability is the support of the user to determine the effect of the future action based on past interaction history. It also deals with the user's ability to know which operations can be performed.

- Provide feedback to user actions and display present states of the system in response to user or system actions to enable the user to predict the future system states; a fast-time simulation of a mathematical model providing information about a future state, or a cause-consequence diagram that projects the progression of a disturbance to the users would facilitate predictability.

Control Strategies

1. Containment

Containment implies restricting the spread of hazard (due to human error) in the event of a failure in any or all of the prior defensive mechanisms.

- While safety interlocks are being overridden, their states should be displayed. The design should require confirmation that the interlocks have been restored before allowing resumption of normal operation.

2. Recoverability

Recoverability is restoring the system to a safe state as quickly as possible. It is also defined as the ability of the user to take corrective action once an error has been recognised.

a Reachability

- Avoid blocking the user from getting to a desired state from some other undesired state.
- Make safety-enhancing actions easy and robust. Stopping an unsafe event should be possible with a single key stroke.

b Forward Error Recovery

- After an emergency stop, require the operator to go through the complete restart sequence.

c Backward Error Recovery

- Provide multiple ways to change from an unsafe state to a safe state.
- Provide compensating (reversing) actions.
- Provide time to reverse from errors.

3. Feedback

Feedback is essential to monitor the effects of user actions, to allow for the detection and correction of errors, and to maintain the alertness of the users.

a Feedback on User Actions

- Provide adequate feedback to keep users in the loop and to update their mental models so that correct actions can be taken when required.
- Provide users with feedback if commands are cancelled (not executed) because of time-outs or for other reasons.
- Provide confirmation of valid data entry and the acceptance and processing of user-entered commands.
- Provide feedback on the effects of actions so as to allow the user to cope with the time delay between the execution of an intention and the observation of its effect.

b Feedback on Errors

- Make errors observable. Provide feedback about actions and the state of the system.
- Provide error messages that distinguish safety-critical states or errors from non-safety critical ones.
- If the automatic system detects an unsafe condition, inform the user of the anomaly detected, the action taken, and the current system configuration.

4. Responsiveness

Responsiveness measures the rate of communication between the system and the user. Response time is generally defined as the duration of time needed by the system to express state changes to the users. Short duration and instantaneous response times are desirable.

- Distinguish processing from failure. Provide status indicator to the user if any processing requires several seconds.

5. Warning

Warning implies to signal the presence and the nature of the hazard to all those likely to be exposed to its dangers.

- Make warning displays brief and simple.
- Dramatic warning devices, such as flashing lights and loud sounds, are often used to indicate potential problems, but too many attention-grabbing signals have a negative effect.
- Minimise spurious signals and alarms. Provide users with straightforward checks to distinguish hazards from faulty instruments.
- Safety-critical alarms should be distinguishable from routine alarms. The form of the alarm should indicate the degree of urgency.

6. Robustness

Robustness is the level of support provided to the user in determining successful achievement and assessment of goals.

- Provide multiple physical devices and logical paths to ensure that a single hardware failure or system error cannot prevent the user from taking action to maintain a system state and avoid hazards.
- Instrumentation meant to help operators deal with a malfunction should not be disabled by the malfunction itself.