Proceedings of the IASTED International Conference
HUMAN-COMPUTER INTERACTION
November 14-16, 2005, Phoenix, AZ, USA

FUNCTIONAL SAFETY FOR HCI – PROPOSAL FOR INTEGRATION OF HUMAN FACTORS WITH THE IEC 61508 STANDARD

Andreas Aas
Bestumveien 62B
NO-0282 Oslo, Norway
Andreas.Aas@idi.ntnu.no
Norwegian University of Science and Technology.

Torbjørn Skramstad
Dept. of computer and information science, NTNU
NO-7491 Trondheim, Norway
Torbjorn.Skramstad@idi.ntnu.no
DNV Research, NO-1322 Høvik, Norway
torbjorn.skramstad@dnv.com

ABSTRACT
An Australian study of 75 fatal aeroplane accidents showed that more than 70% of the accidents involved pilot factors [1], i.e. human factors (HF).

The needs and roles of human operators in safety critical systems are seldom given adequate focus, compared to the potential risk human factors represent, when humans interact with safety critical systems. Therefore, unsatisfactory consideration of human factors and poorly designed Human-Computer Interaction (HCI) may contribute to compromising safety in ways beyond the designers' imagination.

This paper suggests a possible implementation of human factors into the IEC 61508 (International Electrotechnical Commission), a leading industrial standard for functional safety [2]. Implementation of human factors is suggested performed by dividing the user interface into different levels, representing the whole spectrum of the system safety integrity. Then, each level must fulfil specific demands, based on the effects on the related system's or subsystem's safety integrity.

One important issue for an operator to operate a system safe and efficient is to have a correct mental model of the system. To achieve this, the operator depends on a correct system image. Providing a correct system image should be one of the main concerns when designing user interfaces for safety critical systems.

KEY WORDS
HCI, HAZOP, Functional safety, SIL, IEC 61508, System Image.

1. Introduction

Human factors are an important issue regarding the use of safety critical systems. Stress, communication and operational documents (e.g. checklists) are all examples of factors that need to be considered. Nevertheless, the user interface is perhaps the most important part of HCI. The user interface is the operator's "window" into the system. Hence, the user interface is the main dynamical source of information from where the correct system image or current system status can be derived.

This paper is based on an M. Sc. thesis with the title "User interfaces for safety critical systems" [3]. A case study in the thesis, focusing on the user interface of a safety critical software-intensive system used to operate vessels at sea, known as Dynamic Positioning (DP), is used as an example in this paper also.

As the importance of computers increase in our society in general, so is also the situation for computers used in safety critical systems. However, the use of computers seldom excludes the human being, making the system fully automated. This means that HCI should also be carefully considered during the entire life cycle of a system.

Typical examples of safety critical systems are air traffic control systems, control rooms at nuclear power plants and systems used to operate vessels at sea. History shows many examples of why human factors should be considered carefully when designing safety critical systems. Well known accidents like Three Mile Island, Bhopal and Chernobyl [4] are all examples of accidents where a more careful consideration of human factors could have prevented the accident. Or at least could have made the impact on the surroundings less severe.

The use of industrial standards when developing safety critical systems will help prevent or control many potential hardware and software failures. The IEC 61508 has proven very useful for this purpose, but does not aim at eliminating hazards. The standard basically set demands for the safety system(s) controlling a particular and identified hazard, reducing risk to an acceptable level. IEC 61508 can be used as a stand alone standard [5]. Other examples of use are that it is basis for a published nuclear sector standard [5] and that it is applied in the Norwegian petroleum industry [6] with additional guidelines.

IEC 61508 uses a risk based approach to determine the safety integrity requirements of Electrical, Electronic or

Programmable Electronic (E/E/PE) safety-related systems, and includes a number of examples of how this can be done [5].

The primary idea behind IEC 61508 is to divide the "spectrum" of integrity into different levels and then to define requirements for each safety integrity level [2]. These levels, normally four, are referred to as Safety Integrity Levels (SILs). Ranging from SIL 1 to SIL 4, where SIL 4 represents the largest risk, the standard gives guidance to specific targets and requirements for the safety systems for each SIL. The higher risk a hazard represents, the higher SIL is given to the safety system(s) controlling the hazard. And the higher SIL a safety system is given, the stronger requirements it must fulfil. Hence, the higher the SIL fulfilled by the safety system, the less likely the safety system is to fail and the hazard will remain under control.

Secondly, each SIL has two different modes, known as high demand and low demand modes [7]. High demand mode is used for safety systems needed continuously, e.g. the brakes on a car. Low demand mode is used for safety systems that are seldom needed, e.g. the airbags in a car. This is based on the general idea that the more often a safety system is used, the more seldom can it be allowed to fail. It is far more dangerous when the brakes on car fail during a drive, than when it happens to the airbag system.

Reducing risk to the most appropriate level is in IEC 61508 referred to as "As Low As Reasonably Practicable" (ALARP) [8]. If a particular risk is found to be neither so great that it must be refused, nor so small that it is found to be insignificant, it is in the ALARP region. When a risk is in the ALARP region, one should seek to reduce risk to a level where further risk reduction is "impracticable or if its cost is grossly disproportionate to the improvement gained" [8]. This means that risk should be reduced to a level where further reduction is no longer cost efficient.

When all identified hazards are sufficiently under control and risk is reduced to an acceptable level, then functional safety is achieved in compliance with the standard. The standard includes the whole safety lifecycle of a system [2], from development of the overall safety requirements to decommissioning or disposal of the system.

In the IEC 61508, human factors are mainly referred to in the annexes, which are informative, and hence the standard sets no specific demands to considering human factors for achieving compliance with the standard. This means that human factors only "needs to be considered" [2], giving poor guidance to how human factors can influence the total safety integrity of the system.

Most accidents linked to human factors happen because operators don't follow orders given written or orally [4]. This indicates that HCI should be carefully considered during design of safety critical systems, aiming to make human errors impossible, or at least hard to perform or easy to detect for the operators themselves. An extension of IEC 61508 should therefore be considered to set further focus on HCI of safety critical systems.

2. Methods for analysing risk related to HCI

There exist a number of different methods for analysing risk in general. A number of methods are developed for, or can be applied to, analysing human factors, and are therefore also related to HCI. Some methods are simple to use, while others require a great deal of time and effort to achieve the desired results, [3].

There is no single method suitable for analysing risk in all systems in general, and also no single method suitable for analysing all risks related to human factors. The methods applicable depend on the system being analysed, the time and resources available, the maturity of the design, the competence of the performing organization etc. Nevertheless, some methods, both quantitative and qualitative, are briefly presented here to illustrate how they may be applied.

FMEA – Failure Modes and Effects Analysis.

FMEA was developed to predict the reliability of a product. That is that some piece of equipment will operate without failing over a specified period of time or the time it will take before a failure occurs [2]. This technique requires a detailed design, and is suitable for analysing single units or single failures.

FMEA is a quantitative method using forward search. Forward search means that the analyst starts with a piece of equipment and then follows the possible outcomes of different scenarios for this particular piece of equipment. The analyst then constructs an event tree, showing which hazards may occur if the piece of equipment fails. The constructed tree then is the basis for assessing the risks related to the piece of equipment under consideration.

HEART – Human Error Assessment and Reduction Technique

HEART is a quick technique for quantifying human reliability [9], which means that one can quantify human errors in operator tasks. The basic concept of HEART is that an operator will fail accomplishing a specific type of task at a rate based on a set of generic Human Error Probabilities (HEPs) [9] for different types of tasks, given "perfect" conditions.

However, perfect conditions are hardly ever the reality, so after classifying a task, the analyst will find any Error-Producing Conditions (EPC) for the specific scenario under consideration. Such conditions can be high or low training, hostile environment, the possibility of independent checking or testing output etc. [9].

For each EPC, the generic HEPs are multiplied by the EPC which increases the human error probability. HEART provides the following formula to determine the final assessed effect for an EPC [9]:

((HEART effect – 1) * assessed significance) + 1 = final EPC assessed effect

HAZOP – Hazards and Operability analysis.
HAZOP is a qualitative method, aiming at identifying hazards, not avoiding them. It is conducted by a HAZOP-team. Such a team consists of a team leader, a secretary and a number of experts on the system being analysed. The method is a structured walkthrough of the system's documentation, conducted during a series of meetings [4].

The analysis of a system starts with a large piece of equipment, e.g. a gas tank in the documentation. Then the team studies all connections (e.g. gas pipes) this gas tank has with the rest of the system. The part of the system being studied, such as a gas pipe, is referred to as a study node. The team members are then prompted by a set of guidewords for each of these study nodes. For example, when examining a gas pipe, typical questions to be asked are: Could there be no flow? Could there be reverse flow? Etc.

HAZOP encourages all team members to think creatively about all possible hazards related to a study node, prompted by the guide words. The results of the HAZOP-analysis are documented in tables, including existing and suggested protections, where hazards are identified.

3. HAZOP for HCI

In order to analyse a user interface in an efficient and meaningful manner, a suitable method is required. HAZOP was found to be a suitable method [3] due to its flexibility, ease to adapt to user interfaces, its ease of use, ability to identify previously undiscovered hazards, its focus on operability as well as hazards, and that it can be conducted based on existing documentation.

Others have proven that HAZOP can be used for analysing HCI, like for example Redmill [10]. One drawback of Redmill's method is the required development of extra documentation. This documentation is basically the system's design included all human tasks modelled into the same diagrams in a uniform language.

To make the analysis simpler, an easier and less time consuming variant of HAZOP is suggested here. The analysis can be performed directly on an operative user interface, a prototype, sketches or any other available representations found applicable, with support from the system documentation, such as flow diagrams etc. [3].

The team leader must be an expert on HAZOP itself, and the other members of the HAZOP-team must be experts on the system, on HCI, on psychology, or any other field related to HCI. One can either include HCI-experts in the HAZOP-team, or set up a separate HAZOP-team analysing the user interface [3], the latter requiring that the teams' work is synchronized.

Applying HAZOP to HCI requires the guidewords to be adapted to match the properties of user interfaces. The most common guidewords are shown in Table 1, with interpretations matching user interface analysis.

Guideword	Interpretation of guideword for user interface
No	Total denial of the intended purpose of the design. The user interface isn't capable of showing anything of the systems condition for the entity in focus.
More	Quantitative increase. The user interface shows a higher quantitative value for the entity in focus than what the systems true condition is or what is required to achieve a correct system image.
Less	Quantitative decrease. The user interface shows a lower quantitative value for the entity in focus than what the systems true condition is or what is required to achieve a correct system image.
As well as	All purposes of the design are fulfilled, but with additional results. The user interface shows more information about the entity in focus than what is required to achieve a correct system image.
Part of	Qualitative decrease. The user interface shows less information about the entity in focus than what is required to achieve a correct system image.
Opposite	The opposite of the intended purpose of the design is achieved. The user interface shows the opposite information about the entity in focus compared to what is required to achieve a correct system image.
Other than	Complete substitution. The user interface shows other information about the entity in focus than what is required to achieve a correct system image.

Table 1. HAZOP analysis guidewords for user interfaces [3].

Using the guidewords as described in Table 1 will result in some form of documentation. An example is shown in Table 2 [3]. The analysed system's user interface shown in Table 2 mainly consists of a computer screen, a number of buttons used to activate or deactivate different functions, a numeric keyboard, a trackball and a 3-D joystick. The system is referred to as SDP and is manufactured by Kongsberg Maritime, Norway [3].

1. Process unit: **Command** Point 1. Process parameter: **Take**				
Guide-word	Deviation	Consequences	Causes	Suggested solution
No				
More				
Less				
As well as				
Part of	Button must be double clicked to be activated. If it's clicked only once, the intended function will not be activated.	The user will not gain command over the operator console.	The button looks the same as all other buttons, regardless of they must be double or single clicked.	Change shape and/or colour on buttons that must be double clicked.
Opposite				
Other than				

Table 2. Result form for HAZOP analysis [3].

Having described a powerful, yet relatively cheap and efficient method for analysing user interfaces, it's time to see how HCI can be implemented into the IEC 61508.

4. Proposal for integrating HCI into IEC 61508

As for system design in general, designing HCI requires that potential problems and hazards are identified and dealt with as early in the design as possible [11]. This is not further discussed here, as all methods require some sort of documentation to perform the analysis on.

As for safety systems, the parts of the user interface, which in some way can affect the overall system safety, are partitioned into different Safety Integrity Levels (SILs). This means that each relevant part of the user interface is assigned a SIL, ranking the parts based on their potential effect on safety. A proposal for the process of designing user interfaces to be integrated in the IEC 61508 is shown in Figure 1.

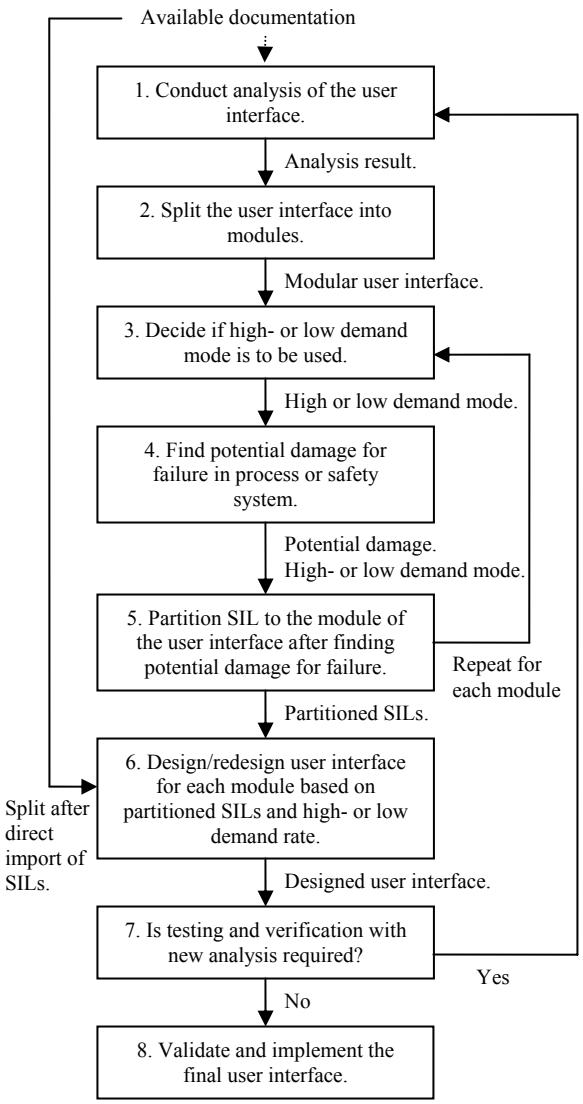

Figure 1. Proposed process for user interface design in IEC 61508 [3].

For user interfaces, high demand mode is used for the parts which must always be visible to the operator(s), for example the representations of a cooling system for a nuclear reactor. Low demand mode is used for parts of the user interface which are not required to be visible for the operator(s) at all times, but will appear or can be retrieved whenever necessary. An example of this is the representations of an emergency system for extra cooling of a nuclear reactor.

When each relevant part of the user interface has been assigned a SIL matching its potential impact on safety, the next step will be to set some requirements to each part, based on their SILs.

Partitioning SILs to parts of the user interface can be done in at least two different ways. The application of these alternatives will be discussed in section 5, here they are merely described.

First, one option is to take the SIL of a safety system (if available), and assigning the SIL for the corresponding part of the user interface to the same SIL. This can be seen as a "direct import" of SIL to the user interface.

The second option is to analyse the user interface, using for example HAZOP for user interfaces or any other suitable method, and then assigning the SIL for each part of the user interface based on the results from this analysis.

There are a number of possible demands or requirements/restrictions one can set to a user interface in a safety critical system. Table 3 shows how the system can be protected from potential hazardous input from the operators at each SIL, illustrating the difference between the SILs. In the thesis from which this is retrieved, a number of possible requirements and restrictions are suggested for each SIL [3].

SIL	Action to be taken
SIL 1	Control of operator input, with warnings for potential unsafe input.
SIL 2	Control of operator input, with warnings for potential unsafe input. Input that can not immediately be accepted must be confirmed by the operator with password, key, key card or similar.
SIL 3	Validation check of operator input, blocking potential unsafe input. Alternatively, two operators can override the blocking using passwords, keys, key cards or similar.
SIL 4	Validation check of operator input, blocking potential unsafe input with no override possibility.

Table 3. Example of SIL demands for user interfaces [3].

As shown in Table 3, the demands for controlling input increases for each SIL. Ranging from warnings for potential unsafe input in SIL 1, to the non-acceptance of input defined as hazardous by the system designer(s) in SIL 4. Note that the demands for SIL 4 exclude the operators' possibility to override the blocking of input. Hence, it is crucial that there under no circumstances and in no system states will be safe or appropriate for an operator to give the input in question when the particular blocking of input is active. Therefore, extra redundancy for SIL 3 may be preferred over SIL 4.

The demands for the SILs for user interfaces should include, but not be limited to the following; protection against potentially harmful input, timing, reversibility, robustness, training, feedback to and diversity for operators and redundancy for the user interface.

5. Discussion

Taking HCI into consideration in the earliest phase of system design, when deciding system concepts, goals trade-offs etc. is necessary to achieve a well working and safe user interface. For example, one can set overall system goals, such as the user interface being safe or encouraging safe actions. Also, management's view on safety is an important element when considering the overall safety of the system. According to Leveson [4], the perhaps most important factor in making a system safe is that the management communicate that they give safety the highest priority. Even though this is very important, further discussion of this issue is left to others to conduct.

The IEC 61508 recognizes the fact that one can not and must not quantify every potential outcome for every potential hazard. Qualitative techniques can also be used, and are in fact recommended [2].

Quantifying all human factors can be impossible, or at least very time and effort consuming. Because of this, qualitative methods like HAZOP may be preferred for analysing user interfaces. However, this does not exclude the use of quantitative techniques where this is found suitable by the person or organisation performing the analysis.

Whether hazard analysis of the user interface is performed or not, the integration of HCI into IEC 61508 as suggested here, requires partitioning of SILs to the relevant parts of the user interface. Analysing the user interface using a method like HAZOP can give advantages as previously unknown hazards are revealed. The best results to achieving functional safety for the user interface can be expected to arise from this approach. This is because a separate analysis will serve as an independent evaluation of the user interface, independent of previous analysis.

However, there is not always time and resources available to perform a separate analysis of the user interface. By applying the approach using direct import of the safety systems' SILs, one can at least partition SILs to the parts of the user interface which are directly related to the safety systems. This approach requires that the safety systems already have defined SILs and the approach is only suitable when the IEC 61508 in its current form is already applied.

Some parts of the user interface, not directly related to any safety systems can also influence the overall system safety. Where this applies, a separate safety analysis of the user interface is highly recommended.

There may be situations where the SIL of one part of the user interface should have a higher SIL than the safety system it represents. This makes no conflict with the IEC 61508. However, partitioning the SIL of a user interface lower than the SIL of the related safety system makes little sense, because this would imply that the SIL of the safety system was set too high in the first place.

We have not yet described how to decide to which SIL a particular part of the user interface should be partitioned. The IEC 61508 includes numeric targets for each SIL, based on the required probability of failure not occurring. Setting numeric targets for user interfaces will be as

difficult as doing the same for human errors, and cannot be set as generalized quantitative measures. One solution to this problem is classifying the required tasks to be performed by the operator based on the tasks' complexity.

The complexity of an operator's tasks can be ranged from simple to complex [12], and can be linked to each SIL as follows:
SIL 4 - Simplest possible task.
SIL 3 - Simple routine task.
SIL 2 - Routine task requiring attention.
SIL 1 - Complex non-routine task.

As described here, very simple tasks can be used for SIL 4, and complex non-routine tasks can be used for SIL 1. The probability of a human being failing in performing a task can be directly related to the SILs. This illustrates the need to design HCI, including the user interface, in a way that matches the complexity of the task and the potential damage if the operator makes an error doing it.

Performing a very simple task, like switching a pump on or off can easily be supported and controlled by the user interface. On the other hand, complex non-routine tasks, perhaps performed more seldom than once a year, increases the possibility for the human operator making an error, and should not be used for the higher SILs.

Usability may conflict with safety. This is inevitable and will occur in some form in all larger safety critical systems. But usability for a safety critical system must include the ability to operate safely on the system. The ease of performing a task for the operator doesn't make sense if accidents arise from that ease. Usability for safety critical systems can therefore be seen as the difficulty to make errors, or the ease of performing a task safely, disregarding the ease of performing the task itself. But this requires that there are no easier and potentially unsafe ways for an operator to achieve the goal of a task, as this would most certainly be used by the operators, making the safe way close to useless.

The focus in this paper has been on the user interface, almost disregarding the working environment where the operators actually interact with the system through the user interface. The working environment must also be considered when partitioning SILs to the user interface, but this is outside the scope of this paper.

6. Conclusion and further work

Human factors and hence, HCI and user interfaces, are important elements to achieve overall functional safety. The work presented in this paper shows that it is possible to fully integrate HCI into IEC 61508. Being the leading industrial standard for developing systems with functional safety, HCI can receive the acquired attention needed to prevent many accidents related to human errors.

Whether used on software- or hardware-based user interfaces, the suggested implementation of HCI into IEC 61508 will apply. Suitable methods may differ as the user interfaces and systems behind change, but in general, this method is applicable to all kinds of user interfaces and all kinds of safety critical systems.

The partitioning of SILs to the different parts of the user interface makes it possible to organize the user interface in respect of the effect the different parts have on system safety. The SILs make it possible to identify which parts of the user interface that have an effect on safety and which do not. But it also ranges the parts influencing safety internally, where SIL 4 requires the most attention and SIL 1 requires less attention.

To fully integrate HCI into IEC 61508, more specific demands for each SIL are needed. Further work on this subject requires extensive research to find appropriate demands for each SIL which can be generalized and applied to all kinds of systems and all kinds of user interfaces.

Being a formalised way to develop user interfaces for safety critical systems, HCI in IEC 61508 will make it required for the developers to take the user interface into careful consideration when designing safety critical systems.

HAZOP is proven to be applicable for analysing user interfaces, but all hazards will not be revealed after an analysis, regardless of which method is applied. The work presented here is no guarantee against human errors. Therefore, especially robustness and reversibility are important to implement in the user interface, and should be included in the IEC 61508 as specific demands for developing user interfaces for safety critical systems.

Complex systems require extensive work to achieve functional safety for the user interface. Complex systems can make it hard for human operators to gain a correct mental model based on the system image. The work presented here does not give much guidance to how the different parts of the user interface should operate together. But an overall evaluation of the user interface should be performed, ensuring that the user interface follows good standards for designing HCI.

Common Cause Failures (CCF) may be hard to identify. CCF is considered in IEC 61508 [13], and their relation to the user interface should also be considered.

The ideas in the work presented can be used for all safety critical systems involving human interaction at some level. The suggested extension of IEC 61508 applies to systems where the standard in its current form is already applied and also "new" systems, not yet compliant with the standard as it is today. Even if the work presented is not implemented in IEC 61508, parts of the suggestions

made can nevertheless be applied where this is found applicable, without interfering with the standard itself.
A strong focus on user interfaces in safety critical systems will result in safer operation of systems where human factors play an important role. Therefore, the implementation of an additional chapter into IEC 61508 for HCI is recommended, as it can give HCI the necessary focus when developing safety critical systems.

References

[1] Secretary of the Department of Transport and Regional Development, Bureau of Air Safety Investigation (BASI), *Special investigation report: Human Factors in Fatal Aircraft Accidents*, Canberra, Australia, April 1996, iii.

[2] D. J. Smith & K. G. L. Simpson, *Functional Safety. A straightforward guide to IEC 61508 and Related Standards* (Oxford: Butterworth-Heinemann, 2001).

[3] A. Aas, *M. Sc. thesis: "User interfaces for safety critical systems"*, Trondheim, Norway: NTNU, 2004.

[4] N. G. Leveson, *Safeware. System safety and computers* (MA, USA: Addison-Wesley, 1995).

[5] IEC, Functional safety and IEC 61508: A basic guide. Geneva, Switzerland, November 2002. 8-11.

[6] The Norwegian Oil Industry Association (OLF), Application of IEC 61508 and IEC 61511 in the Norwegian Petroleum Industry, Revision 2, 2004. 6-7.

[7] IEC 61508-4:1998 including corrigendum April 1999, Functional safety of electrical/electronic/programmable electronic safety-related systems, 19.

[8] IEC 61508-5:1999 including corrigendum April 1999, Functional safety of electrical/electronic/programmable electronic safety-related systems, 16-18.

[9] S.P. Smith, M.D. Harrison, Blending descriptive and numeric analysis in human reliability design, *In Proceedings of the 9th International Workshop on Interactive Systems: Design, Specification, and Verification (DSV-IS 2002), Forbrig, P., Urban, B., Vanderdonckt, J. and Limbourg, Q. (eds.),* York, UK, 2002, 223-237.

[10] F. Redmill, M. Chudleigh, J. Catmur, *System Safety: HAZOP and Software HAZOP* (Chichester: Wiley, 1999).

[11] J. Good & A. Blandford, Integrating HCI concerns into the design of safety critical interactive systems: a case study, IEE colloquium 99/010, PUMA Working paper WP28, 1999.

[12] D. J. Smith, *Reliability, Maintainability and Risk. Practical Methods for Engineers* (Oxford: Butterworth Heinemann, 1993).

[13] IEC 61508-6:2000, Functional safety of electrical/electronic/programmable electronic safety-related systems, 51-62.