PUMA Programmable User Modelling Applications

Incorporating a user-focused failure modes and effects analysis-like technique into the design of safety critical systems

by

Jason Good & Ann Blandford

18th December 1997

WP18

Principal Investigator	Dr Ann Blandford	Middlesex University
Research Fellow	Dr Richard Butterworth	Middlesex University
Academic Collaborator	Dr David Duke	University of York
Research Fellow	Jason Good	Middlesex University
Industrial Collaborator	Sue Milner	Praxis Critical Systems Ltd
Academic collaborator	Dr Richard Young	University of Hertfordshire

http://www.cs.mdx.ac.uk/puma/

Contact : Dr Ann Blandford
School of Computing Science
Middlesex University
Bounds Green Road
London. N11 2NQ. UK
tel : +44 (0)181 362 6163
fax : +44 (0)181 362 6411

Project funded by
EPSRC grant number GR/L00391

Incorporating a user-focused failure modes and effects analysis-like technique into the design of safety critical systems

Jason Good & Ann Blandford

Extended Abstract

18th December 1997

1. Introduction

Failure Modes and Effects Analysis (FMEA) has long been a core part of the hazard analysis procedures conducted in the design of high-reliability or safety-critical hardware systems. More recently, it has been applied to the analysis of software functions to examine the effects of failures in software. However, approaches that account for operator behaviour as a determinant of overall system safety have not been developed to the same degree, and the effect of the operator on system "safety" is often handled in an ad-hoc manner with little rigour. In this paper we propose an FMEA-like approach by analysing the interaction between operator and device. We discuss how such an approach would fit into current hazard and safety analysis procedures, develop the approach using a model of user behaviour that originates from work in cognitive science, and demonstrate how this could be used by working through an example analysis based on a real-world system

1.1 Hazard analysis and FMEA

In high-reliability systems (such as bank transaction processing) or safety-critical systems (such as aircraft and nuclear power plants) there is a long tradition of using structured methods to assess the reliability of the *hardware* aspects of the system (in terms of probabilities of failure) and to assess the consequences of such failures. Taken together such approaches are often referred to as *Hazard Analysis techniques*.

One technique used successfully by hardware engineers is Failure Modes and Effects Analysis (FMEA). This involves taking each hardware component in turn and identifying the *effects* (outcomes) of any conceivable *failure modes* (ways in which the component might fail). The effects will include hazards, and the technique allows an analyst to identify critical components which lead to these hazards. Combining this analysis with established data about component failure rates enables the system to be improved by adding levels of redundancy or replacing components by more reliable ones.

Fairly recently, software engineers have adopted these techniques and adapted them to take account of the more ephemeral nature of software. Sometimes called Functional FMEAs, these are conducted by examining each function within a piece of software and comparing it to a set of guide words to try to generate possible failure modes. These guide words include *no, too high, too low, incomplete, stale, incorrect, wrong sequence, too early, too late, spurious,* etc. The analysis is concerned with the *end-effects* — that is, the outcome at system level resulting from the failure being considered.

This technique is the subject of British Standard BS5760 Part 5 (BSI, 1991) and, when combined with a hardware FMEA, provides a fairly thorough qualitative description of the reliability of the device being analysed, provided it incorporates *non-interactive* software.

Section 5 of the Interim DEF STAN 00-58/1(MOD, undated) describes the complementary use of HAZOP Studies and FMEA, making the following distinction between them:

> *"A HAZOP Study investigates the interactions between components (at any given level of definition) and is carried out by a team.*
> *An FMEA investigates the failures of the components themselves and is often performed by an individual."*

The distinction between a team-based process and an individual effort is clearly important as any "expert team" that has to meet will draw valuable resources away from other areas. The more work that can be undertaken by an individual analyst, the faster and cheaper the process should become, but clearly not all of a safety analysis should or could be carried out by one person.

We take as a starting point the idea that a HAZOP will not explore too deeply into the causes of failure — that such investigations should be taken "off-line" for separate analysis; also, that these analyses can be conducted at various levels of abstraction and, consequently, at various stages in the design process. Whilst this may lead to a degree of repetition of analysis, it ensures that limited HAZOP Studies and/or FMEA analyses can be made early in the lifecycle to highlight areas of safety concern for consideration during the detailed design. Such early analyses can act as prompts in later studies to ensure that broad concerns are not lost when considering details.

1.2 Analysing human error

Various approaches have been taken to this area in the past and detailed accounts of the history of Human Reliability Assessment are given in Villemeur (1992) and Reason (1990). A summary is presented by Dearden and Harrison (1996), who identify 3 classes of techniques.

Many of the early techniques were *quantitative* — that is, they concentrated on giving a numerical estimate of the probability of error (or correct action) as opposed to being concerned with the severity of the outcome. Dearden and Harrison discuss shortcomings of these techniques, including the uncertain effect of "performance shaping factors" upon the numerical estimate; the unreliability of probabilities transferred from

simulator or experimental studies into real-world situations; and the focus on procedure execution, making them more suited to analysing recovery procedures following an initiating event rather than normal activity with a latent error.

Another approach to error analysis is *human error identification*, which uses qualitative techniques to identify what errors are possible. A thorough review of such techniques is given by Kirwan (1992). Dearden and Harrison observe that such methods can help to reduce the cost of human error or even reduce overall risk, but that they often require the formation of expensive, multidisciplinary teams to perform the analysis, and they can often only be used fairly late in the life of the design, so they cannot influence early design decisions.

A final class of analysis techniques are those using *cognitive models* of system operators to explain (qualitatively) the mechanisms leading to human error. As Dearden and Harrison observe, such explanations can be incorporated into the earliest stages of the design, making them good for identifying possible weaknesses in a design and generating improvements.

2. Programmable User Modelling as an approach to analysing human error

Programmable User Modelling Analysis (PUMA) is one of this last class. It is an approach to analysing an interactive system that models the user as a problem solver of intentionally limited power (Young, Green and Simon, 1989). The modelled problem solving is based on established theories of cognition (e.g. Newell, 1990). The approach is based on the premise that the user has *knowledge* — about the current state of the system, about actions and their effects, and about the task, and that the user behaves rationally. That is, the user selects actions to perform (or plans an extended sequence of actions) on the basis of what is known and what the current tasks are. According to this approach, errors arise through the user having incomplete or incorrect knowledge.

A standard PUM analysis consists of up to five stages, the first three of which are essential, and the final two optional (Blandford, Good & Young, 1997):

1. The analyst starts by defining some task scenarios. These tasks are ones that have both domain and device relevance; that is, they are not tasks that relate solely to the device (e.g. “press a button”), but neither are they tasks that involve substantial knowledge about the context of use or organisational goals. Such tasks are typically quite small, and described at a detailed level.

2. The second stage is to identify *conceptual operations* to perform the candidate tasks. A conceptual operation corresponds to an action that is selected under certain conditions to achieve a particular purpose.

3. The third stage is to describe the knowledge the user needs in an Instruction Language (IL). It also involves describing the device in similar terms. One important focus of the analysis while writing the

description is defining how users know things. The process of writing this description may highlight sources of potential difficulty without need for further analysis.

4. If no such difficulties are found then the analyst can proceed to hand simulation. This involves giving an account of how users acquire goals, become committed to operations, execute actions, and update their knowledge of the device state. This analysis may highlight further problems.

5. If considered necessary, the Instruction Language model can be compiled to yield a runnable cognitive model to seek further insights. In practice, constructing a running model generally serves as a notional, rather than an actual, target of the analysis, since the cost of constructing one for any sizeable problem outweighs the likely benefits.

The aim of this enterprise is not to construct an artificial user (with all the real-world knowledge such a user would typically bring to bear on the tasks in hand), but to provide the designer or analyst with a means of identifying minimal requirements on the user's knowledge and capabilities, and considering in detail the consequences of there being gaps or inaccuracies in that knowledge.

2.1 PUM and FMEA

In summary, PUM allows us to take a designer's description of a device, a task and assumptions about a user's knowledge of the device and domain and their cognitive processes. These are used to create a model of how a user may use the device. It is important to note that the analysis is based upon *a single task* - in other words it can be considered a *scenario-based technique*. This makes it highly suited to a FMEA-style analysis.

In generating the model we detail the knowledge needed by the user to carry out the task. Having detailed this, we can examine the implications of certain pieces of knowledge being missing. To take a simple example, if a user has to know a password to proceed with an interaction then the interaction will break down if they do not know the password; this can happen if password-protected equipment is used by several users and one of them (intentionally or maliciously) changes the password. This is quite detailed knowledge, but we can also consider more general things such as possible consequences of a user being unfamiliar with a menu structure, or lacking awareness of certain concepts that are crucial to the interaction.

Exploring such gaps in knowledge allow us to perform a user-centred Failure Modes and Effects Analysis that is similar to a hardware or software FMEA. We can:

- define the *failure modes* by omitting in turn each piece of knowledge which could justifiably be missing;
- suggest *causes* of this piece of knowledge being missing (for example, cluttered display);
- work through to the *end effect* on the system (for example, the interaction stalling at a certain point).

2.2 Method

We propose a 5-stage process, which is repeated as necessary as an iterative design process progresses, as follows:

1. Carry out a HAZOP study of the design, and identify aspects of the design where it is possible that the user lacking certain knowledge or believing incorrect "facts" could result in an undesired outcome.

2. Assess whether there are plausible scenarios where this could occur. If so, carry out a UFMEA analysis of this aspect of the system. Otherwise document the potential problem and note that no plausible occurrences could be envisaged so that they can be reconsidered at a later stage.

3. In carrying out the UFMEA, generate a PUM model of this aspect of the system and record any observations made during creation of the model and during hand simulation.

4. Feed this information back to the HAZOP team for them to assess the safety of this aspect in the light of this analysis.

5. If necessary, amend the design for this aspect of the system and repeat the process.

The example hazard analysis of an interactive system presented below is hypothetical, but is based on the design, analysis and re-design of a real system. For reasons of space, we present a very sketchy outline of the analysis in this abstract, omitting all details.

2.3 Example analysis: a 3D movement control system

The system analysed forms part of a mechanical device which moves in three-dimensional space inside a room, controlled by a human operator. The purpose of the system is to prevent the operator moving the equipment in such away that it hits the floor, walls or ceiling, hits another part of itself, or enters a three-dimensional "safe region". The initial design described is that presented in the design documents at a certain stage of the design; the re-design reflects the final implementation.

All of the input values that specify where the device is within the room (relative to walls, floor and ceiling) are configured upon installation and need not be changed unless the device is relocated. The safe region, however, is user-configurable and, because of the way the system is used in practice, the size of this region may be changed quite frequently (many times a day).

During the HAZOP, one of the questions considered is "Are there circumstances in which the safe region might be violated?" During discussion it is decided that there are various ways in which this may be problematic. For each of these, the HAZOP team develops plausible scenarios. These scenarios are used as a basis for PUM analysis which focuses on the operator's knowledge. Results are fed back to the HAZOP team; by this point, one of the concerns has already been addressed in a re-design (by including the presentation of more "size" information on the display); the outstanding concerns are used as a basis for further re-design and re-analysis.

3. Discussion and further work

We have discussed the need for more rigorous consideration of the user in the development of safety-critical systems, and proposed the development of an FMEA-like approach as a step towards this. This approach can be integrated into existing software design practice and can give theory-based insights into potential operator errors, their possible causes and likely results. As this technique is based on one used for analysing hardware and software, the results can be combined to give a "whole system" FMEA.

As the technique is based on the PUM model of cognition, it focuses on issues where errors in user knowledge can affect the operation of a safety critical system. It would certainly be possible to draw upon other psychological approaches, theories and models to widen the analysis to consider other issues, but that is beyond the scope of the current work.

References

Blandford, A., Good, J. & Young, R. M. (1997) *Programmable User Modelling Analysis for usability evaluation*. PUMA Project document WP11 (see http://www.cs.mdx.ac.uk/puma/)

BSI (1991). *Reliability of Systems, Equipment and Components. Part 5: Guide to failure modes, effects and criticality analysis (FMEA and FMECA).* BS 5760 part 5 1991. BSI Standards.

Dearden, A.M. and Harrison, M.D. *Impact and the Design of the Human-Machine Interface.* In Proceedings of COMPASS '96. pp161-170. IEEE.

Kirwan, B. (1992). *Human error identification in human reliability assessment. Part 1: Overview of approaches.* Applied Ergonomics, 23(5). pp299-318.

MOD (undated) *A guideline for Hazop Studies on systems which include a programmable electronic system.* Interim DEF STAN 00-58/1 (Draft).

Newell, A. (1990) *Unified Theories of Cognition*, Harvard University Press, Cambridge, MA.

Reason, J.T. (1990). *Human Error.* Cambridge University Press, Cambridge.

Villemeur, A. (1992). *Reliability, Availability, Maintainability and Safety Assessment,* volume 2. John Wiley & Sons, Chichester.

Young, R.M., Green, T.R.G. & Simon, T. (1989) 'Programmable user models for predictive evaluation of interface designs' in Bice, K. and Lewis, C. (eds.) *Proceedings of CHI '89,* 15-19, New York : ACM.