Appeared in: *Innovation and Consolidation in Aviation*, G. Edkins & P. Pfister (Eds.), Aldershot, UK: Ashgate, 2003, pp. 255-262. ISBN 0 7546 1999 0

Development of hazard analysis techniques for human-computer systems

Andrew Neal, Michael Humphreys, David Leadbetter and Peter Lindsay
The University of Queensland, Australia

Introduction

Human error is known to be responsible for approximately 80% of all system failures within industries such as aviation, power generation, and mining (Hollnagel, 1993). Many of these errors can be traced back to the design of the human-computer or human-machine system. For example, the London Ambulance Service installed a new computerised dispatch system in 1992 resulting in lengthy delays in the dispatch of ambulances to emergencies (Finklestein & Dowell, 1996). A number of the errors were caused by a slow human-computer interface in which exception messages were not prioritised, queues scrolled off the screen with no means of retrieval and duplicated calls were not identified. In order to overcome these types of design problems, a range of techniques have been developed to analyse the potential for human error within safety-critical systems, and to examine the consequences of errors for the system as a whole.

It is interesting to compare the types of techniques that are used for analysing human error, with the hazard analysis techniques that are used for the design and evaluation of hardware and software. International system safety standards – such as in the defence, railways, and process industries – mandate or highly recommend formal (mathematical) modelling of safety-critical aspects of hardware and software functionality (Commonwealth of Australia, 1998; European Committee for Electrotechnical Standardization, 1995; International Electrotechnical Commisson, 1997). Formal models are used for safety assurance with software and hardware systems, because they are precise, systematic, reproducible and auditable.

By contrast, the techniques currently used for modelling and analysing the safety of Human-Computer Interface (HCI) designs, and operator error rates, are informal. One of the most commonly used methods for safety analysis is Failure-Modes and Effects Analysis (FMEA). Two examples are SHERPA (Systematic Human Error Reduction and Prediction Approach) and THERP (Technique for Human Error Rate Prediction: Kirwan, 1994). In such an FMEA, the designer inspects components of the system and identifies possible human failure modes and their potential effects using a “checklist” of common human failure-modes (Hussey, 1998). These approaches require a subject matter expert to estimate the likelihood of different types of errors occurring. Such judgements are frequently difficult to make, and are inherently subjective. Empirical data regarding the frequency of different types of errors is often not available, or is difficult to collect, particularly for systems that are under development.

There are a number of reasons why formal models are not currently used for modelling the performance of human operators within safety-critical systems. These include:

- difficulty formally modelling the interaction between operators and the computer;
- lack of understanding of the psychological processes responsible for operator error;
- inability to formally specify the antecedent conditions that trigger those processes, and to estimate the resulting likelihood of errors; and
- lack of precise methods for determining system risk due to operator errors.

The aim of the current paper is to describe the first stages of the development of a new methodology for safety assurance. This method includes a formal model of operator performance, and incorporates this model into a formal model of the computer system. The method is designed to be used as a risk analysis tool, allowing the user to estimate the probability of operator errors under different operational scenarios, and evaluate the effect of those errors on the performance of the system as a whole.

The Safe HCI Methodology

The safety assurance methodology involves the following steps:

1. Definition of system concept and scope. This involves identifying the functions that the human-computer system performs, the range of conditions ("operational profiles") that it operates under, and possible system hazards.

2. Modelling of the safety-critical aspects of the system. This involves:

    a) A system risk model, to categorise system failure modes and identify the mechanisms that give rise to system failures and safety hazards and mishaps.

    b) Cognitive models of human error. These models identify the characteristics of the system's operational profile and HCI that enhance the likelihood of error.

    c) A model of human-computer interaction that incorporates the cognitive models of human error within a formal specification of the human-computer system. This model identifies the protective features of the HCI and ways of detecting whether interactions are diverging from safe operation of the system.

3. A series of experiments to collect data for calibrating the risk model and fine-tuning the cognitive models, by varying the operational profiles and measuring the rates of human error, and rates of system hazards.

4. A series of experiments to validate the model of human-computer interaction. This involves using the models to (a) predict error rates and system hazards under a new set of operational profiles, and a range of new HCI design configurations, and (b) empirically testing these hypotheses.

The output of the methodology is a "formula" for calculating system safety risk under varying operational profiles, and recommendations for improving HCI design to reduce risk. One of the key advantages of this type of model is that it allows the analyst to simulate the performance of a system under a wide range of different operational profiles, and design configurations. Given the large variety of potential conditions that any moderately complex system can operate under, it is generally not possible to empirically evaluate the performance of the system under all conditions. Simulation, therefore, provides a powerful and cost-effective tool for risk analysis.

A prototype of this methodology has been developed at The University of Queensland, using a highly simplified air traffic control task. In the following sections, we illustrate the cognitive model that was developed for this task, show how this model can be used to analyse the safety hazards stemming from operator error within this system.

An ATC Case Study

The ATC system

The ATC task that we are using in this research program runs on a personal computer, and is simplified to the point that naïve participants can learn to perform the task with an adequate level of proficiency within a two hour experimental period. The task involves routing aircraft through a series of waypoints, and detecting and preventing conflicts by controlling the speed of the aircraft. The aircraft fly on a small number of fixed routes in straight lines in two dimensions, and the HCI functionality is very simple. Altitude is not represented in this task, and participants do not have control over the route of the aircraft. Participants have to ensure that no aircraft ever violate separation

standards. A five nautical mile separation standard is used in this task. Participants are informed that their primary goal is to ensure that no aircraft pass within five nautical miles of each other. In reality, air traffic controllers are concerned with the orderliness and efficiency of traffic flow, as well as safety. We do not consider these outcomes in the current analysis.

A formal model of this human-computer system has been developed, representing the key functions that are performed by the system as a whole, the states that the system can enter, the nature of the human-computer interface, and the mechanisms by which the operator interacts with the system (see Hussey, Leadbetter, Lindsay, Neal & Humphreys, 2000).

The Cognitive Model

A simplified version of the cognitive model is shown below in Figure 1. Operator performance is modelled as a cyclic process, involving scanning for potential conflicts between aircraft, projecting potential conflicts forward in time to assess whether there will be a violation of separation, prioritising potential conflicts, making a decision, and performing the intended action.

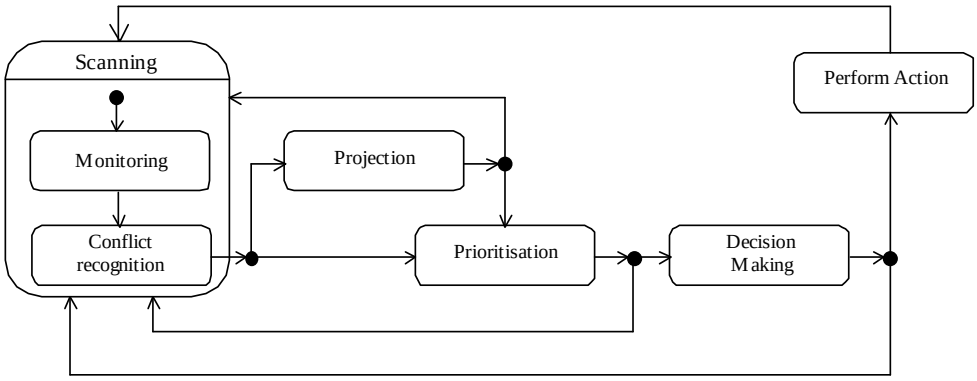

Figure 1: The ATC cognitive model.

The scanning function involves two sub-processes: monitoring and conflict recognition. The participants are assumed to monitor the aircraft within their sector by considering the attributes of pairs of aircraft. These attributes can include the position, speed, and route of either or both aircraft. These attributes can cue conflict recognition in two ways: (1) by retrieving matching examples from memory, and (2) using rules or algorithms. Our participants, therefore, may recognise a conflict because it resembles previously seen conflicts, or because they have developed a set of rules that allow them to calculate whether a pair of aircraft will conflict.

If a potential conflict is recognized, then the participant may either project the event forward in time, or proceed direct to prioritisation. A potential conflict must be projected forward if the participant is unsure when the potential conflict will occur, or if the participant is unsure as to whether the potential conflict is simply a false alarm. By mentally projecting the event forward in time, the participant is able to provide a more accurate estimate as to when and where the conflict may occur. However, if the participant knows when and where the conflict will occur, or the event requires immediate action, the participant proceeds directly to prioritisation.

The prioritisation function assigns a priority to the conflict based on the time that is available for preventative action to be taken. The participant is then assumed to retrieve other current conflicts from short term memory, and to compare the priority of the currently attended conflict with the priorities of other conflicts. If the currently attended conflict has the highest priority, they continue to the decision process, otherwise they return to scanning in order to pick up the higher priority event.

There are two potential outcomes of the decision process: a decision is taken to change the speed of one of the aircraft immediately to prevent the conflict, or the decision is deferred until a later time. If the decision is deferred, then the participant returns to scanning. If the decision is taken, then the participant proceeds to perform the action. The actions are then taken through the HMI, and the participant returns to scanning.

The Error Model

The cognitive model provides a systematic basis for evaluating the potential operator errors within this simplified air traffic control task. Each component within the model is examined to assess the types of failure modes that are possible. Some of the failure modes associated with these components are shown in Figure 2, below.

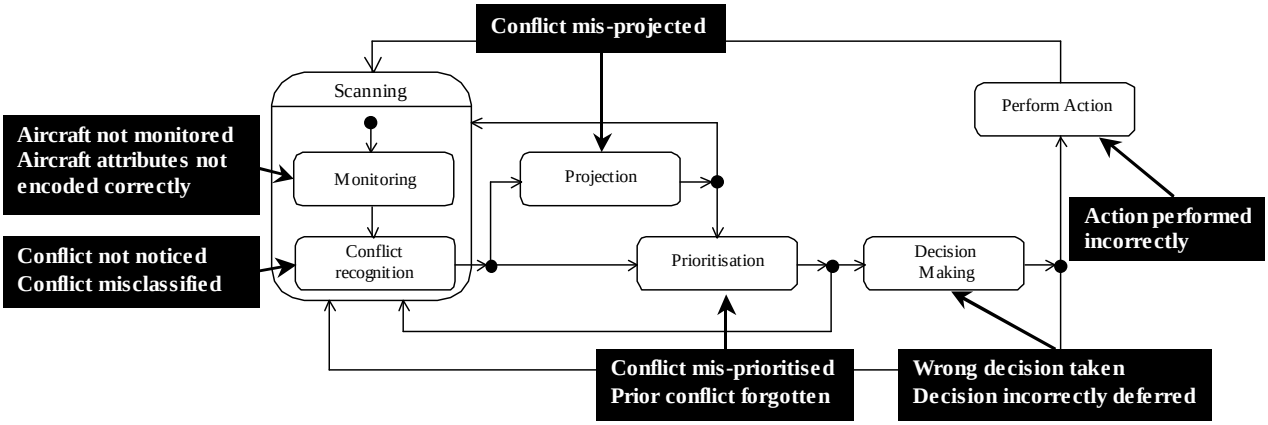

Figure 2: Potential failure modes for each component of the cognitive model.

The principle types of errors associated with monitoring are likely to be a failure to monitor aircraft that are in potential conflict, and a failure to encode the attributes of the aircraft correctly. These types of errors are known to be a major problem for novice air traffic controllers, who are prone to so-called "tunnel vision". Tunnel vision occurs when controllers focus on one highly demanding problem, and fail to systematically scan for other potential problems. These types of errors may also occur under low workload conditions, if the participant has previously attended to the aircraft, but not noticed a conflict.

Conflict recognition can fail in at least two ways: the conflict may not be noticed, or it may be misclassified. These errors can be caused by failures in monitoring, but can also occur even when the participant is actually attending to the aircraft in conflict. For example, a participant may attend to two aircraft that are in potential conflict, but he or she may have seen a number of examples of aircraft in similar circumstances that were not in conflict. In this case, the prior examples in memory may cause the participant to misclassify the event as a "near miss".

The principle error associated with projection is mis-estimation. If a participant projects a pair of aircraft forward in time, they may estimate the separation between the aircraft incorrectly, or they may estimate the time at which the aircraft pass a specific point (eg the point of minimum separation) incorrectly.

Participants can make errors in prioritisation by assigning the wrong priority to the currently attended conflict, or by failing to retrieve one or more of the other current conflicts that are stored in short term memory. The result of this is that the participant either continues to work on the current conflict, when there is a more urgent conflict that needs attending to, or the participant switches to another conflict which should have lower priority.

The major errors associated with the later stages of the model include taking the wrong decision (eg selecting the wrong speed), incorrectly deferring the decision (eg because the estimated time until separation is violated is incorrect), and incorrectly performing the action (eg by selecting the wrong aircraft with the mouse).

Having defined the major types of errors that are likely to affect each component of the model, the next step is to empirically estimate the frequency of these errors, and to identify the major factors that modify the probability of error. The frequency of each error type can be estimated empirically using standard experimental techniques. For example, the baseline error rate for conflict identification can be assessed by asking participants to perform a conflict identification task. This involves asking participants to indicate when they recognise a conflict by pressing a response key as soon as possible. Similarly, the baseline error rate for projection can be estimated by giving participants specific problems, and asking them to project forward.

These experimental studies can also be used to empirically estimate the effects of external factors on error rates. Examples of external factors that may affect the probability of error include:

- Expertise
- Workload
- Memory load
- Fatigue
- HMI design

HCI Redesign

Once developed, the error model can be used to systematically evaluate alternate HCI design options. By integrating the error model into the formal specification of the human-computer system, it is possible to identify the operator error modes that pose the greatest hazard to the safe operation of the system, and the operational conditions under which these errors are most likely to occur. Specific HCI designs can then be developed to address these problems, and the redesigns can be run through the model. For example, our preliminary testing of the ATC task suggests that conflict recognition represents a significant source of error for our participants. There are a range of design options that could be developed to address this problem. These include conflict detection probes that automatically alert participants to potential conflicts, and flight projection tools that allow participants to accurately estimate the time at which aircraft pass specific points. Each design option is evaluated by considering the effect that it should have on each error mode, and re-running the model to see if it significantly enhances the predicted safety score for the system (see Figure 3).

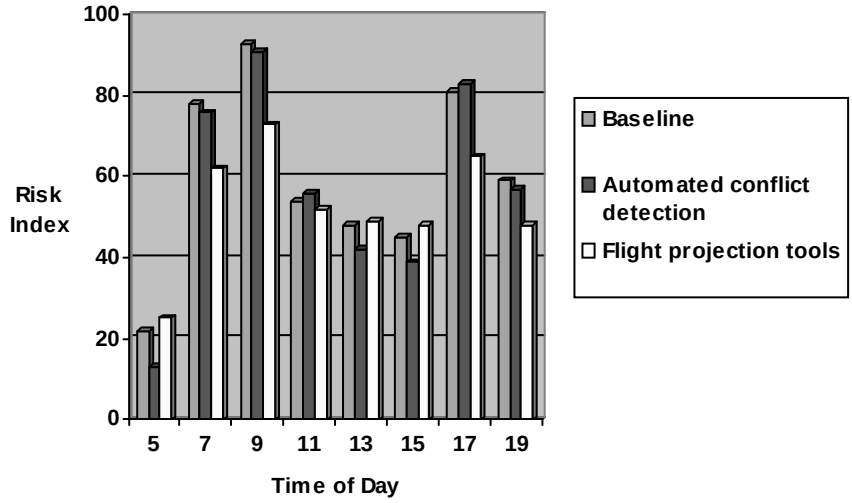

Figure 3: Illustrative example of a risk analysis for alternate HCI designs.

Conclusion

In summary, the current paper has illustrated the ways in which a model of operator cognition can be used for evaluating the potential for human error within human-computer systems. This approach is promising, because it allows the user to simulate the operation of the system as a whole under a wide range of different conditions. This technique can be expanded to incorporate cost models in order to provide a comprehensive evaluation of the cost effectiveness of a range of interventions designed to enhance safety. In this manner, it is possible to compare the cost effectiveness of options, such as redesigning the HCI, changing staffing levels, or providing more training.

References

Commonwealth of Australia. (1998). *Australian Defence Standard DEF (AUST) 5679: The Procurement of Computer-based Safety Critical Systems.* Department of Defence, Canberra.

European Committee for Electrotechnical Standardization. *European Standard prEN 50128: Railway applications; Software for railway control and protection systems.* CENELEC.

Finklestein, A. & Dowell, J. (1996). A Comedy of Errors: the London Ambulance Service case study. In Proceedings of the 8th International Workshop on Software Specification and Design (pp 2-4), IEEE.

Hollnagel, E. (1993). Human Reliability Analysis: Context and Control. Academic Press Limited.

Hussey, A. (1998). Safety Analysis of User-interfaces at Multiple Levels of Interaction, In *Proceedings of the 3rd Australian Workshop on Industrial Experience with Safety Critical Systems and Software* (pp. 41-57) Australian Computer Society.

Hussey, A., Leadbetter, D. Lindsay, P., Neal, A. & Humphreys, M. (2000). *A Method for Analyzing Hazards and Error Rates Related to Operator Activities.* Software Verification Research Centre Technical Report No. 00-25.

International Electrotechnical Commission. *61508: Functional safety of electrical/electronic/programmable electronic safety-related systems.* IEC.

Kirwan, B. (1994). A Guide to Practical Human Reliability Analysis. Taylor and Francis.