SOFTWARE VERIFICATION RESEARCH CENTRE

SCHOOL OF INFORMATION TECHNOLOGY

THE UNIVERSITY OF QUEENSLAND

Queensland 4072
Australia

TECHNICAL REPORT

No. 00-18

Hazard Analysis of Interactive Systems

Andrew Hussey
Brenton Atchison

May 2000

Phone: +61 7 3365 1003
Fax: +61 7 3365 1533

Hazard Analysis of Interactive Systems

Andrew Hussey
Brenton Atchison

Software Verification Research Centre,
School of Information Technology,
University of Queensland, Australia
email: {ahussey,brenton}@svrc.uq.edu.au

Abstract

This report discusses approaches to analysis of safety-critical systems for operator error. The report summarises the existing literature in the area as well as the guidance provided by existing safety-critical system development standards.

Keywords: safety-critical, interactive system, hazard analysis

1 Introduction

This report provides guidance for performing a hazard analysis focussing on operator errors for an interactive system. There are four key steps to performing such a hazard analysis.

1. Task analysis
2. Human error analysis
3. Error reduction measures
4. Residual risk quantification

In the next section we present a generic method for operator safety case preparation that considers each of these steps in turn. Section 3 discusses the literature supporting the method, and we conclude in section 4 by summarising some of the current research issues in the area.

2 Operator Safety Case Preparation

A general method for operator safety case preparation is now presented. The method is derived from a survey of current approaches. More information is presented in Section 3.

Operators are considered to be an integral part of the system, interacting with safety-critical machines via operator interfaces. The operator, machine and operator interface are treated as separate components of the system (see Figure 1), with information passing between the machine and the operator via the operator interface. Whereas operators are concerned with achieving task goals correctly, the operator interface is concerned with correct mediation of information between the operator and machine. Operator and interface safety requirements are determined through analysis of their influence on the operational system.

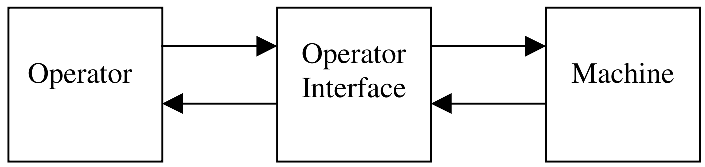

Figure1: The Operator, Operator Interface and Machine Components

The safety-case for the operator components of a safety-critical system should demonstrate that the component safety requirements for the operator are satisfied by the design of the operator procedures, in terms of goals, task steps and concrete operator actions. Further, it should provide confidence that procedures will be free from operator error by analysis of operator characteristics (e.g., skill level and training) and the characteristics of the workplace of which the operator is a part (e.g., noise levels and lighting). The assessment process consists of the following steps.

1. **Task analysis**. The task analysis requires documentation of the procedures (tasks) that the operator performs. The task analysis should provide information on the steps undertaken to perform a procedure, the objects that the operator interacts with in carrying out their duties and the outcomes of the procedure (goals). The task analysis should also give details of the actions that the operator engages in with the operator interface to execute each task step. We discuss task analysis methods further in Section 3.1 of this report.

2. **Human error analysis**. The task analysis should be analysed for potentially hazardous failure mechanisms using known operator error types as a guide. The analysis should investigate both causes and consequences of operator errors. A typical approach is to use a HAZOP style analysis with known error types in place of keywords. Errors are then mapped to system hazards that can result. We discuss error analysis of operator procedures further in Section 3.2 of this report.

3. **Error Reduction**. Once knowledge of operator errors and resultant hazards is gained, measures are identified for controlling (reducing the occurrence of, or preventing) operator error. Typical strategies for error control are as follows.

    1. Prevention by hardware or software changes: automation of tasks and prevention of error by use of interlock devices and behavioural “forcing functions”.
    2. Enhanced error recovery: provide feedback, checking procedures, supervision and automatic monitoring of performance.
    3. Reduce errors at source: improve procedures, training, work environment and interface design.

    The measures identified may relate to the operator or operator interface component. Error reduction measures that are implemented as procedures (e.g., checking and supervision), requirements on the work environment and required operator characteristics (e.g., skill and training) are concerned with the design of the operator component. Measures concerning interface design are concerned with the operator interface component. We discuss techniques for error rate reduction in Section 3.3.

4. **Residual risk quantification**. An assessment is made that the risk associated with operator error in the operational system is acceptably low. For some systems, it is possible to perform such an assessment in a quantitative way, if accurate measures of operator failure rates are known. We discuss quantitative

assessment in detail in Section 3.4.2 of this report. Usually however, it is necessary to base such an assessment on knowledge of factors influencing operator performance. We call these factors Human Performance Factors (HPFs). Such factors include characteristics of the operators and the tasks they perform as well as environmental and workplace conditions.

The Human Performance Factors should be identified and assessed by their influence on operator error. Where the assessment is based on assumed operator or workplace characteristics, further evidence is required that those characteristics can be satisfied for the operational life of the system.

3 Guidelines

3.1 Task Analysis

Tasks are goal-directed activities to transform some given initial state into a goal state. A task can be decomposed into sub-tasks unless the task is itself composed only of elementary actions. Higher-level tasks represent goals and intentions in the task domain [Duke95]. Likewise Carroll and Rosson have used scenarios as design representations [Carroll90]. Each elementary action is concerned with a manipulation to be performed upon an object in the task domain. The task domain is the set of tasks correlated by an overall goal [Stary95] and the objects and elementary actions that compose those tasks.

3.1.1 Purpose of the task analysis

User Task Models (UTM's) facilitate understanding of users and the tasks that are performed in a given domain [Johnson90]. Such models allow identification of requirements and analysis of designs for new requirements and user training needs [Johnson90]. Task models examine the knowledge required to operate a system so that a certain class of tasks may be executed [Hoppe90]. Task models have been referred to as *competence* models because they summarise the knowledge underlying human information processing in human-machine interaction [Hoppe90]. UTMs include Task Knowledge Structures (described below).

Task models alone are often incomplete and ambiguous [Palanque95]. Formal methods that exploit task models are more likely to produce understandable specifications of systems than methods that do not [Bass94], and are less likely to suffer from the completeness and consistency problems of informal task models. Much research has been conducted on the topic of formal notations for describing the allowable action sequence (traces) of an interactive system, e.g., LOTOS [Paterno97], Petri-nets [Palanque96], Object-Z [Hussey98], CSP [Alexander90] and such formal notations are now commonly used for task descriptions.

For the purpose of safety-critical systems, the task analysis must describe procedures for normal operation of the system, maintenance procedures and also procedures for emergency situations [Redmill97]. The description of procedures for normal operation and maintenance should include any recovery steps by which errors of the user are detected and corrected to avoid an accident [Kirwan92]. The task analysis must describe both the logical sequence of actions that the operator engages in, and the detailed physical executions that the operator must perform in order to effect each step in the procedure.

3.1.2 Knowledge Analysis of Tasks

We describe a particular task-analysis method, "Knowledge Analysis of Tasks" (KAT), in more detail. KAT is similar to the GOMS method [Card83]. GOMS stands for Goals (user goals), Operators (elementary actions which compose to form tasks), Methods (procedures for achieving goals, i.e., composed of operators) and Selection rules (for selecting which operator to use). GOMS models the interface from the user perspective, i.e., what the user needs to do to achieve certain results.

KAT is a task analysis method based on task knowledge structures. A task knowledge structure defines all the knowledge a user has of a task. A role defines a collection of tasks that a person occupying that role performs. Within a role, each task will have its own task knowledge structure. Between roles, tasks may have similar task knowledge structures. A task knowledge structure has four components:

1. goals;
2. task procedures;
3. actions and objects; and
4. summary.

The goal element identifies the goals and sub-goals associated with a task. A goal is a state that the user desires. Associated with goals is a plan that maps a path through a series of sub-goals to the desired end goal. Task procedures are executable and define the actions needed to achieve a sub-goal or goal. Task procedures rely on knowledge of objects and actions. Objects are identified by a taxonomic sub-structure that defines object properties and attributes, e.g., class membership, procedures in which the object is used, relationship to other objects and actions that involve it.

KAT identifies the elements of knowledge in a task knowledge structure. The analysis is composed of three activities:

1. collecting data;
2. analysing data; and
3. modelling the task domain.

Knowledge gathering techniques include structured interviews, questionnaires, observation, concurrent and retrospective protocols (verbalisations by the subject of their goals, plans, procedures, objects and actions during and after performing the task). Knowledge is categorised as actions, objects, goals, sub-goals, procedures or actions. Task knowledge components may then be structured in terms of their representativeness and centrality to the task.

3.2 Human Error Analysis

3.2.1 Types of operator error

Phenomenological taxonomies describe errors superficially in terms that refer to observable events. Categories of error include omissions, substitutions and repetitions (the latter two are *commission* errors) [Senders91]. Redmill [Redmill97] lists the following categories of error:

- Action or check made too early or too late;
- Action or check omitted or partially omitted;
- Action too much or too little;
- Action too long or too short;
- Action in wrong direction;
- Right action or check on wrong object;
- Wrong action or check on right object;
- Information not obtained;
- Wrong information obtained; and
- Misalignment (synchronisation error between human and machine).

In addition to the unintentional errors discussed in this Section, accidents may also arise as a result of violations (intentional infringements of safe working practices). However, such violations are not generally a significant source of accidents in the Defence services because Defence personnel are generally highly trained and disciplined.

3.2.2 Determining potentially hazardous operator error

HAZOP (e.g., [Std00-58]) and FMEA (e.g., [StdIEC-1025]) are the predominant techniques for analysing human error. Both techniques use a task analysis as the model of the system that is analysed. HAZOP has been applied widely in the research community to analyse human error (e.g., [Chudleigh93], [Kirwan94], [Leathley97]). Because it is possible to specifically list the expected error types and mechanisms, FMEA has been used in many of the current methods for human error analysis including HEART (Human Error Assessment and Reduction Technique) [Williams86] and THERP (Technique for Human Error Rate Prediction) [Swain83]. These techniques give good analyses of skill and rule-based errors but have difficulty with cognitive decision-making (knowledge-based) errors [Kirwan90]. Because FMEA is widely used in the safety industry, we do not describe the method in further detail here. A typical set of operator error modes is listed under Section 3.2.1 above.

3.2.3 Error mechanisms

Operator error types do not describe the proximate causes of operator failure in terms of human cognition. We summarise the error mechanisms, as given by Reason and Embrey [Reason86]; Whalley [Whalley88] has also compiled a similar list of error mechanisms:

- Failure to consider special circumstances;
- Short cut invoked;
- Stereotype takeover;
- Need for information not prompted;
- Misinterpretation of display;
- Assumption by operator;
- Forget isolated act;
- Mistake among alternatives;
- Place losing error;
- Other slip of memory;
- Motor variability; and
- Topographic or spatial orientation inadequate.

Norman [Norman90a] gives a model of human-machine interaction that he refers to as the “execution-evaluation” model. In Norman's view, errors occur at one or more stages in this cycle of activity; each stage of the cycle defines an activity of the human in the user-machine interaction. Rasmussen [Rasmussen83] has refined Norman’s model to produce the “step-ladder” model. The stepladder model identifies eight stages of decision making (see Figure 2). A goal (e.g., to retain one’s work for later reference) is satisfied by sub-goals (e.g., to save the work to disk). A goal is specified as a procedure (e.g., open the file menu and select the save option) and at a more concrete level as an execution sequence (i.e., in terms of concrete physical actions). The user observes changes to the world as a result of their actions. These are identified with goals, interpreted in terms of their effect and that interpreted effect is evaluated against the goal to determine whether the goal has been satisfied.

Norman categorises errors into two types; slips and mistakes. Slips are concerned with automatic behaviour at the physical execution level. Mistakes are the result of conscious deliberation; a “wrong” procedure is formulated. Similarly, Reason [Reason90] distinguishes between skill-based slips at the execution level and rule-based or knowledge-based mistakes at the procedure level.

Rule-based mistakes: Choice of a wrong strategy to perform a task; primarily caused by memory failures.

Knowledge-based mistakes: Construction of a wrong strategy when a rule is not known.

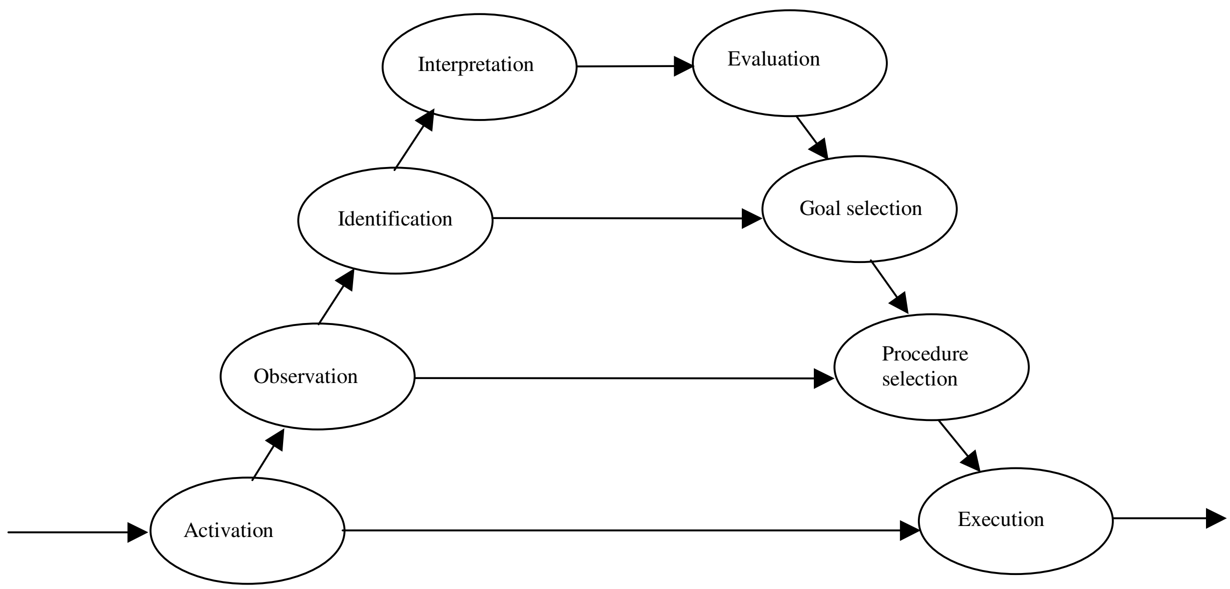

Figure 2: Rasmussen's "step-ladder" model

Errors arise when decision makers take short-cuts in the decision process, e.g., using rule-based routines when knowledge-based decision is demanded by the novelty of a situation [Reason86] (in Figure 2, a short-cut from Observation to Procedure selection, omitting the "higher" knowledge-based decision processes).

Norman describes three main classes of slips; we summarise each and give examples.

Capture errors:
Capture errors occur whenever two different action sequences have their initial stages in common, with one sequence being unfamiliar and the other being well practised. It is rare for the unfamiliar sequence to capture the familiar one, but common for the familiar sequence to "capture" the unfamiliar. For example, driving to the office rather than the store on a Sunday (the *intent* to go to the store remains unchanged).

Description errors:
Description errors are similar to capture errors but result in performing the right action on the wrong object, because the wrong object shares some characteristics with the correct object (as opposed to performing the wrong action on the right object, which is characteristic of capture errors). Description errors occur most frequently when the wrong and right objects are physically near each other. For example, pouring coffee into a nearby glass rather than the *intended* coffee cup (the action sequence fails to distinguish cup from glass).

Mode errors:
Mode errors occur when devices have various modes of operation and an action has different meanings according to the mode. Mode errors are particularly likely when the mode is not visible. For example, pressing a button on a multi-function calculator when it is in trigonometric mode, rather than the expected and desired "normal" mode.

3.3 Error reduction

In the case of an error occurring, the user-interface should provide simple mechanisms to correct the system state before hazards or accidents result (i.e., undo mechanisms). A more effective solution is to prevent the error occurring. The likelihood of slips can be reduced by considering the tasks provided, physical actions to perform task steps, the information provided and the way in which that information is presented. A user-interface can be designed to reduce the opportunity for slips by minimising like execution prefixes (particularly between frequently used non-critical sequences and less frequently used, safety-critical sequences). Norman [Norman90a] calls features that *prevent* slips or mistakes "forcing functions" because they force a user to choose a safe sequence of actions.

The strategies to address operator errors have been summarised by Kirwan [Kirwan90]:

1. **Prevention by hardware or software changes**: automation of tasks and use of interlock devices and behavioural "forcing functions" to prevent error.
2. **Enhanced error recovery**: provide feedback, checking procedures, supervision and automatic monitoring of performance.
3. **Reduce errors at source**: improve procedures, training and interface design.

Automation alone is not sufficient to remove human error. Bainbridge [Redmill97] describes the following issues concerning such automation:

- the system is more complicated hence the designer has more potential to introduce errors into the system during the design process;
- the operator still has to do those tasks that the designer could not automate;
- the operator's role is supervisory but with reduced information about the system;
- the operator may become de-skilled but is expected to intervene when the automated system fails.

Bainbridge [Bainbridge87] maintains that it is not possible for even a highly motivated user to maintain attention toward a source of information on which little happens for more than half an hour. Hence it is humanly impossible to carry out the basic monitoring function needed to detect unlikely abnormalities. In addition, the operator will rarely be able to check in real-time the decisions made by the computer, instead relying on a meta-level analysis of the `acceptability' of the computers actions. Such monitoring for abnormalities must therefore be done by the system itself and abnormalities brought to the operator's attention via alarms. In general, a function should be automated if [Mill92]:

- performing the function involves danger to an operator [Mill92];
- performing the function requires exceptional skill, e.g., when the response time is far shorter than a human can normally achieve;
- performing the function requires tedious or repetitive work.

In general a function should not be fully automated (i.e., a human should be included in the control loop with responsibility for decisions) if a decision must be made that:

- cannot be reduced to uncomplicated algorithms;
- involves fuzzy logic or qualitative evaluation;
- requires shape or pattern recognition.

Appropriate design of automatic systems should assume the existence of error, continually provide feedback, continually interact with operators in an effective manner and allow for the worst situations possible [Norman90b].

3.4 Risk Assessment

3.4.1 Qualitative Assessment

Errors arise from characteristics of the operator interface, the individual, human cognition generally and the organisation. Rasmussen [Rasmussen82] calls such characteristics “Performance Shaping Factors” but in this report, we use the term Human Performance Factors (HPFs) to emphasise that the factors affect *human* performance. HPFs capture the underlying causes of operator error and are triggers activating error mechanisms. In Reason’s GEMS (Generic Error Modelling System) model [Reason86] HPFs are associated with skill, rule and knowledge-based activities. Redmill [Redmill97]has produced a categorised list of HPFs:

> **Task demands and characteristics:** Frequency, workload, duration, interaction with other tasks, perceptual, physical, memory, attention required, vigilance required.
>
> **Instructions and procedures:** Accuracy, sufficiency, clarity, level of detail, meaning, readability, ease of use, applicability, format, selection and location, revision.
>
> **Environment:** Temperature, humidity, noise, vibration, lighting, work space, movement restriction, operator control of environment.
>
> **Displays and controls:** Compatibility, ease of operation, reliability, feedback, sufficiency, location, readability, identification, distinctiveness.
>
> **Stresses:** Time pressure, workload, fatigue, monotony, isolation, distractions, shift work incentives.
>
> **Individual:** Capacities, training and experience, skills and knowledge, personality, physical condition, attitudes, motivation, risk perception.
>
> **Socio-technical:** Staffing adequacy, work hours and breaks, resource availability, social pressures, conflicts, team structure, communications, roles and responsibilities, rewards and benefits, attitude to safety.

Many human errors are a product of the work context, e.g., poorly designed procedures, unclear allocation of responsibilities, lack of knowledge of training, low morale, poor equipment design, time pressures etc. The work context can be traced to organisational policies and decisions. Such organisational decisions are latent failures that contribute indirectly to accidents [Reason90]. The focus is on management decision making, safety management and issues such as safety culture, participation, competence, control and communication [Redmill97].

3.4.2 Quantitative Assessment

Human reliability quantification techniques aim to quantify the Human Error Probability (HEP) which is defined as: number of errors per number of opportunities for error. Numerous quantification methods have been devised, including HEART, THERP and SLIM (Success Likelihood Index Method) [Embrey84]. HPFs similar to those given by Reason [Reason86] appear as a factor in most of the available estimation methods, e.g., HEART, THERP [Kirwan90]. Estimation methods such as THERP use event trees with recovery paths to take account of the potential for operators to recover from previous errors. Screening of potential human errors helps identify where the major effort in the quantification analysis should be applied. For example, the SHARP (Systematic Human Action Reliability Procedure) method screens out human errors that are unlikely to lead to accidents by considering co-effectors, effects of errors and broad probabilities for classes of error [Kirwan90].

As an example of a human reliability quantification technique, we briefly describe the THERP method, which is widely used in industry [Kirwan90]. The quantification part of THERP is composed of:

1. A database of human errors and their likelihoods;
2. A dependency model which calculates the degree of dependence between two operator actions;
3. An event tree approach to producing HEPs;
4. Assessment of error recovery paths.

An event tree is constructed for the task under consideration, representing the sequence of events and possible failures at each branch. Nominal HEPs are assigned to errors using the database. Human Performance Factors are applied to modify the nominal HEPs provided by the database. Error recovery paths are added to the tree where possible and the overall error rate for the task is quantified.

A major weakness of most of the likelihood prediction methods is the data source for error rate prediction. If a database of error rates is used, as in HEART and THERP, the applicability of such a database to the particular task that is being analysed is questionable. A database of error likelihoods in combination with HPFs is likely to produce only a very rough estimate of the probability of a particular class of human error. Use of expert judgement to determine likelihood, as in SLIM and Absolute Probability Judgement, also results in poor reproducibility and requires that a rationale be recorded for the likelihoods that are generated. Because the error likelihood estimates produced by current methods suffer from low accuracy, they should be regarded as providing a ranking of error reduction measures, rather than as a measure of achievement of error likelihood goals.

4 Outstanding Research Issues

Truly accurate methods for predicting human error rates are yet to emerge. While databases of error likelihoods are relatively straightforward to apply, they can only give rough estimates of the likely error rate for any particular circumstance. Expert judgement may take account of particular circumstances better, but is likely to exhibit significant variation, and the effort required to apply methods involving expert judgement is likely to be much greater.

Automation also presents a challenge to system designers. On the one hand, automation is necessary for the safe operation of many systems such as aircraft and power plants. However, when a failure does occur, automation can mask the defect, allowing consequences to magnify until the automated system can no longer retain control. The challenge for designers of safety-critical systems is to automate without impoverishing the operator’s mental model. The operator needs to retain sufficient understanding of the system and sufficient skill to correct failures when they occur, before hazards are realised; to some extent, designers are still grappling with how to do this.

5 Acknowledgements

We gratefully acknowledge the support of the Department of Defence which funded this work under the Defence Software Acquisition Reform project which is investigating enhancements to the Australian Defence Standard Def(Aust) 5679 as part of an initiative to improve techniques for safety-critical system development in the Australian defence industry.

6 References

[Alexander90] H. Alexander. Structuring Dialogues using CSP, In M. Harrison and H. Thimbleby, editors, *Formal Methods in Human-Computer Interaction*, chapter 9, pages 274-295, Cambridge University Press, 1990.

[Bainbridge87] L. Bainbridge. Ironies of Automation, In J. Rasmussen, K. Duncan and J. Leplat, editors, New Technology and Human Error, chapter 24, pages 271-283, John Wiley and Sons, 1987.

[Bass94] L. Bass. Working Group on Formal Methods in HCI and Software Engineering, In R. N. Taylor and J. Coutaz, editors, *ICSE Workshop on SE-HCI: Joint Research Issues: Sorrento, Italy*, pages 14-16, Springer-Verlag, 1994.

[Card83] S. Card, T. P. Morgan and A. Newell. *The Psychology of Human-Computer Interaction*. Lawrence Erlbaum Associates, 1983.

[Carroll90] J. M. Carroll and M. B, Rosson. Human-Computer Interaction Scenarios as a Design Representation, In *HICSS-23: Hawaii International Conference on System Sciences*, pages 556-561, IEEE Computer Society Press, 1990.

[Chudleigh93] M. F. Chudleigh and J. N. Clare. The benefits of SUSI: Safety Analysis of User System Interaction, In J. Gorski, editor, SAFECOMP'93: Proceedings of the 12th International Conference on Computer Safety, Reliability and Security, pages 123-132, Springer-Verlag, 1993.

[Duke95] D. J. Duke and M. D. Harrison. Mapping User Requirements to Implementations. *Software Engineering Journal*, 10(1): 13-20, 1995.

[Embrey84] D. E. Embrey, P. Humphreys, E. A. Rosa, B. Kirwan and K. Rea. *SLIM-MAUD: an Approach to Assessing Human Error Probabilities Using Structured Expert Judgement*. USNRC Report Nureg/CR-3518. Washington, DC: USNRC, 1984.

[Hoppe90] H. U. Hoppe. A Grammar-Based Approach to Unifying Task-Oriented and System-Oriented Interface Descriptions, In D. Ackermann and M. J. Tauber, editors, *Mental Models and Human-Computer Interaction 1*, pages 353-373, Elsevier Science, 1990.

[Hussey98] A. Hussey and D. Carrington. Which widgets? Deriving implementations from formal user-interface specifications, In P. Markopoulos and D. Duce, editors, *DSV-IS '98*, pages 239-258, Springer, 1998.

[Johnson90] P. Johnson, K. Drake and S. Wilson. A Framework for Integrating UIMS and User Task Models in the Design of User Interfaces, In D. A. Duce and M. R. Gomes and F. R. A. Hopgood and J. R. Lee, editors, *User Interface Management and Design: Proceedings of the Workshop on User Interface Management Systems and Environments*, chapter 20, pages 203-216, Springer-Verlag, 1990.

[Kirwan90] B. Kirwan. Human Reliability Assessment, In J. Wilson and E. N. Corlett, editors, *Evaluation of Human Work*, chapter 28, Taylor and Francis, 1990.

[Kirwan92] B. Kirwan and L. K. Ainsworth. *A Guide to Task Analysis*. Taylor and Francis, 1992.

[Kirwan94] B. Kirwan. *A Guide to Practical Human Reliability Analysis*. Taylor and Francis, 1994.

[Leathley97] B. A. Leathley. HAZOP Approach to Allocation of Function in Safety-critical Systems. In E. Fallon, M. Hogan, L. Bannon and J. McCarthy, *ALLFN'97: Vol 1*, pages 331-343. IEA Press, 1997.

[Mill92] R. C. Mill. *Human Factors in Process Operations*. Institution of Chemical Engineers, 1992.

[Norman90a] D. A. Norman. The Design of Everyday Things. Doubleday, 1990.

[Norman90b] D. A. Norman. The 'problem' with automation: inappropriate feedback and interaction, not 'over-automation', *Philosophical Transactions of the Royal Society of London, Series B*, 327(1241): 585-593, 1990.

[Palanque95] P. Palanque, R. Bastide and V. Senges. Validating Interactive System Design through the Verification of Formal Task and System Models, In L. J. Bass and C. Unger, editors, *Engineering for Human-Computer Interaction*, chapter 11, pages 189-212, Chapman and Hall, 1995.

[Palanque96] P. Palanque and R. Bastide. Task Models – System Models: a Formal Bridge over the Gap. In D. Benyon and P. Palanque, *Critical Issues in User Interface Systems Engineering*, chapter 4, pages 65-79. Springer, 1996.

[Paterno97] F. Paterno, C. Mancini and S. Meniconi. ConcurTaskTrees: A Diagrammatic Notation for Specifying Task Models, In, S. Howard, J. Hammond and G. Lindgaard, editors, *Human-Computer Interaction INTERACT'97*, pages 362-369, Chapman and Hall, 1997.

[Rasmussen82] J. Rasmussen. Human errors: A taxonomy for describing human malfunction in industrial installations. *Journal of Occupational Accidents*, 4: 311-335, 1982.

[Rasmussen83] J. Rasmussen. Skills, Rules and Knowledge: Signals, Signs and Symbols and Other Distinctions in Human Performance Models. *IEEE Transactions on Systems, Man and Cybernetics*, SMC-13(3): 257-266, 1983.

[Reason86] J. Reason and D. Embrey. Human Factors Principles Relevant to the Modelling of Human Errors in Abnormal Conditions of Nuclear and Major Hazardous Installations. Report for the European Atomic Energy Community, 1986.

[Reason90] J. Reason. *Human Error*. Cambridge University Press, 1990.

[Redmill97] F. Redmill and J. Rajan, editors. *Human Factors in Safety-Critical Systems*. Butterworth Heinemann, 1997.

[Senders91] J. W. Senders and N. P. Moray. *Human Error: Cause, Prediction and Reduction*. Lawrence Erlbaum Associates, 1991.

[Stary95] C. Stary and A. Pasztor. LUIS – A Logic for Task-Oriented User Interface Specification. *International Journal of Intelligent Systems*, 10(2): 201-231, 1995.

[Std00-58] UK Ministry of Defence , Draft Interim Defence Standard 00-58/1: A Guideline for HAZOP Studies on Systems which include a Programmable Electronic System, 1995.

[StdIEC-1025] International Electrotechnical Commission, International Standard CEI IEC 1025. Fault Tree Analysis, 1990.

[Swain83] A. D. Swain and H. E. Guttmann. *A Handbook of Human Reliability Analysis and Emphasis on Nuclear Power Plant Applications*. USNRC Report Nureg/CR-1278. Washington, DC: USNRC, 1983.

[Whalley88] S. P. Whalley. Minimising the cause of human error. In G. P. Libberton, editor, *10th Advances in Reliability Technology Symposium*. Elsevier, 1988.

[Williams86] J. C. Williams. HEART – a proposed method for assessing and reducing human error. In *Proceedings of the 9th Advances in Reliability Technology Symposium*. University of Bradford, 1986