Developments in Risk-based Approaches to Safety

Proceedings of the Fourteenth Safety-critical Systems Symposium, Bristol, UK, 7-9 February 2006

Edited by
Felix Redmill and
Tom Anderson

Springer

Safety-Critical Systems Club

Developments in Risk-based Approaches to Safety

Related titles:

Towards System Safety
Proceedings of the Seventh Safety-critical Systems Symposium, Huntingdon, UK, 1999
Redmill and Anderson (Eds)
1-85233-064-3

Lessons in System Safety
Proceedings of the Eighth Safety-critical Systems Symposium, Southampton, UK, 2000
Redmill and Anderson (Eds)
1-85233-249-2

Aspects of Safety Management
Proceedings of the Ninth Safety-critical Systems Symposium, Bristol, UK, 2001
Redmill and Anderson (Eds)
1-85233-411-8

Components of System Safety
Proceedings of the Tenth Safety-critical Systems Symposium, Southampton, UK, 2002
Redmill and Anderson (Eds)
1-85233-561-0

Current Issues in Safety-critical Systems
Proceedings of the Eleventh Safety-critical Systems Symposium, Bristol, UK, 2003
Redmill and Anderson (Eds)
1-85233-696-X

Practical Elements of Safety
Proceedings of the Twelfth Safety-critical Systems Symposium, Birmingham, UK, 2004
Redmill and Anderson (Eds)
1-85233-800-8

Constituents of Modern System-safety Thinking
Proceedings of the Thirteenth Safety-critical Systems Symposium, Southampton, UK, 2005
Redmill and Anderson (Eds)
1-85233-952-7

Felix Redmill and Tom Anderson (Eds)

Developments in Risk-based Approaches to Safety

Proceedings of the Fourteenth Safety-critical Systems Symposium, Bristol, UK, 7-9 February 2006

Safety-Critical Systems Club

BAE SYSTEMS

Springer

Felix Redmill
Redmill Consultancy, 22 Onslow Gardens, London, N10 3JU

Tom Anderson
Centre for Software Reliability, University of Newcastle,
Newcastle upon Tyne, NE1 7RU

British Library Cataloguing in Publication Data
A catalogue record for this book is available from the British Library

ISBN-10: 1-84628-333-7 Printed on acid-free paper
ISBN-13: 978-1-84628-333-8

Printed in the United Kingdom

9 8 7 6 5 4 3 2 1

Springer Science+Business Media
springer.com

PREFACE

Each February, the Safety-critical Systems Symposium (SSS) hosts a one-day tutorial followed by two days of paper presentations. Annually, the papers provide a mix of industrial experience and research results, and address the most critical topics in the field of safety-critical systems. This year, the focus is on recent developments in risk-based approaches, and the papers report on these in a number of areas.

A topic of continuing interest and increasing importance is that of the safety case, and in recent years papers at the Symposium have highlighted its principles and the nature of its contents. This year there are two sessions on the subject. The papers in the first report on experience of developing safety cases, and they offer advice on the process; those in the second give suggestions on the safety case's evolutionary requirements and directions.

Other perennial subjects are risk and software safety. Three papers report on directions that risk analysis have taken or could take, and two provide interesting insights into language development and the creation of systems for complex control functions.

On the academic side, three papers address the use of new software technologies. They raise questions as to when such technologies are ready for application in the field of safety-critical systems. The need to consider them in the context of safety principles, taking a risk-based approach, is emphasised.

Finally, there is a section on management risk, a subject that is both important and neglected. It is hoped that both practitioners and academics in our field will carry out further work on this subject.

Each year, the organisation of SSS depends heavily on the authors who prepare and present their papers. Without them, there would be no Symposium, and without useful content in the papers, there would be no successful Symposium. We therefore extend our thanks to the authors and their companies for their time and intellectual application, and for responding to our editing demands with grace and good will. We also (and again) thank Joan Atkinson for being a continuing mainstay of the event.

FR & TA
November 2005

THE SAFETY-CRITICAL SYSTEMS CLUB

organiser of the

Safety-critical Systems Symposium

What is the Club?

The Safety-Critical Systems Club exists to raise awareness of safety issues in the field of safety-critical systems and to facilitate the transfer of safety technology from wherever it exists. It is an independent, non-profit organisation that co-operates with all bodies involved with safety-critical systems.

History

The Club was inaugurated in 1991 under the sponsorship of the UK's Department of Trade and Industry (DTI) and the Engineering and Physical Sciences Research Council (EPSRC). Its secretariat is at the Centre for Software Reliability (CSR) in the University of Newcastle upon Tyne, and its Co-ordinator is Felix Redmill of Redmill Consultancy.

Since 1994 the Club has been self-sufficient, but it retains the active support of the DTI and EPSRC, as well as that of the Health and Safety Executive, the Institution of Electrical Engineers, and the British Computer Society. All of these bodies are represented on the Club's Steering Group.

What does the Club do?

The Club achieves its goals of awareness-raising and technology transfer by focusing on current and emerging practices in safety engineering, software engineering, and standards that relate to safety in processes and products. Its activities include:

- Running the annual Safety-critical Systems Symposium each February (the first was in 1993), with Proceedings published by Springer-Verlag;
- Organising a number of 1- and 2-day seminars each year;
- Providing tutorials on relevant subjects;
- Publishing a newsletter, *Safety Systems*, three times each year (since 1991), in January, May and September.

How does the Club help?

The Club brings together technical and managerial personnel within all sectors of the safety-critical community. Its events provide education and training in principles and techniques, and it facilitates the dispersion of lessons within and between industry sectors. It promotes an inter-disciplinary approach to safety engineering and management and provides

a forum for experienced practitioners to meet each other and for the exposure of newcomers to the safety-critical systems industry.

The Club facilitates communication among researchers, the transfer of technology from researchers to users, feedback from users, and the communication of experience between users. It provides a meeting point for industry and academia, a forum for the presentation of the results of relevant projects, and a means of learning and keeping up-to-date in the field.

The Club thus helps to achieve more effective research, a more rapid and effective transfer and use of technology, the identification of best practice, the definition of requirements for education and training, and the dissemination of information. Importantly, it does this within a 'club' atmosphere rather than a commercial environment.

Membership

Members pay a reduced fee (well below a commercial level) for events and receive the newsletter and other mailed information. Without sponsorship, the Club depends on members' subscriptions, which can be paid at the first meeting attended.

To join, please contact Mrs Joan Atkinson at: Centre for Software Reliability, University of Newcastle upon Tyne, NE1 7RU; Telephone: 0191 221 2222; Fax: 0191 222 7995; Email: csr@newcastle.ac.uk

CONTENTS LIST

SOFTWARE SAFETY

NEW TECHNOLOGIES IN SAFETY-CRITICAL SYSTEMS

ADDING DIMENSIONS TO SAFETY CASES

TUTORIAL

People and Systems: Striking a Safe Balance between Human and Machine

Carl Sandom,
iSys Integrity,
Gillingham (Dorset), UK

Derek Fowler
Independent Safety Consultant,
Henley on Thames, UK

Abstract

Humans may be viewed as being merely fallible operators of machines; however, that technology-centred view can easily understate the ability of the human to perform tasks which most machines are incapable of doing and to intervene in the event of failure. On the other hand, an overly human-centred view may not take full advantage of the ability of machines to carry out numerically-complex, repetitive tasks consistently and at relatively high speed, and to provide alerts in the event of failure on the part of the human. Somewhere between these extremes lies a more balanced, integrated approach in which the best (and worst) characteristics of human and machine are fully recognised in the development of safe system solutions.

This paper, produced in support of a tutorial entitled: 'System Safety Requirements for People, Procedures and Equipment', given at the Safety-critical Systems Symposium 2006, presents a generic approach for the specification and realisation of safety requirements for both technical and human elements of safety-related systems.

1 Introduction

In the absence of a holistic approach to system safety assessment, it is tempting to concentrate safety assessment effort on what we understand or think we understand (such as hardware and software) and to adopt a 'head in the sand' approach to the human factors which are often perceived as too difficult. Humans are often the major causal factor for hazards in safety-related systems (Sandom 2002) and yet human failures often don't receive proportionate attention in safety analyses. On the other hand, human operators also often provide substantial mitigation between

machine-originated hazards and their associated accidents; yet this too is often overlooked or, conversely, sometimes over-stated.

It is well-established that in some application sectors humans are the major cause of accidents or safety incidents; however, this can lead to erroneous conclusions. Taking the human 'out of the loop' may not be the panacea that it first appears unless we fully understand, for example:

- The potential for equipment failures to cause accidents can be hidden by human mitigation of those failures.
- Humans often perform far less well in monitoring roles than they do if fully involved and occupied.
- Increased automation inevitably leads to de-skilling of the human operator and the ability of the human to mitigate the effects of equipment failure is often impaired.

Apart from a preoccupation with reliability and integrity issues, the development of safety-related equipment is relatively well understood and well covered by process-based safety standards including IEC 61508 (system and software), DO-254 (hardware) and DO-178B (software). However, the role of human factors in system development is far less understood and receives little coverage in the popular safety standards. It is difficult to see how overall system safety can be demonstrated (or even achieved) except through actual operating experience. Safety is not just a matter of system reliability and an argument is made here for safety requirements, including those for human sub-systems, to include functionality and performance as well as the integrity of each safety function.

Some safety-related systems (e.g nuclear reactors) are categorised as such simply because they pose an unacceptable safety risk to their environment and they require additional protection systems to contain that risk within an acceptable level. In contrast, systems such as Air Traffic Control or Railway Network Control are designed specifically to provide risk reduction and can be likened to one big protection system. This paper presents a generic approach for the specification and realisation of safety requirements for the technical and human elements of both types of safety-related systems. The term 'realisation' is used here to cover all activities associated with requirements implementation, validation and verification.

The paper presents a pragmatic methodology to fully integrate human factors analyses with safety engineering analyses to take account of both human and technology capabilities and limitations, thereby addressing the major risks to systems safety. The approach presented here addresses the specification of both Operational-level and System-level safety requirements down to the allocation of functions and safety requirements to subsystems comprising equipment, people and procedures.

However, in order to ensure that such safety requirements are correctly specified, we first need to understand the fundamental nature of safety and safety requirements.

2 Safety Fundamentals

Safety is commonly defined as freedom from unacceptable risk of harm (or accident). One very useful view of safety, and of safety assessment, is the 'barrier model' illustrated in Figure 1 using Air Traffic Management (ATM) as an example.

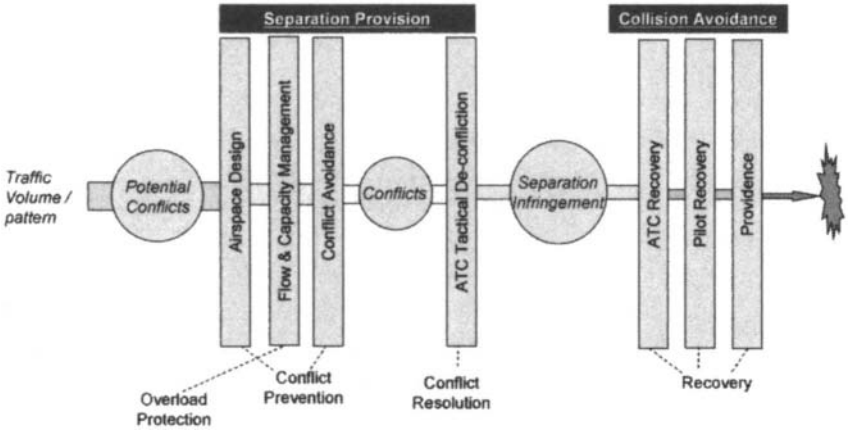


Figure 1. Barrier Model (adapted from Reason 1997)

On the right-hand side of the model is the accident that we are seeking to avoid. In ATM terms, harm is normally taken to be a collision between two aircraft or between one aircraft and a ground-based obstacle – for simplicity we will consider only the case of a possible mid-air collision between two aircraft.

On the left-hand side is the threat posed by the presence of aircraft in the airspace. Intervening between the threat and the accident depends on the presence and effectiveness of a series of barriers. In general, the avoidance of mid-air collisions is dependent primarily on the maintenance of appropriate separation between aircraft or, if that fails, by collision avoidance. Aircraft separation is provided by:

- *Airspace design*: structuring the airspace so as to keep aircraft apart spatially, in the lateral and/or vertical dimensions.
- *Conflict avoidance*: planning the routing and timing of individual flights so that the aircraft, if they followed their planned trajectories, would not pass each other within the prescribed minimum separation.
- *Conflict resolution*: detecting conflicts when they do occur and resolving the situation by changing the heading, altitude or speed of the aircraft appropriately.

In order to prevent overload of the above barriers, the flow of traffic is maintained within the declared capacity of the *Separation Provision* service. *Collision Avoidance* is intended to recover the situation only for those potential accidents that *Separation Provision* has not removed from the system. In general, these may be considered as:

- *Air Traffic Control Recovery* mechanisms – human and/or machine-based safety nets.
- *Pilot Recovery* mechanisms – again, human and/or machine-based safety nets.
- *Providence* – i.e pure chance.

One very important thing that the above barriers have in common is that none of them (neither singly nor in combination) is 100% effective even when working to full specification. This leads us to some crucial conclusions regarding safety, as illustrated in Figure 2:

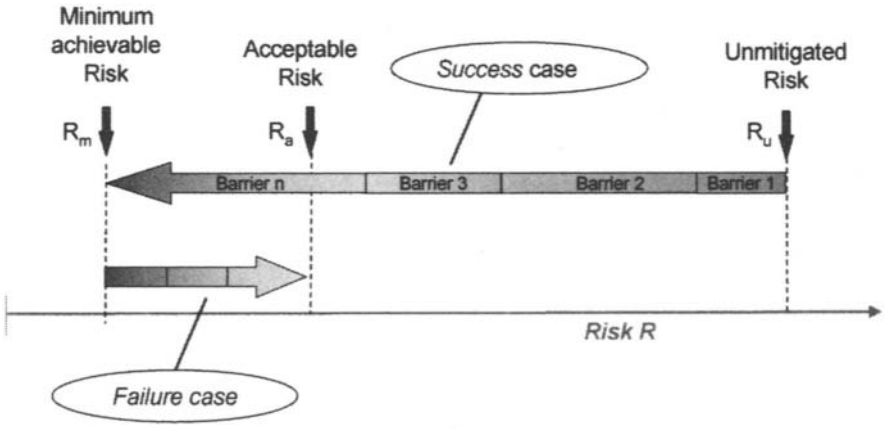


Figure 2. General Risk Model

- Firstly, when risk exists at an unacceptable level (Ru), barriers need to be provided in order to mitigate that risk.
- Secondly, risk cannot be eliminated totally (unless the threat is removed entirely) and the minimum level to which risk can be reduced (Rm) is determined by the desired properties of the barriers – e.g functionality, accuracy, capacity, speed of response etc.
- Thirdly, the risk-reduction effectiveness of a barrier is itself reduced by the undesired properties of the barrier – unreliability, unavailability etc – causing risk to rise somewhat.

Clearly the net risk must lie at or below the acceptable level (Ra). Thus, if we consider a system to include the associated barriers, any safety assessment of that system must address two key issues:

- How safe it is when the barriers are working to specification, in the absence of failure – the *success case.*
- How less safe it is in the event of failure, or partial failure, of a barrier – the *failure case.*

There is a widespread view (unfortunately reinforced by some safety standards) that safety is largely a matter of reliability despite the fact that theory and experience have shown this to be far too narrow a view of safety (see Sandom and Fowler 2003). What the success case tells us is that one of the first considerations in assessing system safety must be whether the functionality and performance properties of the system are adequate to achieve substantially better than an acceptable level of risk.

Once the success case is established, only then is it worthwhile considering the failure case and the increase in risk associated with the failure-related properties of the system. This leads directly to the conclusion that Safety Requirements must take two forms:

- Those relating to the required function and performance, of the barriers – herein referred to as Functional Safety requirements.
- Those relating to the required reliability, availability and integrity, of the barriers – herein referred to as Safety Integrity requirements.

The rest of this paper describes a framework for the specification of Safety Requirements, for a system comprising equipment, people and procedures, using aspects of ATM to illustrate the safety requirements specification process.

3 Safety Requirements Specification

Figure 3 shows a representation of the safety requirements specification process based on a hierarchical framework. An explanation of the five principal levels of Figure 3, appropriate to the development of safety properties, is given as follows:

- The *Operational Environment* (or domain) into which the service is provided. In ATM, the airspace structure and rules, and users of the ATM service, exist at this level and full account must be taken of the properties of the operational domain in the safety specification of the lower levels in the hierarchy.
- The *Service Level*, defined by the barrier model (see Figure 1). Safety targets for the service may be specified at this level.
- The so-called *Abstract Operational Level* at which the barriers that fall within the system boundary are decomposed into abstract safety functions; those safety functions that are entirely independent of whether they are provided by humans and/or equipment. It is at this level that hazards are

defined and *Tolerable Hazard Occurrence Rates* (THORs) are set in order to limit the frequency of occurrence of those hazards sufficiently to satisfy the safety targets.

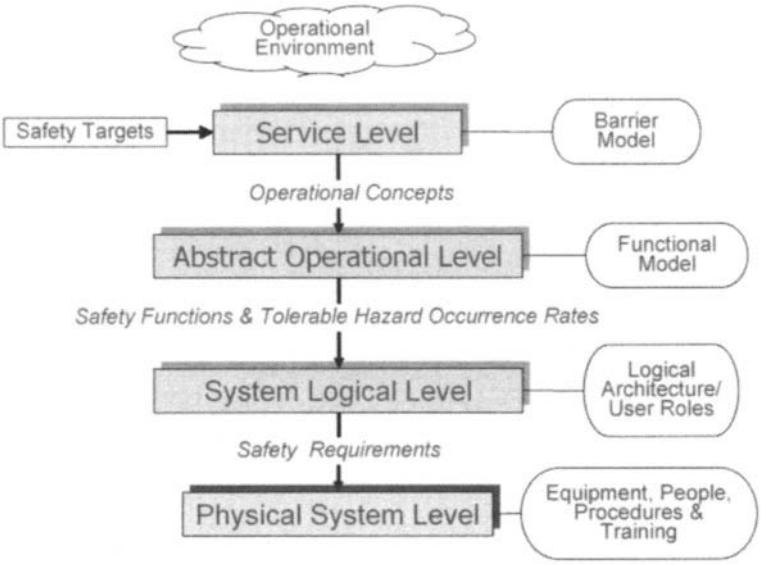


Figure 3. Safety Requirements Hierarchy

- The *System Logical Level* at which the safety functions are allocated to the various elements of the system logical architecture, plus the tasks to be performed by generic human-operator roles – the causes of the hazards are identified at this level, as are the Safety Integrity Requirements that limit the frequency of occurrence of each cause such that the THORs are satisfied; although at this level the distinction between human and machine is made, the safety requirements which emerge from it are still independent of the actual physical implementation.
- The *Physical System Level* - comprising the physical sub-systems, implemented typically in equipment (hardware and software), people (operational and maintenance) and procedures (operational and maintenance). It is at this level that the satisfaction of the safety requirements is demonstrated, possibly via further stages of safety requirements decomposition.

A representation of the relationship between Hazards, Causes and Consequences is the Bow-Tie model, shown in Figure 4, in which all the causes of a hazard are linked directly to the possible outcomes (i.e consequences) in a single structure.

Event Tree Analysis (ETA) is used where appropriate [1] to model all the possible outcomes of a hazard taking account of the mitigations (usually external to the system element in question) that could be used to break an accident sequence should a hazard occur. Working from left to right, each branch of the Event Tree represents a mitigation to which probabilities can be applied in order to express the relative likelihood of success (S) or failure (F) of the mitigation.

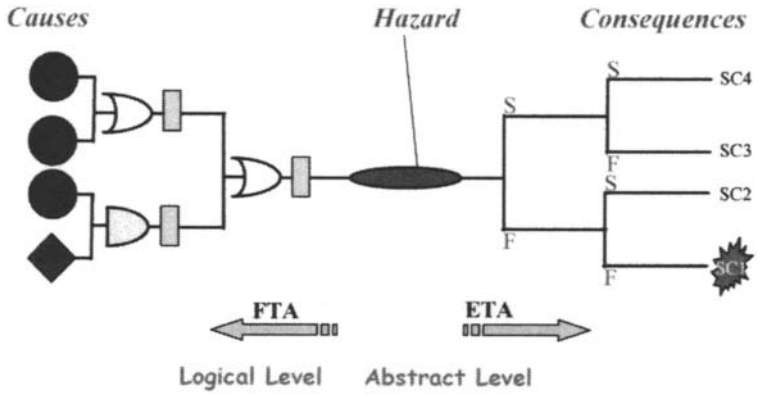


Figure 4. Bow Tie Model

The severities of the various outcomes are categorised - in this case, on a scale of 1 to 4. If safety targets are set for each of these categories, then the THOR for the hazard can be set such that these targets are met, taking account of the probability of success of the various mitigations.

Fault Tree Analysis (FTA) is used to model all the possible ways in which a given hazard could arise from failure within the system element in question, taking account of the mitigations (internal to that system element) that could be used to prevent such failures leading to the occurrence of the hazard. Given the THOR for the hazard, the frequency at which each of the lowest-level events in the Fault Tree are allowed to occur can be determined; each of those frequencies is the Safety Integrity Requirement for that event. The process of developing Safety Requirements is explained in more detail in the following paragraphs.

3.1 Operational Level - Safety Functions and THORs

The first step is to determine what Safety Functions need to be provided at the service level, and to specify the FSR including the performance required of them (e.g accuracy, capacity, timeliness etc, but excluding integrity), in order for safety targets to be met. Figure 5 shows that the Safety Functions are in fact a functional

[1] Usually, ETA is appropriate when there are several possible mitigations for a particular Hazard

description of the elements of the Barrier Model – in this case a simple functional model of the barrier ATC Tactical De-confliction is used to illustrate the point.

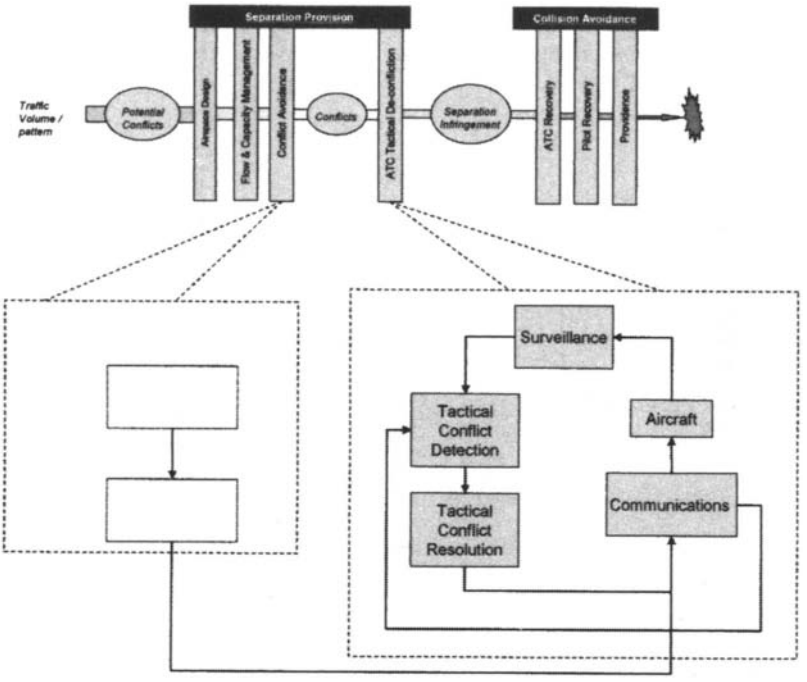


Figure 5. Derivation of Safety Functions

It is necessary at this stage to carry out some form of performance-risk assessment in order to show that specified safety functions are sufficient to reduce the risk to a level (Rm) well below the Safety Targets – i.e minimum acceptable level (Ra) - as indicated in Figure 6. R_a-R_m in Figue 6 represents that portion of the Safety Target, which can be allocated to (functional) failure – clearly these must be a realistic figures otherwise there is no point in proceeding further.

The potential failure modes of the Safety Functions (i.e Hazards) are analysed using the Bow Tie approach, described above, and THORs are specified to limit the allowable rate of occurrence of each Hazard such that the aggregate risk associated with all the Hazards is within the value of R_a-R_m, taking account of any mitigations that are identified during the process.

It is very important in this process that all mitigations are captured as either:

- Additional Safety Functions and corresponding tolerable probability of failure for the provision of deliberate mitigations of the consequences of the identified Hazards.
- Operational Domain Knowledge for any circumstantial mitigations (e.g those arising as a matter of pure chance).

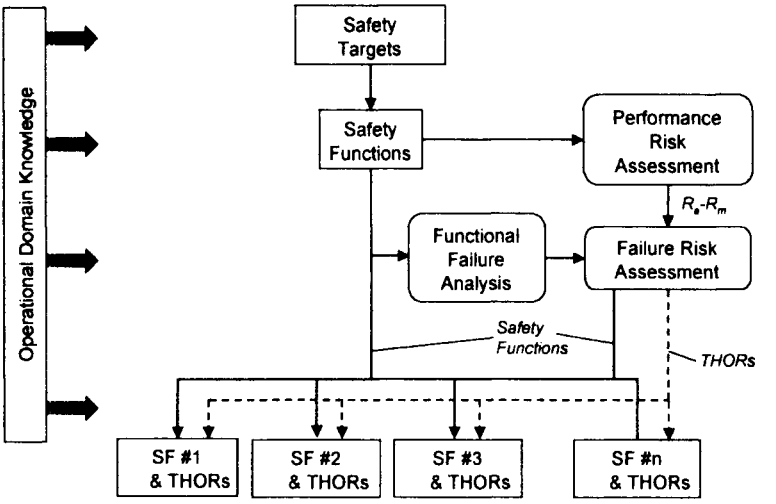


Figure 6. Operational-Level Safety Functions and THORs

3.2 System Logical Level

System Level safety requirements are specified at a logical architecture level – i.e taking into account the distinction between equipment and human elements of the system design but still independent of the actual physical solution.
A generic process for specifying primary and derived safety requirements (the latter through analysing system failure) is illustrated in Figure 7 and it is similar to that for the Operational level, as described above and shown in Figure 6.

Primary system safety requirements stem from an allocation of the service-level safety functions to the subsystem(s) on which they are to be implemented. The example illustration in Figure 7 shows typical ATM equipment sub-systems (Air-Ground-Air communications, Radar Data Processing, Flight Data Processing, and Display) and human-based subsystems (Executive and Planning controllers).

A discussion on the *safe* initial allocation of function between human and machine will be given later in the paper. The hazards and risks associated with failure of each subsystem may be assessed, using the broad Bow Tie approach described above, any mitigations are identified and allocated (as domain knowledge or additional safety functions, as appropriate), and the safety integrity requirements for each subsystem determined.
The safety properties determined from this part of the process being known collectively as derived safety requirements. The outputs from this stage are therefore:

- *Safety functions* to be implemented by each subsystem, and the performance required of them.
- Specification of the *interactions and interfaces* between the subsystems.
- *Safety integrity requirements* for each subsystem.

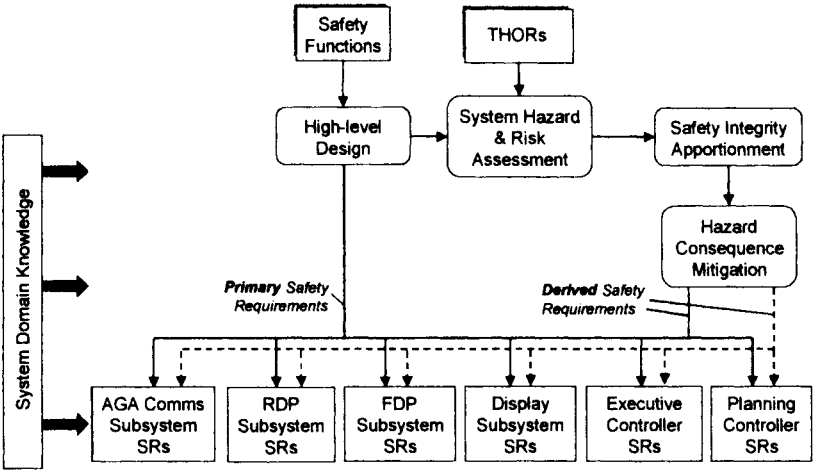


Figure 7. Safety Requirements Specification

A key point here is that the subsystems comprise both technical and human subsystems and the specific methods and techniques used to assess the hazards and risks associated with failure of each subsystem will necessarily be different.

3.3 System Physical Level

As discussed above, the *Physical System Level* comprises the physical sub-systems implemented typically in equipment (hardware and software), people (operational and maintenance) and procedures (operational and maintenance). It is at the physical level that the satisfaction of the safety requirements is demonstrated, possibly via further stages of safety requirements decomposition.

The engineering methods and techniques used for demonstrating the satisfaction of equipment safety requirements (e.g Fault Tree Analysis, Event Tree Analysis, Zonal Hazard Analysis etc.) are relatively well understood by the wider safety engineering community compared with those for people and procedures and will therefore not be discussed further here. The remainder of this paper will discuss how the above approach to safety requirements specification and realisation can be developed in the case of human-based subsystems, using Human Factors methods and techniques.

4 Human Safety Requirements

Human Factors (HF) is a discipline that covers the social, organizational and individual human factors aspects of a system in its context of use (i.e real time). HF analyses primarily address the need to match technology with humans

operating within a specified environment, in order to meet the Operational–level safety requirements.

Previous discussions here on safety requirements have indicated that the scope of system safety analyses must address the system, service and operational environment. This vast scope presents a challenge for the systems engineer who needs to consider the safety-related aspects of the entire system and then to focus the often limited resources available on the most critical system functions.

The human can be both a positive and a negative influence on system safety and humans can alternatively be considered as 'hazard' or 'hero' depending upon the circumstances of the specific system interaction. Ideally, an interdisciplinary approach should be taken to safety-related systems development through the focused application of HF and Systems Engineering methods and techniques – this approach has been referred to as Human Factors Engineering (HFE) (Sandom and Harvey 2004).

4.1 Pragmatic HFE Approach

Broadly, what is required is a *pragmatic* approach to the application of HF methods and techniques for human safety requirements specification at the Logical Level and the demonstration of satisfaction of human safety requirements at the Physical Level. Figure 8 shows different HF analyses that can be undertaken for the specification of human safety requirements (function, performance and integrity) and the realisation of those requirements and their contribution (both success and failure) to safety assurance typically provided by a system safety case.

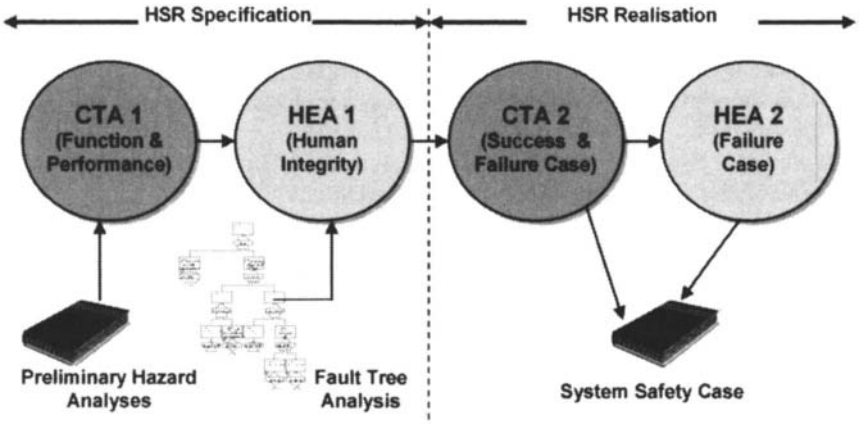


Figure 8. Safety-Related HFE Analyses

Figure 8 shows two different safety-related HF analyses described as Critical Task Analysis (CTA should not be confused here with cognitive task analysis) and Human Error Analysis (HEA).

CTA and HEA are high-level descriptions of analyses which may be undertaken using single or multiple combinations of the various HF Task Analysis,

Human Error Identification or Human Reliability Analysis methods and techniques available.

It is important to note that the CTA deals only with the *safety-critical* tasks and likewise HEA deals only with *safety-critical* human errors. Other HF analyses may have a wider scope to address usability issues which are not directly safety-related. Both CTA and HEA analyses should therefore be planned to ensure that there is no unwanted (and costly) overlap with any wider HF programme.

Typically, two iterations of each analysis should be undertaken to cover human requirements specification and realisation phases and, as the analyses become more focused, the results from each one will inform and focus the other. In addition, these HF activities are entirely complementary as CTA and HEA are bottom-up and top-down analysis techniques respectively (from a hazard to human event perspective).This combination of top-down and bottom-up analyses significantly increases the probability of identifying inconsistencies in the individual techniques and thus enhances safety assurance.

Referring to the safety requirements hierarchy shown in Figure 3, the Operational Level deals with abstract functions with no consideration of implementation details and it follows that there are no specific human factors to consider at that level. CTA and HEA analyses are therefore directed specifically to address the human factors at the system Logical and Physical levels. At the Logical System Level (for each allocated Human SR) safety-related human factors issues may be addressed by undertaking:

- A CTA to validate allocated *human* tasks taking into account procedures and equipment design.
- The specification of human *performance* requirements through an initial CTA.
- The specification of Human Integrity Targets through a HEA of physical system interactions directed by initial system hazard analyses.

At the Physical System Level (for each implemented Human SR) safety-related human factors issues may be addressed by undertaking:

- A detailed CTA to verify human tasks and performance taking into account procedure and equipment design.
- Realisation of HE probability claims through a refined analysis of physical system interactions directed by detailed system Fault Tree Analyses.

4.2 Success and Failure Cases

The Risk Model in Figure 2 makes a clear distinction between success and failure and relates that to the acceptable level of risk at the overall system level using people, procedures and equipment to implement system functionality.

Likewise, a clear distinction needs to be made between the human *success* and *failure* cases as follows:

- The *success* case – the main intention is to assess whether the tasks allocated to the human can be undertaken safely and to identify all the support (e.g procedures, tools etc.) that the human would require while undertaking those tasks.
- The *failure* case – the intention is to identify human error potential and assess reliability when specifically related to the dangerous human errors of commission or omission. In addition, the failure case must identify any human tasks arising from the need to mitigate machine failure.

Figure 9 shows the high-level issues for consideration when making initial decisions relating to the logical safety requirements specification and implementation.

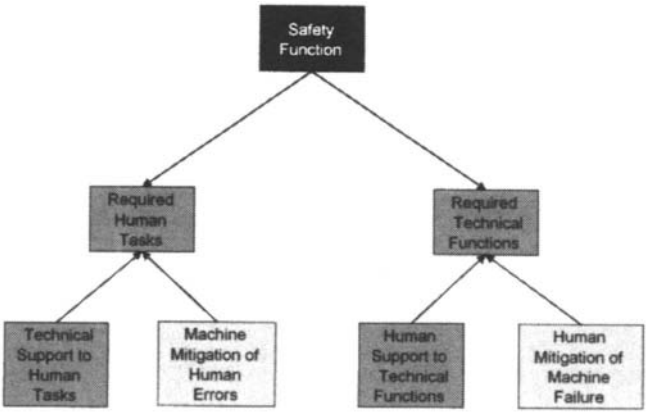


Figure 9. Allocation of Safety Functions

Figure 9 shows the system-level *success case* requirements for tasks and functions which are typically as follows:

- Determination of which Safety Functions should be allocated primarily to the human (as tasks) or machine (as equipment Functions), taking into account the characteristics of the Safety Function.
- Identify what additional human Tasks are needed to support the machine – e.g operation, insertion of data etc.
- Identify what additional equipment Functions are needed to support human performance and achievement of the required Tasks (e.g information, computation etc).

In addition, Figure 9 shows the high-level *failure case* requirement for tasks and functions which can be summarised as follows:

- Technical mitigations of potential human errors.
- Human mitigation.

A summary of success and failure from different perspectives is given in Table 1 and it can be seen that a human success case requires the specification of achievable human tasks to include the successful provision of human mitigation for technical failures where possible.

Case Type	Human View	Technical View	System View
SUCCESS	Human Tasks	Technical Functions	Absence of failure
FAILURE	Human Error (of success tasks AND human tasks for mitigation of technical failures).	Technical Failure (of main functions AND functions for mitigation of human errors	Failure (of Tasks, Functions AND mitigations)

Table 1. Success and Failure Case Summary.

The remainder of this paper will examine the broad issues relating to specific HF methods and techniques that can be used to undertake CTA and HEA which aim to generate detailed evidence to support the human success and human failure cases contributing to the overall system safety assurance.

4.3 The Success Case – Human as Hero

The human success case is built upon the evidence provided by CTA activities undertaken during both the requirements specification and realisation phases of systems development. CTA is a general term applied to the process that identifies and examines task performed by humans, or groups of humans, as they interact with systems. Task Analysis (TA) is a method supported by a number of specific techniques to collect and organize information to build a detailed picture of the system from the human perspective (for comprehensive coverage of TA techniques see Kirwan & Ainsworth 1992). CTA can be used to focus various TA techniques on specific safety issues rather then examining the system as a whole.

CTA seeks to promote appropriate job and task design, suitable physical environments and workspaces, human-machine interfaces and the appropriate selection, training and motivation of the humans involved. At the detailed level CTA examines how the design of human-computer interactions can foster the efficient transmission of information between the human and machine, in a form suitable for the task demands and human physical and cognitive capabilities.

CTA activities can be characterized as being undertaken for one or more of the following broadly defined purposes:

- Allocation of Function.
- Interface design or assessment.
- Task and procedure design or assessment.
- Personnel selection.
- Operability and workload assessment.
- Training requirements or assessment.

For each of these analyses there are specific methods and approaches that are the most appropriate and these are often selected based upon familiarity with the techniques and the aim of the analysis.

The human success case must be built upon two main activities relating to the system safety requirements specification which are the initial Allocation of Function between human and machine and an initial CTA of the functions (or tasks) allocated to the human subsystems to determine what constitutes successful human task performance requirements. Both of these activities are examined here in more detail.

4.3.1 Allocation of (Safety) Functions

The allocation of functions between humans and machines, and defining the extent of operator involvement in the control of the system is a critical activity in safety-related systems. Figure 7 shows a general process for deriving the subsystem safety requirements from a high-level architectural design.

An important feature of Figure 7 is that the high-level design must take into consideration the human factors in the initial allocation of Safety Functions. Too often, this decision is based upon technical capability and the human is allocated whatever functionality can't be implemented in hardware or software, regardless of the suitability of the human to undertake the resultant tasks.

The production of a high-level architectural design requires initial decisions to be made on the allocation of functions to human or equipment sub-systems, in full knowledge of the safety risks involved. Functional allocation decisions need to be informed by good human factors principles and yet the allocation of function is still considered exclusively an ergonomics problem by many systems developers.

The first step is to allocate the abstract operational-level Safety Functions on to the logical model; at this point it is helpful to have a broad notion of how the human and machine will interact in delivering the Safety Functions. The early work of Fitts (1951) was often used to derive MABA-MABA (Men Are Better At-Machines Are Better At) lists that were typically restricted to considerations of either the human or the machine performing each individual function. However, since Fitts' early work, it has become apparent that many functions in complex systems require apportionment of the function between *both* human and machine.

An extensive discussion on functional allocation is beyond the scope of this paper; however, for a detailed review of task allocation techniques see Kirwan and Ainsworth (1992).

4.3.2 Critical Task Analysis

A CTA can be undertaken to identify and analyse the human performance issues in critical operational tasks *as defined for successful interaction.* The initial CTA should focus on human performance aspects relating to the design of the human tasks including high-function cognitive functions such as: attention; vigilance; situation awareness etc.

CTA is a bottom-up technique used broadly to analyse the relationships between system hazards (identified by the System Hazard Assessment in Figure 7) and operational tasks and the HMI design. The analysis works in a bottom-up fashion from operational tasks, related to base events, to identified service-level hazards.

A CTA can concentrate initially on the identification and analysis of the relationships between system hazards and safety-related operational tasks. This analysis will enable both the PHA and TAs to be checked for consistency, providing confidence in subsequent safety assurance claims. Any deficiencies - such as hazards with no related operational tasks or operational tasks (deemed as safety-related by subject matter experts) with no relationship to identified hazards - can be highlighted.

The analysis will also look for opportunities for hazard mitigation through identification of human error potential and improved information presentation by comparing the TA with HMI design guidelines from appropriate sectors. In summary, the CTA will enable the safety-related system developer to:

- Define the allocated safety functions in terms of human operator tasks, including potential mitigations to be provided by the Operator in the event of failure of technical subsystems.
- Capture the interactions and interfaces between the human and equipment subsystems.
- Determine task skills, knowledge and procedure requirements and record these as additional functional safety requirements.
- Confirm feasibility regarding human capabilities performance and reallocate inappropriate tasks to equipment (i.e tools, automation etc) as functional safety requirements.
- Identify training requirements and record these as functional safety requirements.
- Determine human information requirements and human-machine interaction requirements and record these as functional safety requirements.

4.4 The Failure Case – Human as Hazard

The human failure case is built upon the evidence provided by additional CTA and HEA activities undertaken during both the requirements specification and realisation phases of systems development. Broadly, the CTA is undertaken for the specification and realisation of the human tasks (including performance

requirements) required to mitigate against technical failures. The term 'realisation' is used to cover all activities associated with requirements implementation, validation and verification. An HEA is undertaken to achieve the following:

- The specification and realisation of Human Integrity Targets relating to the success-case human tasks.
- The specification and realisation of Human Integrity Targets relating to the human tasks required to mitigate against technical failures.

Human subsystems must be specified and acceptable Human Integrity Targets specified for the identified sources of human error. In addition, the validation and verification of the achievement of the allocated Human Integrity Targets for each human subsystems is also required (this may include procedures as well as people).

Figure 7 shows a generic process for deriving both technical and human system-level safety requirements from a high-level architectural design. However, the specific processes for determining primary safety requirements and producing derived safety requirements will necessarily be based upon on different analysis techniques when dealing with human rather than technical subsystems.

Figure 10 (an adaptation of the generic Figure 7) shows the high-level process for deriving system-level safety requirements for humans using HEA to generate integrity requirements based upon an analysis of human failure.

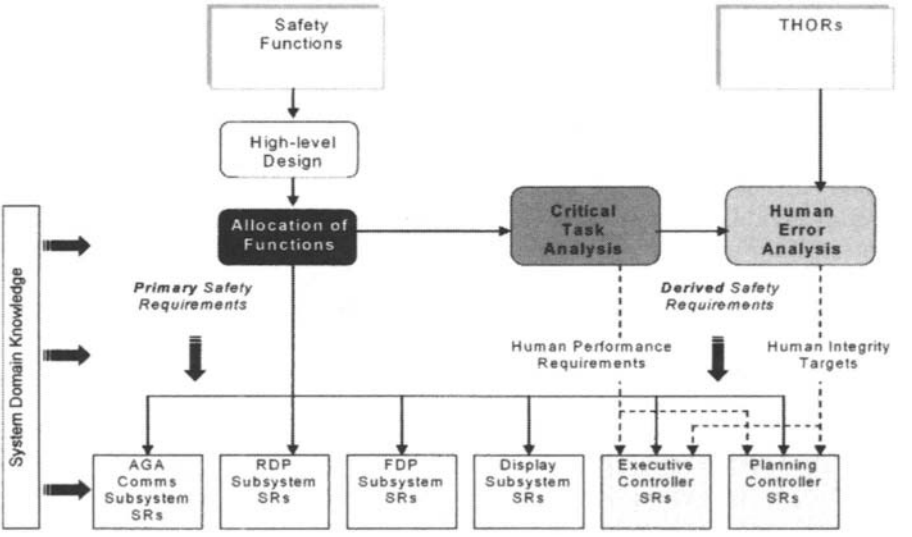


Figure 10. Human Safety Requirements Specification

HEA analysis is a top-down technique used to model the relationship between critical human failures and hazards, and the mitigating aspects of the system design.

An HEA should be undertaken using a two stage process of Human Error Identification (informed by the CTA) followed by a Human Reliability Assessment (informed by other safety analyses such as FTA etc.) which can be either qualitative or quantitative as required. Both of these activities are examined here in more detail.

4.4.1 Human Error Identification

Historically, the emphasis in Human Reliability Analysis (HRA) has been on techniques for the derivation of Human Error Probabilities (HEPs) for use in systems analysis techniques such as FTA. However, HEA should be an integrated process that includes a systematic and rigorous qualitative analysis to *identify* the nature of the errors that can arise prior to any attempt at quantification. This qualitative Human Error Identification (HEI) must ensure that no significant failures are omitted from the analysis.

It is widely recognised that there are considerable uncertainties in the quantitative data available for inclusion in HRA. However, as long as the qualitative error identification process is sufficiently comprehensive, valuable insights will emerge with regard to the sources of risk, and where limited resources should be most cost effectively applied in minimising these risks.

4.4.2 Human Reliability Analysis

The derivation of quantitative human integrity targets is difficult and HRA techniques have attempted to address this issue (see Kirwan 1994). However, much of the HRA research has been dominated by assumptions that apply to technical systems and arguably these do not translate well to human systems. While the failure probability of hardware can be largely predicted by its basic design and its level of use, human error probabilities are influenced by a much wider range of contextual factors, such as the quality of the training, the design of the equipment and the level of distractions.

The terms 'Performance Shaping Factors', 'Performance Influencing Factors' or 'Error Producing Conditions' are often used interchangeably to refer to the direct and indirect factors that influence the likelihood that a task will be performed successfully.

A pragmatic method of addressing this issue is to undertake a HRA focused specifically on the basic human events identified by the system safety analyses and in particular from the system Fault Tree Analyses. For systems, which typically have a high degree of operator interaction, many basic FTA events will be identified as human interactions. Once each fault tree is modelled, predictive, quantitative failure data can be input at the bottom from Availability and Reliability data for all hardware and software base events. By subtracting these

values from the associated hazard target, quantitative Human Integrity Targets (HITs) can then be calculated for each critical human event.

An HEA would then focus on developing specific safety arguments for each basic human event to provide evidence that the HITs can be achieved.
For critical areas, where the HEA reveals that the HITs are unrealistic, mitigations can be re-assessed and recommendations developed for further action. In this way, no predictions are being made about the human error rates; rather, the HITs are derived from the remaining integrity requirements once the hardware and software failure data is input and an analysis is undertaken to ascertain if the remaining human integrity requirements are realistic.

5 Conclusions

This paper has examined problems associated with the specification and realisation of functional safety requirements for the human elements of a system for which a target level of safety is specified at the service level. It was shown that the high-level allocation of functions to hardware, software or humans must be done by taking human performance and limitations into account and a generic approach was presented for the specification of both service-level and system-level safety requirements down to the allocation of functions and safety requirements to subsystems.

The process for the specification of human subsystem safety requirements is no different to software or hardware; although it is arguably considerably harder due to the difficulties associated with the immense scope and variety of issues affecting the reliable performance of human tasks. This paper has examined issues relating to the consideration of human subsystem safety and has outlined the scope and activities necessary for a comprehensive human factors safety analysis. A pragmatic method was introduced that advocates the application of focused Human Factors techniques to the assurance of safety for human subsystems.

The relative difficulties associated with the specification, implementation, validation and verification of human safety requirements, compared with safety requirements for hardware and software, should not be underestimated and this paper has not addressed many of these difficulties in detail. However, this paper has outlined a high-level approach for a focused and integrated application of Human Factors analyses for the specification and realisation of human subsystem safety requirements.

References

Fitts P M (1951). Human Engineering for an Effective Air Navigation and Traffic Control System, National Research Council, Washington D.C., 1951

Kirwan B (1994). A Guide to Practical Human Reliability Assessment, Taylor & Francis, 1994

Kirwan B and Ainsworth L K (Eds.) (1992). A Guide to Task Analysis, Taylor and Francis, 1992

Reason J (1997). Managing the Risks of Organizational Accidents, Ashgate Publishing

Sandom C and Harvey R S (Eds.) (2004). Human Factors for Engineers, IEE Publishing, London

Sandom C and Fowler D (2003). Hitting the Target - Realising Safety in Human Subsystems, Proceedings of the 21st International System Safety Conference, Ottawa, Canada, August 2003

Sandom C (2002). Human Factors Considerations for System Safety, in Components of System Safety, Redmill F and Anderson T (Eds.), proceedings of 10th Safety Critical Systems Symposium, 5th-7th February 2002 Southampton, Springer-Verlag, UK, February 2002

NEW APPROACHES TO RISK ASSESSMENT

Risk Assessment for M42 Active Traffic Management

Max Halbert
Cambridge Consultants Limited[1]
Cambridge, England

Steve Tucker
Highways Agency
Bristol, England

Abstract

The M42 Active Traffic Management project will introduce controlled use of the hard shoulder to the UK motorway. This paper describes some of the challenges encountered and solutions adopted in carrying out risk assessment on this project. These include the development of a methodology for demonstrating, in advance of opening the scheme, that the safety of the modified motorway will be equivalent to or better than that of the original.

1 Introduction

The M42 Active Traffic Management (ATM) project was established as a pilot project to test a number of innovative operational regimes for improving the performance of busy motorways without resorting to road widening. The most significant new feature, due to go live in 2006, is the controlled use of the hard shoulder to relieve congestion during busy periods. To implement such a feature, new programmable electronic systems (PES) had to be developed, and it was recognised that it would be necessary to follow a safety programme in accordance with IEC 61508 (IEC 1998). However, as no industry specific version of the standard was available for intelligent transport systems (ITS), it was necessary to return to first principles in choosing appropriate methodologies for hazard identification and risk assessment.

The Highways Agency (HA) has many safety standards, but these are mainly concerned with road layout, road construction, signage and other issues of a civil engineering nature. The ATM project extends the level of control that an HA ITS exerts over the movement of traffic. As such, it was subject to considerable

[1] The views expressed in this paper are those of the authors and are not necessarily those of Cambridge Consultants Ltd or the Highways Agency.

stakeholder interest, with parliamentary questions being tabled on a regular basis. It was therefore the first major road project in this country to involve the production of a formal safety case.

This paper describes some of the challenges that were encountered in running the ATM safety programme and describes the solutions that were adopted. In particular, it focuses on the approach to estimating risk in this environment and to demonstrating that the safety targets would be met.

An outline of the paper is as follows. Section 2 introduces the key features of M42 ATM. Section 3 discusses the setting of a safety target for the project – this was given the acronym GALE (Globally At Least Equivalent) and had a notable impact on the approach taken to hazard identification and risk assessment. Section 4 outlines the general approach taken to the safety programme. This leads to descriptions of specific aspects, namely hazard identification (section 5), the choice of a risk assessment methodology (section 6), the methodology for demonstrating GALE (section 7), management of safety information (section 8) and the safety case (section 9).

2 Description of M42 Active Traffic Management

The M42 is a major route forming part of the strategic road network around Birmingham distributing both national and local traffic. The section of the M42 between J3a (M40) and J7 (M6) is under increasing pressure from traffic growth. Amongst other destinations, it provides access to the National Exhibition Centre (NEC), Birmingham International Airport and Business Parks. It is therefore important that congestion on this section of motorway is minimised and that journey times are as reliable as possible.

In July 2001 the Minister for Transport announced the implementation of a pilot project on the M42 between Junction 3a and 7. The project, ATM, will form a key element in the delivery of the Government's Ten Year Plan for Transport. It will also enable the HA to take forward its function as 'Network Operator'.

ATM consists of a number of new Operational Regimes that will work in combination with each other to target and resolve specific traffic problems identified on the network. The most innovative of these is Mandatory 4-lane Variable Speed Limits (4L VSL), which involves the controlled use of the hard shoulder during busy periods. Key features of the ATM scheme are:

- Lightweight gantries positioned nominally every 500m. Each gantry carries one message sign (capable of displaying both text and pictograms) and an Advanced Message Indicator (AMI) over each lane (for displaying speed restrictions, lane divert arrows and red X stop signals).
- Emergency Refuge Areas (ERAs), nominally at 500m intervals. These are the size of a standard lay-by and are intended for use during breakdowns and other emergencies. An Emergency Roadside Telephone (ERT) is located in each ERA.
- A set of 192 fixed cameras to survey the hard shoulder and ERAs before opening and a further 19 pan-tilt-zoom cameras for general surveillance.

- MIDAS (Motorway Incident Detection and Automatic Signalling System) loops at 100m intervals (compared with 500m on other motorways) for more accurate and rapid response to the onset of congestion or incidents.
- Digital speed enforcement cameras.

The hard shoulder will be used for running between but not through junctions, i.e. for exiting at the next junction. It will normally only be opened during peak morning and evening periods when the traffic flows reach a defined threshold, prior to flow breakdown. An operator will make the opening decision on the basis of traffic flow information from MIDAS. In exceptional circumstances, the hard shoulder may also be opened during the day, e.g. to facilitate the flow of traffic if there has been an incident in the offside lanes.

Before opening the hard shoulder on a link (where a link is the stretch of carriageway from one junction to the next), a mandatory 50 mph maximum will be set on all lanes of that link. This maximum will remain throughout the time that the hard shoulder is open.

In order to avoid opening the hard shoulder when a parked vehicle or some other obstruction is present, the hard shoulder will be opened a section at a time, starting at the exit and working in an upstream direction. A section is the distance between two sets of overhead signals, which is nominally 500m. Before opening each section, the operator will automatically be presented with a sequence of images from the fixed cameras, and required to confirm that the section is clear. If a parked vehicle or other obstacle is encountered or there is an equipment failure, the process pauses with only the sections downstream of the obstruction opened. The procedure will be able to resume when the problem has been cleared.

Closure of the hard shoulder commences when the traffic flow is sufficiently low to flow freely in three lanes. Again an operator makes this decision on the basis of traffic flow information from MIDAS. The closing sequence starts at the beginning of the link and moves downstream in the direction of the traffic. The speed of the closing sequence is designed to ensure that vehicles already in the hard shoulder will be able to continue to the next exit.

3 Safety Target

One of the first questions to be addressed in the safety programme was the definition of the safety target for the project. IEC 61508 does not directly address the question of how to do this, although many of its examples are based on the assumption that risks will be reduced As Low as Reasonably Practicable (ALARP). This is understandable given that applying the ALARP principle is a duty of all employers under the Health and Safety at Work etc Act 1974 (HSWA).

However, the HSWA does not place a duty on highways authorities to achieve ALARP as far as members of the public are concerned. If the ALARP principle were applied to roads to the same extent as the railways, for example, the costs could increase substantially. Furthermore, it is unlikely that the measures that would be

needed to achieve ALARP would be acceptable to the general public. Witness, for example, the ongoing objections to the proliferation of 'safety cameras'.

The public seems to expect that roads will demonstrate a gradual improvement in safety over time, without a significant loss of throughput or journey time. The government has reflected this in (DfT 2000), which sets out a 10-year plan to reduce the number of people killed or seriously injured (KSI) on roads in Great Britain using a range of measures, including safer drivers, safer vehicles and safer infrastructure. To achieve such a plan, most new road schemes need to be no less safe than their predecessors and ideally show an improvement in safety.

In the light of these considerations, the safety target agreed for the M42 ATM project was given the acronym GALE, for Globally At Least Equivalent. This was defined to mean that the M42 motorway, with ATM in operation, should present a level of risk less than or equal to that experienced by users of the M42 prior to the commencement of the construction of ATM.

Adoption of the GALE principle means that if there are circumstances where the risk associated with a particular hazard increases, the system remains acceptable if it can be shown that risks in other areas have been reduced by an equivalent or greater amount. However, the principle does not remove the need to assess risk on a hazard by hazard basis or to seek to apply mitigation measures where reasonably practical. If a risk reduction measure can reasonably be put in place, even if it is not necessary to achieve the overall safety objective, the project is expected to consider applying this risk reduction.

The GALE principle is also to be applied to specific road user groups, as far as is practical. Examples of road user groups include car users, heavy goods vehicle (HGV) drivers, motorcyclists, disabled drivers or passengers, recovery organisations, traffic officers and maintenance personnel. Applying the GALE principle to each group means that it is not acceptable to balance an increased risk for one group, say motorcyclists, by reducing it in another, say HGV drivers.

4 General Approach to Safety Programme

The general approach to safety taken by the M42 ATM project was a risk-based approach, in accordance with IEC 61508 and many other safety standards. In summary, this process involved the following steps:

- Define the scope (boundaries) of the system under consideration
- Identify hazards and their causes
- Estimate the risk of each hazard, as a function of the likelihood or frequency of the hazard and its potential consequences
- Identify candidate risk reduction measures for each hazard
- Choose which risk reduction measures to implement, with priority given to the hazards with the highest risks
- As risk reduction measures are adopted, revise risk assessments accordingly
- Evaluate whether GALE is likely to be achieved and if not seek further opportunities for risk reduction
- Produce safety case to show that acceptable level of safety is achieved.

Safety was defined to be the responsibility of everyone on the project and safety activities were undertaken within and across all work streams. In addition, a team was nominated to support the safety activities and lead the safety analysis work. Key roles were defined for the Managing Consultant, the Safety Workstream Manager and the Project Safety Manager. A Safety Champion was also appointed to provide safety guidance and mentoring as appropriate.

The adoption of the GALE principle was a key driver in the choice of safety approach throughout the M42 ATM project. Every one of the steps listed above required subtly different approaches to those commonly adopted. Further challenges were also encountered in applying this methodology to the road environment for the first time.

In this paper, we will describe the approach to risk assessment that was adopted for the M42 ATM project and highlight features of it that proved to be successful. In particular, we will show how each of the following questions were addressed:

- Hazard identification. What range of hazards must be considered in order to be able to compare safety between two different schemes? What will we treat as the 'top-level' hazards?
- Risk assessment methodology. How can we score the risk in a way that is conducive to making comparisons? How do we overcome the problem that so many risks are governed by human behaviour? How do we deal with the lack of data on so many motorway hazards?
- Demonstration of safety target. How can we show in advance of starting operation that the GALE target is likely to be achieved?
- Managing the process. How do we ensure that safety-related information from the many strands of a multi-disciplinary, multi-partner project is managed successfully?

5 Hazard Identification

5.1 Definition of System Boundary

One of the first steps recommended in IEC 61508 is to determine the boundary of the Equipment Under Control (EUC) and the EUC control system. Since IEC 61508 is directed towards programmable electronics systems, it is natural to think of the road and its users as the EUC and the ITS as the control system. Under such a model, it is also natural to consider modelling hazards at the boundary of the control system, e.g. at the level of errors in the display of signs and signals.

It rapidly became evident that this model would not suffice and that a much broader definition of the system boundary was necessary. Some of the reasons for this were:

- The project was responsible for delivering not just ITS equipment, but also infrastructure and operational regimes

- The consequences of failure in the ITS were heavily dependent on the design of the infrastructure and the definition of the operational regimes – the interaction between them was complex and it was impossible to analyse them in isolation
- A wide range of risk reduction measures could be identified for each hazard, including infrastructure measures (e.g. changing fixed signs and road markings), operational measures (e.g. defining procedures for control room operators or on-road resources) and technical measures (e.g. specifying test procedures for detecting failures)
- The need to demonstrate GALE involved developing a global view of the risk.

5.2 Definition of top-level hazards

A question closely related to that of defining the system boundary is the question of what should be regarded as the top-level hazards.

Consider first a simple model of an accident. If an accident is analysed, it is normally possible to identify a sequence of events leading up to the accident. First there is an initiating event (a cause), perhaps in conjunction with one or more exacerbating conditions. This creates a hazard – a condition or event that has potential to do harm. Once a hazard arises, it may sometimes – but not always – lead to an accident.

When carrying out hazard identification, choices are often encountered as to what should be regarded as the 'top-level' hazards. For example, consider an accident where a speeding driver runs into the back of another vehicle, causing serious injuries. What do we regard as the hazard? Here are some candidates that might be considered:

- *Vehicle collides with another vehicle*, i.e. at the level of the undesirable outcome. Although it is valid to regard this as a hazard, we found that modelling a hazard at this level was too complex to be helpful. There are a multitude of reasons why two vehicles may collide. Some may overlap with other hazards, such as hitting a pedestrian. The possible mitigations vary greatly depending on the causes. We therefore labelled this as an 'accident' and used accidents as a means of grouping and organising the hazards.
- *Vehicle travels too fast for prevailing road conditions*, i.e. the point at which driver behaviour deviates from the ideal. This was the level that we chose to adopt as the top-level hazard. The number of causes is generally manageable, and the possible risk reduction measures are more obvious. More importantly, it enables us to highlight the differences between hazards that apply to the baseline and those that apply to ATM.
- *System fails to warn of queue ahead*, i.e. a deviation in the output of the system. This may be appropriate when the project's responsibility is limited to the delivery of equipment. For the ATM project however, whose deliveries also included operational regimes, operator procedures and substantial amounts of infrastructure, it was important to consider a much wider range of hazards, including those related to driver and operator behaviour. Equipment failures were therefore generally modelled as hazard 'causes'.

5.3 Representing hazards to facilitate the GALE comparison

In order to demonstrate GALE, it was necessary to carry out risk assessment for both the baseline motorway (i.e. prior to constructing ATM) and the motorway with ATM in operation. In order to facilitate such a comparison, the hazard identification exercise had to meet a number of conditions:

- As far as possible, all hazards that apply to standard motorways should be identified, not just those associated with the new scheme. We therefore included hazards such as 'individual vehicle travels too fast', 'unsafe lane changing' and 'vehicle stops in running lane'. It was interesting that we were able to discern reasons why ATM could have some impact on nearly all of these hazards. Indeed, because the new scheme introduces new hazards, it is essential to improve many of the typical motorway hazards in order to achieve GALE.
- Hazards should be defined in a way that minimises any overlap between them. This is necessary to avoid double-counting when making the comparison.
- Each hazard should be defined in such a way that it has maximum commonality between the baseline and ATM. This facilitates making comparisons at the level of individual hazards. Hazards that are entirely new to ATM should be modelled as separate hazards, rather than being embraced by some broader more complex hazard.
- Each hazard should be defined as precisely as possible. This should include stating clearly which causes are included, which operational modes are applicable and which populations are affected.

Initial hazard identification was carried out in various workshops, using well-known techniques such as the Hazard and Operability (HAZOP) study. The initial set of hazards that were identified did not readily meet the above conditions. A considerable amount of work ensued to rationalise the hazards, deciding which could be regarded as top-level and which could be treated as causes. This involved combining some hazards and sub-dividing others. This process continued to some extent throughout the project, new causes being identified each time a change to the system or operational regimes was proposed.

6 Choosing a Risk Assessment Methodology

6.1 Applicability of existing methods

6.1.1 Risk Matrix

Many standards, including IEC 61508, suggest the familiar risk matrix approach, such as in Table 1, for deciding whether risks are tolerable or not. The Roman numerals I to IV represent risk classes, where risk class I is intolerable and IV is acceptable. Although IEC 61508 is clear that this is an example only, we have observed many projects which have attempted to apply this matrix directly without question.

Frequency	Consequence			
	Catastrophic	Critical	Marginal	Negligible
Frequent	I	I	I	II
Probable	I	I	II	III
Occasional	I	II	III	III
Remote	II	III	III	IV
Improbable	III	III	IV	IV
Incredible	IV	IV	IV	IV

Table 1 Example risk matrix from IEC 61508 Part 5

This approach pre-supposes that it is possible to determine independently for each hazard whether the risk is tolerable or not. This may be applicable when the safety target is ALARP, but not so when it is GALE. The risk matrix approach also overlooks the issue that, if there are many hazards that fall into a particular risk class, this should be regarded as less tolerable than if there is only a single hazard in that class.

6.1.2 Controllability Method

Another methodology that was considered was the controllability method as described in (MIRA 2000) and (MIRA 2004). Controllability provides a qualitative assessment of the ability of any category of user to control the safety of the situation after a dangerous failure of an ITS. This method assigns each hazard to one of the following controllability classes: uncontrollable, difficult to control, debilitating, distracting, nuisance only. This is then mapped directly onto the Safety Integrity Level (SIL) that is required of the ITS. The controllability method deliberately makes no attempt to specify the final effect explicitly, or to identify the probability of occurrence of a dangerous failure. It is independent of the number of units deployed so that, say, a high volume equipment manufacturer will use the same SIL for the same system in the same application, as a low volume manufacturer.

An attempt to apply the controllability method suggested that the ITS equipment would fall into controllability class 'distracting' and hence should be SIL1. The ATM project was not just delivering equipment, however, and deriving a SIL was only a minor part of what was needed from a risk assessment methodology. The controllability method appears to be most applicable when only the ITS is within the system boundary. For the ATM risk analysis, the system boundary was necessarily much broader, encompassing control room operators, on-road resources, emergency services, roadside infrastructure and the operational regimes themselves.

6.2 Requirements of Risk Assessment Methodology for ATM

In choosing a risk assessment methodology for ATM, we aimed to meet the following requirements:

- The method should be simple to understand and apply

- The method should enable risks to be compared with each other – this is necessary to make it clear which are in greatest need of risk reduction
- The method should enable combining of risks to produce an estimate of total risk – this is required to produce a GALE comparison
- The method must be applicable to the broad range of top-level hazards encountered on a motorway
- The method should be able to take advantage of objective analysis where this is possible, but also accommodate the uncertainties associated with human behaviour.

6.3 Basics of methodology

The basic measure of risk for a particular hazard is in terms of the accidents that it causes:

$$risk = (\ accident\ frequency)\ x\ (accident\ severity) \tag{1}$$

Since it is difficult to estimate risk with great precision, we decided to allocate values in bands, rather than assign precise numbers. This is in common with the qualitative methods that are advocated in many of the safety standards. We also decided that the definition of each successive band would be a factor of 10 apart.

The output of our risk assessment method was to be a 'risk index'. The risk index is a value related to the logarithm of the actual risk, as follows:

$$risk\ index = \log_{10}(\ accident\ frequency) + \log_{10}(accident\ severity) + c \tag{2}$$

where c is a constant which reflects the fact that the risk index is a relative rather than absolute measurement. The actual value of the risk index depends on the definition of the frequency and severity bands. When allocated by table, the risk index is normally rounded to the nearest integer, although in principle rounding is not necessary. The risk tables that were used will be presented in section 6.7.

The use of a logarithmic measure satisfies the requirement for simplicity. The value was obtained by simple addition of numbers in tables and the result was typically an integer in the range 0 –11. The project team readily understood that an increase of one represented a 10-fold increase in risk, and that ten hazards in one band were approximately equivalent to one hazard in the next band up. This is considerably easier than working with linear measures and dealing with numbers such as 1×10^{-9}.

Using a semi-quantitative method also satisfies the requirement to be able to add risk measures together. To produce a global risk index for an entire scheme, it is in principle simply necessary to sum the antilogarithms of the risk indices for each hazard and take the log again.

$$Global\ Risk\ Index = \log_{10} \sum_i 10^{R_i} \tag{3}$$

where R_i is the risk of hazard i.

Although the basic principles are simple, the process of defining the bands and assigning values provoked several challenging questions. For example:

- How can severity be scored when a range of outcomes is possible?
- How can frequency be estimated given a shortage of data? Also, are qualitative methods reliable enough when humans are so poor at perceiving risk?
- How can we deal with the fact that some hazards behave like events (where the risk depends on the frequency of these events) and some behave like states (where the risk depends on the duration of the state)?

These questions are addressed in the following sub-sections.

6.4 How can severity be scored when a range of outcomes is possible?

When accidents do occur, there is a big variation in their severity. The same cause may sometimes result in damage-only accidents and at other times result in fatalities. This is due to the many degrees of freedom in a motorway system, including large variations in human behaviour, vehicle types and road conditions.

An examination of available accident data illustrates this variation in severity. The following table shows the number of casualties of different severities on British motorways in 2003, obtained from (DfT 2004).

Accident severity	All motorways	Ratio wrt fatalities
Fatal	217	1
Serious injuries	1,234	6
Slight injuries	12,578	58
Damage-only[2]	Not available	200 – 300

Table 2 Motorway casualties by severity 2003

For scoring severity, most safety standards propose a table similar to that in Table 3.

Severity classification	Interpretation
Catastrophic	Multiple deaths or serious injuries
Critical	A single death or serious injury
Marginal	A single injury
Negligible	Little or no potential of injury

Table 3 Typical severity classification scheme found in many safety standards

Applying a table like this is a problem for motorways. If a particular hazard is capable of producing a wide range of outcomes, which category is chosen? If we choose the most severe outcome, most motorway hazards will be critical or

[2] Damage-only statistics are not collected nationally. This ratio estimate was obtained by analysing records from the M42.

catastrophic. If on the other hand, we choose the most likely, most motorway hazards will be marginal or negligible.

One approach that has sometimes been adopted is to assess the risk separately for each severity. This typically yields three or four risk estimates for each hazard. Unfortunately, this is difficult to use when attempting to rank hazards. It also adds little value, because the best that can be done in estimating the relative frequencies of each outcome is to make estimates based on the ratios in Table 2.

To resolve this problem, we decided to evaluate the severity according to the expected distribution of outcomes. The figures in Table 2 show the expected range of severities for all motorway casualties. It is reasonable to suppose that many of the most common motorway hazards will produce outcomes with similar ratios, whilst some will be more inclined to produce fatalities and others will be more inclined to produce damage-only accidents. We therefore adopted a three-step scoring scheme for severity, as shown in Table 4.

Severity classification	Interpretation	Risk index
Severe	The proportion of accidents that are fatal is expected to be much higher than average.	2
Average	The distribution of accidents (i.e. ratio of damage-only to fatal) is expected to be similar to the motorway average.	1
Minor	The proportion of accidents that are fatal is expected to be much lower than average.	0

Table 4 Scoring scheme for severity of accident

As a rule of thumb, most hazards involving people in vehicles were scored as 'Average', as this is the most common type of motorway accident. Hazards involving pedestrians or motorcyclists, who are less well protected, were scored as 'Severe'. Hazards were scored 'Minor' when they involved vehicles at low speeds, or with low speed differentials. There was some suspicion that hazards scored in the 'Severe' and 'Minor' classes were not fully 10 times more or less severe than average. We therefore tested the sensitivity of the final results to a lower spread of severity.

6.5 How can frequency be estimated when there is a shortage of data?

Most of the statistics published on road accidents in Great Britain are obtained from the national database of road accidents, commonly referred to as STATS19. This database, which is compiled from police accident reports, includes figures for fatalities, critical injuries and slight injuries. It does not include damage-only accidents. The underlying cause of the accident is also entered into the database, but much of this detail is not available through the published results. Whilst this data is useful, it does not help in the estimation of completely new hazards and does not map well onto the breakdown of hazards that facilitates a comparison between the ATM and baseline motorways.

Having made the decision to model hazards at the level where driver behaviour deviates from the ideal, which may not result in an accident, the accident frequency can be broken into two factors, as follows:

$$\textit{Accident frequency} = (\textit{frequency of hazard}) \times (\textit{probability of hazard leading to accident}) \quad (4)$$

Taking the logarithm of both sides, this yields a number that can be directly inserted into the calculation of the risk index:

$$\log_{10}(\textit{Accident frequency}) = \log_{10}(\textit{frequency of hazard}) + \log_{10}(\textit{probability of hazard leading to accident}) \quad (5)$$

To assign values to this, tables can be drawn up which allocate the two factors into bands, where each band is a factor of 10 apart.

The following sections describe how estimating each of these two factors requires quite a different approach – one biased towards the analytical and the other tending to the intuitive. This proved to be a very powerful technique, which played to the strengths of each approach.

6.5.1 Estimating the hazard frequency

The hazard frequency can very often be derived by objective or analytical methods, making use of available data and simple probability models. It is often necessary to make assumptions, but these can be stated clearly and if necessary steps taken to verify the assumptions. Verifying assumptions typically involves collection of additional data, together with modelling or simulation. Some assumptions can only be tested when some way through the programme, perhaps only after the scheme has opened. However, by stating them clearly, a monitoring programme can be planned to collect the necessary information as soon as it becomes available.

To give an example, one of the most important new hazards is that of a vehicle being stopped on the hard shoulder when the hard shoulder is opened. How often is this likely to happen? By collecting data on typical stoppage patterns on the hard shoulder and making assumptions about what proportion of these would move to the ERAs, it was possible to model this with some confidence. Some of the results from this exercise were surprising. For example, it had been planned to check that the hard shoulder was clear in advance of opening by getting a traffic officer to drive through the section. However, the survey of stopping behaviours on the hard shoulder showed that a high number of short duration 'comfort' stops take place. On the ATM section of the M42 it is predicted that there will be 100 such stops per day, of average duration 3 minutes. This leads to a significant probability that a vehicle will stop on the hard shoulder between the time of the drive through and the opening of the hard shoulder. It even opens up the possibility of a vehicle stopping between the time of a camera scan and opening of the hard shoulder. It is therefore necessary to scan the hard shoulder with cameras immediately before opening, such that the

time delay between scanning and opening is kept very short, ideally well under a minute.

The fact that some participants found the results counter-intuitive shows the strength of separating out the parameters that can be measured and analysed. It seems that people are generally poor at estimating frequencies by intuition, particularly for events that only occur when multiple conditions apply.

6.5.2 Estimating the probability of accident

The probability of accident is much more difficult to model analytically because there are humans in the loop and human reliability is notoriously difficult to evaluate. One can look at a particular hazard and have no idea whether it will lead to an accident on 10% of occasions, 1% of occasions or 0.1% of occasions. However, it is much easier to sense intuitively how hazardous one event is relative to another event. For example, is a pedestrian more likely to have an accident when crossing from one side of a motorway to the other or when getting out of his car on the hard shoulder? Provided that one does not have to consider how often each situation will arise, it is easy to conclude that it is much more hazardous to cross six lanes of moving traffic than to be temporarily alongside one lane of moving traffic.

Because of these features this factor was typically estimated by a team of people in a workshop. The scoring scheme was qualitative only, and the hazards were scored relative to each other. Visibility was maintained of all hazards being evaluated and the team regularly checked that the scores were being applied in a consistent manner. In making the decisions, a checklist of questions were considered, such as:

- What action must be taken by a driver to avoid an accident, e.g. must they take some positive action such as stop or change lanes, or is slowing down and concentrating sufficient?
- How quickly must a driver respond to avoid an accident when encountering the hazardous situation?
- What indications would alert the driver to the presence of the hazard?
- In the case of a sign or signal failure, how credible is the erroneous signal – is it reinforced or contradicted by surrounding signals? Here errors that affect isolated signs and signals can be distinguished from those that cause the entire sequence of signals to be wrong.

Note that these questions have much in common with those used by the controllability method. In effect, the question is the same – what is the loss of control experienced when the hazardous event arises?

6.6 Hazardous events vs. hazardous states

Having carried out the initial hazard identification, it was observed that, while some hazards are events, many others are better regarded as hazardous states. The hazardous state may be initiated by a hazardous event, but the problems caused

depend on how long the state persists. For example, when a signal fails (a hazardous event) and therefore displays the wrong value (a hazardous state), it does so for a certain amount of time (an example is a signal failing to blank when it was displaying a red X Stop aspect). The longer it is wrong, the more drivers are influenced by it, and the more likely it is to cause an accident.

Hazardous states also produce accidents with a certain frequency. However, the two-part estimate of accident frequency is more conveniently treated as the product of the following two factors:

- First factor: probability of hazardous state being present
- Second factor: frequency with which hazardous state leads to an accident

Again, the first factor is best estimated by analysis, the second by intuition.

The use of hazardous states was not essential, but largely a matter of convenience, depending on which parameters were best estimated by analytical methods. For a few hazards, it seemed equally reasonable to score them as hazardous states or hazardous events. In some cases, the hazard identification process had generated two hazards that combined to produce an accident. For example, 'vehicle parked on the hard shoulder' is a hazardous state that leads to an accident if the hazardous event 'vehicle drives down hard shoulder when closed' occurs. To avoid double counting, only one of these hazards was included in the global risk scores. To mitigate the hazard, however, measures that reduce either the incidence of vehicles parked on the hard shoulder or the incidence of vehicles driving down the hard shoulder can be equally effective.

6.7 Risk estimation tables

For completeness, we show below the tables used for risk estimation on the ATM project.

$$\text{Risk index for hazardous event, } R_E = F_E + P_E + S \qquad (6)$$

where:

- F_E is the log of the frequency of the hazardous event given by Table 5
- P_E is the log of the probability that the hazardous event causes an accident, given by Table 6
- S is the log of the accident severity given previously in Table 4.

Frequency classification	Occurrences/year on M42 ATM section		F_E
Very frequent	10000	Hourly	6
Frequent	1000	A few times a day	5
Probable	100	Every few days	4
Occasional	10	Monthly	3
Remote	1	Annually	2
Improbable	0.1	Every 10 years	1
Incredible	0.01	Every 100 years	0

Table 5 Classification of hazard frequencies, where the hazard is regarded as an event

Classification	Interpretation	P_E
Probable	It is probable that this hazard, if it occurs, will cause an accident.	3
Occasional	This hazard, if it occurs, will occasionally cause an accident	2
Remote	There is a remote chance that this hazard, if it occurs, will cause an accident	1
Improbable	It is improbable that this hazard, if it occurs, will cause an accident	0

Table 6 Classification for 'Probability that hazardous event causes an accident'

Similarly the risk index for a hazardous state, $R_S = P_S + F_S + S$

where:

- P_S is the log of the probability that the hazardous state is present given in Table 7
- F_S is log of the rate at which accidents occur if the hazardous state is present given in Table 8
- S is the log of the severity, which was already given in Table 4.

Likelihood classification	Value on ATM section of M42		P_S
Very frequent	10	Ten occurrences at any time	6
Frequent	1	One occurrence at any time	5
Probable	0.1	5 weeks per year	4
Occasional	0.01	3 days per year	3
Remote	0.001	9 hours per year	2
Improbable	0.0001	50 minutes per year	1
Incredible	0.00001	5 minutes per year	0

Table 7 Classification of probability that hazardous state is present

Classification	Interpretation	F_S
Probable	This hazard, if present, will frequently cause an accident.	3
Occasional	This hazard, if present, will occasionally cause an accident	2
Remote	This hazard, if present, will infrequently cause an accident	1
Improbable	This hazard, if present, will rarely cause an accident	0

Table 8 Classification for 'Rate at which hazardous state causes an accident'

Because the above scheme does not define the relationship between hazardous events and states, we prefixed each risk score with the letter E or S to indicate which type of hazard it was. Thus a risk score might be quoted as E08 or S07 and references to the 'E09 hazards' soon became part of the project vocabulary. It was also widely appreciated that an E08 hazard represented a 10-fold increase in risk over an E07 hazard and that ten E07 hazards were equivalent to one E08 hazard.

Finally, it was necessary to remember that an E08 hazard was not necessarily equivalent to an S08 hazard. Insights into the relative weighting of hazardous states and events only emerged after most of the hazards had been scored.

7 Methodology for Demonstration of GALE

The ATM project has planned an extensive programme of before and after monitoring. Ultimately, it may be possible to use accident statistics to show that the motorway with ATM is at least as safe as it was before. However, it will be many years into the operation before statistically significant accident statistics are produced.

In advance of opening ATM, however, it is necessary to produce a safety case which gives confidence that GALE will be achieved. This section describes how this has been achieved using the risk assessment methodology described in the previous section.

7.1 Demonstration of GALE for all users

Consider first the meaning of the risk index that is yielded by our risk assessment methodology. For a given hazard, the antilogarithm of the risk index is essentially proportional to the frequency of serious accidents caused by that hazard on the ATM section of the M42. The sum of the antilog of the risk indices for all hazards is proportional to the total number of serious accidents expected on the ATM section of the M42. This is only true of course, if the following conditions are met:

- All significant hazards have been listed;
- Each hazard is defined in such a way that there is minimal overlap between each of the hazards. This necessitates stating clearly the conditions under which each hazard can arise (e.g. the operational mode, the type of road user, etc.) and which causes are responsible for each hazard.

Given that it is possible to generate a global risk score from the risk estimates for each hazard, it is theoretically possible to generate and compare two risk scores, one for the baseline and one with ATM in place. Of course, there are practical difficulties to overcome, such as:

- Risk estimates for individual hazards are rounded to the nearest factor of 10. There are very few risk mitigations which can change the risk enough to give a different result.
- The use of numbers can be misleading and open to misuse. They have a tendency to imply a level of precision that is not justified.

The approach that we adopted for demonstrating GALE on ATM involves the following steps:

- Carry out hazard identification for both the baseline and ATM versions of the motorway. Define the top-level hazards such that they cover all significant sources of risk, that there is minimal overlap between them and such that there is

maximum commonality between the baseline and ATM representations. In general, this yields a set of hazards that affect both the baseline and ATM, a (small) set that apply to the baseline only and another set that are new to ATM.

- Carry out risk estimation for each hazard, as it would apply to ATM (or for baseline only hazards, as it would apply to the baseline).
- For each hazard that applies to both ATM and the baseline, estimate how much the risk would differ from that of ATM. As with all risk estimation, there is uncertainty in such an estimate, so in practice we merely assigned one of five values as defined as in Table 9. In assigning these scores, we also recorded all the reasoning behind them.

Score	Interpretation
+ +	ATM risk more than a factor of 2 greater than baseline
+	ATM risk more than the baseline, up to a factor of 2
=	ATM risk similar to baseline
-	ATM risk less than the baseline, up to a factor of 2
- -	ATM risk less than the baseline by at least a factor of 2

Table 9 Scoring scheme for comparing ATM and baseline risks of individual hazards

- By making assumptions about the numerical value of the above scores, it is possible to construct a spreadsheet that calculates a global risk score for both the baseline and ATM versions of the motorway. This spreadsheet can also be used to test the sensitivity to other uncertainties, e.g. the effect of changing the scores for individual hazards, the effect of assumptions made about the relative value of event and state hazards, or the effect of risk reduction effort on the large scoring hazards.

To avoid the problems of attributing excessive precision to numbers, the final GALE argument, as presented in the safety case, is given in qualitative rather than quantitative terms. By examining the spreadsheet comparisons, it becomes very clear which hazards have the most significant impact on the comparison. The GALE argument is then presented in the following format:

1. ATM reduces the risk of a number of normal motorway hazards. A list of the hazards that have the most significant effect on the GALE comparison is provided, together with the reasons why ATM reduces the risk.
2. ATM introduces a number of new hazards to the operation of the motorway. However, various mitigating measures are being adopted to reduce the risk of these hazards. A list of the most significant new hazards is provided, together with a description of the mitigating measures being adopted.
3. ATM is capable of meeting the GALE target because the improvement to normal motorway hazards counterbalances the effect of the hazards introduced by virtue of ATM. The safety case reader is encouraged to examine the reasons and decide for him/herself whether the qualitative argument presented is persuasive.

It is possible that the GALE comparison, when first carried out, is not convincing. In this case, the risk scoring methodology makes it immediately apparent which hazards have the greatest impact on the safety of the scheme. This prompts two types of activity:

- A search for further risk reduction measures to apply to those hazards;
- A search for more data, to provide better understanding of those hazards.

7.2 Demonstration of GALE for specific road user groups

Having completed the overall GALE comparison, there remains the task of evaluating GALE for specific road user groups. This is necessary because it was agreed that the safety of one group of users would not be achieved at the expense of another. For each group of importance, this involves the following steps:

- Determine which hazards apply to that group.
- Decide what proportion of the risk for each hazard applies to that group. Adjust the risk index accordingly.
- Check whether the comparison between baseline and ATM is different for that group. In general, we found that this was the same.
- Compute a global risk score for both the baseline and ATM cases.
- Determine which hazards have the greatest effect on the comparison and construct a qualitative argument.

It was interesting to note that certain hazards which did not have a large impact on the global risk score became prominent when considering the risk to specific road user groups. An example of this was the effect that the additional equipment required for ATM could potentially have on the amount of maintenance required. This then prompted further work to reduce the risk to maintenance personnel, for example by arranging for much of the work to be possible from the emergency refuge areas and by implementing a permit to access system to avoid having maintenance personnel on site when controlled use of the hard shoulder is likely to commence.

8 Management of Safety Analysis Information

On a project of this scale, which is multi-disciplinary in nature and involves contractors from many different companies, a great deal of information accumulates that is relevant to safety. It is vital that this is managed in an efficient manner.

For this project, we developed an appropriate hazard log tool early in the programme and used this to maintain all safety information. Key features of the hazard log were:

- It was operated by web browser, obviating the need to install special software on each user's computer. With team members from many different organisations, working at many different locations, this was the only practicable way to ensure accessibility for all team members.

- A full audit trail of any changes to the hazard log was automatically maintained. This enabled many people to be permitted to contribute to the log without risk of losing information.
- Accidents, hazards and causes were represented in a structured manner, which enabled the relationships between them to be seen clearly.
- Behind each entry, it was possible to include an unlimited amount of supporting information, included attached documents. When entering risk assessments, the full rationale was provided along with the result. This proved vital whenever the risk assessments were reviewed, as it is easy to forget important considerations that were taken into account at the time. By stating explicitly the underlying assumptions, it is possible to refine the estimates progressively over time as further information becomes available. This is impossible if the risk assessments are stored in a crude database, such as a spreadsheet, where there is no provision for storing the rationale.
- As the need for mitigations were identified, these were entered into the hazard log as safety requirements. The tool supported a many to many relationship between hazards and safety requirements. It also provided for tracking the progress of safety requirements, including the acceptance of the requirement, the entry of a verification plan and the entry of verification evidence.
- The hazard log provided for the management of actions or tasks. A task could be associated with any other entry in the hazard log (including hazards, causes, accidents and safety requirements). Each task supported a workflow that involved assigning it an owner and requiring that person to enter a task plan and the results of task progress.

To oversee the safety programme, a 'Hazard Control and Review Committee' was formed. This was chaired by the Safety Champion and contained representatives of the project sponsor, the managing consultant and the work stream managers. This committee reviewed all risk assessments, tasks and safety requirements. The hazard log was central to this process – a live connection to the log was projected in the meetings and the content reviewed directly. This enabled convenient navigation around the hazard log as questions arose. The review status of each entry was also recorded in the hazard log, simplifying the task of identifying which entries required review.

9 Safety Case

The first version of a safety case for the ATM project has now been written. This is presented in four volumes – a top-level system safety case, with sub-level safety cases for operations, telematics and infrastructure.

Several versions of the safety case have been planned:

- A 'design' version, which was released at completion of the design. Writing this helped to identify any weaknesses or omissions in the design or the design

process in sufficient time to enable them to be rectified. It also provided an early opportunity for people to comment on the structure of the safety case.

- An 'as-built' version, to be released and approved prior to the commencement of controlled use of the hard shoulder. This version will contain the results of the GALE comparison, based on the risk assessment methodology described above.
- Later versions, to be updated based on the experience of operating the scheme.

The strength of the tool support provided by the hazard log has greatly simplified the writing of the initial version of the safety case and will be invaluable in updating it as future versions are released. By ensuring that all actions and safety requirements are closed out in the hazard log, we are able to gain considerable confidence that important safety issues have not been overlooked or forgotten.

10 Conclusions

The M42 ATM project was the first major HA project to require an IEC 61508 safety programme and a formal safety case. This paper has described the risk assessment methodology that was developed to suit the particular features of the motorway environment. This methodology differed somewhat from the examples given in most safety standards, for the following reasons:

- The safety target was different – rather than requiring that the risk of each hazard be reduced to ALARP, it required that the global risk be equivalent to or better than the preceding system (GALE). The hazard identification and risk assessment methodologies had to be capable of demonstrating whether this target was likely to be met.
- For each hazard, there is potentially a wide range of accident severities. This is difficult to represent properly by traditional risk matrix methods.
- The effect of the 'human in the loop', along with a lack of data, makes it difficult to estimate accident frequencies. This was dealt with by dividing the accident frequency into two factors: the hazard frequency, which could be estimated most effectively by objective analytical means and the probability of accident, which was estimated in a relative sense by more intuitive processes.

The safety work was given high visibility in the project, through the use of a hazard log which was accessible to all project team members and which managed safety-related tasks through to completion. The wealth of information captured within the tool considerably simplified the task of writing the safety case.

The risk assessment methodology developed for the M42 ATM project was very successful in serving the needs of the project. It is believed that, with suitable redefinition of the frequency tables, the methodology would be applicable to many other ITS projects. However, other methodologies have also been trialled within the HA and further work is required before a standard is adopted.

Another valuable output of the work is the extensive set of hazards and causes that have been identified. These have been carefully defined and organised, so as to minimise the overlap between them. From this work, it should be relatively easy to derive a generic set of motorway hazards that can be used as the basis for many different future projects.

References

DfT (2000). "Tomorrow's Roads - Safer for Everyone", March 2000, available on http://www.dft.gov.uk.

DfT (2004). Department for Transport, "Road Casualties in Great Britain: 2003", Table 24, June 2004, available on http://www.dft.gov.uk.

IEC (1998). International Standard IEC 61508, "Functional safety of electrical/ electronic/programmable electronic safety-related systems" (7 parts), 1998.

MIRA (2000). MIRA and University of Leeds, "Framework for the Development and Assessment of Safety-Related UTMC Systems", UTMC22, Release version 1.0, March 2000.

MIRA (2004). MISRA Technical Report, "Hazard classification for moving vehicle hazards – controllability', Version 1, May 2004.

Safety Risk Assessment by Monte Carlo Simulation of Complex Safety Critical Operations

Henk A.P. Blom, Sybert H. Stroeve and Hans H. de Jong
blom@nlr.nl; stroeve@nlr.nl; hdejong@nlr.nl
National Aerospace Laboratory NLR
Amsterdam, The Netherlands

Abstract

This paper gives an overview of performing safety risk assessment of a safety critical operation with support of Monte Carlo (MC) simulation. The approach is outlined for an air traffic example involving aircraft departing from a runway, which is occasionally crossed by taxiing aircraft. At the airport considered, a Runway Incursion Alert System (RIAS) is installed to warn the air traffic controller in case of impending runway incursions. The paper explains the key issues to be mastered in performing a MC simulation supported safety risk assessment of this kind of operation. To begin with, one has to develop an appropriate simulation model, and a sound way to speed up the MC simulation based on this model. Complementary, one has to validate the simulation model versus the real operation, and the simulation supported approach has to be embedded within the safety risk assessment of the total operation. For this application example MC simulation results are given and the way of feedback to the design of the operation is outlined.

1 Introduction

Among the class of complex and safety critical industries, air traffic is an interesting example that poses exceptional challenges to advanced design. By its very nature, each aircraft has its own crew, and each crew is communicating with several human operators in different air traffic management (ATM) and airline operational control (AOC) centres on the ground in order to timely receive instructions critical to a safe flight. In addition, from an organisational perspective, air traffic involves interactions between many stake holders: pilots, air traffic controllers, airline operation centres, airport authorities, government regulators and the public travelling. Figure 1 highlights this characteristic feature of interplay between distributed decision making and safety criticality both for air traffic and for other complex or safety-critical industries, such as finance and nuclear and chemical plants. Among the safety critical industries, air traffic stands out regarding the many distributed levels of interactions in control and decision making.

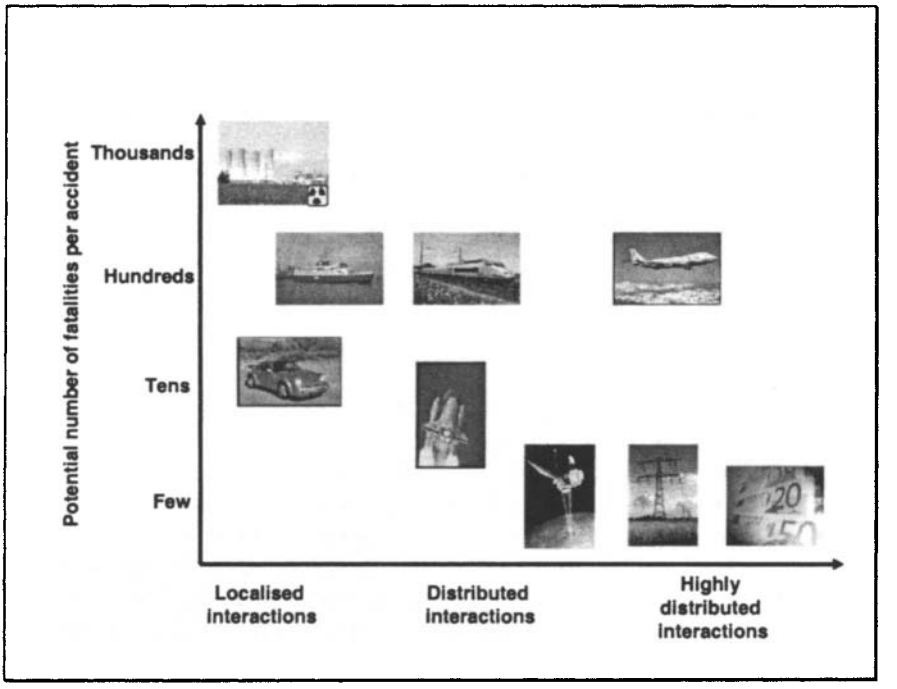


Figure 1: Air traffic compared with other complex and/or safety-critical industries in terms of potential number of fatalities per accident and the level of distributed interactions

The implication is that safety of air traffic is the result of interactions between multiple human operators, procedures (including spacing and separation criteria), and technical systems (hardware and software) all of which are highly distributed. Since safety depends crucially on the interactions between the various elements of a system, providing safety is more than making sure that each of these elements function properly. It is imperative to understand the safety impact of these interactions, particularly in relation to non-nominal situations.

Traditional ATM design approaches tend to be bottom-up, that is starting from developing concept elements aimed at increasing capacity, and next to extend the design with safety features. The advantage of the traditional approach is that advanced design developments can be organised around the clusters of individual elements, i.e., the communication cluster, the navigation cluster, the surveillance cluster, the automation tools cluster, the controllers/pilots and their human machine interfaces (HMIs), the advanced procedures, etcetera. The disadvantage of this traditional approach is that it fails to fully address the impact of interactions between controllers, pilots and ATM systems on safety.

A goal oriented approach would be to design ATM such that safety has been built in at the capacity-level required. From this perspective, safety assessment forms a primary source of feedback (Figure 2) in the development of advanced ATM

designs. An early guidance of ATM design development on safety grounds can potentially avoid a costly redevelopment program, or an implementation program that turns out to be ineffective. Although understanding this idea is principally not very difficult, it can be brought into practice only when an ATM safety assessment approach is available that provides appropriate feedback to the ATM designers from an early stage of the concept development (Figure 2). This feedback should provide information on which safety-capacity issues are the main contributor to unsafety.

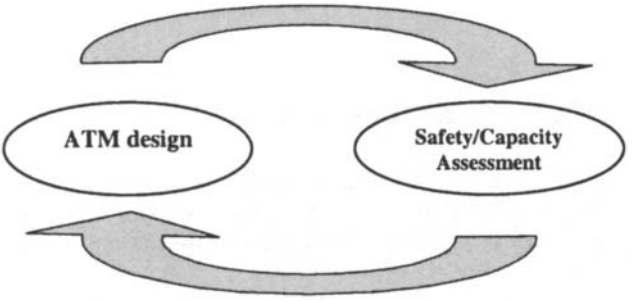


Figure 2: Safety feedback based ATM design.

For oceanic air traffic, the civil aviation community has developed a mathematical model to estimate mid-air collision risk levels as a function of spacing values in route structures (ICAO, 1988). This model is known as the Reich collision risk model; it assumes that the physical shape of each aircraft is a box, having a fixed orientation, and the collision risk between two aircraft is approximated by an expression that has proven to be of practical use in designing route structures (Hsu, 1981). Apart from the approximation, the most severe shortcoming is that the Reich model does not adequately cover situations where interaction between pilots and controllers play a crucial role, e.g. when controllers monitor the air traffic through surveillance systems and provide tactical instructions to the aircraft crews. In order to improve this situation, NLR has developed a safety risk assessment methodology which provides safety risk feedback to advanced air traffic operation design. The resulting safety risk assessment methodology has been named TOPAZ, which stands for Traffic Organization and Perturbation AnalyZer (Blom, 2001a). Within TOPAZ, Petri net modelling and Monte Carlo simulation has proven to deserve a key role in modelling and assessment of advanced air traffic operations on safety risk (Bakker and Blom, 1993; Blom et al., 2001b, 2003a,b,c; Everdij&Blom, 2002, 2003, 2005; Stroeve et al., 2003; Blom&Stroeve, 2004). In this respect it is relevant to notice that the use of Petri nets has been shown to work well in modelling safety critical operations in nuclear and chemical industries (e.g. Labeau et al., 2000).

The aim of this paper is to explain how the TOPAZ methodology effectively uses Monte Carlo simulation in safety risk assessment of an advanced air traffic operation. Emphasis is on how Monte Carlo simulation of safety risk works and how this is embedded within a complete safety risk assessment cycle.

This paper is organized as follows. First, section 2 provides an overview of the steps of the TOPAZ safety risk assessment cycle and for which step Monte Carlo simulation is of direct use. Next, section 3 provides an overview of how to develop a Monte Carlo simulation model of a given operation. In order to keep the explanation concrete, a particular example is introduced first. Subsequently section 4 provides an overview of key issues that have to be taken into account when using a Monte Carlo simulation supported safety risk assessment. Section 5 presents Monte Carlo simulation results for the particular example identified in section 3. Finally, conclusions are drawn in section 6.

2 Safety Risk Assessment Steps

An overview of the steps in a TOPAZ safety risk assessment cycle is given in Figure 3. Although the cycle itself is very much in line with the established risk assessment steps (e.g. Kumamoto and Henley, 1996), some of these steps differ significantly.

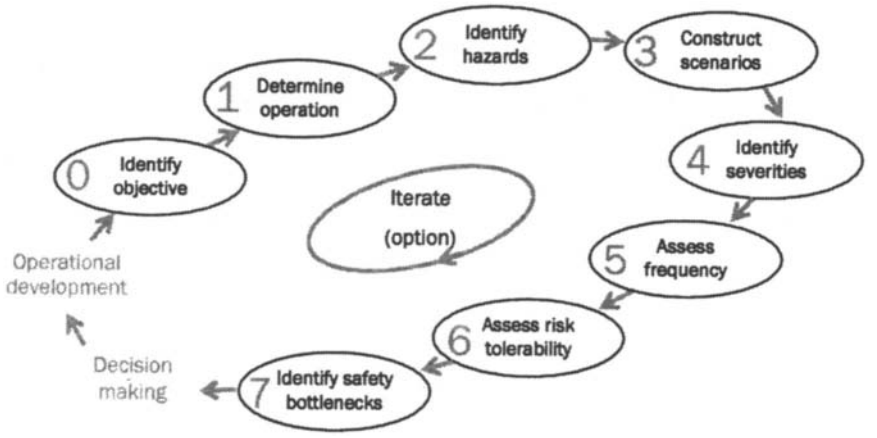


Figure 3: Steps in TOPAZ safety risk assessment cycle.

In step 0, the objective of the assessment is determined, as well as the safety context, the scope and the level of detail of the assessment. The actual safety assessment starts by determining the operation that is assessed (step 1). Next, hazards associated with the operation are identified (step 2), and aggregated into safety relevant scenarios (step 3). Using severity and frequency assessments (steps 4 and 5), the safety risk associated with each safety relevant scenario is classified (step 6). For each safety relevant scenario with a (possibly) unacceptable safety risk, the main sources contributing to unsafety (safety bottlenecks) are identified (step 7), which help operational concept developers to learn for which safety issues they should

develop improvements in the ATM design. If the ATM design is changed, a new safety risk assessment cycle of the operation should be performed in order to investigate how much the risk posed by previous safety issues has been decreased, but also to assess any new safety issues that may have been introduced by the enhancements themselves.
The following subsections present the risk assessment steps of a TOPAZ cycle in more detail. Then it also becomes clear that Monte Carlo simulation plays a key role in step 5: assess frequency.

Step 0: Identify objective
Before starting the actual safety assessment, the objective and scope of the assessment are set. This should be done in close co-operation with the decision makers and designers of the advanced operation. Also, the safety context must be made clear, such that the assessment is performed in line with the appropriate safety regulatory framework.
An important issue for setting the safety context is the choice of safety criteria with respect to which the assessment is performed. Depending of the application, such criteria are defined for particular *flight condition* categories (e.g. flight phases or sub-phases) and for particular *severity* categories (e.g. accident, serious incident). Typically, within the chosen context, these criteria define which *flight condition/severity* categories have to be evaluated and which frequency level forms the Target Level of Safety (TLS) threshold per *flight condition/severity* category.

Step 1: Determine operation
Step 1 serves for the safety assessors to obtain a complete and concise overview of the operation, and to freeze this description during each safety assessment cycle. Main input to step 1 is a description of the operational concept from the designers, while its output is a sufficiently complete, structured, consistent and concise description of the operation considered. The operation should be described in generic terms, the description should provide any particular operational assumption to be used in the safety assessment, and the description has to be verified by the operational concept designers. Typically during this step, holes and inconsistencies in the concept as developed are also identified and immediately fed back to the design team

Step 2: Identify hazards
The term hazard is used in the wide sense; i.e. an event or situation with possibly negative effects on safety. Such a non-nominal event or situation may evolve into danger, or may hamper the resolution of the danger, possibly in combination with other hazards or under certain conditions. The goal of step 2 is to identify as many and diverse hazards as possible. Hazard identification brainstorming sessions are used as primary means to identify (novel) hazards.
In system engineering, the functional approach to hazard identification is well-known. In this approach it is attempted to determine all possible failure conditions and their effects, for each function that plays a role in the operation, including the human operator tasks. Unfortunately, the approach cannot identify all hazards related

to an operation that involves human operators. An important reason for this is that the performance of air traffic controllers and pilots depend on their (subjective) situational awareness. From a human cognition perspective a particular act by an air traffic controller or pilot can be logical, while from a function allocation perspective the particular act may be incorrect. Such incidents are often called "errors of commission" (Sträter et al., 2004). An example of error of commision in the crossing operation is that because of the complicated taxiway structure, the pilot thinks to be taxiing far from the runway, while in reality, he starts crossing the runway without noticing any of the runway signs.

Another well-known technique of hazard identification is the HAZOP (HAZard and OPerability) method. With this method, hazards are identified and analyzed using sessions with operational experts. At the same time, the experts come up with potential solutions and measures to cope with the identified hazards (Kletz, 1999). The advantage of HAZOP with respect to the functional approach is that also non-functional hazards are identified during the brainstorm with operational experts. However, in applying HAZOP, one needs to take care that hazard analysis and solution activities do not disturb the hazard identification process, which could leave certain hazards unidentified or inappropriately "solved". Leaving such latent hazards in a design typically is known to be very costly in safety critical operation.

Based on the experience gained in using the hazard identification part of HAZOP in a large number of safety analyses and on scientific studies of brainstorming, NLR has developed a method of hazard identification for air traffic operations – by means of pure brainstorming sessions (De Jong, 2004). In such a session no analysis is done and solutions are explicitly not considered. An important complementary source is formed by hazards identified in previous studies on related operations. For this purpose, hazards identified in earlier studies are collected in a TOPAZ database.

Step 3: Construct scenarios

When the list of hazards is as complete as reasonably practicable, it is processed to deal with duplicate, overlapping, similar and ambiguously described hazards. First, per flight condition selected in Step 0, the relevant scenarios which may result from the hazards are to be identified using a full list of potentially relevant scenarios, such as for instance 'conflict between two aircraft merging onto one route' or 'aircraft encounters wake vortex of parallel departure'. Per *flight condition*, each potentially relevant scenario is subsequently used as crystallisation point upon which all applicable hazards and their combined effects are fitted. If hazards are not appropriately addressed by the crystals developed so far, then additional crystallisation points are defined. The output of such crystallisation process is a bundle of event/condition sequences and effects per crystallisation point, and these are referred to as a safety relevant scenario. This way of constructing scenarios aims to bring into account all relevant ways in which a hazard can play a role in each *flight condition/severity* category.

In order to cope with the complexity of the various possible causes and results, clusters of similarly crystallised hazards are identified. A cluster of hazards could for instance be the set of 'events causing a missed approach to deviate from the normal path'. An example is given in Figure 4. It should also be noticed that the same cluster

of hazards may play a role in one or more safety relevant scenerios.
Each of the identified hazards can be of the following types:

- a root hazard (cluster), which may cause a safety relevant scenario; or
- a resolution hazard (cluster), which may complicate the resolution of a safety relevant scenario.

For an appropriate safety risk assessment, all combinations of root and resolution hazards have to be evaluated in the next steps.

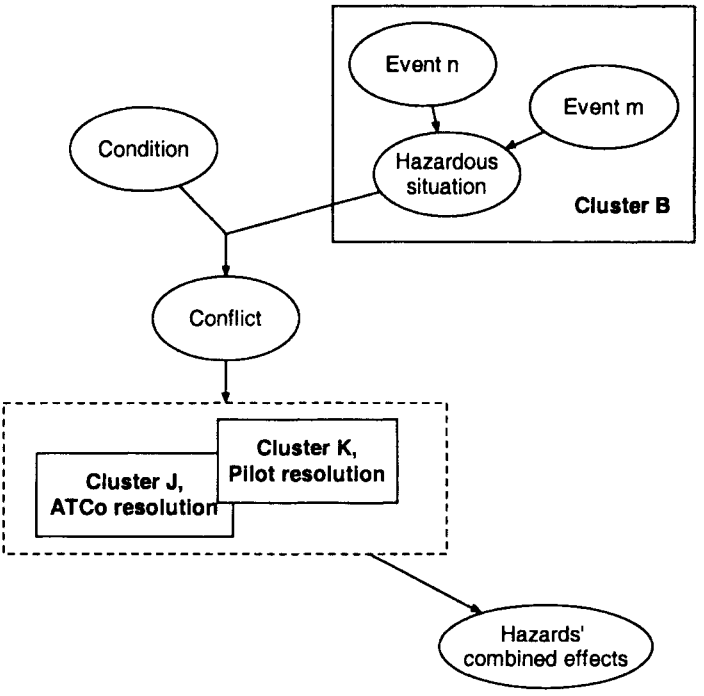


Figure 4: Example of a safety relevant scenario diagram.

Step 4: Identify severities

For each of the safety relevant scenarios identified in step 3, it is determined which of the *severity* categories selected in step 0 are applicable to its possible effects. Safety experts should assess which of the severities are applicable for each safety relevant scenario, by consultation of and review by operational experts. For each safety relevant scenario the effects and their severities depend on many factors, such as the conditions under which the scenario starts and evolves, the geometry of the scenario, and the possibilities of (timely) resolution of the conflict. Therefore, a range of *severities* may apply to a safety relevant scenario. If necessary, the structuring of the events in the safety relevant scenarios of step 3 are updated such that each applicable *severity* category is linked to the occurrence of specific event sequences.

Step 5: Assess frequency

Next, for each possible severity outcome of each safety relevant scenario, the occurrence frequency is evaluated by making use of an appropriate tree per safety relevant scenario. The probability of the top event in the tree is expressed as a sum of a product of probabilities of applicable conditional events. For assessing the factors in these trees, primary sources of data are operational experts and databases. Examples of databases are aviation safety databases, local controller reporting system(s), et cetera. For appropriate use of such data dedicated operational expertise is taken into account. Hence, important input for the frequency assessments is always formed by interviews with operational experts (including experienced pilots and controllers) who are as much as is possible familiar with the operation under consideration. Qualitative expressions are to be translated in quantitative numbers when the selected safety criteria of step 0 also are expressed in numbers. Complicating factors in assessing at once the frequency of a conflict ending in a given severity can be that there is often little or no experience with the new operation, and that the situation may involve several variables. This holds especially for the more severe outcomes of a safety relevant scenario, since these situations occur rarely, and consequently little information is at hands about the behaviour of air traffic controllers and pilots in such situations. For these difficult safety relevant scenarios it is logical to make use of Monte Carlo simulation of safety risk. This approach has three clear advantages: 1) the quality of the risk estimate improves; 2) it is possible to estimate a 95% confidence interval; and 3) once a MC simulation tool for a particular application has been developed it can be re-used for assessing safety risk for similar applications. The next sections explain for an example safety risk assessment by MC simulation.

Step 6: Assess risk tolerability

The aim of this step is to assess the tolerability of the risk for each of the *flight condition/severity* categories selected in step 0. First the total risk per *flight condition/severity* category is determined by summing over the assessed risk contributions per safety relevant scenario for that *flight condition/severity* category. This summation takes into account both the expected value and the 95% confidence interval of the risk summation. Next for each *flight condition/severity* category the total risk expected value and the 95% confidence interval are compared against the TLS selected in step 0.

Step 7: Identify safety bottlenecks

From the risk tolerability assessment, it follows which safety relevant scenario(s) contribute(s) most to the expected value and the 95% confidence interval of the risks that has been qualified as being not below the TLS. For each safety relevant scenario the hazards or conditions that contribute most to the high risk level or confidence interval are identified and localised during step 7. These are referred to as the safety bottlenecks. If desired, this may also be done for assessed risk levels that are just below the TLS. The identification and localisation of safety bottlenecks is important as it gives operational concept designers directions for searching potential risk

mitigating measures of the operation, and it gives the safety assessment experts the hazards and conditions for which the reduction of uncertainty has priority.

3 Monte Carlo Simulation Model

3.1 Active Runway Crossing Example

The Monte Carlo simulation-based risk assessment approach will be illustrated for an active runway crossing operation. This example accounts for a number of interacting human agents (pilots and controllers). The runway configuration of the active runway crossing operation considered is shown in Figure 5. The configuration takes into account one runway, named runway A, with holdings for using the runway from two sides (A1 and A2) and with crossings (C1, C2, D1 and D2) and exits (E1, E2, E3 and E4). The crossings enable traffic between the aprons and a second runway, named runway B. Each crossing has remotely controlled stopbars on both sides of the runway. Also the holdings have remotely controlled stopbars and each exit has a fixed stopbar.

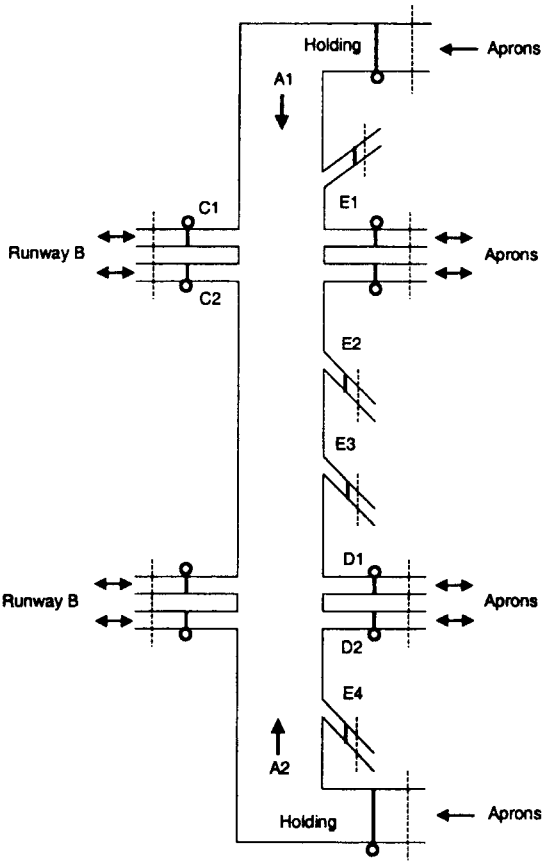


Figure 5: Runway configuration of active runway crossing procedure.

The involved human operators include the start-up controller, the ground controller, the runway A controller, the runway B controller, the departure controller, and the pilots flying (PF's) and pilots not flying (PNF's) of aircraft taking off and aircraft crossing. Communication between controllers and aircraft crews is via standard VHF R/T (Very High Frequency Receiver/Transmitter). Monitoring by the controllers can be by direct visual observation under sufficiently good visibility conditions; it is supported by ground radar surveillance. The runway A controller is supported by a runway incursion alert system and a stopbar violation alert system. The runway A controller manages the remotely controlled stopbars and the runway lighting. Monitoring by the aircraft crews is by visual observation, supported by the VHF R/T party-line effect.
In the runway crossing operation considered, the control over the crossing aircraft is transferred from the ground controller or the runway B controller (depending on the direction of the runway crossing operation) to the runway A controller. If the runway A controller is aware that the runway is not used for a take-off, the crew of an aircraft intending to cross is cleared to do so and subsequently the appropriate remotely controlled stopbar is switched off. The PNF of the crossing aircraft acknowledges the clearance and the PF subsequently initiates the runway crossing. When the crossing aircraft has vacated the runway, then the PNF reports this to the runway A Controller. Finally, the control over the aircraft is transferred from the runway A controller to either the runway B controller or the ground controller.

3.2 Safety Relevant Scenarios

Prior to the development of a quantitative accident risk model for the active runway crossing operation considered, all risk assessment steps had been performed using an expert-based approach. In this study the following safety relevant scenarios were found:

- Scenario I: Aircraft erroneously in take-off and crossing aircraft on runway;
- Scenario II: Aircraft erroneously crossing and other aircraft in take-off;
- Scenario III: Aircraft taking off and runway unexpectedly occupied;
- Scenario IV: Aircraft crossing and runway unexpectedly occupied by aircraft;
- Scenario V: Aircraft crossing and vehicle on runway;
- Scenario VI: Collision between aircraft sliding off runway and aircraft near crossing;
- Scenario VII: Aircraft taking off and vehicle crossing;
- Scenario VIII: Jet-blast from one aircraft to another; and
- Scenario IX: Conflict between aircraft overrunning/climbing out low and aircraft using a nearby taxiway.

From this expert-based study it followed that of all identified safety relevant scenarios, for scenarios I, II and III it was difficult to assess the risk sufficiently accurate using an expert based approach. For these three scenarios it is therefore useful to assess the risk through performing Monte Carlo simulations.
In this paper, we focus on the details of a Monte Carlo simulation accident risk

model for scenario II. In this scenario there is one aircraft that takes off and has been allowed to do so and there is one aircraft that crosses the runway while it should not. Taxiing along a straight line over one of the standard runway crossings (i.e., via C1, C2, D1 or D2 in Figure 5) is considered.

3.3 Multi-Agent Situation Awareness in the Simulation Model

The safe organisation of co-operation between pilots and controllers in air traffic depends to a large extent on the "picture" or situation awareness (SA) maintained by each of the pilots and controllers. When a difference, even a small one, sneaks into the individual pictures and remains unrecognised, this may create unnoticed miscommunication and a subsequent propagation and increase in differences between the individual pictures. Eventually the situation may spiral out of control, with potentially catastrophic results. Hence any mismatch between individual pictures forms a serious hazardous condition in maintaining a safe organisation. Many hazards identified for the runway crossing operation were of this type.

Endsley (1995) has defined human SA as the perception of elements in the environment within a volume of time and space, the comprehension of their meaning, and the projection of their status in the near future. Stroeve et al. (2003) and Blom and Stroeve (2004) have captured these perception, comprehension and projection notions of SA mathematically in terms of three components: State SA, Mode SA and Intent SA. They also extended this single (human) agent SA concept to a multi-agent SA concept for operations involving multiple humans and systems, inclusive the basic updating mechanisms of such multi-agent SA.

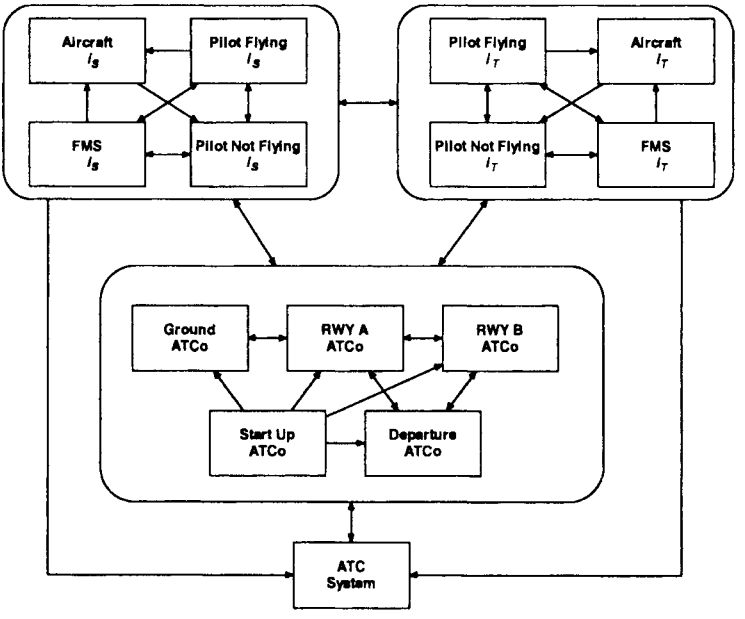


Figure 6: Relations between agents identified for the active runway crossing operation.

As depicted in Figure 6, for the active runway crossing operation we identified a need to model 10 agent types (7 humans and 3 systems) and their interactions:

- Pilots flying;
- Pilots not flying;
- (Each) aircraft;
- Aircraft's flight management systems (FMS);
- Runway A controller;
- Runway B controller;
- Ground controller;
- Departure controller;
- Start-up controller;
- ATC system, which we broadly define to include airport manoeuvre control systems, air traffic communication and surveillance systems, airport configuration and environmental conditions.

3.4 Dynamic Stochastic Modelling

The Monte Carlo simulations are based on dynamic stochastic models of all relevant agents. These simulation models are mathematically specified using the Dynamically Coloured Petri Net (DCPN) formalism (Everdij and Blom, 2003, 2005). A high-level overview of the agents modelled is provided next.

Taking-off Aircraft

The model of the taking-off aircraft represents the ground run, airborne transition and airborne climb-out phases and includes the possibility of a rejected take-off. The taking-off aircraft initiates its take-off from a position near the runway threshold and may have a small initial velocity. The aircraft may have diminished acceleration or deceleration power. Two types of aircraft are included in the model: medium-weight aircraft and heavy-weight aircraft.

Taxiing Aircraft

The model of the taxiing aircraft represents aircraft movements (hold, acceleration, constant speed, deceleration) during taxiing. The taxiing aircraft enters the taxiway leading to a runway crossing at a position close to the remotely controlled stopbar, with a normal taxiing speed or initiates taxiing from stance. The entrance time of the crossing aircraft is uniformly distributed around the take-off start time. The taxiing aircraft may have diminished deceleration power. Two types of aircraft are included in the model: medium-weight aircraft and heavy-weight aircraft.

Pilot Flying of Taking-off Aircraft

Initially, the pilot flying (PF) of a taking-off aircraft has the SA that taking-off is allowed and initiates a take-off. During the take-off the PF monitors the traffic situation on the runway visually and via the VHF communication channel. The PF starts a collision avoiding braking action if a crossing aircraft is observed within a critical distance from the runway centre-line or in reaction to a call of the controller, and if it is decided that braking will stop the aircraft in front of the crossing aircraft.

Further details of taking-off aircraft PF model are given by (Stroeve et al., 2003).

Pilot Flying of Taxiing Aircraft
Initially, the PF expects that the next airport way-point is either a regular taxiway or a runway crossing. In the former case the PF proceeds taxiing and in the latter case the PF may have the SA that crossing is allowed. The characteristics of the visual monitoring process of the PF depend on the intent SA. In case of awareness of a conflict, either due to own visual observation or due to a controller call, the PF stops the aircraft, unless it is already within a critical distance from the runway centre-line. Further details of taxiing aircraft PF model are given by (Stroeve et al., 2003).

Runway Controller
The runway A controller visually monitors the traffic and has support from a stopbar violation alert and a runway incursion alert. If the controller is aware that a taxiing aircraft has passed the stopbar, a hold clearance is given to both taxiing and taking off aircraft. Further details of the runway controller model are given by (Stroeve et al., 2003).

Radar Surveillance System
The model of the radar surveillance system represents position and velocity estimates for both aircraft. There is a probability that radar surveillance is not available, resulting in track loss. Radar surveillance data is used as basis for ATC stopbar violation alerting and ATC runway incursion alerting.

ATC Alerts
Two types of ATC alerts are included in the model: a stopbar violation alert and a runway incursion alert. A stopbar violation alert is presented to the controller if surveillance data indicates that an aircraft has passed an active stopbar. There is a probability that the stopbar violation alert system does not function, implying that there will be no alert. A runway incursion alert is presented to the controller if radar surveillance data indicates that the taxiing aircraft is within a critical distance of the runway centre-line and the taking-off aircraft has exceeded a velocity threshold in front of the runway crossing. There is a probability that the runway incursion alert system does not function, implying that there will be no alert.

VHF Communication Systems
The model for the VHF communication system between the runway controller and the aircraft crews accounts for the communication system of the aircraft, the communication system of the controller, the tower communication system, the frequency selection of aircraft communication system and the VHF communication medium. The nominal status of these communication systems accounts for direct non-delaying communication. The model accounts for a probability of delay in or failure of the communication systems.

4 Use of Simulation Model in Risk Assessment

Once the simulation model has been specified, there are several important aspects that have to be taken into account during the preparation, execution and interpretation of the Monte Carlo simulations. This section explains these aspects.

4.1 Does the Simulation Model Cover the Identified Hazards?

During step 2 of the safety assessment cycle, a lengthy list of hazards, including non-nominal situations, has been identified. These hazards contribute individually and possibly in combination with other hazards to the safety risk of the operation considered. Hence it is quite important to verify prior to performing the simulations that the hazards identified in step 2 of the assessment cycle are covered by the model. The verification process consists of specifying per hazard how it is captured by the simulation model. A special class of hazards is formed by the situation awareness related hazards. Table 1 shows three of such situation awareness related hazards and includes a short explanation how these hazards are covered by the simulation model.

Table 1: Examples of situation awareness related hazards and their simulation model.

Pilots become confused about their location at the airport because of complexity of the airport layout.	State SA of the PF of a taxiing aircraft is that its aircraft is at a location that differs from the actual location.
Crew of taxiing aircraft is lost and therefore not aware of starting to cross a runway.	Intent SA of PF is and stays taxiing while PF starts crossing the runway.
RIAS is switched off by maintenance and controllers are not informed.	RIAS working or not is not connected to Mode SA of controllers.

Inevitably this verification of each hazard against the model will lead to the identification of hazards that are not (yet) covered by the simulation model. For non-covered hazards the simulation model developers should consider to further extend the simulation model prior to performing Monte Carlo simulations.

4.2 Parametrisation of the Simulation Model

During the mathematical specification of the simulation model there is no need to bother about the correct parameter values to be used during the Monte Carlo simulation. Of course, this is addressed prior to running the simulations. In principle there are three kinds of sources for parameter values. The ideal source would consist of sufficient statistical data that has been gathered under the various contextual conditions for which the risk assessment has to be performed. In practice such ideal sources almost never exist. Instead one typically has to work with limited statistical data that has been gathered under different conditions. Fortunately there often are two complementary sources: domain expertise and scientific expertise (on safety and human factors). In the context of Monte Carlo

simulation this means one fuses statistical and expertise sources into a probability density function for the possible values of each parameter. Typically the mean or mode of such a density function is then used as the best estimate of the parameter value to be used when running the Monte Carlo simulation.

4.3 Speeding up Monte Carlo Simulations

Air traffic is a very safe means of transport. Consequently, the risk of collision between two aircraft is extremely low. The assessment of such low collision risk values through straightforward Monte Carlo simulation would need extremely lengthy computer simulation periods. In order to reduce this to practicable periods, five to six orders of magnitude in speeding up the Monte Carlo simulation are needed. The basis for realizing such speed-up factors in Monte Carlo simulation consists of decomposing accident risk simulations in a sequence of conditional Monte Carlo simulations, and then to combine the results of these conditional simulations into the assessed collision risk value. For the evaluation of logical systems good decomposition methods can often be obtained by Fault and Event Tree Analysis. Because air traffic operations involve all kinds of dependent, dynamic and concurrent feedback loops, these logic-based risk decomposition methods cannot be applied without adopting severe approximations, typically by assuming that events/entities happen/act independent of each other.
The stochastic analysis framework, that has shown its value in financial mathematics (e.g. Glasserman, 2004), is exploited by the TOPAZ methodology to develop Monte Carlo simulation models and appropriate speed-up factors by risk decomposition. The power of these stochastic analysis tools lies in their capability to model and analyse in a proper way the arbitrary stochastic event sequences (including dependent events) and the conditional probabilities of such event sequences in stochastic dynamic processes (Blom et al., 2003c; Krystul&Blom, 2004). By using these tools from stochastic analysis, a Monte Carlo simulation based risk assessment can mathematically be decomposed into a well-defined sequence of conditional Monte Carlo simulations together with a subsequent composition of the total risk out of these conditional simulation results. The latter composition typically consists of a tree with conditional probabilities to be assessed at the leaves, and nodes which either add or multiply the probabilities coming from the subbranches of that node. Within TOPAZ such a tree is referred to as a collision risk tree (Blom et al., 2001, 2003).

For the active runway crossing example, the particular conditions taken into account for this risk decomposition are:

- The type of each aircraft (either a medium-weight or a heavy-weight);
- The intent SA of the PF of a crossing aircraft concerning the next way-point (*Taxiway/Crossing*) and concerning allowance of runway crossing (*Allowed/Not Allowed*);
- The alert systems (functioning well or not);
- The remotely controlled stopbar (functioning well or not); and
- The communication systems (functioning well or not).

Based on the simulation model and the accident risk decomposition, Monte Carlo simulation software is developed to evaluate the event probabilities and the conditional collision risks, and to compose this with the help of the collision risk tree into the collision risk value assessed for the simulation model.

4.4 Validation of the assessed risk level

For operations as complex as the active runway example considered, a simulation model will always differ from reality. Hence, validation of the MC simulation results does not mean that one should try to show that the model is perfect. Rather one should identify the differences between the simulation model and reality, and subsequently analyse what the effects of these differences are in terms of *bias* and *uncertainty* at the assessed risk level of the model. If the bias and uncertainty fall within acceptable bounds, then the assesed risk levels are valid for the specified application. Otherwise one should improve the MC simulation model on those aspects causing the largest *bias* and *uncertainty* influence on the assessed risk level.
Five types of differences between simulation model and the real operation can be distinguished (Everdij and Blom, 2002):

- Numerical approximations;
- Parameter values;
- Assumptions on the model structure;
- Non-covered hazards;
- Differences between the real operational concept and the operational concept modelled.

Thinking in terms of these differences makes it possible to consider the validation problem as a problem of making the differences specific, assessing each difference and its effect on the collision risk, and subsequently decide if this is accurate enough (valid) or not (invalid) for the purpose aimed at. The effects of differences on the collision risk can mathematically be expressed in terms of bias and uncertainty that has to be taken into account when using the simulation model assessed risk value for decisions about reality:

- *Bias*. The accident risk as defined by the simulation model is systematically higher or lower than it is for the real operation.
- *Uncertainty*. In addition to a systematic bias, the differences between simulation model and reality may induce uncertainty in the difference between the safety risk of the real operation and the safety risk resulting from the simulation model.

With this, the validation of a simulation based accident risk assessment has largely become a bias and uncertainty assessment process. Within TOPAZ, a bias and uncertainty assessment method has been developed which consists of the following steps:

- Identify all differences between the simulation model and reality;
- Assess how large these differences are, or how often they happen;
- Assess the sensitivity (or elasticity) of the risk outcome of the simulation model to changes in parameter values;

- Assess the effect of each difference on the risk outcome, using model sensitivity knowledge and complementary statistical and/or expert knowledge;
- Combine the joint effects of all differences in bias and uncertainty factors, and compensate the risk value of the model with these bias and uncertainty factors.

The result is an expected value of risk for the real operation, including a 95% confidence interval of other possible risk values. If the bias or the 95% confidence interval of the combined effects, or the bias and uncertainty of individual differences is too large, then these differences have to be taken into account in the decision making process regarding the acceptability and/or further design of the operation considered.

5 Monte Carlo Simulation Results

This section presents collision risk results obtained by Monte Carlo simulation with a computer implementation of the mathematical model of the active runway example of section 3. In order to relate these results to an actual operation, a bias and uncertainty assessment remains to be performed; however, this falls outside the scope of this paper.

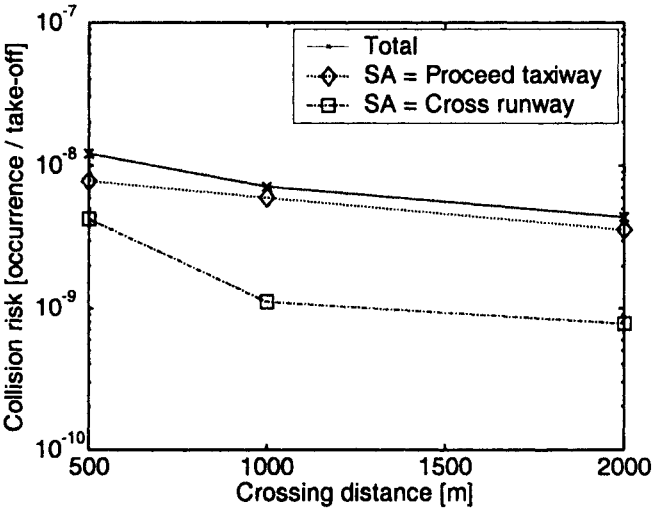


Figure 7: Contributions to the total collision risk by the simulation model for the cases that the SA of the PF of the taxiing aircraft is to proceed on a taxiway, or to cross the runway.

5.1 Assessed risk levels

Figure 7 shows the accident risk as function of the position of the runway crossing with respect to the runway threshold. The probability of a collision decreases for positions of the crossing distances further from the threshold. Figure 7 also shows the decomposition of the total risk for the cases that the pilot flying of the taxiing

aircraft either thinks to be proceeding on a normal taxiway (without being aware to be heading to a runway crossing) or where the pilot intends to cross the runway (without being aware that crossing is currently not allowed). The largest contribution to the risk is from the situation that the pilot thinks to be proceeding on a normal taxiway. The relative size of this contribution depends on the crossing distance and varies from 64% for crossing at 500 m to about 83% for crossing at 1000 or 2000 m.

A more complete overview of the contributions to the collision risk is provided by a projected version of the collision risk tree in Figure 8. It shows the contributions of events related to the situation awareness of the pilot of the taxiing aircraft (*Cross runway/Proceed runway*) and the functioning of ATC alert and communication systems (*Up/Down*). The collision risk results in the leaves of the tree are the product of the probability of the event combination indicated and the Monte Carlo simulation based collision risk given the event combination. The results in Figure 8 show that the risk is dominated by situations with a pilot flying of a taxiing aircraft having an erroneous situation awareness and the ATC alert and communication systems working nominally.

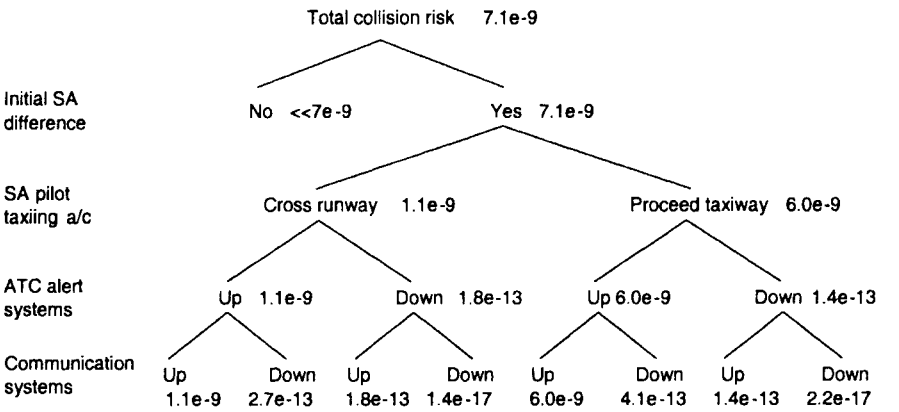


Figure 8: Projected version of the collision risk tree for the active runway crossing example, showing the contributions to the collision risk for various combinations of events related to pilot situation awareness and functioning of ATC alert and communication systems. The values are for a crossing distance of 1000 m.

5.2 Who contributes to safety risk reduction?

Based on results of the accident risk model, it is possible to attain insight in the accident risk reducing performance of involved human operators and technical systems. Table 2 shows conditional collision risks for the situation that an aircraft taxies towards a runway crossing at a distance of 1000 m from the runway threshold while the pilot has the situation awareness to taxi on a normal taxiway. The conditional collision risks in Table 2 refer to cases where the model either does ('yes') or does not ('no') involve the indicated human operators actively monitoring for traffic conflicts. A risk reduction percentage is determined by comparing the

conditional collision risk with the situation in which none of the human operators is actively monitoring. In this case, a collision is only avoided by the lucky circumstances that the taxiing aircraft just passes in front of or behind the taking-off aircraft (case 0 in Table 2). From the results in Table 2 a number of model-based insights into the operation can be attained:

- It follows from case 1 that 99.8% of the accidents can be prevented by the combined effort of all human operators and alert systems.
- It follows from a comparison of cases 1 and 5 that in the normal situation that all human operators are actively monitoring, ATC alert systems (runway incursion or stopbar violation) have a modest effect on the achieved risk.
- It follows from a comparison of cases 1 and 4, and cases 5 and 8, that the risk reduction that can be achieved by the tower controller in addition to the risk reduction of both pilots is very small.
- It follows from comparison of cases 1 and 3, and cases 5 and 7 that the pilot of the taxiing aircraft has the largest capability to prevent a collision in this context. Thus, resolution of the conflict is most likely to be by the human operator whose wrong situation awareness initiated the conflict.

Table 2: Risk reduction achieved in the simulation model by various combinations of involved human operators when the PF of a taxiing aircraft intends to proceed on a normal taxiway under good visibility (crossing is at 1000 m from runway threshold.)

0	no	no	no	$8.9\ 10^{-2}$	-
1	yes	yes	yes	$1.7\ 10^{-4}$	99.8%
2	yes	no	yes	$4.0\ 10^{-4}$	99.6%
3	no	yes	yes	$9.4\ 10^{-3}$	89.4%
4	yes	yes	no	$2.3\ 10^{-4}$	99.7%
5	yes	yes	yes	$2.2\ 10^{-4}$	99.8%
6	yes	no	yes	$1.7\ 10^{-3}$	98.1%
7	no	yes	yes	$1.1\ 10^{-2}$	87.9%
8	yes	yes	no	$2.3\ 10^{-4}$	99.7%

5.3 Comparison against expert based results

In the earlier conducted expert based safety risk assessment of the active runway crossing operation, it was concluded that both the pilots and the runway controller make large contributions to the prevention of a collision in the scenario aircraft erroneously crossing and other aircraft in take-off. In hindsight, it can be concluded that in the expert based safety risk assessment, the total effect of the pilots and the runway controller in preventing a collision turns out to be overestimated under good visibility condition. It is the simulation based approach that makes clear that although the runway controller identifies a good share of the conflicts, its

contribution to timely conflict resolution is relatively small. One significant part of the instruction issued by the runway controller appears to concern conflicts that are already solved by the pilots. And another significant part of the instructions issued by the runway controller appear to arrive too late for the pilots to successfully avoid a collision. Because of this, the effective contribution by the runway controller towards reducing collision risk is relatively small.

6 Concluding remarks

This paper has given an overview of performing safety risk assessment and providing feedback to the design of advanced air traffic operations with support of Monte Carlo simulation. The motivation for developing such a Monte Carlo simulation approach towards safety risk assessment was the identified need for modelling stochastic dynamic events and interactions between multiple agents (humans and systems) in advanced air traffic operations. The distributed and dynamical interactions pose even greater challenges than those seen in, for instance, nuclear and chemical industries (e.g. Labeau et al., 2000). The paper has explained the key issues to be mastered in performing a Monte Carlo simulation supported safety risk assessment of air traffic operations, and how this fits within a full safety risk assessment cycle. The steps to be followed in developing an appropriate Monte Carlo simulation model has been outlined, including a short overview of multi-agent situation awareness modelling, which plays a key role in the safe organization of cooperation between many pilots and controllers in air traffic. The paper also has explained the need for using stochastic analysis tools in order to develop the necessary speed-up of the Monte Carlo simulations, and has shown a feasible way to validate the simulation model versus the real operation. This assessment approach has been applied to an air traffic example involving aircraft departing from a runway that is occasionally crossed by taxiing aircraft. The results obtained demonstrate the feasibility and value of performing Monte Carlo simulation in accident risk assessment for safety relevant scenarios that are difficult to assess expert based, because of many interacting agents.

References

G.J. Bakker and H.A.P. Blom, (1993). 'Air Traffic Collision Risk Modeling, Proc. 32nd IEEE Conf. on Decision and Control, pp. 1464-1469

H.A.P. Blom G.J. Bakker, P.J.G. Blanker, J. Daams, M.H.C. Everdij and M.B. Klompstra (2001a). Accident risk assessment for advanced air traffic management. In: Donohue GL and Zellweger AG (eds.), Air Transport Systems Engineering, AIAA, pp. 463-480.

H.A.P. Blom, J. Daams and H.B. Nijhuis (2001b), Human cognition modelling in air traffic management safety assessment, Eds: G.L. Donohue and A.G. Zellweger, Air transport systems engineering, AIAA, pp. 481-511.

H.A.P. Blom, S.H. Stroeve, M.H.C. Everdij and M.N.J. van der Park (2003a), Human cognition performance model to evaluate safe spacing in air traffic, Human Factors and Aerospace Safety, Vol. 3, pp. 59-82

H.A.P. Blom, M.B. Klompstra and G.J. Bakker (2003b), Accident risk assessment of simultaneous converging instrument approaches, Air Traffic Control Quarterly, Vol. 11, pp. 123-155.

H.A.P. Blom, G.J. Bakker, M.H.C. Everdij and M.N.J. van der Park (2003c), *Collision risk modelling of air traffic*, Proc. European Control Conf. 2003 (ECC03), Cambridge, UK.

H.A.P. Blom and S.H. Stroeve (2004). Multi-agent situation awareness error evolution in air traffic. Proc. 7th Conference on Probabilistic Safety Assessment & Management, Berlin, Germany

H.H. De Jong (2004). Guidelines for the identification of hazards; How to make unimaginable hazards imaginable? National Aerospace Laboratory NLR, Contract report for EUROCONTROL, NLR-CR-2004-094

M.R. Endsley (1995). Towards a theory of situation awareness in dynamic systems. Human Factors, 37(1): 32-64

M.H.C. Everdij and H.A.P. Blom (2002), Bias and uncertainty in accident risk assessment. National Aerospace Laboratory NLR, NLR-TR-2002-137.

M.H.C. Everdij and H.A.P. Blom (2003). Petri-nets and hybrid-state Markov processes in a power-hierarchy of dependability models. In: Engel, Gueguen, Zaytoon (eds.), Analysis and design of hybrid systems, Elsevier, pp. 313-318

M.H.C. Everdij and H.A.P. Blom (2005), Piecewise deterministic Markov processes represented by dynamically coloured Petri nets, Stochastics Vol. 77, pp.1-29

P. Glasserman (2004), Monte Carlo methods in financial engineering, Springer.

D.A. Hsu (1981). 'The evaluation of aircraft collision probabilities at intersecting air routes', J. of Navigation, Vol.34, pp.78-102

ICAO (1988). Review of the General Concept of Separation Panel, 6th meeting, Doc 9536, Volume 1, ICAO, Montreal.

T. Kletz (1999), Hazop and Hazan; identifying and assessing process industry hazards, The Institution of Chemical Engineers, 4th ed.

J. Krystul and H.A.P. Blom (2004). Monte Carlo simulations of rare events in hybrid systems. Hybridge report D8.3, http://hosted.nlr.nl/public/hosted-sites/hybridge/

H. Kumamoto and E.J. Henley (1996), Probabilistic Risk Assessment and management for engineers and scientists, IEEE Press.

P.E. Labeau, C. Smidts and S. Swaminathan (2000), Dynamic reliability: towards an in-tegrated platform for probabilistic risk assessment, J. Reliability Engineering and System Safety, Vol. 68, pp. 219-254

O. Sträter, V. Dang, B. Kaufer and A. Daniels (2004). On the way to assess errors of commission. Reliability Engineering and System Safety 83:129-138

S.H. Stroeve, H.A.P. Blom and M.N.J. Van der Park (2003). Multi-agent situation awareness error evolution in accident risk modelling. 5th USA/Europe ATM R&D Seminar, Budapest

So how do you make a full ALARP justification? Introducing the Accident Tetrahedron as a guide for Approaching Completeness.

Richard Maguire, B.Eng, M.Sc, C.Eng, MIMechE
SE Validation Limited
rlm@sevalidation.com

Abstract

One of the fundamentals of safety engineering is the need not merely to achieve safety, but to demonstrate its achievement in advance [Redmill and Anderson 2005]. There are now a significant number of standards and taxonomies to follow in order to create a demonstration medium for indicating the level of achieved safety, the residual risks and how both are to be managed through the life of the system.

However, it is most difficult to obtain easy to use advice and tools for being able to demonstrate completely that risks are as low as reasonably practicable (ALARP). It is accepted that each project's handling of ALARP has to be specific to the particular project risks involved, and so bespoke guidance is not the objective of this text. This paper seeks to introduce a simple tool, well utilised and accepted in another safety field, but developed from a specific use into having generic scope that will be of useful value to safety practitioners looking for dedicated ALARP advice. The tool is called the Accident Tetrahedron.

Keywords
ALARP, Exposure, Assets, Hazards, Accident Tetrahedron

1. Introduction

Contemporary risk management follows a maturing path to the establishment, acceptance and management of a level of risk that is deemed tolerable and as low as reasonably practicable (ALARP). The recent issue of military standards [MoD 2004] describes six processes for risk management; hazard identification, hazard analysis, risk estimation, risk and ALARP evaluation, risk reduction and risk acceptance. Whilst these are not the universal descriptions of the processes involved, the underlying principles are consistent with other procedures and handbooks, for example IEC 61508, JSP 454 and Mil Stan 882D.

Guidance on demonstrating ALARP is given in the UK by both military [MoD 2004] and civilian organisations [HSE 2003]. The MoD guidance on demonstrating ALARP suggests the following principles;

- Show that the sum of all risks from the system is in the broadly acceptable class.
- If that isn't possible, then show that the risks are not in the intolerable class.
- Identify risks that may be addressed by the application of good practice.
- Those risks not addressed by good practice need risk reduction techniques applied.
- Cost benefit analysis needs to be undertaken, with costs balanced against loss prevention.
- A grossly disproportionate factor of costs should be applied according to risk level.

The HSE guidance [HSE 2003] concentrates on the design phase, where it feels there is maximum potential for reducing risks. Their design guidance indicates the following principles;

- Carry out risk assessment and management in accordance with good design principles.
- Risks should be considered over the life of the facility and all affected groups considered.
- Use appropriate standards, codes, and good practices with any deviations justified.
- Identify practicable risk reduction measures and their implementation, unless the implementation is demonstrated as not *reasonably* practicable.

The HSE also makes some important statements of principle when considering ALARP [HSE 2001];

> "The zone between the unacceptable and broadly acceptable regions is called the tolerable region. Risks in that region are typical of the risks from activities that people are prepared to tolerate in order to secure benefits in the expectation that the nature and level of the risks are properly assessed and the results used properly to determine control measures; the residual risks are not unduly high and kept as low as reasonably practicable (the ALARP principle); and the risks are periodically reviewed to ensure that they still meet the ALARP criteria, for example, by ascertaining whether further or new controls need to be introduced to take into account changes over time, such as new knowledge about the risk or the availability of new techniques for reducing or eliminating risks."

The ALARP demonstration needs to be recorded in the safety documents produced, but the question remains; how can one ensure that the ALARP consideration is as complete as possible? Whilst fairly well accepted in the UK,

ALARP demonstration is seen by many international viewers as difficult to verify and there is the perception that claims that all practicable risk reduction has been done, may be made without an appropriate effort [Bibb 2005]. Recent developments in The Medical Device Risk Management Standard (ISO 14971) indicate that the consideration of ALARP may be deleted from the new 2nd edition due to the completeness question [Bibb 2005].

Discussions concerning the problem of demonstrating completeness in risk assessments, and therefore ALARP, have even reached parliament. Medical discussions have highlighted that a risk appraisal process that excludes what are held by some stakeholders to be important factors, may fail to secure the crucial property of stakeholder confidence. It follows from this that, by instilling a misleading impression of completeness, robustness or rigour, risk assessments based on such incomplete risk characterisation may leave regulators and business highly exposed to a subsequent backlash on the part of the excluded parties [GeneWatch 1999].

A solution to the ALARP completeness question is proposed in the rest of this paper, generalising a specific analogy that already exists in the fire and safety domain.

2. Review of the Fire Safety Triangle

For many years the concept of fire initiation was symbolised by the Triangle of Combustion, that represents fuel, heat, and oxygen. For a fire to propagate, a fourth element must be present – that of a sustained exothermic chemical reaction. Essentially all four elements must be present for fire to progress; fuel, heat, oxygen and a chemical chain reaction. This has led onto the definition of the Fire Tetrahedron in national fire brigades, where there are four elements considered being essential for a fire to develop. Removal of any one of these essential elements will result in the collapse of the propagation of the fire [Sutton 2004].

A fire begins via some external initiation source from a system. As the three initial components are brought together, molecular excitation increases. If the conditions of the system are sufficient, a self-sustaining chain reaction between the elements occurs. This will continue the propagation process and the resulting reaction will escalate without the need for the original source [Sutton 2004]. Once propagation has occurred, it will continue until;

- Sufficient fuel has been removed (or consumed).
- Sufficient oxygen has been removed (or consumed).
- The temperature has been sufficiently reduced.
- The chemical-system chain reaction has been broken.

These are the four methods by which all of the available fire extinguishers control fires from candles to forest-fires. A pictorial presentation of the fire tetrahedron has been developed from the fire triangle [Sutton 2004]. For ease of

two-dimensional representation, this has been done using four linked equilateral triangles, the diagram is presented in Figure 1.

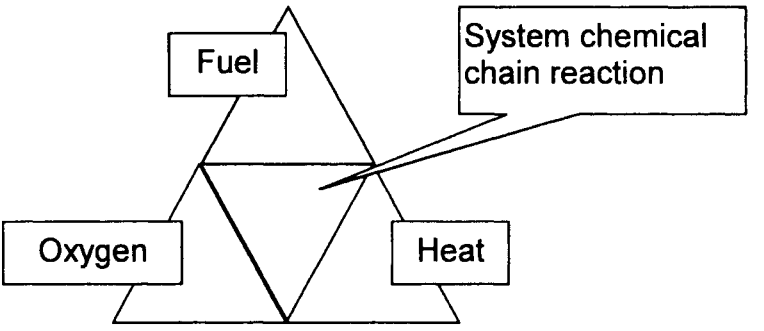


Figure 1: The Fire Tetrahedron

3. Introducing the Accident Tetrahedron

The first steps in being able to prevent accidents from propagating is to understand the combination of factors that can initiate them, and what causes them to escalate [Ontario 1999]. The generally accepted theories of accident causation, for example Heinrich's domino theory [Heinrich 1931] and Reason's organisational accident theory [Reason 1997] may use different terminology, but they do all have common themes;

- The immediate cause of an injury is not the same thing as the cause of the accident.
- Several causation factors usually combine to cause an accident.
- Accidents are unintended effects on persons or other objects of value.

Many texts cite accidents as occurring when humans suffer an exposure to hazards. It should be noted that the definition of accident might also include equipment, valuable assets, societal assets and the environment etc. This citation of exposure to hazards may be used as a generic approach to assessing accidents and their prevention. Transferring the ideas of the fire triangle and tetrahedron leads to a strong tool for considering accident initiation and propagation. More importantly, the tool provides a route to demonstrate ALARP arguments in a systematic way and can be used to provide a route to proposing a justification for a completeness argument.

Developing the parallels with the original fire tetrahedron further indicates that accident initiation should be prevented if;

- Potential exposure is sufficiently reduced in either the temporal or spatial frame of reference (so that vulnerable entities and hazards are prevented from sharing the same space and time, for example personal protective face equipment when operating a cutting machine).
- The number of people or value of assets used is sufficiently reduced (such that the consequences are below the threshold of a system's definition of accident, for example that the destruction of a remotely operated vehicle may be regarded differently to the destruction of a human operator).
- The severity of the hazard is sufficiently reduced (to a level that is tolerable and accepted, for example a lower strength or non-toxic material is used in some operation or system).

and also the accident propagation should be prevented if;

- The reaction chain within a system can be broken (deliberate and planned intervention as an accident sequence develops to halt or reduce the progress of the event).

Consider the new representation in Figure 2.

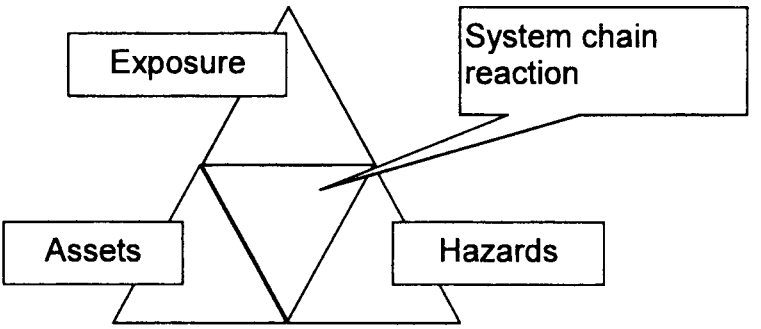


Figure 2: The Accident Tetrahedron

In the specific 'fire' case, the system chain reaction refers to the nature of the combustion being a sustaining exothermic chemical reaction (for example, rapid oxidation) that continues to provide heat into the fire system [Schmitt 1985]. It is the essential component for ignition to progress on to combustion and is not limited to the provision of heat. It may be the case that the chemical reaction also produces oxygen, or potentially more fuel, for example in a nuclear based reaction process. The combustion mechanism as a system is well researched and understood, so the system chain reaction can also be specifically targeted for fire prevention. In the more generic case of accident-systems as a whole and in other specific safety fields, this may not be the case. The concept of recovery analysis is already commonly used to highlight methodologies that may be used to prevent accident propagation by directly affecting the accident system's chain of reaction. For example, in the area of industrial pollution, the dispersal of an accidental spillage can be well understood

with flow directions, migration, and run-off knowledge. Containment practices are well established in the nuclear industry and also in the field of water science and technology, where recovery analysis methods have been studied with appropriate barriers introduced along release migration routes [Rossi and Ettala 1996].

4. Use in ALARP justifications

The fire tetrahedron indicates that if any of the elements is sufficiently reduced the propagation of the fire ceases and the full flame phase of the fire is prevented. One can envisage fire safety case reports using the fire tetrahedron as a basis, where for each operational mode of, for example, an off-shore fire management system, where the occurrence and significance of each tetrahedral element is judged to determine a level of risk priority. In turn each operational mode of a system is assessed to see if one or more of the four tetrahedral elements can be removed completely, or reduced to a level that would prevent fire propagation.

This is in-fact what happens in many health and safety fire assessments – the necessity and quantity of fuel stored is reviewed and reduced, rich sources of oxygen are heavily controlled, and sources of heat, sparks or naked flames are prohibited. Chemical chain reactions are more complicated to control, but protective atmospheres and drenching systems can be used, although they may also impact on the quantity or dilution of oxygen present as well. As is often the case, one prevention method can often have an influence on more than one of the essential elements.

Using the developed accident tetrahedron and similarly focussing on preventing each component from being of sufficient size or magnitude to allow initiation to progress, can lead to a series of assessments that can be used in ALARP justifications. For a complete ALARP justification, it will also be necessary to show that no further alternative mitigation is justified on effort/cost/disruption grounds. This tool and method allows the user to develop convincing evidence that all strands of possible mitigation have been considered.

As examples of use consider the following range of risks as shown in Table 1;

- The disposal of excavated explosives on military sites.
- The transport of nuclear material for re-processing.
- The cutting down of a large tree in a public park.

Assessing each example with ALARP arguments in mind and using the accident tetrahedron as a guide enables the systematic recording of the strategies that have been put in place to manage the risk in a reasonably practicable way. It is suggested that the prompts of the four components of the tetrahedron may be shown to be 'improving practice' for contributing to a more complete justification that a risk condition may be considered ALARP.

NOTE: The examples below are not meant to be complete ALARP assessments, but may be taken as indicative of the concept of use. The risk situations considered may be chosen as required and could even be different phases of the same procedure.

Risk situation	***Exposure reduction methods*** ***(Temporal and/or spatial factors between hazards and assets)***	***Hazard reduction methods*** ***(Quantity, nature or severity factors)***	***People / Asset reduction methods*** ***(Factors to reduce number or value of assets exposed to hazards)***
Explosives disposal	Disposal carried out using single detonation. Temporary safe refuges used as protection. Protective equipment used.	*Limited control here due to nature of possible explosive types.*	Task planning undertaken. Carried out by single highly trained person or even a remotely operated vehicle (ROV). Controlled entry to sites.
Nuclear transport	Transport at night preferred. Protective casing used on transporters.	Quantities below critical mass and critical exposure levels transported separately.	Carried out by as small a team as possible. Low population areas and routes used. Police escort used as warning.
Tree cutting	Cutting time is kept to a minimum. Protective equipment used.	Wood is cut away in small masses at a time rather than in one large mass.	Task planning undertaken. Cutting carried out by single, trained person. Local areas evacuated or park area closed to public.

Table 1 : Example ALARP arguments

Of course the fourth aspect to the tetrahedron needs to be considered – the chain of system reactions and sequence of responses to an accident once initiated and in progress. This aspect needs to deal with the accident once it is under way, to reduce the duration or severity, rather than to prevent it occurring in the first place. There are several strategies in place for this already in industry, perhaps under the name of 'crisis planning' or 'major accident planning'. For example;

- Wider area evacuation strategies.
- Trained response teams and first-aiders.
- Attendance of dedicated emergency personnel.
- Provision and placement of containment reservoirs.
- Provision of 'safe' areas.
- Communication and alarm procedures.

When using the Accident Tetrahedron as guidance, the system reaction reduction strategies also need to be demonstrated as having been considered and put in place, or argued that they are not reasonably practicable. For our three examples above, potential ALARP arguments for the system reaction aspects once an accident has started could be as shown in Table 2. Again these should not be taken as totally complete – you can probably think of more, the accident tetrahedron gives focus and structure to the thought process. Note: For a more usable presentation these tables may be combined – they are shown separately here for discussion and construction purposes only.

Risk situation	***System Reaction Methods (Quantity, nature or severity factors)***
Explosives disposal	Water deluge system; provision of safe areas for non-participants; decontamination system on-site; medical facilities; evacuation plan; event rehearsal.
Nuclear transport	Police / Army escort including vehicle mechanics; decontamination equipment carried in separate vehicle; spare transport vehicles along route; medics in convoy; communication equipment carried; event rehearsal.
Tree cutting	First aid kit carried; Emergency shut off available; working in pairs; event rehearsal.

Table 2 : Example ALARP arguments

5. Consistency with contemporary techniques

Whilst the Accident Tetrahedron method might be seen as yet another technique to be added to the tool set of the safety practitioner, it is suggested that the procedure actually gives some order to the wealth of tools already in use. An exercise in contrast-and-compare is useful to demonstrate the positioning of the proposed tool.

There does not appear to be any over-arching techniques for identifying and considering all potential mitigation strategies for use in ALARP justifications. There are certainly many individual techniques for identifying and/or assessing hazards – HAZOP, FMEA, SWIFT, various taxonomy checklists [MoD 2002] and other guidance [Maguire 2005]. There are some wider processes in dedicated industry areas e.g Control Of Major Accident Hazards (COMAH) in the UK and Health And Safety Planning (HASP and e-HASP) in the US. These existing techniques focus at

a particular level of analysis, specific to fairly narrow fields, although there isn't anything particularly preventing their use elsewhere. Usually they have a driving concern for the reduction of the risk itself with a focus on a human injury impact component. For their purposes, they are perfectly acceptable. There appears to be no generic safety assessment methodology that brings in system response, time at risk, valuable assets and the environment used in a particular system, whatever the hazardous field of engineering. As such the analysis effort of these diverse areas does not necessarily equate in depth or strength to that for risk assessment on its own. By way of a contrast, the Accident Tetrahedron accepts that all four components are equally essential for accident propagation and so all are worthy of equal attention.

UK HSE guidance [HSE, 2004] gives some indication of procedure to demonstrate ALARP, but does not give specific guidance on 'determining what additional risk reduction measures, beyond relevant good practice standards, may be implemented'. However, the guidance does state that 'Arguments based upon `strength in depth` concepts such as Layers of Protection (LOPA) or `Lines of Defence` may be used when these have been developed sufficiently', but then doesn't go on to define 'sufficiently', citing that research is under way. This paper suggests that the four components of the Accident Tetrahedron may be considered as a new concept of separate lines of defence or layers of protection, particularly with respect to the analysis of system response reactions.

A closer look at some of the wider safety analysis techniques indicates that they do have some relationship with exposure and value of assets. The concept of risk analysis matrices [MoD 2004] explicitly includes impact analysis and frequency of exposure, and uses these to determine the criticality of the risk on a hazard by hazard basis. It does not, nor does it claim to, consider the role of on-going system reaction and response. In comparison, the Accident Tetrahedron explicitly includes these factors.

These discussions are not meant as criticisms of current practice, just comments on the current state of the art. The Accident Tetrahedron does not contradict any of the currently accepted practices; indeed it does appear to utilise, organise and be consistent with many of the individual techniques in use. The Tetrahedron can be used to give a framework to risk analysis and a systematic order to the organisation of safety arguments. As such, the author can recommend its use in the field of safety analysis without significantly changing accepted practice.

6. Discussion on the use of costs in ALARP

For a complete ALARP assessment some notion of cost also has to be considered. In its simplest form an analysis uses only financial measures, having to allocate representative financial values on more intangible costs and benefits brings about increasing sophistication. When assessing ALARP arguments, there are two areas of benefits where a value should be required; avoidance of the negative and promotion of the positive. The third area for a complete cost/benefit analysis is the cost of implementing the actions that gain the benefits.

The cost of developing and implementing a solution is wholly dependent on the nature of the required tasks from re-design and re-assessment to re-work and new procedures. These costs will be different for each part of the Accident Tetrahedron. However, here the new tool can be of further help by providing a usable framework for organising the construct of accident prevention costs into the four tetrahedral aspects. The implementation of a solution should demonstrably consider the costs for all four areas. Specifically, modifying the chain of corporate system reactions may be more difficult and expensive in effort than in direct costs, where as affecting the spatial separation between a worker and a hazard, would probably have reasonably high direct costs and require only a little effort. The elicitation of actual costs in the four areas is not a research area for this paper and further study on measuring the costs of system response modification is still required.

7. Formulation of the Completeness Strategy

Through this paper a specific fire analysis tool has been *reverse-engineered* to have a more general capability, with utility across much of the field of risk evaluation and management. Two developments have identified a method for organising many of the existing practices in hazard and safety management; and indicated a basis for a claim of approaching completeness in risk ALARP arguments. These two developments can be formulated into a definite strategy for claiming completeness of risk ALARP arguments.

- Part One: The accident situation and system have to be analysed for safety factors using the four aspects defined in the accident tetrahedron – exposure, assets, hazards and system response to the initiation of an accident. Evidence may be presented in tabular form or referenced to specific reports. The assessment and review methods should be recorded (including duration and participants), as it is likely that the evidence compiled in each aspect, may not be of equal volume. Every accident situation should have contributions in each area, any absence should be viewed as falling short of completeness.
- Part Two: The claim for completeness of assessment and risks being ALARP can be made providing it can be demonstrated through evidence that all four areas of the accident tetrahedron have been equally assessed to a degree appropriate to the credible accident scenarios developed. The greater the risk of the credible accident, the more evidence in all four areas is required. For additional probity of the claim, independent assessment may be taken. It must also be shown that further reduction in risk is not reasonably practicable.

The resulting completeness argument must always be subject to regular reviews. It is likely that new technological developments will offer safer working practices, for example an increase in the utility of remotely operated vehicles and cutting equipment, or better body protection armour. It may just be the case that existing, expensive (and therefore not reasonably practicable) protection methods become cheaper or more easily available.

8. Summary

A discussion on a new approach and tool for ALARP justifications has been produced. The approach has been derived by taking an accepted and specific method already in use for fire prevention, and enlarging its area of use to the more generic field of accidents to humans, equipment and other valuable assets. The tool represents an easy-to-use and systematic method for guiding the demonstration of ALARP. Some brief examples of use have been utilised, and a new strategy for claiming ALARP completeness has been developed.

It is suggested that the four components of the Accident Tetrahedron may be considered as a new concept of separate lines of defence or layers of protection when considering ALARP justifications, all of which must be considered during risk assessments. Further research is recommended on the costs of mitigation methods across the four areas of the tetrahedron. A specific retrospective application would also have benefit.

The author is aware of at least one equipment safety case intending to use this method for ALARP justifications in the future, the intent has already been independently endorsed.

References

Bibb (2005). The medical device risk management standard – an update. The Safety-Critical Systems Club Newsletter, Vol. 14 No. 3, May 2005.

GeneWatch (1999). Appendix 32 to the Minutes of Evidence II, Scientific Advisory System : Genetically Modified Foods. Select Committee on Science and Technology, House of Commons, May 1999.

Heinrich (1931). Heinrich HW, Peterson D & Roos N, Industrial Accident Prevention, 5th Edition, Mcgraw Hill, New York, 1980. ISBN 0-07028-061-4.

HSE (2001). Reducing Risks, Protecting People – HSE's Decision Making Process. HMSO Norwich, 2001. ISBN 0-7176-2151-0.

HSE (2003). Policy and Guidance on Reducing Risks As Low As Reasonable Practicable in Design, The Health and Safety Executive, June 2003. (http://www.hse.gov.uk/risk/theory/alarp3.htm)

HSE (2004). Guidance on 'as low as reasonably practicable' (ALARP) decisions in Control Of Major Accident Hazards (COMAH), The Health and Safety Executive, 2004. (http://www.hse.gov.uk/comah/circular/perm12.htm)

Maguire (2005). Introduction to Safety, SE Validation, 2005. ISBN 0-95501-070-5.

MoD (2002). An Introduction to System Safety Management and Assurance, LSSO, MoD, 2002. (http://www.ams.mod.uk/ams/content/docs/syssafmn/intro.pdf)

MoD (2004). Safety Management Requirements for Defence Systems Part 1, Interim Defence Standard 00:56 Issue 3, MoD, December 2004.

Ontario (1999). Ontario's Basic Certification Training Program – Participants Manual, Ontario Workplace Safety and Insurance Board, 1999.

Reason (1997). Managing the Risks of Organizational Accidents, Ashgate, 1997. ISBN 1-84014-105-0.

Redmill & Anderson (2005). Preface to Constituents of Modern System-Safety Thinking, Proceedings of the 13th Safety-Critical Systems Symposium, February 2005, Springer-Verlag, London. ISBN 1-85233-952-7.

Rossi & Ettala (1996). Recovery Analysis in Risk Management of Hazardous Materials, Water Science and Technology Vol 33 No 2, IWA Publishing, 1996.

Schmitt (1985). Pyrophoric Materials Handbook : Flammable Metals and Materials, C.R. Schmitt, Editied by J. Schmitt, Saber-Towson. 1996 (http://saber.towson.edu/~schmitt/pyro/)

Sutton (2004). Notes for Guidance, The Fire Safety Advice Centre, Merseyside Fire Liaison Panel, 2004.

EXPERIENCE OF DEVELOPING SAFETY CASES

Safety Case Practice - Meet the Challenge

Werner Winkelbauer, Gabriele Schedl and Andreas Gerstinger
FREQUENTIS GmbH,
Vienna, Austria

Abstract

FREQUENTIS is a producer of voice and data communication systems in many different areas like air traffic control (ATC), public transport, maritime and public safety. Our customers are spread around the world. Most of our products are used for safety related or critical tasks. Therefore our customers often demand or need a safety case, partly to be allowed by the authorities to "go live" with their system.

This is a particular challenge for our safety work, as different standards in varying depth of compliance are to be fulfilled. Due to limited budgets caused by a rather hard competition on the market, it is necessary to perform the work as effective as possible.

In this paper we want to show, how we perform safety programmes for our voice communication systems, how we try to reduce our efforts with maximum output in spite of the difficulties of different business areas with different standards (which occasionally change quite substantially over time). This includes the presentation of our safety management system, hazard log, internal trainings, safety analyses and the production of safety cases.

1 History

The history of evolution of safety engineering at FREQUENTIS – a telecommunications company with approximately 500 employees - started with reliability calculations (reliability block diagrams – RBDs – put into Reliability, Availability and Maintainability Modelling and Prediction Reports – RAM MPRs), which are strongly safety related for voice communication systems in air traffic control, and hardware Failure Mode, Effects and Criticality Analyses (FMECA). This evolution was, and still is, heavily driven by more and more stringent customer requirements due to the increasing complexity of the built systems and the rising awareness of the public and the authorities regarding safety.

The following steps were the creation of an internal, companywide hazard log, the unification of software quality management with safety engineering and the integration of independent verification and validation, test tool development,

hardware compliance and the process ownership for the development process. This way all safety related engineering disciplines are combined in an independent department with the reporting line directly to the upper management (see Figure 1).

A major problem, due to different business areas and customers around the world (with dissimilar national laws), are considerably varying requirements and expectations of safety engineering which is partly reflected in the big number of standards which had to be taken into account in time:

DoD 2167, MIL-Std 498, IEEE 12207, Mil-Std 882c, IEC1508, IEC 61508, Def Stan 00-55, Def Stan 00-56, CENELEC 50126, 50128, 50129,...

Those challenges lead to continuous improvement of the development and the safety processes and to the development of various supporting tools.

Now the department has experience with complete safety programs for projects comprising a full safety assessment process concluded with a safety case report.

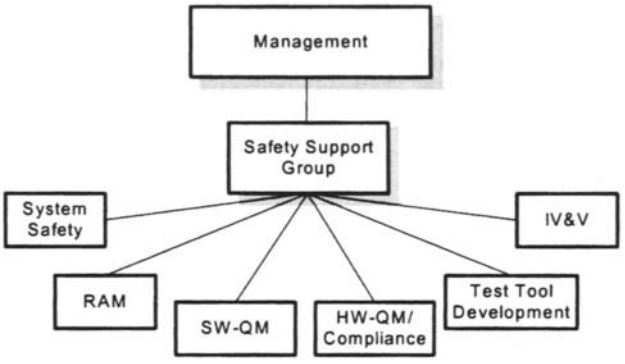


Figure 1. Safety Support Group Organization.

2 Safety Management System

To be able to perform all necessary tasks within the company, a comprehensive safety management system was introduced. Its main components are briefly described in the following paragraphs.

2.1 Safety Policy

As it is essential, that the importance of safety is understood both by the very top of the company and by all other employees, an internal safety policy, valid for every single person in the company, was written down, including a statement from the CEO about the importance of safety and his commitment to the implementation of safety:

*"**Achievement of satisfactory safety in Voice Communication has the highest priority over commercial, environmental, operational or social pressures.***
I am committed to implementing a policy of safety awareness and to ensuring adequate resources to maintain and improve safety.
Suggestions on safety related subjects to the management by anyone involved in the projects are actively encouraged. Safety is the responsibility of all of us, all the time at everything we do.
Safety has to be improved continuously, in co-operation with customers, suppliers and authorities.
Hazard prevention is to be performed by means of hazard logging, maintaining effective communications to customers and suppliers and the staff's awareness of risks and safety responsibilities."

Thus the safety department has the full support to drive changes and improvements in the whole company which are necessary for success.

2.2 Trainings

The implementation of the safety policy is supported by several internal trainings which can be accessed by every employee, comprising system safety, software quality management with focus on software safety, reliability engineering and a special training programme called the Safety Certificate.

The Safety Certificate is an extensive training programme consisting of several mandatory and optional modules like "Foundation to System Safety", "Hazard Identification and Management", "Software Safety" and "Safety Case", an examination and an upgrade module to renew the validity of the certificate after two years.

It is intended that only those employees who hold a certificate are allowed to work for safety critical projects.

To gain the necessary knowledge themselves the safety specialists participate in external trainings (in the USA and Great Britain) and safety conferences and are members of relevant societies (e.g. the International System Safety Society and the Safety-Critical Systems Club) as there are not many possibilities for a complete safety education in central Europe yet. An additional goal of these activities is the early recognition of new or revised legal requirements.

2.3 Hazard Log

The main goal of the internal companywide hazard log is to act well in advance instead of reacting to problems, which is both a safety benefit and a commercial one, as we all know about the cost explosion of problem solving over lifecycle time.

The hazard log is a database containing all our systems at customer sites and all known hazards with respective data. After contract award, new projects are entered into this database. When a new hazard arises, information is gathered by the safety

department and passed on to all departments which could possibly be influenced. Then the respective development department is instructed to solve the problem. Every hazard, once defined, stays in the hazard log, even if it is closed company-wide, as well as a project remains in it over its whole lifecycle.

Hazards are assigned to all projects or systems where they might possibly contribute to accidents. As soon as a new project is acquired, all known hazards of the corresponding product family are checked for applicability. All open hazards of the same product are automatically assigned. It is then the task of the project manager either to show that this hazard is not applicable or to implement the solution when available.

2.4 Hazard Checklist

Out of the hazard log a checklist was created, asking for the root causes of these hazards. The questions are assigned to different roles in a project where development is performed. The respective employees get their questions on a sheet and have to answer these questions and return the filled in and signed sheets before the system integration phase begins.

This serves as an aid to prevent repetition of the same hazardous errors by different people as the developers get the information and have to think about the specific problems.

2.5 Failure Reporting System

It is under the responsibility of all employees to pass on all necessary information which could affect safety in any respect to their managers and to the safety department. Additionally information is controlled via the web tool ERRSYS, a company-wide error-tracking tool for all kinds of errors and incidents with an incorporated workflow for the management of these errors. This tool is mandatory to be used beginning at the latest with system integration.

The ERRSYS database is scanned for hazardous entries by safety personnel. All this information then serves as input for the hazard log.

2.6 Safety Working Group

The Safety Working Group regularly performs meetings with participants from different development teams, the quality management department, the manufacturing team and the safety department. Information is passed on to the management board, the project management department, the maintenance department and all heads of the various development teams.

The objectives are to enable company wide hazard processing, to pass on and discuss information, to identify risks at development projects as early as possible and to have an information board for results of monitoring activities and subcontractor evaluations.

One of the main goals, apart from hazard processing, is the establishment of an information network with decision making competence.

2.7 Safety Monitoring

The latest addition to the process is Safety Monitoring for development projects. The objective is to assure compliance to the agreed processes, traceability, performance of safety reviews, achievement of project milestones in time and that way to reduce the overall risk. All findings are reported to the head of the development department to give him a quick overview of all development projects in work.

Safety Monitoring is implemented mainly with the help of several meetings where the necessary development process level is determined, the implementation of the planned tasks is supervised and finally the lessons learned are discussed.

2.8 Product Release Process

A comprehensive product release process ensures that products are very mature when released. Parallel to the comprehensive quality management process the safety process starts with general safety requirements which are checked for applicability and allocated to the project respectively. It continues with several tasks like performance of an Functional Hazard Assessment, production of an hardware RAM Modelling and Prediction Report and a Failure Modes, Effects and Criticality Analysis for a typical configuration and the use of the previously mentioned hazard checklist. Finally all issues of the product release checklist are to be fulfilled to get the official release.

3 Safety Analysis Techniques

To be able to analyse a system, you have to create a suitable model. It is possible to view a system from multiple points of view. The main types of models are the physical model, which details what the system is physically made of (e.g. platform, sub-system 1, sub-system 2... controls, sensors, actuators, mechanics) and the functional model which explains, what the system does (function 1, function 2, ... , function n; sub-function 1.1, 1.2, ...).

Usually the physical model can be mapped to the functional one (which functions are performed in which hardware components). Safety always has to consider both the physical and the functional view.

Figure 2 gives an overview of some analysis techniques used in the context of safety engineering which are briefly detailed in the following.

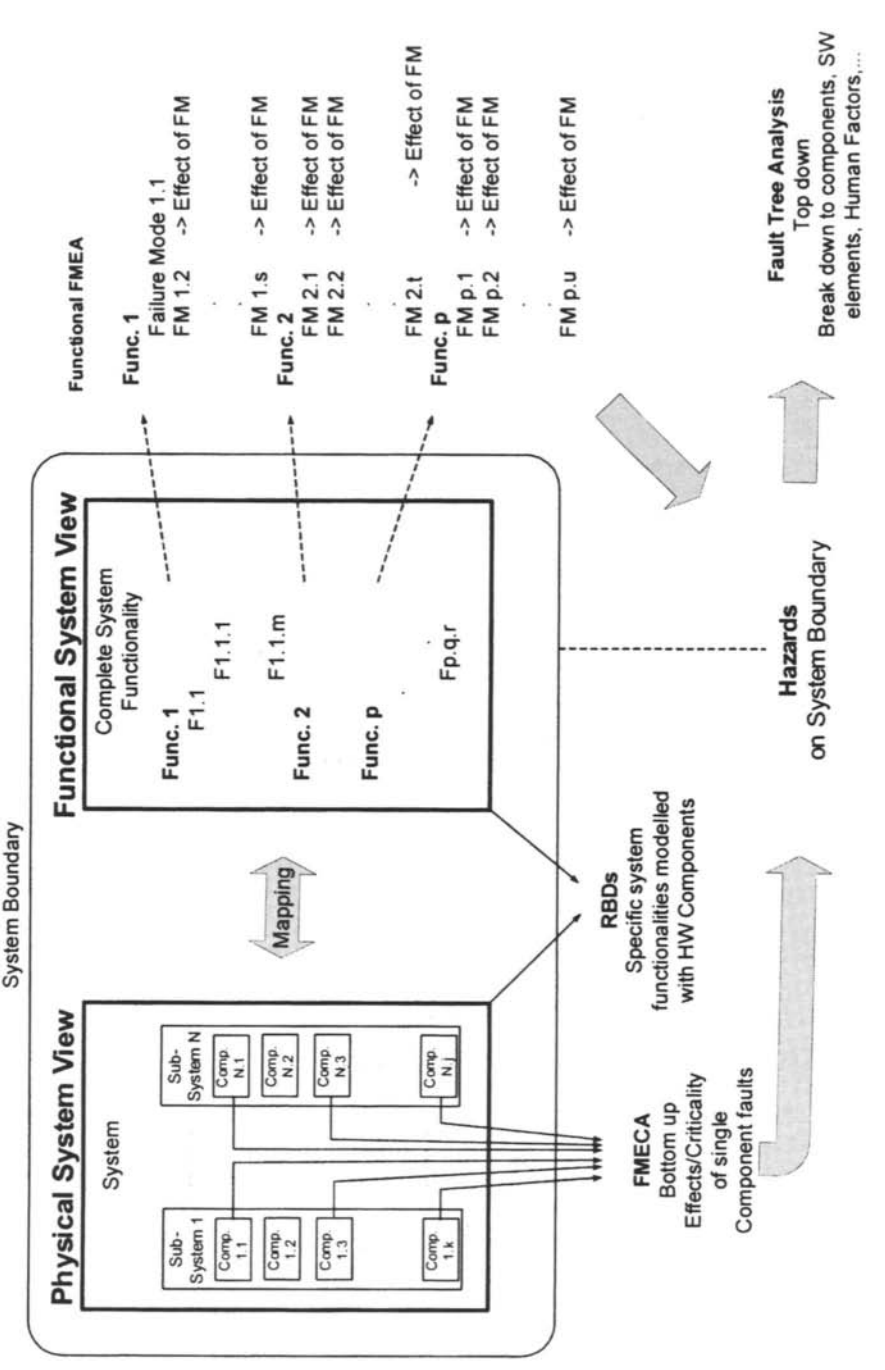


Figure 2. Safety Techniques Overview.

3.1 Failure Modes, Effects and Criticality Analysis

The hardware Failure Modes, Effects and Criticality Analysis (FMECA) is an inductive, bottom up analysis technique to study the effects of single failures of physical components on system operation and classifies potential failures according to their severity and probability of occurrence. Results are presented in tabular form. This way you can identify single points of failure and additionally provide early criteria for maintenance planning analysis, Logistic Support Analysis (LSA), test planning etc. and identify maintainability design features requiring corrective action.

3.2 Reliability Block Diagrams

With a reliability block diagram (RBD) the physical configuration of a functional operation is described. It models "what is necessary for success" for defined functions and gives reliability, availability and maintainability (RAM) figures as a result. RBDs can be used for design decisions (which design/configuration will reach the required RAM targets) or verification (does the system reach the required RAM targets) and for logistic support calculations (for repairable systems).

3.3 Functional Failure Modes and Effects Analysis

The Functional Failure Modes and Effects Analysis (Functional FMEA) or Functional Failure Analysis (as it is usually called in the USA) is typically performed to support safety analysis efforts early in the lifecycle and intended for iterative application. It tries to find all hypothetical failure modes to the defined functions of the considered system and assesses the operational effect thereof. That way functional failure modes that may be eliminated/mitigated by functional level design changes can be found, the confidence in the overall design concept is strengthened and areas requiring risk reduction can be identified.

3.4 Fault Tree Analysis

Fault Tree Analysis (FTA) is a well known and widely used safety tool, implementing a deductive, top down approach. It starts with a top level hazard, which has to be known in advance and "works the way down" through all causal factors of this hazard, combined with Boolean Logic (mainly AND and OR gates). It can consider hardware, software and human errors and identifies both single and multiple points of failure. Both a quantitative and qualitative analysis is possible.

4 Safety Process

It is a well-known fact that all kinds of traffic flow (air, rail, road, sea) increased rapidly in the past decades and are still growing. This caused also a noticeable – sometimes strong – increase in the amount of requirements on traffic control systems, including voice communication systems, which are now often realised by very complex systems.

As a result these systems now include more and more networks of highly sophisticated hardware and especially software preventing all outcomes of environmental, technical or operational influences to be considered by one single person. Therefore an organised, structured safety process has become necessary which is also reflected by many customer requirements concerning compliance to respective safety standards.

Due to the fact that FREQUENTIS has customers around the world, many different standards have to be followed. This lead to the definition of a generic safety process that comprises the basic principles of all those standards as most of them anyway differ more in wording than in content. This process is then tailored for specific projects and customers.

The following standards were considered:

- The European SAF.ET1.ST01.1000-POL-01-00: EATMP SAFETY POLICY
- The European SAF.ET1.ST03.1000-MAN-01-00: AIR NAVIGATION SYSTEM SAFETY ASSESSMENT METHODOLOGY
- The international standard IEC 61508: Functional Safety of electrical/electronic/programmable electronic safety-related systems
- The USA's MIL-STD 882c,d: System Safety Program Requirements
- The British Def-Stan 00-56: Safety Management Requirements for Defence Systems
- The British Def-Stan 00-55: Requirements for Safety Related Software in Defence Equipment
- The European (CENELEC) EN 50126: Railway applications: The specification and demonstration of Reliability, Availability, Maintainability and Safety (RAMS)
- The European (CENELEC) EN 50128: Railway applications – Communications, signalling and processing systems – Software for railway control and protection systems
- The European (CENELEC) EN 50129: Railway Applications: Safety related electronic systems for signalling

4.1 Generic Safety Process

Figure 3 gives an overview of the generic safety process, linked to the respective project phases. Time progress is from the left to the right (in the direction of the arrows). The figure details the following:

- Main Objectives of Safety Process Phase: What is intended to be reached in that phase

- Tools, Techniques: Which tools and techniques are mainly used during that phase
- Inputs: Which inputs are necessary to perform the tasks of this phase
- Outputs: What is the general outcome
- Reports, Document: What is the outcome in the form of documentation

4.2 Safety Process Phases

In the following the safety process phases, as shown and linked to the project phases in Figure 3, are briefly detailed.

4.2.1 Planning Phase

In the planning phase the customer requirements have to be assessed and the respective process and resources planning is performed and detailed in a System Safety Plan.

4.2.2 Preliminary Hazard Identification

The safety core-process itself starts with the Preliminary Hazard Identification (PHI) and the Preliminary Hazard Assessment (PHA). During that phase a preliminary hazard list with severities is created via brainstorming and the use of historical data and checklists. Outputs are the preliminary hazard list, including severities and hazard target rates, and initial development process integrity level allocations as detailed in various standards, e.g.: Safety Integrity Level (SIL) in IEC 61508 or CENELEC EN 50128.

4.2.3 Functional Hazard Assessment

The Functional Hazard Assessment (FHA) asks the question: "How safe does the system need to be?" considering the required functionality and the specific environmental context of the system. A typically used technique in that phase is the Functional Failure Modes and Effects Analysis (Functional FMEA) to find all theoretically possible failure modes which then can be traced to hazards.

The preliminary hazard list is revised and safety requirements are derived.

4.2.4 Preliminary System Safety Assessment

The Preliminary System Safety Assessment (PSSA) asks the question: "Does the proposed design reach the safety objectives?"

The causes of hazards and functional failures are broken down, e.g. via Fault Tree Analysis (FTA). Other typical techniques are the Failure Modes, Effects and Criticality Analysis (FMECA) and the production of a Reliability Availability Maintainability Modelling and Prediction Report (RAM MPR), containing reliability block diagrams of the system.

This can lead to further requirements, e.g. that additional redundancy is necessary to meet the hazard target rates.

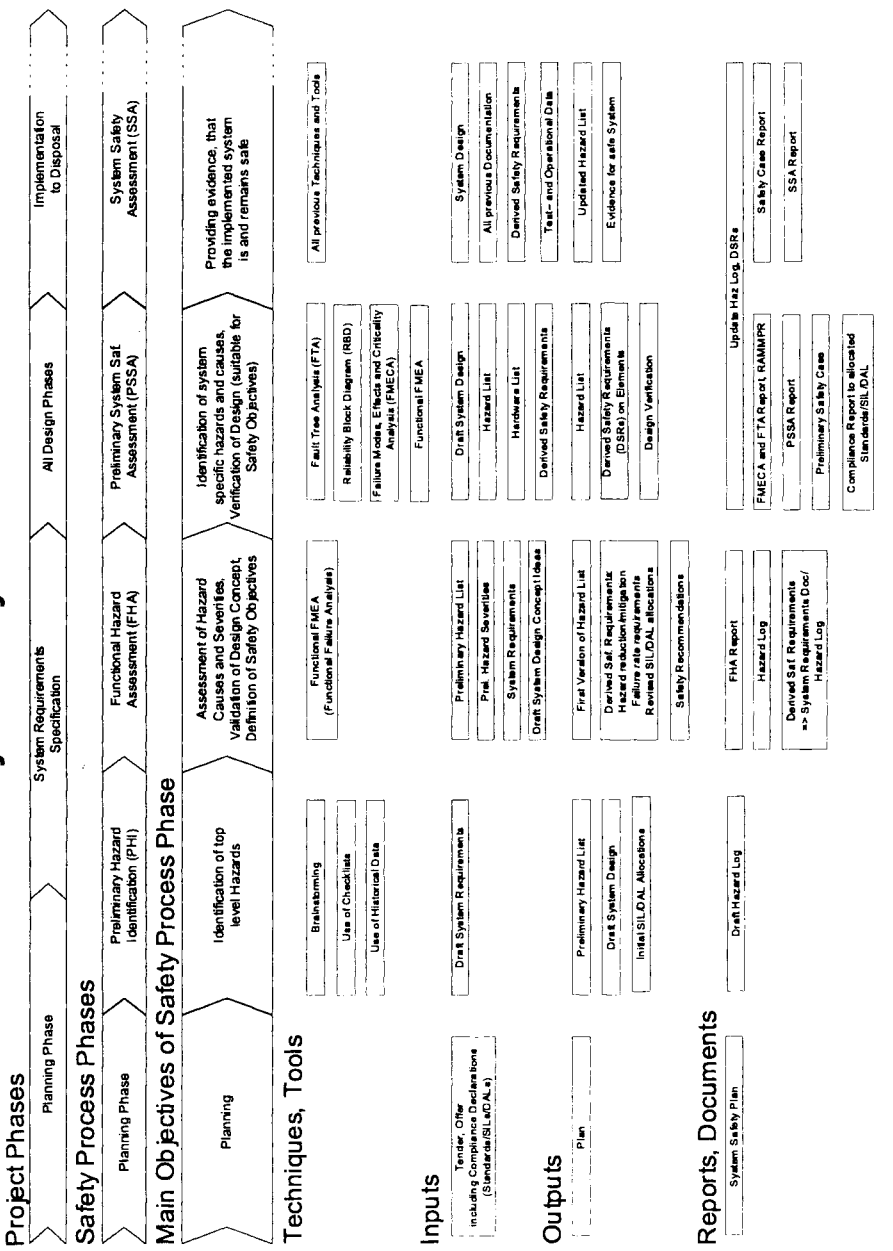


Figure 3. Safety Process.

4.2.5 System Safety Assessment

The System Safety Assessment (SSA) asks the question: "Does the system as implemented achieve tolerable risk?"
All previously performed analyses are updated with the latest available data and all safety targets and safety requirements have to be verified, whether they are met. Finally a Safety Case Report is produced. This report is a living document, which has to be kept up-to-date during the whole life-cycle of the system, especially when there are changes at the system or the environment.

5 Safety Case

5.1 Purpose of a Safety Case

The Safety Case provides a justification that the considered system or equipment is safe to be used or deployed in a specific operational environment. One example of a definition is taken from the British Defence Standard JSP 430:
"A safety case is a comprehensive and structured set of safety documentation which is aimed to ensure that the safety of a specific vessel or equipment can be demonstrated by reference to:
Safety arrangements and organisation
Safety analyses
Compliance with the standards and best practice
Acceptance tests
Audits
Inspections
Feedback
Provision made for safe use including emergency arrangements"

It is often necessary to produce a safety case before the authorities allow a system or installation to be operated. It is to note, though, that despite the fact that the certifiers assess and approve the safety case, the liability usually remains at the developers and operators – the certification does not transfer it.

5.2 Contents of a Safety Case

The exact contents depend on the specific regulatory environment, but the following chapters are key elements of most standards:

5.2.1 Scope
The chapter "Scope" details the principal objectives of the safety case, the key requirements and standards, possible relationships to other safety cases, e.g. software and system safety case and high level assumptions and limits.

5.2.2 System Description
The system description gives an overview of the system sufficient to understand the principal objectives. Additional system information is provided at the points, where it is required.

5.2.3 System Hazards
An excerpt of the current hazard log and possibly a presentation of the "Key Hazards", including a short description for each hazard and a reference forward to where each hazard is addressed are given in the chapter "System Hazards".

5.2.4 Safety Requirements
This chapter details all standards which have to be addressed, failure rate targets, allocated safety integrity levels, a description of requirements handed over from other safety cases and emerging from the hazard analysis and a reference forward to where each requirement is addressed. It also includes operational safety requirements.

5.2.5 Risk Assessment
This gives a description of how the risk associated with identified hazards was determined, a summary of the system risks in each risk category, an explanation of each SIL allocated to system/software elements and a summary of the procedures that were applied for each risk category.

5.2.6 Hazard Control / Risk Reduction Measures
For each identified hazard, the measures that were taken to control/reduce/mitigate the risk and a reference forward to the safety analysis or to tests as evidence of the sufficiency of measures taken are given.

5.2.7 Safety Analysis / Test
Here a description of the key techniques involved, a reference to procedures and test schedules etc., a summary of results obtained and how they relate to the goals made in the previous section (e.g. calculated reliability figures, reports of "no anomalies" etc.), a top level fault tree and a representative FMEA table are detailed.

5.2.8 Safety Management System
This chapter shows the allocation of roles and responsibilities, education and experience of development and/or operational staff and an overview of activities described in the Safety Plan (e.g. Reviews, Audit Procedures ...).

5.2.9 Development Process Justification
An overview of the system development process – describing key phases, tools, techniques, programming languages used – and a justification of appropriateness for the allocated SILs and risk classes (e.g. references to tool qualifications, compliance with recommendations in the standards) etc. are given.

5.2.10 Conclusion

The conclusion shall contain a clear and concise statement of the principle reasons why the system is acceptably safe, expressed at the top level. The safety case should be written to convince the reader!

5.3 Argumentation Structure

The basic safety argument is given by showing that all risks are controlled and reduced. But we also need to consider how thorough the analysis was, whether all the hazards were identified, if the assumptions about the operational environment are credible, whether other design strategies have been overlooked, if the role of the human in operation and maintenance was considered and so on. This is especially important at early stages of the development.

Normally two elements are distinguished: High level arguments (HLA), which represent the principles on which safety is based on and supporting evidence (SE). High level arguments are e.g. redundancy strategy, maintenance policy, design for human error tolerance and identification of safety requirements. Supporting evidence are e.g. results of relevant assessments, system safety analysis, system tests, pre-operational testing and the verification that requirements are met. High level arguments give the context to make supporting evidence relevant.

The basic structure of how to argue, that a system is safe, looks as detailed in Figure 4, using the Goal Structure Notation. The Goal Structure Notation is a graphical representation to give a clear picture of the argumentation principles.

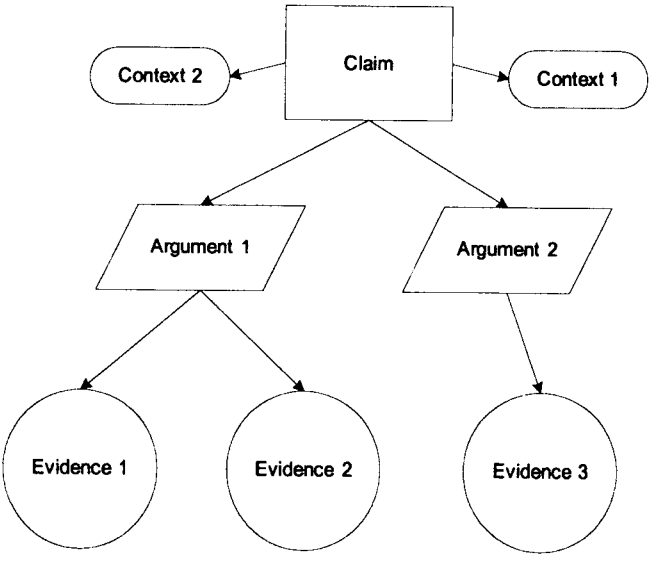


Figure 4. Safety Case Argumentation Structure.

A claim is a statement that shall be shown to be correct e.g. "The risk associated with the system is acceptable". Claims are based on arguments, which are a way of

reasoning this claim, e.g. "Testing was performed satisfactorily". Evidences are all basic facts that support the arguments e.g. "The test results have been verified and no catastrophic errors were found". If necessary, claims can be structured into sub-claims, sub-sub-claims...

The context details the environment in which the argumentation structure is built up. If the context changes, the argumentation has to be re-assessed, whether it is still valid in the new context.

5.3.1 Types of Arguments

Arguments can be:

- A deterministic or analytical application of predetermined rules to derive a true/false claim (given some initial assumptions), e.g. a formal proof (compliance to specification, safety property), an execution time analysis, exhaustive tests...
- A probabilistic quantitative statistical reasoning, to establish a numerical level, e.g. Mean Time To Failure (MTTF), Mean Time To Repair (MTTR), reliability testing,...
- A qualitative compliance with arbitrary rules that have an indirect link to the desired attributes, e.g. compliance with quality management standards, safety standards, staff skills and experience,...

A typical basic structure is shown in Figure 5.

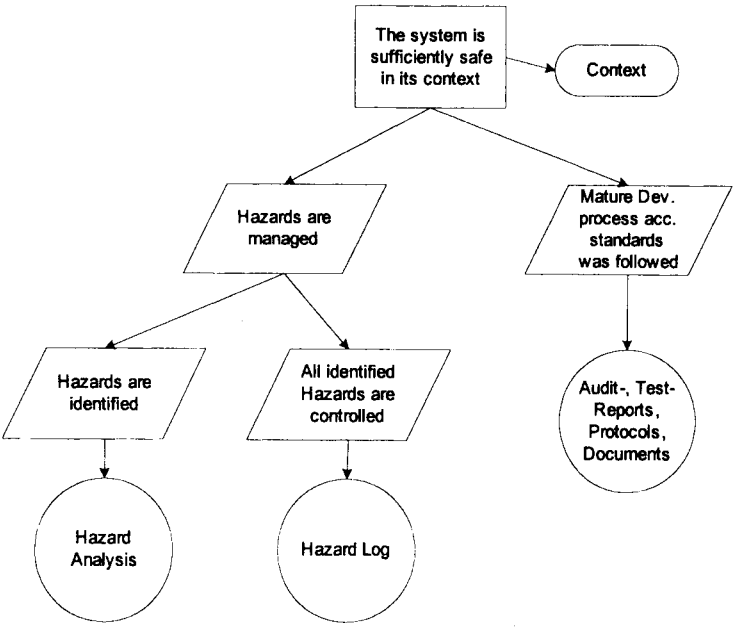


Figure 5. Safety Case Basic Structure.

Other typical basic approaches are:

- Process and Product: The system is safe as there is a safe development process and a safe finished product
- Functional Breakdown: The system is safe as all system functions are safe
- ALARP: The system is safe as all risks are reduced ALARP
- At Least As Safe: The new system is safe as it is at least as safe as an existing system

Many safety cases use combinations of these forms.

5.4 Sources of Evidence

The arguments themselves can utilise evidence from the following main sources:

- The design
- The development processes
- Simulated experience (via reliability testing)
- Prior field experience (proven in use)
- Organisational issues, safety management, competency
- Testing
- Formal analysis

The choice of arguments will depend on the availability of such evidence, e.g. arguments for reliability might be based on field experience for an established design, and on the development process and reliability testing for a new design.

Obtaining suitable evidence is a crucial factor for the quality of the safety case. It is ideal if most of the evidence is already available prior to the generation of the safety case – such that it does not have to be created specifically for the safety case. Therefore, suitable company processes ideally produce some of the evidence automatically. For example, the test reports which are generated in the course of a product development in any case, can already be used as evidence for the safety case if they are in the right structure.

For Commercial Off The Shelf (COTS) products it is often quite difficult to get proper evidence. A typical example is a computer operating system where you usually get no information on the development process, no proper documentation and test reports, only some "sales style" statements, how reliable the system is. Experience nevertheless shows that many current operating systems have a huge variety of functions but very limited stability and reliability. In such a case additional mitigating strategies such as the reduction of the operating system to the basic instruction set which is needed, external watchdogs, external redundancies and comprehensive system level testing have to be employed to argue that the system is safe. Operational experience with the system in similar applications can give further assurance for your argumentation.

The same applies to COTS hardware especially in the computer market, where you often get no relevant data as well. Experience based assumptions for reliability figures and, again, system level testing, operational experience and system field data can fill some of the gaps.

On the other hand, our systems usually are declared as COTS, as we have a basic product, designed for re-use, which is configured and sometimes adapted to the customer's needs. This sometimes narrows the need for comprehensive safety justifications a little down.

Due to the possible adaptations, the use of field data might be limited, but still can show a good basis and gives, together with many direct evidences, trust in the system's behaviour.

5.5 Safety Case Maintenance

A safety case has to be maintained throughout the whole lifecycle of a system as changes in the system and the environment could affect the validity of the used argumentation. Such changes could be:

- Changes to the system itself
- Changes in operational requirements
- Changes to the implementation and assurance technologies
- Physical deterioration of the equipment
- Changes to safety criteria, standards and the regulatory environment
- New technical knowledge and the feedback of experience
- Changes to the safety case process, people and technical resources
- Changes in organisational structures and responsibilities

Typical problems at the maintenance of a safety case, in addition to common problems of large, changing documents with distributed information sources and users, are:

- Difficulty in recognising the importance of changes to the safety case: In conventional cases it can be difficult to discern the objectives, the evidences and the contexts.
- Difficulty in identifying the indirect impact of change: The question is, when to change the safety case and when not to change and how to reason the decision.
- Insufficient information recorded to support the change process.
- Implicit Assumptions: If assumptions are not exactly recorded in the safety case it is difficult to notice, when these pre-conditions change and how this influences the argumentation.

5.6 Key Issues of a Safety Case

The following issues were found to be essential when performing a safety program, finished with a safety case report:

- Safety tasks should be carried out by demonstrably competent individuals and organisations.
- Safety management should be implemented as a key element of a harmonised, integrated systems engineering approach.

- Safety culture is very important. It will not be possible to implement proper safety work without the awareness and consciousness of all involved people. It is the product of individual and group values, attitudes, perceptions, competencies and patterns of behaviour that determine the commitment to, and the style and proficiency of, safety management.
- The quality of safety management and the associated safety culture is a factor in the confidence in the evidence.
- You should identify all credible hazards and accidents and the risks associated with them.
- You should monitor defect/failure reports and incident/accident/near-miss reports and implement remedial actions.

5.7 Problems of Safety Cases

Some of the typical problems during the production and approval of a safety case are detailed in the following sub-chapters.

5.7.1 Level of Detail

One very essential question at a safety case is always, what level of detail is to be chosen regarding the documentation, the reports, the analyses and generally with respect to the exact fulfilment of the requirements of the standards. If the work is performed too extensively, broken down into too much detail it would be commercially unfeasible. For large, sophisticated systems the safety case can become a tremendous compilation of documents, which is very difficult to be put into a clear, readable structure and to be kept consistent and up to date. This becomes even more a burning issue as the considered systems get more and more complex.

A full safety case takes a lot of co-ordinating effort as there can be a large amount of supporting evidence from potentially many sources which have to be combined in a useful way.

No customer would or often even could pay these efforts if you cannot show, that they have a reasonable cost-benefit ratio. Nevertheless the requirements and standards have to be fulfilled and the authorities have to be satisfied.

5.7.2 Early Contact to Approving Body and ISA

One issue of vital importance is to establish an early contact to the approving body and the Independent Safety Auditor (ISA) and work in close co-operation from the very beginning to avoid that safety related issues are raised very late and cause lots of additional effort.

Sometimes a certain reluctance to sign a safety case or parts of it can be seen as nobody wants to be responsible in case of an accident. It is therefore essential to insist on the naming of a respective contact person with sufficient know-how and decision making competence by the customer or authority. All roles and

responsibilities have to be clear. Otherwise many decisions will take a very long time as it is extremely time consuming if a question has to be passed on in a chain to the decision maker and to wait, until the final decision comes back. Ultimately project milestones can be exceeded if safety targets are not agreed early enough or the safety case is not accepted due to arguments about the safety verification.

This is often combined with time schedules, which are quite un-reasonable to be fulfilled from the very beginning and can cause sloppy safety work and possibly an unsafe product due to the big pressure.

5.7.3 Un-reasonable Time Schedule

At many projects the customer demands a time schedule, which prevents proper safety work to be done.

5.7.4 Process versus Goal Based Standards

Even though the basic principles of the safety process usually are the same, there is one major difference in the approach of standards: Process describing standards versus goal based standards. Current practice is moving towards goal based standards, that is, they say what you must do, not how you have to do it. This is easier if you have respective experience and if you know, what exactly the approving bodies demand, but it is a big problem if you are relatively new in this field. On the other hand there are many more and more stringent process describing standards, which all in themselves may be consistent, but no provider can comply fully with all of them, especially if there are many unreasonable requirements. These are often caused by the fact that the standards are written for one specific context, maybe with one specific technology in mind but have to be applied in a different situation. One example is the CENELEC standard EN 50128, which demands 100% code coverage at module testing. This is useful for small systems with a high number of pieces produced, like a train detection system, but is very hard to comply to for large, complex systems with reasonable effort. At least, nearly no customer wants to pay for that.

Many times it is questionable, whether the intentions of the authors are understood by the approving authorities. Basically a standard has to be adapted to the specific context it is used in, but sometimes certification bodies take them "word for word".

5.7.5 Confusing Scene of Standards

The scene of standards can no more be overlooked by a single person. Many customers require "all applicable standards", but no one can tell you, which one. These are often "suicidal" requirements – they have to be accepted to get the contract but lead the supplier into a situation where he depends on the goodwill of the customer or authority to get the final acceptance for the project, as there is always any applicable standard he is not fully compliant to.

More guidance from standardisation bodies and authorities would be helpful, overviews, which standards are equivalent to others, and which are applicable for which situations/systems/environments.

5.7.6 Company Internal Problems

Ideally a safety case should be developed along with the product development. In reality it is often generated too late ("after the fact") when you can no more intervene if e.g. supporting evidence is not produced in a proper manner.

This is especially a problem, if there is a lack of understanding of the necessity of the safety case by the employees, which needs a lot of motivation and convincing beforehand.

This can also lead to the problem, that the design does not meet some safety requirements, as the designer did not understand the reasons behind.

5.7.7 Structural Clarity

A major issue regarding the document is the structural clarity. As a safety case needs a lot of supporting documentation usually a lot of references are made. Care has to be taken by the author, that there are not too many references of references as it can get impossible to have a complete view and in addition to avoid circular references.

A graphical representation of the underlying structure supports the readability of the safety case and helps to get the overview of the whole argumentative body.

5.8 Practical Example of a Safety Case Structure

This example (Figure 6) shows the basic structure of an example safety case in the goal structure notation, with one branch ("Arguments over Direct Evidence, Product related", Figure 7 and Figure 8) described in more detail. It can be seen, that the safety management system and safety process, including the verification of safety requirements, as detailed above serve as essential evidence for main argumentation branches. Of course there has to be sufficient evidence from the general management and development processes as well.

6 Conclusion

We at FREQUENTIS believe to have found a feasible and useful way of dealing with safety cases with the help of our safety management system and the safety process

- to provide the necessary evidences for a safety case,
- to fulfil the requirements from the standards, the customers and the authorities,
- to have a reasonable cost-benefit ratio to ensure economic competitiveness and
- to control and improve the real system safety of our products.

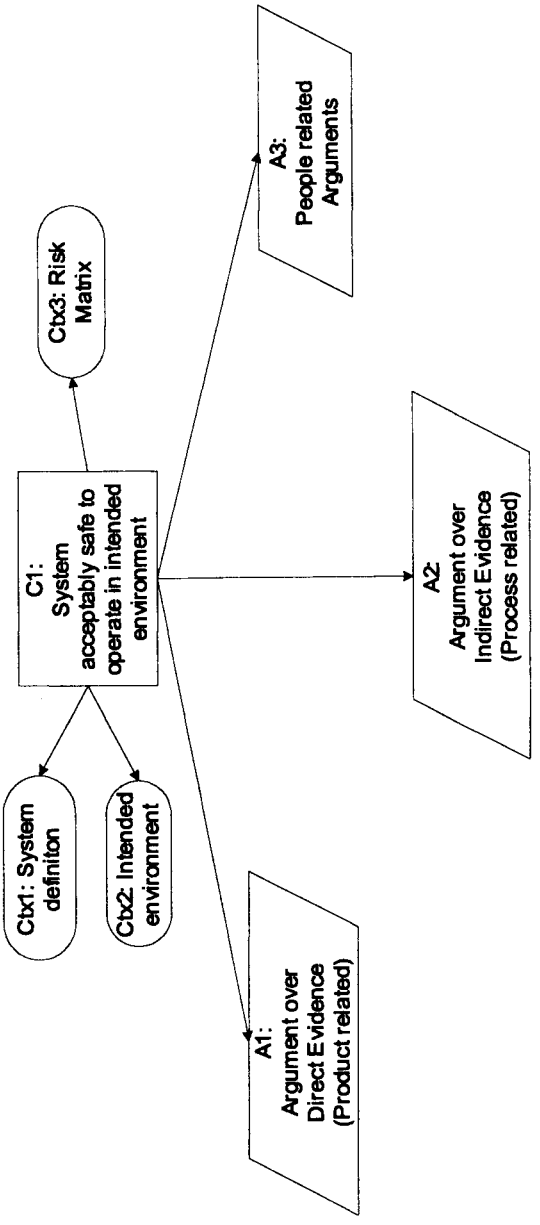


Figure 6. Example Safety Case Structure, Top Level Claim.

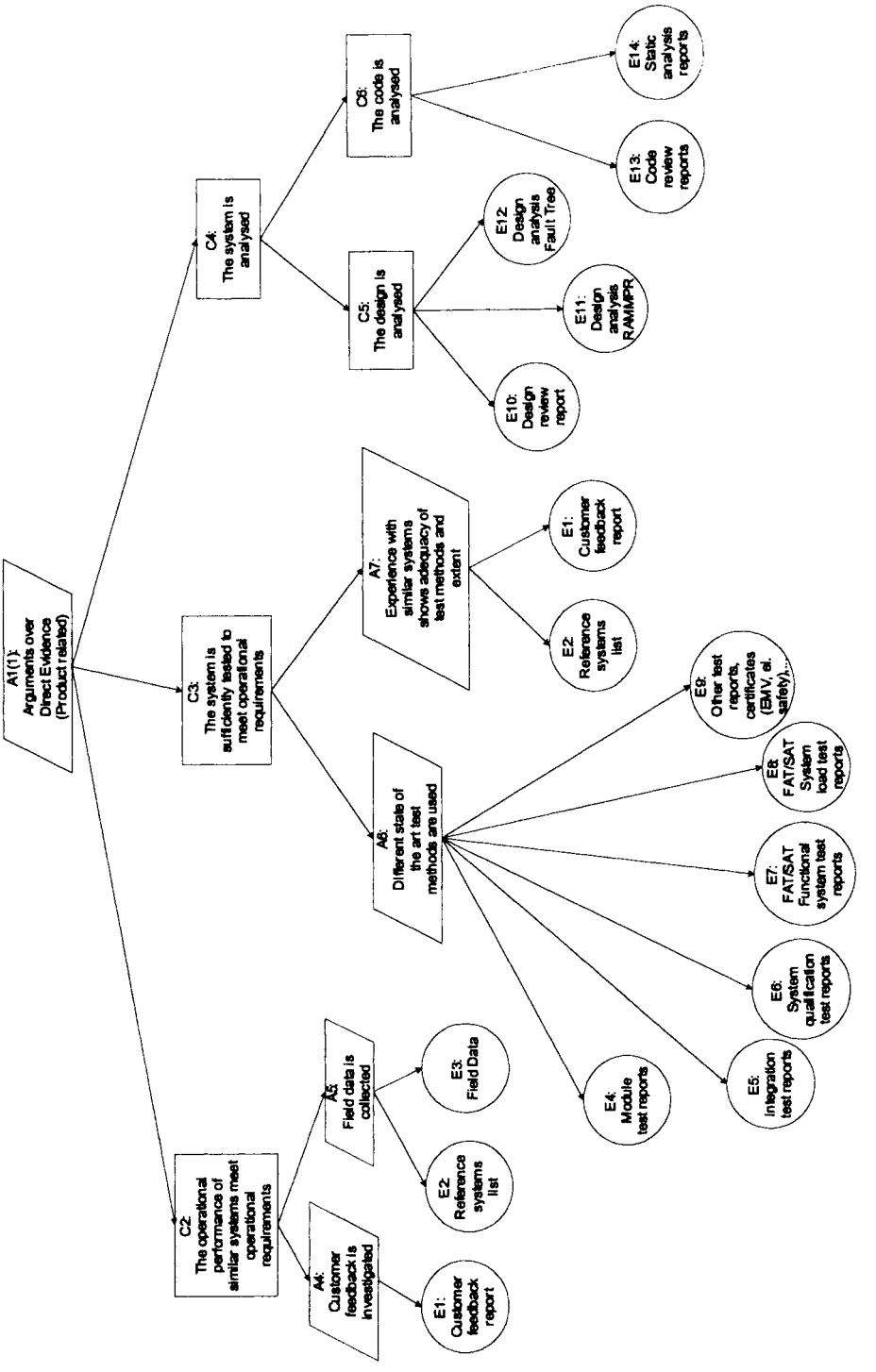


Figure 7. Branch A1(1) of the Example Safety Case Structure.

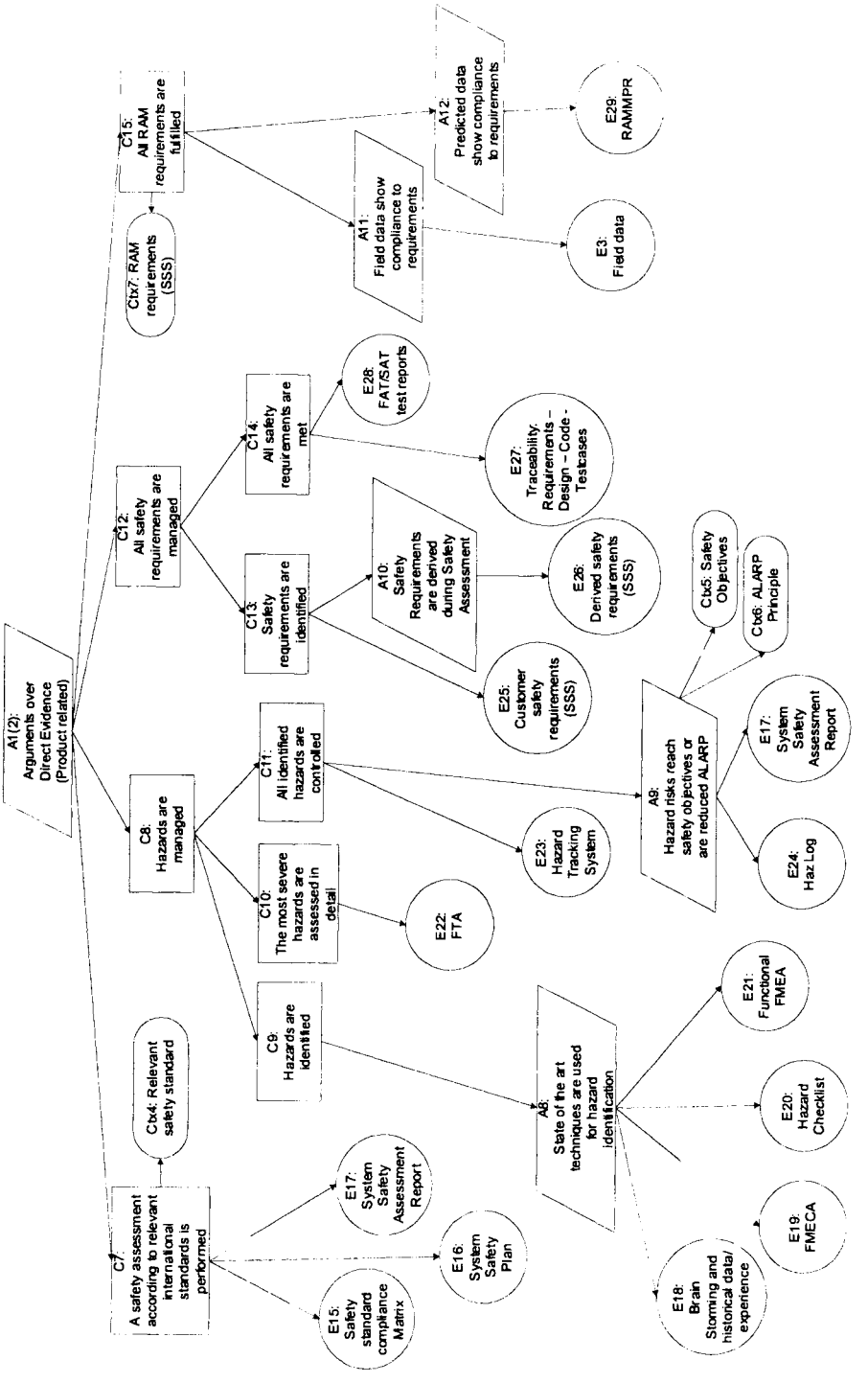


Figure 8. Branch A1(2) of the Example Safety Case Structure.

Safety Case Development - a Practical Guide

Derek Fowler
Independent Safety Consultant, Henley on Thames, UK

Dr Bernd Tiemeyer
EUROCONTROL, Safety Enhancement Business Division, Brussels, Belgium

Abstract

This paper provides guidance on the development of Safety Cases as a means of demonstrating the safety of a safety-related service or new/modified system. It is aimed at those, employed on projects or in service-provider organisations, which have to:

- Produce Safety Cases – eg safety practitioners;
- Approve Safety Cases – eg programme managers and senior line management within service-provision organisations;
- Review Safety Cases – eg safety department staff.

The aim is to achieve sound, well-presented Safety Cases through the adoption of a logical, rigorous, consistent and accurate approach that is based on good safety practice.

The paper is based on the experience of EUROCONTROL in the development of Safety Cases, and captured in its recently revised Safety Case Development Manual. Although the material is orientated towards the business of EUROCONTROL (ie Air Traffic Management) we believe that is readily adaptable to other application sectors.

1 Introduction

From a historical perspective, the following, fairly succinct, and certainly authoritative, statement on Safety Cases comes from Lord Justice Cullen's report on the Public Inquiry into the Piper Alpha Oil Platform Disaster (Cullen 1990):

"Primarily the Safety Case is a matter of ensuring that every company produces a formal safety assessment to ***assure itself*** *that its operations are safe.*

Only secondarily is it a matter of demonstrating this to a regulatory body. That said such a demonstration both meets a legitimate expectation of the workforce and the public and provides a sound basis for regulatory control."

In the field of Air Traffic Management (ATM), for example, service providers are required to ensure the safety of air traffic, in respect of those parts of the ATM system and supporting services within their managerial control. Implicit in this

obligation is a "burden of proof" on those with managerial responsibility to ***demonstrate*** positively that an acceptable level of safety is achieved.

This leads us into the primary purpose of a Safety Case; broadly, it is the documented means by which those who are accountable for on-going service provision, or for managing changes to that service and/or underlying system[1], assure ***themselves***, that those services or changes deliver an acceptable level of safety.

As the main objective of safety regulation is to ensure that those who are accountable for safety discharge their responsibilities properly, then it follows that a safety case which serves the above primary purpose should also provide an adequate means of obtaining regulatory approval for the service or project concerned.

The next section examines the nature of Safety Cases, including issues such as what a Safety Case actually is, how it differs from a Safety Assessment report and what is meant by assurance, as called for by (Cullen 1990).

2 The Nature of Safety Cases

The idea of a Safety Case came originally from a legal case. Under an adversarial legal system, cases are prepared by both the prosecution and the defence. Each case is presented as a series of arguments, stemming from an overall claim of guilt or innocence, followed by the presentation of evidence to show that each strand of the argument is true.

Exactly the same principle applies to Safety Cases, except that the overall claim is invariably that something (eg service or system) is *safe*. It has been suggested that a Safety Case is a pre-emptive case for the "defence" – that seemingly cynical view can actually be a useful way of approaching the development of a Safety Case and serves as a reminder that it is not a task to be undertaken lightly! However, this analogy breaks down in one very important respect - for a Safety Case, the *burden of proof* rests with the "defence" – ie it is up to the authors of the Safety Case to prove that something is safe, rather than for some other body to prove that it isn't safe.

For a legal case, there are rules that have to be followed concerning the applicability and quality of evidence, and the means by which it was obtained. Whilst it is necessary to show that the rules and related procedures have been adhered to completely and correctly, it would clearly be absurd for a prosecuting authority to base the whole case on this – ie solely on the assertion that because the rules and procedures have been followed, person 'A' must be guilty. Rather, the most compelling evidence would be that which showed how the products (or outputs) of applying those rules and procedures demonstrate guilt; evidence of compliance with the rules and procedures would then be used to back up (ie give more credibility to) the main evidence.

---

[1] The term 'System' as used throughout this paper includes equipment, people and procedures, in the context of a defined operational environment. In the case of ATM, the operational environment includes, the structure and rules of the airspace, pilots, aircraft, airborne electronic equipment (including collision-avoidance systems) etc.

Again, we have a clear equivalence in the Safety Case. It would not be sufficient to claim that because we have carried out a safety assessment of system 'A' in accordance with procedure XYZ then system 'A' must be safe. Rather, we must present arguments and evidence that the products (results) of the safety assessment actually show that system 'A' is safe and use evidence about the safety assessment process to give increased confidence in the main evidence.

To summarise so far:

- in a similar manner to a legal case, a Safety Case is based on argument and evidence;
- best evidence comes directly from the products (or outputs) resulting from the application of appropriate processes; evidence about the processes themselves is used mainly to provide backing (to give credibility to) the product-based evidence;
- there is a prima facie presumption of lack of safety.

Having set the scene, the rest of the paper describes what EUROCONTROL, on the basis of collective experience of developing a number of Safety Cases, sees as current good practice[2], in the field of ATM. A fairly brief, theoretical treatment is supported by practical guidance and examples.

3 Safety Case Essentials

3.1 Types of Safety Case

Basically, there are two types of Safety Case[3]:

- those which are intended to demonstrate the ***on-going*** safety of a service and/or system - very much the (Cullen 1990) view;
- those which are intended to demonstrate the safety of a significant ***change*** to a service and/or system.

In EUROCONTROL, the former is known as a Unit Safety Case and the latter as a Project (or System) Safety Case. They are interrelated, as explained below, but since they are somewhat different in approach it is important to decide which is applicable in a particular situation.

Whereas it is sensible to produce a (Project) Safety Case whenever a substantial change[4] to an existing safety-related system (including the introduction of a new system) is to be undertaken, if that is all that we do - ie we do no also establish the absolute safety of the on-going service – then there is a risk that such changes are

---

[2] Captured in a Safety Case Development Manual (EUROCONTROL 2005), developed in conjunction with customers, suppliers and ATM service providers around Europe

[3] There are other kinds of Safety Case but most, if not all, of these are variations on the two basic types

[4] The definition of "substantial change" in this respect is very industry / application-specific and should be defined in the relevant Safety Management System.

being "built on sand". Thus because Project Safety Cases are usually incremental in nature, they must be predicated on a validated assumption (or, better still, on evidence from the corresponding Unit Safety Case) that the pre-change situation is itself safe.

Therefore, in order to provide a solid foundation for change, every provider of a safety-related service / facility should have, and maintain, a Unit Safety Case which shows that the on-going, day-to-day operations are safe. In order to show also that such operations will remain safe indefinitely, a Unit Safety Case should also include argument and evidence that processes are in place to ensure that all changes to the service and/or system are managed safely through, inter alia, Project Safety Cases.

Project Safety Cases are used to update, and usually subsumed into, Unit Safety Cases; of course, both must be specifically designed to facilitate such updates.

3.2 Safety Argument Essentials

This paragraph presents the essential points to be observed in the construction of Safety Arguments. The approach draws on current good practice without prescribing a particular methodology, and is supported by examples in the Appendix to this paper.

A Safety Argument is a statement (or a set of statements) that is used to claim that the service or system concerned is *safe*, and should be developed as follows.

The Safety Argument must start with a top-level statement (Claim) about what the Safety Case is trying to demonstrate in relation to the safety of the service or system. The Claim must be supported by:

- **Safety Criteria**, which define what is 'safe' in the context of the Claim;
- for Project Safety Cases, the **Justification** for introducing the change to the service or system concerned;
- the **Operational Context** for the Claim;
- any fundamental **Assumptions** on which the Claim relies.

The decomposition of the Claim into lower-level Arguments provides the essential links between the Claim and the wealth of Evidence needed to show that the Claim is valid. In performing this decomposition, it is important that:

- each Argument in the structure is expressed as a simple predicate – ie a statement that can be only true or false;
- the Argument structure does not contain any indirect or inconclusive Arguments;
- the set of Arguments at each level of decomposition is necessary and sufficient to show that the parent Argument is true;
- a valid counter-Argument, which would negate the parent Argument, does not exist;
- where the rationale for decomposition of an Argument into lower-level Arguments is not self evident, it is explained by supporting text;

- the number of levels of decomposition is appropriate to the complexity of the Safety Case and/or supporting Evidence;
- each branch of the Safety Argument structure is terminated in supporting Evidence;
- there is a clear distinction between, and correct use of, Direct (product-based) and Backing (process-based) Arguments and related Evidence.

Further guidance on the structuring of Safety Arguments is given in paragraph 4.3 and generic ATM examples are presented in the Appendix to this paper.

3.3 Safety Evidence Essentials

This paragraph presents the essential points to be observed in the collation, review and presentation of Safety Evidence.

Safety Evidence is information, based on established fact or expert judgement, which is presented to show that the Safety Argument to which it relates is valid (ie is true).

The essential rules of Evidence are as follows:

- Evidence must be presented only to the degree and extent necessary to support the related Argument;
- Evidence must be clear, conclusive and, wherever possible, objective;
- the type of Evidence – from safety analysis, design, simulation, test, previous usage, compliance with standards etc – must be appropriate to the Argument;
- the rigour of the Evidence must be appropriate to the associated risk;
- Evidence must actually relate to the correct configuration of the system under consideration.

Further guidance on the gathering, assessing and presenting Evidence is given in paragraph 4.4.

4 Developing a Safety Case

4.1 Safety Cases and the Project Lifecycle

A simplified view of a typical project lifecycle is shown in Figure 1 below.

"Safety Considerations" are the documented results of a process to identify, as soon as possible after a mature Operational Concept has been developed, the main safety issues associated with a Project and to help in deciding whether a full Safety Plan and Safety Case are required.

Building on the Safety Considerations, the initial Safety Argument should be as complete as possible and at least sufficient to form the basis of the Safety Plan. It also provides the starting point, and framework, for the development of the Project Safety Case.

The Safety Plan specifies the safety activities (mainly the gathering and assessment of Evidence) to be conducted throughout the project lifecycle and the allocation of responsibilities for their execution.

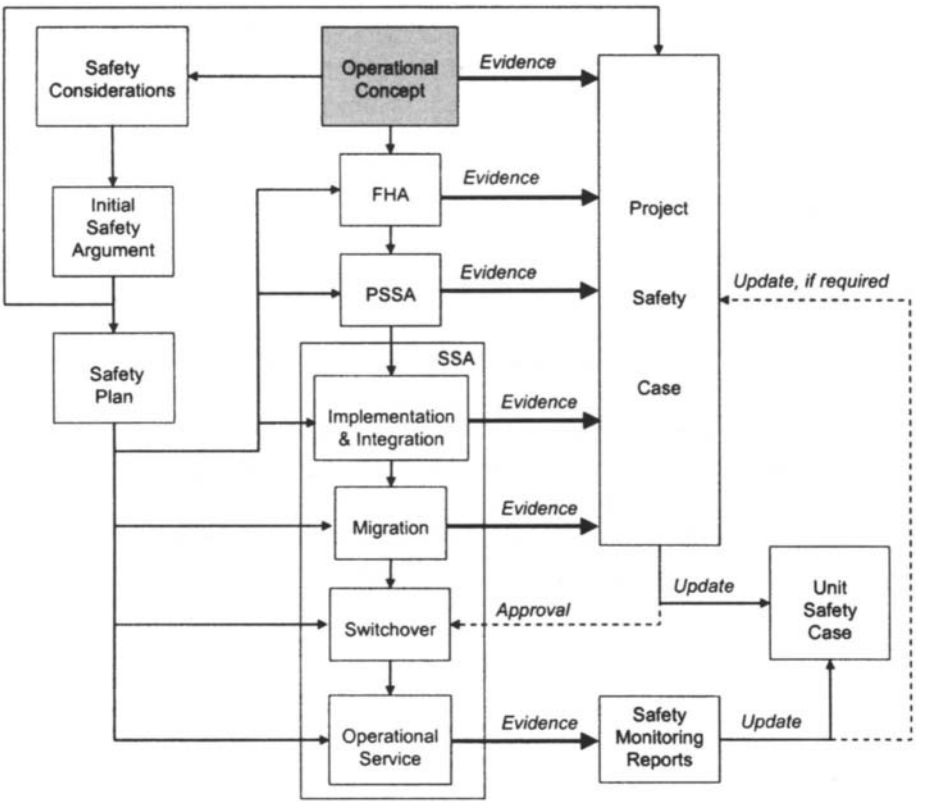


Figure 1: Safety Lifecycle

The three main phases of safety assessment – Functional Hazard Assessment (FHA), Preliminary System Safety Assessment (PSSA) and initial stages of System Safety Assessment (SSA) - provide much of the Evidence needed for the Project Safety Case.

Migration is the phase that covers all the preparation needed in order to bring the new / modified system – ie the subject of the Safety Case – into operational service, including risk assessment and planning for the moment of Switchover. Switchover of the operational service to the new/modified system would normally be preceded by finalisation and, where applicable, regulatory approval of the Project Safety Case.

Because most, if not all, of the preceding safety assessment work is predictive in nature, it is important that further assurance of the safety is obtained from what is actually achieved in operational service. If the operational experience differs

significantly from the results of the predictive safety assessment, it may be necessary to review and update the Project Safety Case.

Once a satisfactory steady state has been achieved, it would be appropriate to update the Unit Safety Case (if one exists) with the information from the Project Safety Case thus establishing a new safety baseline for the on-going service.

Decommissioning of a system, at the end of its operational life, is not shown explicitly on Figure 1, but may be thought of as a special case of a change.

4.2 Determining the Safety Criteria

4.2.1 General Considerations

Safety Criteria are essential to the definition of what is *safe* in the context of the top-level Safety Claim. Basically, they fall into three categories as follows:

- *Absolute*: compliance with a defined target – eg a numerical Target Level of Safety (TLS) – or portion thereof. Such criteria are usually quantitative;
- *Relative*: compared to an existing (or previous) level of safety. Such criteria may be quantitative or qualitative;
- *Reductive*: where the risk is required to be reduced as far as reasonably practicable. Such criteria are usually qualitative.

In general, absolute criteria are preferred since satisfaction of them does not depend on proof of past safety achievement and such proof may be difficult if a suitable baseline does not exist or sufficient historical data is not available. However, in some cases, there may be a problem in establishing what would be a suitable target on which to base the criterion because either:

- a regulatory target has not been set for the operational environment concerned; or
- for Project Safety Cases, it may not be feasible to determine what portion of the overall target it would be reasonable to allocate to the part of the system concerned.

As an alternative to the absolute approach, a relative Safety Argument (ie based on a relative criterion) could be use for a Project Safety Case[5] if:

- a well-defined baseline, prior to the introduction of (or change to) a 'system', could be established; and
- it can be shown, or at least reasonably be assumed, that the baseline situation was acceptably safe.

A reductive approach is normally used in addition to one (or both) of the above criteria. It is an important basis for in-service safety monitoring – especially regarding incident investigation and corrective action.

In European ATM, there are three regulatory requirements that guide the decision as to which criteria to use in a given situation:

---

[5] For Unit Safety Cases an absolute approach should always be the primary criterion.

- an overarching safety goal which requires that, for the indefinite future, risk shall not increase and preferably decrease, relative to historical achievement;
- a numerical safety target which defines a maximum tolerable accident rate (ie aggregate risk) for the provision of an ATM service, for design purposes only;
- a requirement on ATM service providers to ensure that risk is reduced as far as reasonably practicable, on an on-going basis.

4.2.2 Use of Risk Classification Schemes

It is not uncommon to find risk classification schemes (RCS) used as criteria on which to base absolute Arguments. However, experience has shown that the numerical basis for setting risk targets in such schemes is sometimes arbitrary and that, in any case, lack of understanding of how the targets were originally derived often leads to inappropriate use. If an RCS is used, it should be done with great caution and full consideration should be given to:

- where the probability/frequency values used in the scheme came from and whether they are (still) valid;
- at what level in the system hierarchy the values apply;
- to what operational environment the values apply – eg, in ATM, the type of airspace, traffic patterns, traffic density, spatial dimension, phase of flight etc;
- how aggregate risk could be deduced from analysis of individual hazards, in segments of the total system.

EUROCONTROL warns against the possible misuse of RCSs, unless the user has a clear understanding of the above issues and how to address them.

4.3 Constructing a Safety Argument

Since the Safety Argument forms the framework of a Safety Case, it is important that the Argument is set out in a rigorous, hierarchical and well-structured and easily-understood way.

One possible way of creating an Argument with the said properties is to use Goal-structuring Notation (GSN), developed by the University of York (Kelly 1998); this provides a graphical means of setting out hierarchical safety arguments, with textural annotations and references to supporting Evidence.

The logical approach of GSN, if correctly applied, brings some rigour into the process of deriving safety arguments and provides the means for capturing essential explanatory material, including assumptions, context and justifications, within the argument framework.

Figure 2 below shows, in an adapted form of GSN, a specimen *Argument* and *Evidence* structure to illustrate the GSN symbology most commonly used in EUROCONTROL ATM Safety Cases.

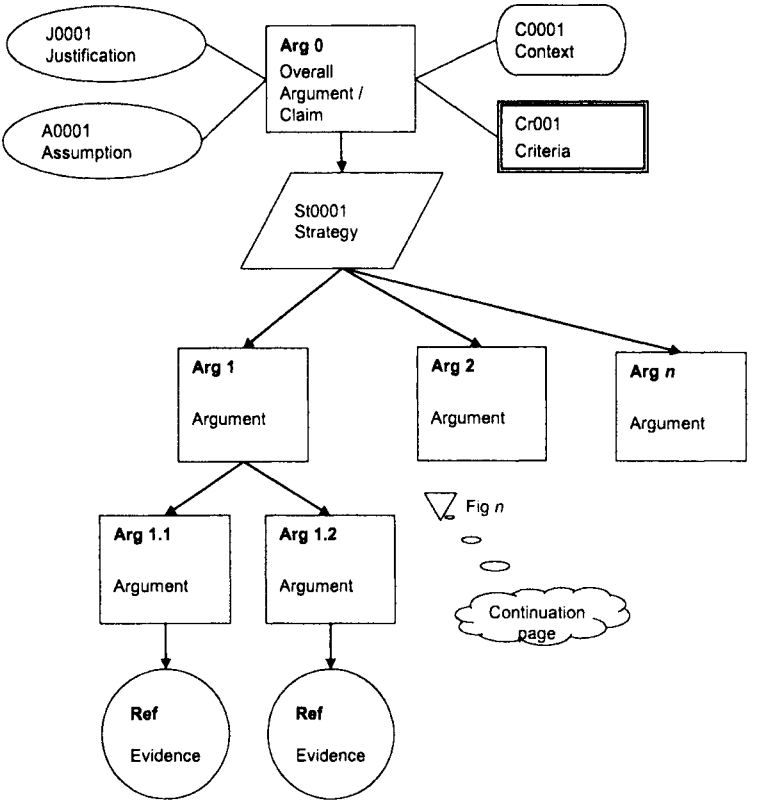
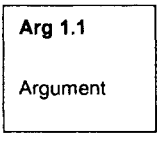


Figure 2. Commonly-used GSN Symbology

The key to the various symbols is as follows.



An *Argument* should take the form of a simple predicate - ie a statement which can be shown to be only true or false.

GSN provides for the structured, logical decomposition of *Arguments* into lower-level *Arguments*. For an *Argument* structure to be *sufficient*, it is essential to ensure that, at each level of decomposition:

- the set of *Arguments* covers everything that is needed in order to show that the parent *Argument* is true;
- there is no valid (negative) *Argument* that could undermine the parent *Argument*.

In Figure 2, for example, if it can be shown that **Arg 1** is satisfied by the combination of **Arg 1.1** and **Arg 1.2**, then we need to show that **Arg 1.1** and **Arg 1.2** are true in order to show that **Arg 1** is true.

If this principle is applied rigorously all the way down through and across a GSN

structure, then it is necessary to show only that each Argument at the bottom of the structure is satisfied (ie shown to be true) in order to assert that the top-level *Claim* has been satisfied. Satisfaction of the lowest-level *Arguments* is the purpose of *Evidence*.

Unnecessary (or misplaced) Arguments do not in themselves invalidate an Argument structure; however, they can seriously detract from a clear understanding of the essential Arguments and should be avoided. The cover-up method illustrated in Figure 3 below can be used to identify unnecessary and misplaced Arguments.

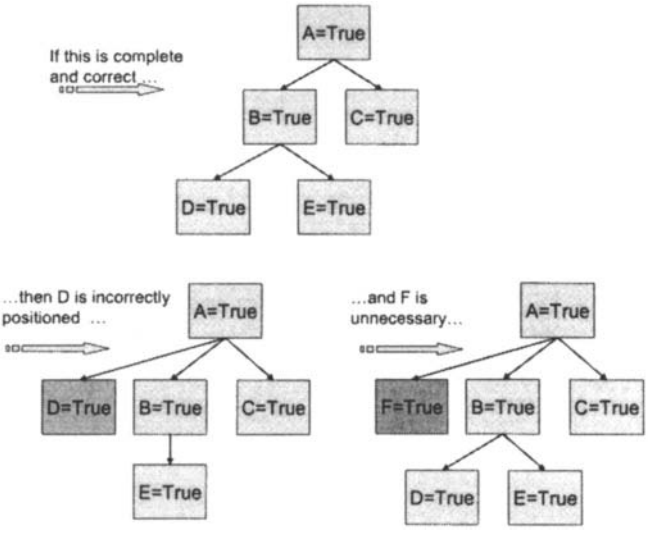
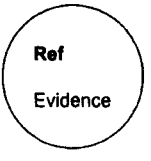
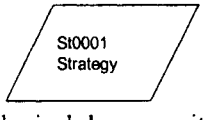


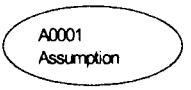
Figure 3. Checking the Argument Structure



It follows from the above that, for an *Argument* structure to be considered to be complete, every branch must be terminated in a reference to the item of *Evidence* that supports the *Argument* to which it is attached. *Evidence* therefore must be appropriate to, and necessary to support, the related *Argument* - spurious *Evidence* (ie information which is not relevant to, and or does not support, an *Argument*) must be avoided since it would serve only to confuse the "picture". Evidence must also be sufficient to support the related *Argument* - inadequate evidence undermines the related *Argument* and, consequently, all the connected higher levels of the structure.

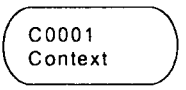


Strategies are a useful means of adding "comment" to the structure to explain, for example, how the decomposition will develop. They are <u>not</u> predicates and do <u>not</u> form part of the logical decomposition of the *Argument*; rather, they are there purely for explanation of the decomposition.



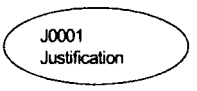
An *Assumption* is a statement whose validity has to be relied upon in order to make an *Argument.*

Assumptions may also be attached to other GSN elements including *Strategies* and *Evidence.*

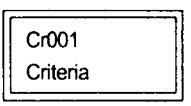


Context provides information necessary for an *Argument* (or other GSN element) to be understood or amplified.

Context may include a statement which limits the scope of an *Argument* in some way.



A *Justification* is used to give a rationale for the use or satisfaction of a particular Argument or *Strategy.* More specifically, it can be used to justify the change that is the subject of the Safety Case



Criteria are the means by which the satisfaction of an *Argument* can be checked. They are used, for example, for defining what an *acceptable* level of risk is.

In numbering the elements of a GSN structure, it is recommended that:

- *Arguments* be numbered hierarchically (eg, **Arg 1.1**) in order to reflect their logical structure;
- *Strategies*, *Assumptions*, *Context*, and *Criteria* be numbered sequentially (eg, **St0001**) since they embellish, but do NOT form part of, the logical structure;
- *Evidence* be numbered according to its source reference and that the *Evidence* 'bubble' contains a brief indication of the form that the *Evidence* takes.

4.4 Gathering, Assessing and Presenting Safety Evidence

Evidence is the heart of every case and ultimately it is on the quality and completeness of the Evidence that the validity of a Safety Case depends. Of course, a well-structured Safety Argument is very important but only insofar as it provides the context for, and thus facilitates interpretation of, the Evidence.

In decomposing the Safety Arguments, the following two main types of Argument (and related Evidence) are used:

- that which shows that a particular objective has been achieved (ie that a higher level Argument or Claim has been satisfied) – this is referred to as ***Direct*** Argument and Evidence;
- that which shows that the Direct evidence is trustworthy (ie that it can be relied upon) – this is referred to as Backing Argument and Evidence.

Direct Evidence may be thought of as being that which relies directly on the observable properties of a ***product*** (ie the output of a process), supporting a logical Argument as to how the product satisfies its safety objectives or requirements, as appropriate.

Backing Evidence is obtained from the properties of the ***processes*** by which Direct Evidence was obtained, and shows that those processes, tools and techniques, human resources etc were appropriate, adequate and properly deployed.

The points below expand upon the "essential rules" outlined in paragraph 3.3 above.

Evidence must be presented only to the degree and extent necessary to support the related Argument. The issue here is that, in the context of an Argument-based approach, any "Evidence" which is unrelated to a part of that Argument is not only of no value but could also serve as a distraction from those aspects of the Safety Case that are relevant.

Evidence should be sufficient, as follows:

- it is bad (but unfortunately not uncommon!) practice to present an element of a structured Argument and then refer to a mass of information as "Evidence" to substantiate the Argument;
- it is vital to the integrity of the Safety Case that the Evidence be presented in such a way that is clear to the reader that the Evidence does actually show the related Argument to be true, "beyond all reasonable doubt";
- where Evidence is contained in appendices or external documents, a summary justifying the adequacy of the Evidence should be presented (in the Safety Case) along with the associated Argument. It is not sufficient to merely reference the Evidence with statements such as "Evidence to support the Argument is presented in ....";
- wherever possible, Evidence should consist of proven facts – eg the results of a well-established process such as simulation and testing. Only where such objective Evidence is not available should Evidence based on expert opinion be used, and then only when the credentials of the expert(s) and the means of eliciting the opinion are adequate and have been presented as Backing Evidence;
- the type of Evidence, from safety analysis, design, simulation, test, previous usage etc, must be appropriate to the Argument – see paragraph 4.5 and 4.6 below;
- the rigour of the Evidence must be appropriate to the associated risk. This is the principle behind the Assurance Level concept in a number of standards, covering software, procedures and human aspects.;
- Evidence must relate to the configuration of the system and the operational environment under consideration.

How the above should be applied specifically to the two main stages of the safety development lifecycle – *requirements determination* and *requirements satisfaction* - and is discussed respectively in paragraphs 4.5and 4.6 below – see also (Fowler,

Tiemeyer and Eaton, 2001).

4.5 Evidence – Safety Requirements Determination

To paraphrase (EUROCONTROL 2001a), Safety Requirements are means by which the necessary risk reduction measures identified in the hazard and risk analysis are rigorously specified. *Necessary* in this context means necessary in order to achieve the required safety levels, as defined by the Safety Criteria (see above).

The primary purpose of ATM is to reduce the risk of accident to air traffic that would otherwise exist. The amount of risk reduction is determined primarily by the functionality and performance of the ATM systems elements, including equipment, people and procedures. However, failure within the ATM system can cause risk to increase again, either by reduction in functionality or performance, or by the introduction of new risk caused by corruption of the outputs of ATM functions.

Therefore, in order to achieve a net safety benefit from ATM, the reduction in risk provided by the desired properties of ATM (ie functional and performance) needs to be substantially greater that any increase due to the undesired properties (ie failure). In EUROCONTROL, we express this essential distinction between the two sets of safety properties in terms of the *success* approach, in which we address the question "is it safe when it is working to specification?" and the *failure* approach in which we address the question "is it still safe if it fails?". This point is emphasised because of a popular misconception that safety is dependent mainly on integrity, whereas neglect of functionality and performance can lead to systems that are "reliably unsafe" – ie for a given set of circumstances, will provide inadequate function and/or performance, consistently! [6]

It follows therefore that Safety Cases are critically dependent on the determination and satisfaction of a complete and correct set of Safety Requirements in which system functionality and performance are appropriately considered alongside system integrity.

Direct Evidence of Safety Requirements Determination is concerned with the requirements themselves and should show, inter alia, that:

- all relevant Hazards have been identified;
- the potential outcomes of the Hazards have been categorised correctly;
- Safety Requirements have been specified to control the Hazards, such that the Safety Criteria are satisfied.

The key issue here is to ensure that the Safety Requirements are complete – ie that all risks are taken into account. It would not be sufficient to show that the Safety Requirements satisfy the Safety Criteria if those Safety Requirements were based on an incomplete / incorrect hazard assessment.

Backing Evidence of Safety Requirements Determination is concerned with the process of deriving the requirements and should show, inter alia, that:

---

[6] See also (Leveson 2001)

- The Safety Requirements were determined using an established and appropriate process;
- the techniques and tools used to support the Safety Requirements Determination were verified and validated;
- The Safety Requirements Determination process was executed by suitably competent and experienced personnel.

The FHA and PSSA stages of the EUROCONTROL Air Navigation System Safety Assessment Methodology (EUROCONTROL 2004a) provides an appropriate and sound process for the determination of ATM Safety Requirements – demonstration of adherence to the FHA and PSSA processes could therefore be used as *Backing* Evidence as in the first bullet point above.

4.6 Evidence – Safety Requirements Satisfaction

Evidence of Safety Requirements satisfaction may be used from three main sources, as follows:

- Service Experience of previous usage
- Verification and Validation
- Compliance with Standards

Service Experience is data from previous operational use of the product concerned. *Direct* Evidence is concerned with analysis of data from Service Experience and what the results of that analysis showed in terms of satisfaction of the safety requirements. *Backing* Evidence is concerned with showing that the environment from which the data was obtained is sufficiently similar to that to which the re-used product will be subjected, that adequate performance-assessment and fault-recording processes were in place when the product was originally deployed, and that the analysis of the outputs of those processes was adequate and properly carried out.

In assessing and presenting Direct Evidence from Service Experience, it is important to ensure that:

- an analysis process, with pass/fail criteria, was specified for each aspect of the product safety requirement whose satisfaction is being justified using service experience;
- the analysis of the service records shows that the criteria for each aspect of the product safety requirement, whose satisfaction is being justified using service experience, have been met;
- all of the details relevant to the argument being made (eg of length of service, history of modifications, list of users) are included in the Evidence;
- any product capabilities that are not necessary to satisfy the Safety Requirements cannot have an adverse effect on the safe operation of the system.

In assessing and presenting Backing Evidence from Service Experience, it is important to ensure, inter alia, that:

- the subject of the Safety Case and the product for which the Service

Experience Evidence is available are identical or sufficiently similar;

- the conditions of use of the product for which the Service Experience is available is taken into account in the analysis;
- the proposed operational environment and the operational environment for which the Service Experience Evidence is available are identical or sufficiently similar;
- any changes made to the operational environment, conditions of use, or product during the period of the Service Experience are analysed to determine whether those changes alter the applicability of the data obtained from Service Experience for the period preceding the changes;
- all aspects of those product functions whose safety requirements that are being justified from Service Experience have been exercised in the (previously) deployed product;
- the extent of the Service Experience is sufficient to demonstrate that each aspect of the product safety requirement has been met;
- a Defect Reporting, Analysis and Corrective Action System (DRACAS) is in place for the deployed product, and is operated in a reliable manner, and is adequate to support the Service Experience Evidence;
- the procedures and tools used to support the creation and analysis of Service Experience Evidence were verified and validated;
- for all reported failures of an aspect in the product component, the underlying fault has been corrected, or it has been shown that the fault is not relevant because it has no safety impact;
- the collection and analysis of Service Experience Evidence was done by suitably competent and experienced personnel.

Evidence from system Verification and Validation (V&V) may be based on, inter alia, analysis and/or testing.

Analysis, in this context, covers any proof of requirements satisfaction that is obtained from the design or other representation of the product, including models, prototypes, software source code etc. It includes, for example, simulation, formal proof, hardware reliability prediction, inspection, and software static and dynamic code analysis.

Testing is restricted largely to tests of the final product in an environment which is as close as possible to the operational environment. Its purpose, broadly, is to demonstrate that what has been built satisfies the requirements, and it is used to supplement (sometimes replace) *Analysis*.

It is beyond the scope of this paper to discuss the relative merits of analysis and testing, or of the various techniques within those two broad categories. Suffice it to say that a Safety Case should set out clear justifications of the selected techniques according to the nature and integrity required of the system to which the Safety Case applies. The following guidance is however given concerning the principal requirements of *Direct* and *Backing* V&V Evidence.

However obtained, *Direct* evidence is concerned with the output of the V&V

processes, and should include, as a minimum:

- specifications of what V&V activities were carried out;
- evidence that the V&V activities and pass/fail criteria were sufficient to demonstrate that the related requirements were satisfied;
- the results of the V&V activities;
- analysis of the results to show that all the specified pass/fail criteria were met;
- explanation and justification of any discrepancies in the results.

Whether obtained from analysis or testing, *Backing* evidence is concerned with the V&V processes themselves, and should include, as a minimum Evidence that:

- the processes were specified and performed independently from design;
- the methods and techniques used are appropriate and adequate, for the properties of the product under consideration;
- the tools used to support the processes were verified and validated to a level appropriate for the assigned assurance level and were properly used;
- the V&V processes were properly and completely executed, and the guidance, procedures, and standards were adhered to;
- for previously existing V&V evidence, obtained for COTS or re-used products, the evidence is entirely valid for the new system application;
- any differences between the operational and V&V environments were identified, and the impact on the results were assessed and justified.

Evidence of compliance with standards can be a significant contribution to the safety case. However, the way in which adherence to a particular standard can be used to demonstrate compliance with Safety Requirements will depend on the nature of the standard itself.

Product standards specify precisely what is required of a specific item of equipment in terms of function, performance, integrity and, in some cases, form and fit. A good example is the Arinc 700 series of standards, which define digital avionics systems and equipment installed on civil aircraft. Currently, product standards are not common in ATM.

Compliance with product standards could be used as *Direct* Evidence of system safety, subject to it being shown that the standard was appropriate to the particular application and to the provision of sufficient *Backing* Evidence concerning the adequacy of the process by which compliance was demonstrated.

At the other end of the spectrum, are standards which address the processes of development and manufacture – non-safety examples range from the very broadly based ISO 9000 series to the more specific ED-78A (Guidelines for the Approval of the Provision and Use of ATS Supported by Data Communications) and ED-109 (Guidelines for CNS/ATM System Software Integrity Assurance). In none of these cases would it be appropriate to certify a product against them, from a safety viewpoint; however, compliance with such standards, especially the more specific ones, could provide excellent Backing Evidence for safety requirements

determination and/or satisfaction.

The distinction between product- and process-based safety assurance is clearly fundamental since the former is concerned with getting the right product and the latter with getting the product right.

4.7 Format, Structure and Layout of the Safety Case

Table 1 below provides notes on a suggested Safety Case layout. Examples, relating to EATM can be found in the EUROCONTROL Pre- and Post-Implementation Safety Cases for RVSM, (EUROCONTROL 2001b) and (EUROCONTROL 20054b) respectively.

Executive Summary	This should provide the reader with an overview of what the Safety Case is about, what it is trying to show and for whom, a summary of the conclusions and caveats (see below), and recommendations (if any).
Introduction:	The Introduction should include:
Background	• an outline of, for example, the historical circumstances which led to the need for, and development of, the Safety Case;
Aim	• a simple statement of the aim – ie **what** the Safety Case seeks to demonstrate. It should be related directly to the top-level Claim (see below);
Purpose	• the purpose of the Safety Case – ie **why**, and **for whom**, it has been produced;
Scope	• the scope and boundary of the Safety Case. It is important to explain what is included and what is not included;
Layout	• the purpose of each of the sections of the document. In general, the main part of the document should be structured along the lines of the Safety Argument.
Service / System Description	This should provide a description of the system to which the Safety Case applies, including its operational environment, interfaces and boundaries of responsibility.
Overall Safety Argument	This section should describe and explain the highest levels of the Safety Argument structure, including:
Claim	• the Claim – ie the top-level statement which asserts that the service / system (etc) is *safe*;
Criteria	• the Safety Criteria which define what is meant

	by safe in the context of the Claim;
Context	• a description of the operational context to which the Safety Case applies;
Justification	• the justification for the change, where the Safety Case addresses a change to a service and/or system that is <u>not</u> being made mainly for reasons of improving safety, and therefore potentially for incurring some risk;
Principal Safety Arguments	• the principal Safety Arguments – ie the first level of decomposition of the top-level Claim – these should be reasoned and well structured, showing how the Safety Criteria are satisfied and the rationale for the approach taken in the decomposition;
High-level Assumptions	• the key Assumptions on which the highest levels of the Safety Argument critically depend – for example, the level of risk prior to the introduction of a change is acceptable. Other Assumptions, applicable to the lower levels of the Safety Argument structure should be included in the Assumptions section – see below.
Safety Argument and Evidence sections	These sections should present each of the principal Safety Arguments (see above) in turn, together with the supporting Evidence which shows that each of the Arguments is valid. It is recommended that, where applicable, each section be structured as follows: • Objective (of the section) – related directly to the principal Safety Argument; • Strategy (breakdown of the principal Safety Argument into lower-level arguments); • Rationale (for the Strategy); • Lower-level Arguments / Evidence; • Conclusions (of section).
Assumptions	All the Assumptions on which the Safety Case depends, including the high-level Assumptions mentioned above, should be presented directly, and/or by reference. Assumptions usually relate to matters outside of the direct control of the organisation responsible for the Safety Case but which are essential to the completeness and/or correctness of the Safety Case. Each Assumption must be shown to be valid or at least reasonable according to the

	circumstances.
Issues	Any outstanding safety issues that must be resolved before the Claim can be considered to be valid should be listed, together with the responsibilities and timescales for clearing them.
Limitations	Any Limitations or restrictions that need to be placed on the deployment and/or operation of the system should be stated and explained.
Conclusions	The Conclusions should not merely repeat the conclusions from each previous section. Rather, the main Conclusion should refer to the original Claim and, if applicable, reassert its validity, subject to the following caveats: • the Scope – especially what the Safety Case does not cover; • the operational Context to which the Safety Case applies; • the Assumptions that have had to be made; • the outstanding Issues; • any Limitations placed on the deployment and/or operation of the service / system.
Recommendations	Recommendations are not mandatory and any that are made should not be temporary in nature. For example, it might be appropriate to make recommendations on the use of the Safety Case by its recipients, but not concerning its approval. Recommendations must not contain any statements that would undermine, or add further caveats to, the Conclusions.

Table 1. Safety Case Outline Layout.

5 Conclusions

The paper provides guidance on the development of Safety Cases as a means of demonstrating the safety of a safety-related service (usually by means of a *Unit* Safety Case) or new/modified system (usually by means of a *Project* Safety Case).

In a similar way to the presentation of a legal case, the importance of a clear, well formed and unambiguous argument, supported by appropriate, and conclusive evidence has been emphasised. Guidance on both aspects of a Safety Case is given in the paper.

A Safety Case is commonly founded on showing that a system (or service) has been specified to be safe and that such specifications have been satisfied in

implementation. It is stressed that both requirements specification and requirements satisfaction need to demonstrate that the system / service is safe when working normally (the success viewpoint) and remains safe when taking account of the reality that it will periodically fail in some way (the failure viewpoint).

Examples (adapted from actual Safety Cases) are provided to illustrate the use of the GSN in structuring Safety Cases, currently favoured by EUROCONTROL.

6 Acknowledgements

The authors would like to take this opportunity to thank Patrick Mana (EUROCONTROL) for his contribution to producing the Safety Case Development Manual (EUROCONTROL 2005), and would like to extend these thanks to those who participated in the review process, which led to current version of the Manual.

The views expressed in this paper are those of the authors and do not necessarily represent official EUROCONTROL policy.

7 References

Cullen 1990 The Public Inquiry into the Piper Alpha Disaster, Volumes 1 & 2, November 1990, HMSO Publications Centre ISBN 0-10-113102-X.

EUROCONTROL 2001a EUROCONTROL Safety Regulatory Requirement 4: Risk Assessment and Mitigation in Air Traffic Management, Edition 1.0, 5 Apr 01.

EUROCONTROL 2001b The EUR RVSM Pre-Implementation Safety Case, Edition 2.0, 14 August 2001.

EUROCONTROL 2004a Air Navigation System Safety Assessment Methodology, SAF.ET1.ST03.1000-MAN-01, 30 April 2004, Edition: 2.0.

EUROCONTROL 2004b The EUR RVSM Post-Implementation Safety Case, Edition 2.0, 28 July 2004.

EUROCONTROL 2005 Safety Case Development Manual, Edition 2.0, 28 September 2005.

Fowler 2001 Fowler D, Tiemeyer B and Eaton A, Safety Assurance of Air Traffic Management and Similarly Complex Systems, Proceedings of the 19th International System Safety Conference, Huntsville, USA, September 2001.

Kelly 1998 Arguing Safety - A Systematic Approach to Managing Safety Cases, T.P. Kelly, University of York, YCST 99/05, September 1998.

Leveson 2001 The Role of Software in Recent Aerospace Accidents, Nancy G Leveson, Proceedings of the 19th International System Safety Conference, Huntsville, Alabama, USA Sep 01.

Appendix to Safety Case Development - a Practical Guide

A.1 Example Application – A "Project" Safety Case

Figures A.1 to A.11 below show a structured Safety Argument for a hypothetical major change ("SGxy") to an ATM service.

The structure is intentionally not complete in all areas of the decomposition; however, it is intended to be sufficient, in breadth and depth, to illustrate the use of the GSN notation. A commentary on the development of the Safety Argument is also provided below. This commentary is also not exhaustive but is intended to bring out all the main points concerning the application of GSN.

The Safety Argument starts, in Figure A.1, with the top-level *Claim* **(Arg0)** that the ATM service, following the change, will be *acceptably safe*.

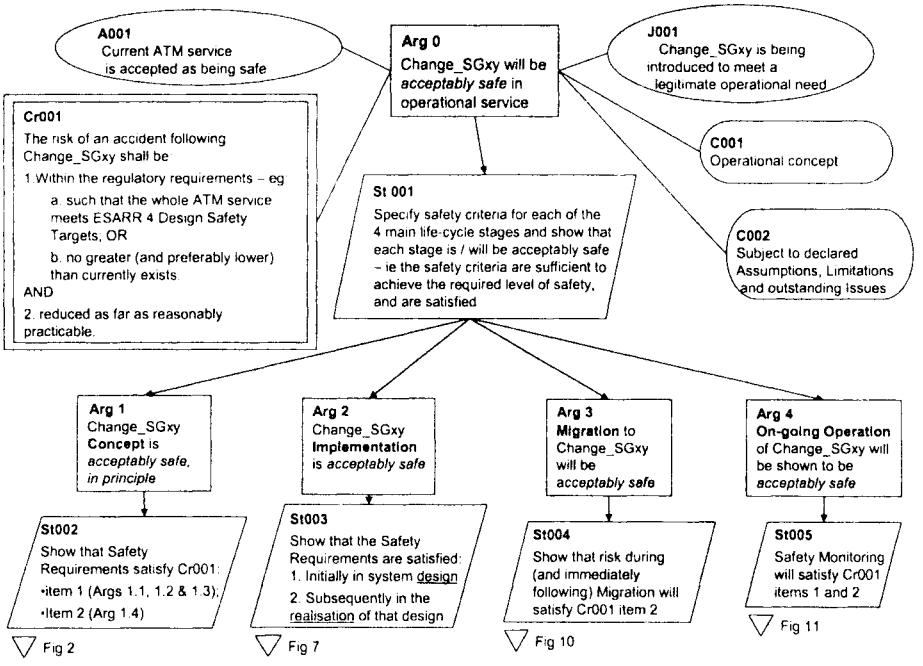


Figure A.1. Arg 0: Safety Argument

J001 indicates that the change is justified operationally and this justification would need to be elaborated in the Safety Case.

C001 provides an essential marker that the change itself needs to be defined in terms of the ATM service / system and accompanying operational concept – such

descriptions would need to be provided in the related Safety Case.

Acceptably safe is defined by three criteria summarised in **Cr001**. These criteria reflect the three main ways of expressing a Safety Argument – ie:

- Absolutely: as compliance with a (numerical) target level of safety;
- Relatively: in relation to the pre-change level of safety;
- Reductively: risk to be further reduced as far as reasonably practicable;

The first two bullets are alternative ways of expressing a typical regulatory minimum safety level[7] and specify what is sometimes known as *tolerable* risk. In the further development of this example, only the absolute criterion is actually used, and is supported by the reductive criterion in order to specify what is sometimes known as an *acceptable* level of risk.

If a relative *Argument*, were to be used it would be necessary to establish that the pre-change baseline is safe. This is illustrated by **A001** on Figure A.1.

As indicated in *Strategy* **St001**, *Claim* **Arg0** is decomposed into four principal Arguments which, in this case, relate to the four main, contiguous stages of the lifecycle of the Change. The outcome of each stage is argued to be acceptably safe and **St002** to **St005** are used to indicate, by reference to **Cr001**, what is defined as acceptably safe for each stage:

- **Arg1** (through **St002**) asserts that the Change is acceptably safe in principle – ie subject to subsequent complete and correct implementation of the Safety Requirements;
- **Arg2** (through **St003**) asserts that the Implementation of the Change is acceptably safe, through satisfaction of the Safety Requirements, and that the rigour of the Assurance (ie lower-level *Arguments* and *Evidence*) to support this is appropriate to the risk associated with the Change;
- **Arg3** (through **St004**) asserts, in effect, that the Migration from the current state to the post-Change state will not endanger the on-going operational service. The change in tense in Arg3 is deliberate since the Safety Argument would be expected to be finalised once all the Implementation and Migration steps, except the final "switchover" to the new state, had been completed satisfactorily. Note that, because of the short time for which the service is at risk, during Migration, only Criterion Cr001, item 2 can be applied to this *Argument*;
- **Arg4** (through **St005**) asserts that the monitoring of the on-going operational service, post Migration, will be sufficient to confirm that the Change is acceptably safe.

Arg1 focuses on the output of the Concept stage of the lifecycle – ie a set of Safety Requirements for the Change that ultimately satisfy the three safety criteria which define an acceptable level of safety.

Arg1 is achieved through a two-fold *Strategy* (see Figure A.2), which uses the principle of *Direct Evidence* and *Backing Evidence*, as follows:

---

[7] It would not normally be necessary to comply with both.

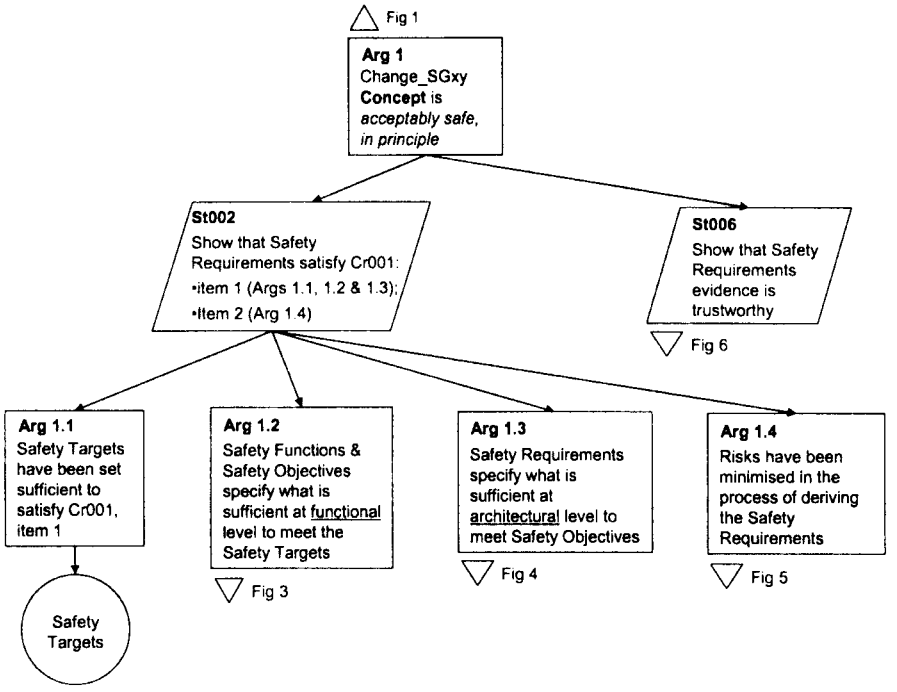


Figure A.2. Arg 1: Safety of the "Change SGxy" Concept

- **St002** shows, through a sequential set of Arguments (**Arg 1.1** to **Arg 1.5**), that the eventual outputs of the Concept phase – the Safety Requirements – satisfy the three safety Criteria. This is clearly a *Direct* approach since it is concerned with the outputs of each stage in the sequence, rather than with the processes that produce those outputs;
- **St006** shows that the *Direct* Evidence is trustworthy – ie it can be relied upon. The Arguments to achieve **St006** are shown in Figure A.6 below, and are considered to be of the *Backing* type since they are concerned with the processes that produce the above outputs, rather than with the outputs themselves (ie they are complementary to **St002**).

Arg 1.1 is not decomposed further in this example but would need to show, through lower-level Arguments and Evidence, that the Safety Targets expand upon, and satisfy the safety criteria specified in **Cr001**. **Arg1.2** to **Arg1.4** are decomposed below, in Figures A.3 to A.5 respectively.

In Figure A.3, the *Context* (**C004**) for **Arg1.2** is a Functional Hazard Assessment (FHA) associated with the Change. **C005** is simply a reminder that the FHA must encompass all aspects of the Change.

Arg 1.2.1 to **Arg 1.2.6** relate to the outputs of the main stages of a typical FHA. Safety Functions are concerned with specifying the desired (correct) operation of a

system in order to provide safe ATM services (known as the "success case") whereas Safety Objectives govern the "failure case" by limiting the frequency of occurrence of each hazard.

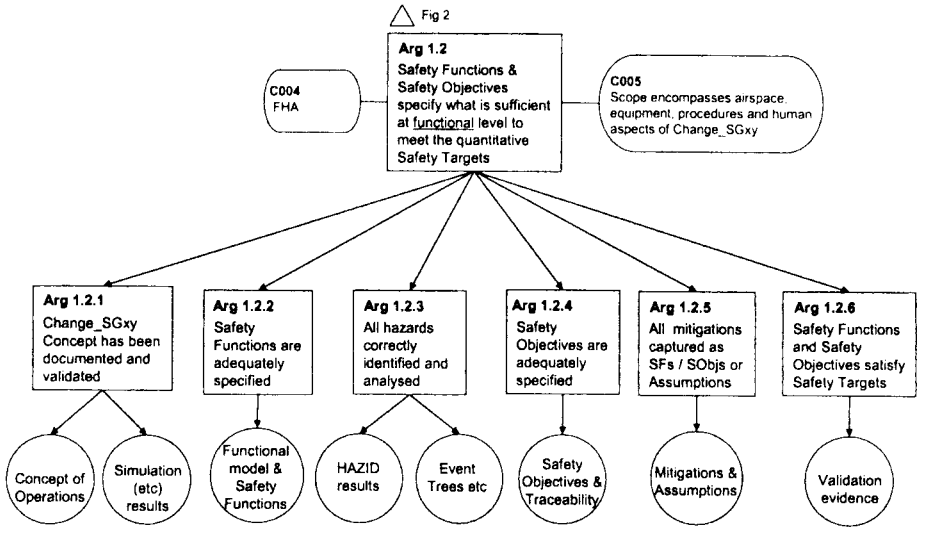


Figure A.3. Arg 1.2: Safety Functions and Safety Objectives

The type of *Evidence* expected to be provided to support each strand of the *Argument* is also shown Figure A.3.

The use of the term "adequately" in **Arg 1.2.2** and **Arg 1.2.4** illustrates what is sometimes a fine distinction between *Direct* and *Backing Evidence*. In general:

- If the *Argument / Evidence* is concerned with observable attributes of an output (product) then it should be considered to be *Direct* – for example, traceability of Safety Objectives back to Safety Functions and Safety Targets would be *Direct* since it would be observable (with the assistance of cross-referencing) from the Safety Objectives, Safety Functions and Safety Targets themselves;
- On the other hand, if the *Argument / Evidence* cannot be deduced from observable attributes of an output itself, but is related only to the process, then it should be considered to be *Backing* – for example it would be impossible to deduce from a set of Safety Objectives that they had been developed by a team with Appropriate expertise – see Figure A.6 below.

The decomposition of **Arg1.3** (see Figure A.4 below) is similar in principle to that for Arg1.2 above. The Context (**C004**) is the Preliminary System Safety Assessment (PSSA) – ie the derivation of Safety Requirements, expressed at the logical-architecture level.

St007 again emphasises the importance of considering the safety of the system when it is working (the "success case", expressed in terms of Safety Requirements

for function and performance) as well as when it fails (expressed in terms of Safety Requirements for reliability and integrity).

The type of *Evidence* expected to be provided to support each strand of the *Argument* is also shown in Figure A.4.

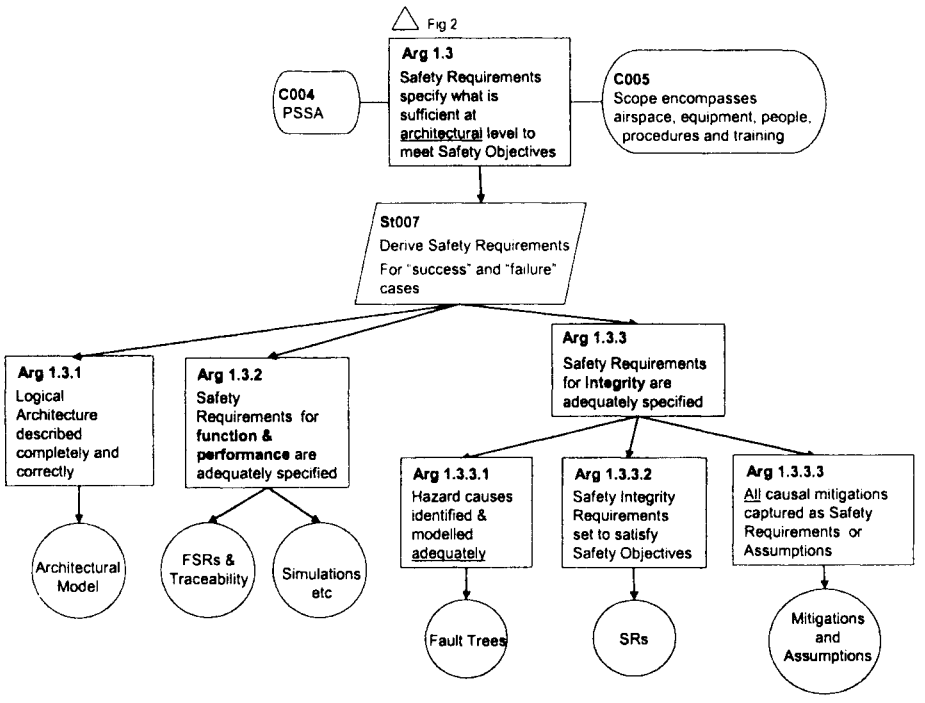


Figure A.4. Arg 1.3: Safety Requirements

Arg1.4 (see Figure A.5 below) presents the Argument and Evidence that the qualitative Safety Targets have been satisfied via the processes that led to the Safety Requirements for Change SGxy.

The difficulty with **Arg 1.4.1** is that most changes in ATM involve some inherent risk because the service in general needs to respond to an ever increasing demand on its capacity to deliver. Therefore, it is necessary to find safety benefits – in the form of removal or mitigation of areas of risk – to offset the inherent risk of change. In most cases the relative Argument involved has to be made on the basis of qualitative Evidence.

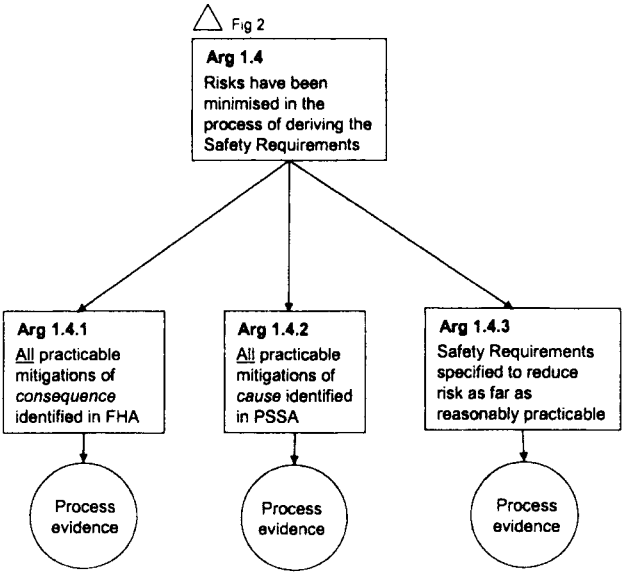


Figure A.5. Argument 1.4: Satisfaction of Qualitative Safety Targets

Arg 1.4.3 is intended to show that a (properly conducted) FHA and PSSA will yield safety requirements that, when implemented, will result in risk that has been reduced as far as reasonably practicable, at that stage[8].

As with most *Backing* Evidence, **St006**, in Figure A.6 below, is based on arguing the adequacy of the processes (including techniques and tools) involved and on the competence of the personnel who executed those processes. In practice, some of the Arguments may need to be decomposed to a lower level of detail than shown in this example.

Figure A.7 below addresses the Implementation of Change SGxy, in two stages: physical-level design and realisation of the design in the physical system – these are further decomposed below in Figure A.8 and 9 respectively.

In this example, **Arg2** is decomposed only far enough to show the possible elements of the ATM system that might be involved.

For the Implementation of Airspace Design, ATC Procedures and Operational Training, most of the Evidence of compliance with the Safety Requirements comes at the Design stage – ie under **Arg2.1**.

[8] The reduction of risk as far as reasonably practicable is covered further, and probably more effectively, in **Arg3** below

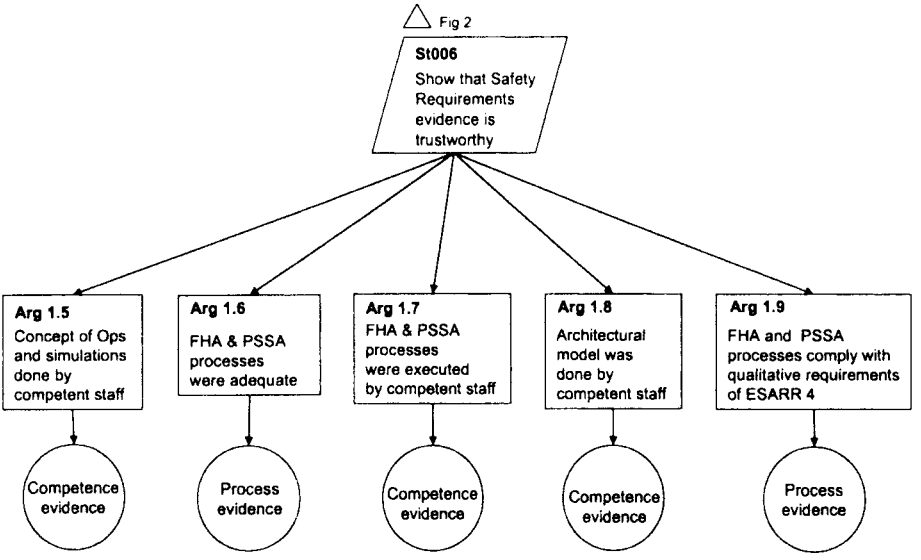


Figure A.6. St006: Safety of the Concept (Backing)

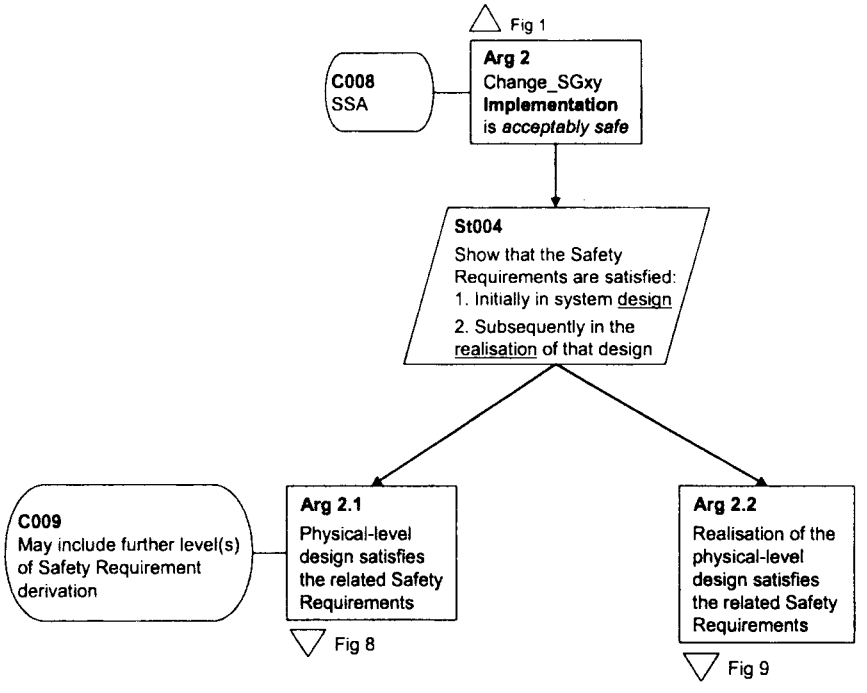


Figure A.7. Arg2: Safety of the Implementation

For the Equipment Implementation aspects, the *Evidence* of compliance with the Safety Requirements should also come from the Design stage – ie under **Arg2.1** – but should be further substantially supported by testing in the subsequent Realisation stage – ie under **Arg2.2**.

The decomposition of **Arg2.1** would need to include *Backing* assurance covering the adequacy of the processes, tools and techniques employed in the design and realisation, and of the competence of the personnel involved. Full use should be made of existing operational and engineering development procedures in the organisation's quality and safety management systems.

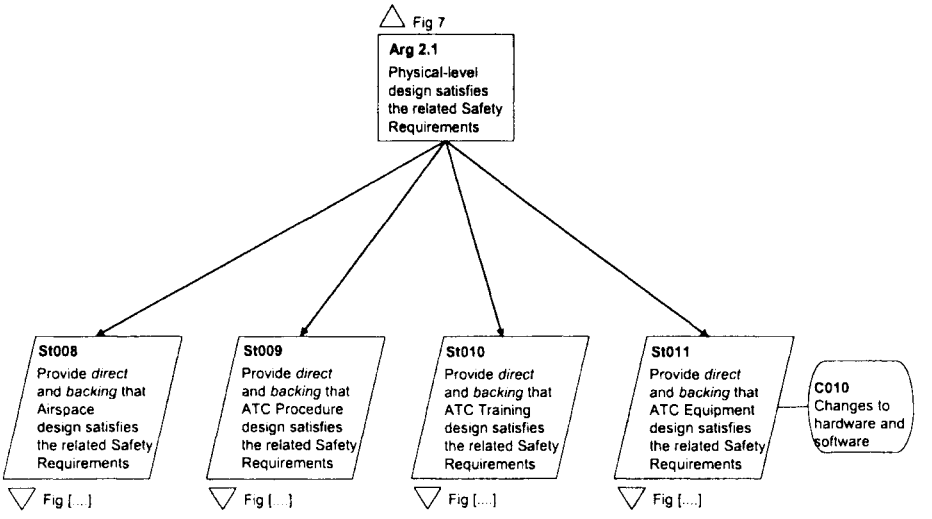


Figure A.8. Arg2.1: Safety of Design

The decomposition of **Arg2.3** (Realisation of Design) mirrors that for Arg2.1 and is shown in Figure A.9 below. In the case of the equipment aspects of Realisation, most of the *Evidence* will come from analysis and testing. The *Backing* for this is not decomposed herein but should address the V&V requirements covered in section 4.6 of the main body of the paper.

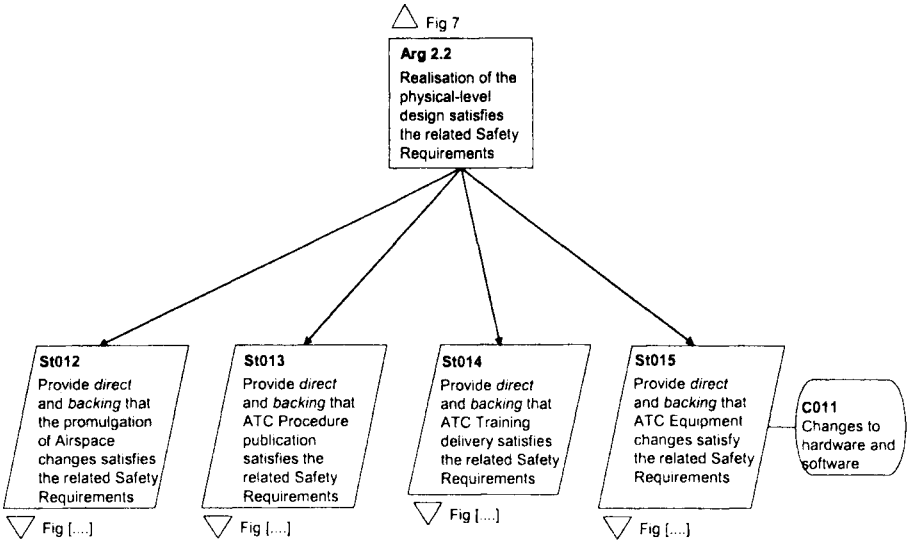


Figure A.9. Arg2.2: Safety of Realisation

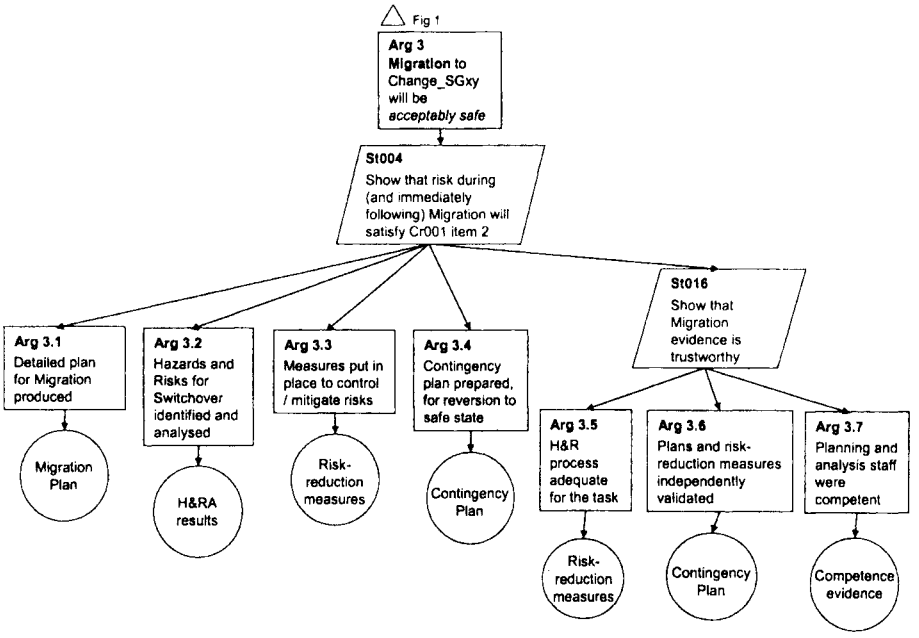


Figure A.10. Arg3: Safety During Migration

Clearly, in introducing a major change (or new system) the safety of the existing ATM service must be preserved during the period of Migration from the pre-change to post-change state.

Figure A.10 shows a typical decomposition of the *Argument*, with supporting *Evidence*, covering both the *Direct* and *Backing* aspects.

Arg 4 in effect recognises that Evidence provided under **Arg1** to **Arg3** is necessarily predictive in nature and needs to be confirmed by Evidence of what is actually achieved in practice, from a safety perspective. This is illustrated in the decomposition of **Arg 4**, in Figure A.11.

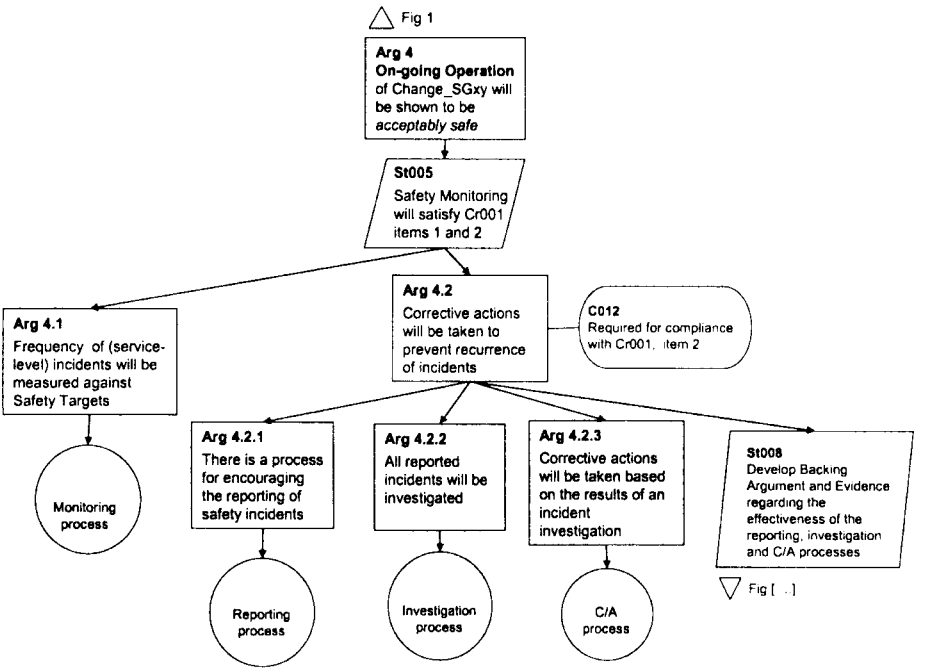


Figure A.11. Arg4: Safety Monitoring

A.2 Example Application – a Unit Safety Case

Unit Safety Case is a commonly used term for the Safety Case for an on-going operational service. Figure A.12 below shows the high-level Safety Argument for this example application of GSN, for a hypothetical Air Traffic Services Unit (ATSU).

Arg 0 is the overall Claim, equivalent to that for Change "SGxy" in Figure A.1 above. **C001** defines the type(s) of service provided and **C002** is a reminder that the

full operational environment – eg airspace boundaries, structure, classification, rules, aircraft-separation minima etc – needs to be fully described in order to define the Context in which the Claim is being made.

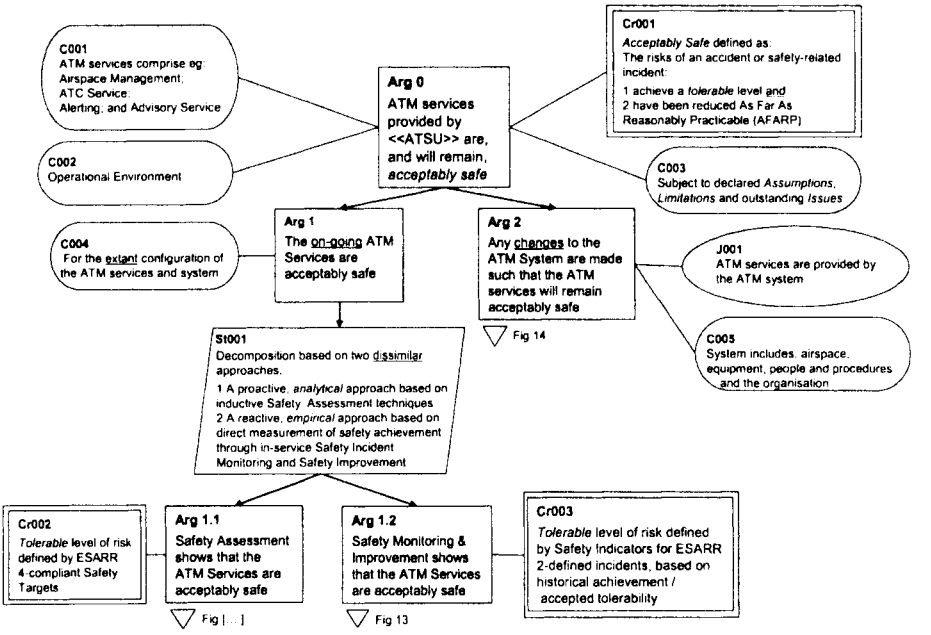


Figure A.12. Arg0: Overall Safety Argument for a Unit Safety Case

C003 is a reminder that the eventual conclusion of the Safety Case will probably be subject to certain Assumptions and outstanding Issues that need to be addressed and possibly to some Limitations on the ATM service(s).

The definition of what is acceptably safe is captured in **Cr001**,– note that item 1 (as elaborated in **Cr002** and **Cr003**) is an absolute measure, as is appropriate to an on-going service.

The Claim (Arg 0) is decomposed into two principal Safety Arguments (**Arg 1** and **Arg 2**) that, in effect, the services are safe "today" (ie for the current system baseline – **C004** refers) and will remain so because any changes to the baseline will be managed so as to maintain the safety of the services.

The decomposition of **Arg 1** is very similar to that for "Change SGxy" but, generally, on a much larger scale; in other words, this part of the Unit Safety Case (although not related to change) treats the Unit as a large ATM system for which:

- Safety Requirements (for the system) are derived and satisfied in a predictive Safety Assessment (**Arg 1.1**);
- actual safety achievement is monitored and improved through empirical

Safety Monitoring (**Arg 1.2**).

Arg 1.1 is not decomposed further herein but should follow a pattern similar to the equivalent Argument for Change "SGxy" except that the underlying safety assessment activities should be carried out for the ATSU as a whole.

The decomposition of **Arg 1.2** , shown in Figure A.13 below, is the equivalent to that for Arg 4 for "Change SGxy" shown in Figure A.11 above, except that the context for the former is the "present" time, rather than the future.

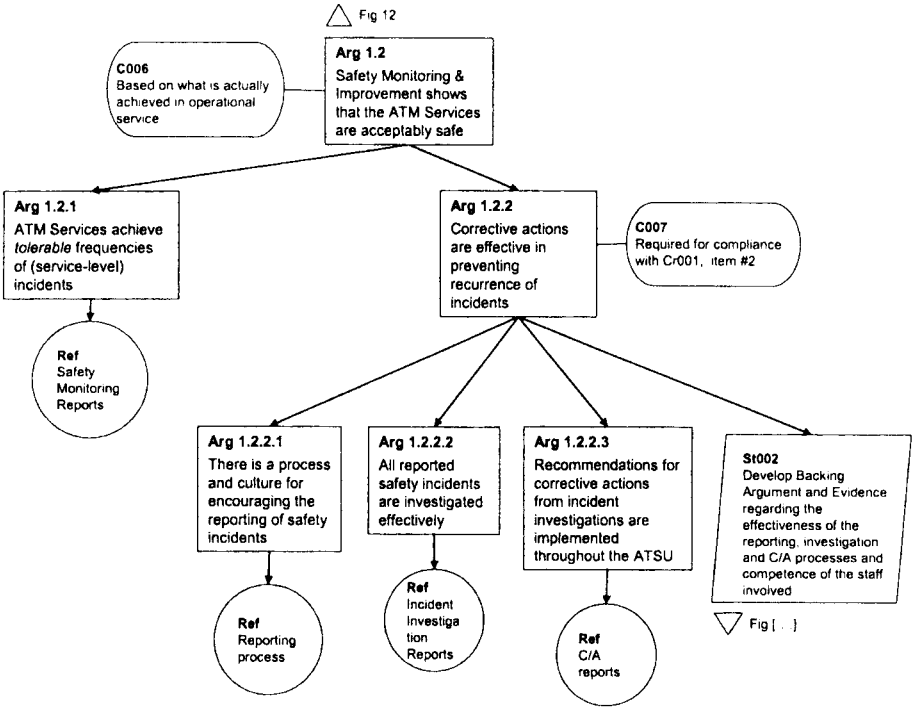


Figure A.13. Safety Monitoring and Improvement

For most *Unit* Safety Cases the system baseline is not fixed but is updated periodically by *Project* Safety Cases produced for significant changes – eg Change SGxy above.

Arg 2, decomposed in part in Figure A.14 below, is concerned with showing that all the necessary processes are in place (and are properly executed) to ensure that such changes are managed safely in terms of the on-going service – both during the period of introducing the change ("Migration") and in the subsequent in-service period.

Note that this is one of the few situations in which processes are used as *Direct* Evidence. Adherence to those same processes would be used as *Backing* Evidence

in the related Project Safety Cases.

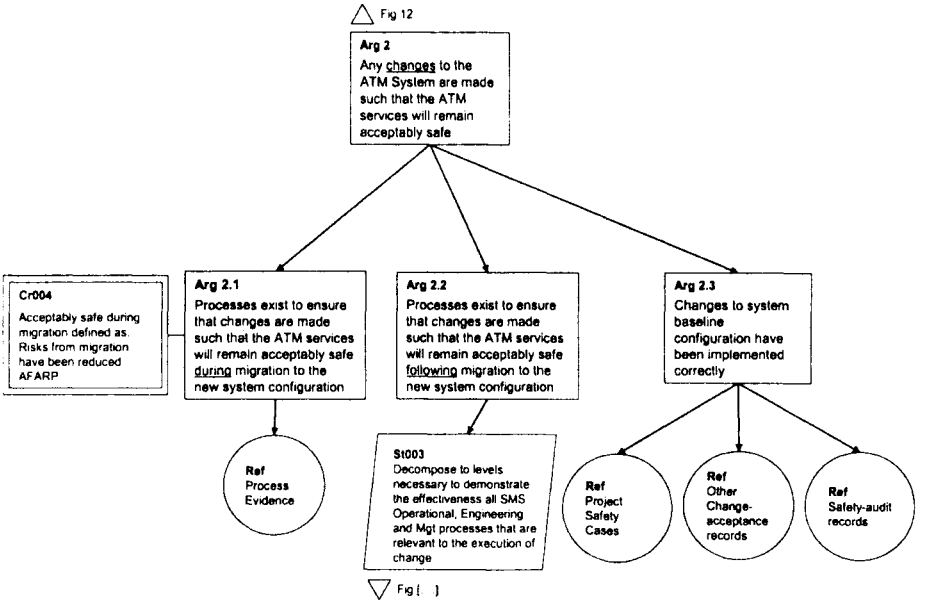


Figure A.14. Arg2; Change Management

MANAGEMENT INFLUENCE ON SAFETY

Governing Safety Management

Andrew Vickers
Praxis High Integrity Systems
Bath, England

Abstract

It is a requirement of UK law that organisations discharge the Heath and Safety at Work etc Act 1974. Duties within the act include ensuring that as far as is reasonably practicable an organisation's undertakings do not expose persons to unacceptable risk. The Act requires an organisation to declare intent in this regard through a company level policy that must be bought off by the organisation's senior management. In order for senior management to be comfortable that they are in fact discharging their duties with regards to the Act the effectiveness of systems that implement this policy must be measured and corrective action taken where deficiencies are identified. The behaviour of senior management should therefore influence an organisation's approach to safety management. This influence is both in terms of policy setting and in terms of measurement, or governance, as the business responds to changing context. In this paper the author identifies a number of principles that can be used by senior management to influence how safety management is carried out with a view to supporting the implementation of the Heath and Safety at Work etc Act 1974. The principles have been drawn from direct experience of assessing industrial systems for engineering governance and of identifying common vulnerabilities that organisations face.

1 Introduction

1.1 Background

It is a requirement of UK law that organisations discharge the Heath and Safety at Work etc Act 1974 (H&S@W etc Act 1974). Duties within the act include ensuring that as far as is reasonably practicable an organisation's undertakings do not expose persons to unacceptable risk. The Act requires an organisation to declare intent in this regard through a company level policy that must be bought off by the organisation's senior management. In order for senior management to be

comfortable that they are in fact discharging their duties with regards to the Act the effectiveness of systems that implement this policy must be measured and corrective action taken where deficiencies are identified. The behaviour of senior management must therefore influence an organisation's approach to safety management both in terms of policy setting and in terms of measurement, or governance.

Praxis HIS (and its recently merged sister company Aspect Assessment) is a specialist provider of engineering services to the critical systems industry. Over the last two years the author has been engaged in a number of assignments related to the governance of safety management for complex safety critical systems, as well as the governance of his own organisation – firstly Aspect Assessment and latterly Praxis HIS. A number of these case studies are outlined within this paper to provide the context for the presentation of a number of principles that support the effective governance of safety management.

1.2 Purpose

The purpose of this paper is to present a number of governance principles that can be adopted by senior management responsible for discharging certain duties within the Health and Safety at Work Act etc 1974. These governance principles specifically relate to discharging the duties for ensuring that as far as is reasonably practicable an organisation's undertakings do not expose persons to unacceptable risk.

1.3 Scope

The principles presented in this paper are drawn from experience gained from working with a particular class of organisation. The following are indicative characteristics of the organisations that have been the basis for the case studies.

- Large multi-million pound (and in two cases multi-billion pound) businesses.
- Developers or operators of complex safety critical platforms.
- Run by responsible senior management with a positive approach to discharging safety duties.
- Users of complex systems engineering processes, operating over significant periods (minimum 10 years), having to deal with change and complex regulatory environments.
- Owners of safety policies, engineering frameworks, etc.

The author makes no observation on the applicability of the derived principles to other classes of organisation. However, if anything, the challenges presented in this paper (if they exist at all for the simpler organisations) are likely to be much easier to solve in simpler smaller organisations and the principles are likely to be more easily implementable.

1.4 Structure

This Section provides the background, purpose, scope and structure of the paper. Section 2 presents a definition of governance and identifies the distinguishing features between governance and management. Section 3 sets a context for the four case studies that provide the source background to Sections 4 thru 7 which present the individual case studies. Section 8 draws together the lessons from the case studies into a shorter set of vulnerabilities. Section 9 presents some proposed principles for dealing with the vulnerabilities before Section 10 makes some comments on the use of requirements engineering to support the implementation of these principles. Section 11 concludes.

2 Governance versus Management

It is conventional to make use of systems of work to implement some business activity reliably, repeatedly, and systematically. A management system is usually needed to control significant business activities. It is likely that if that management system is absent, then the underlying business activity will quickly cease to function.

Governance is different to management. The Institute of Directors defines corporate governance (IoD 2004) as the "...rigorous supervision of management". The absence of a governance system will not necessarily cause the supervised management system to stop working, nor necessarily to stop the underlying business activities from occurring. What will happen when there is an absence of a governance system is that the underlying business activity may drift away from that which is most desirable for the business. The business will also find it harder to adapt to changing circumstance or to control its overall risk.

Activities are governed therefore in order to manage business level issues such as overall progress, deployment of scare resource, and different types of risk. This is true for all business functions and particularly so for safety management where the absence of effective systems can have significant consequences. It is not practicable for most organisations to govern everything, it is therefore necessary to define governance controls and to turn them on and off according to business circumstance. This paper explores some of the issues surrounding corporate governance of safety.

3 Introduction to the Case Studies

In order to arrive at the views presented in this paper four significant case studies have been drawn upon. Each of the case studies is presented anonymously. Client identity is withheld due to the sensitivity of the work. Such an approach does not invalidate the material presented in this paper as the general principles presented reflect a view that is independent of the particular case studies, and it is these

principles that are the key contribution within the paper. The generic background to the case studies does however provide a basic context and is worth setting down.

Each of the case studies has been drawn from work carried out at the organisational level, as distinct from the project level. Each piece of work was motivated by senior management concern over liabilities or risk to which the particular organisation may have been exposed and a positive proactive desire to instigate remedial action where deficiencies were identified. Each case study involved some form of multi-perspective review, analysis, discussion of results, and remediation programme.

The following sections present each study, commencing with some short background to provide a context for the study before the key challenges that inhibit clear governance of safety management are presented. It is these challenges that have led to the identification of the governance principles presented at the end of the paper.

4 Case Study 1: Management Vulnerability

4.1 Background

Case Study 1 concerned a large established organisation with capability to design and manufacture a large variety of safety-critical systems and platforms (systems of systems). The Engineering Director of the organisation required confidence that the organisation was discharging its health and safety liabilities. In many cases the organisation had mature practices, sometimes certified, but the question remained as to whether these were the right practices and the right certifications. In order to inform this view, a broad-spectrum review of the business was carried out including design, manufacture, personnel, and facilities management.

4.2 Challenge – Organisational Requirements

Across the organisation there were no underlying documented management system requirements for safety that linked in detail particular safety systems to the high level safety policy. It was therefore not possible to determine why this particular set of systems was implemented as against any other, nor what the relationships and handovers were across peer safety management systems. A consequence of this was that it was not readily possible to determine completeness through audit of these systems against the actual demands of the business.

From a governance perspective, the implications were that the Engineering Director was unable to determine how corporate safety risk was managed by the cooperating business units from top to bottom, nor how this particular coordination of cooperating business units discharged the overall corporate requirement. This in turn meant that coverage was difficult to confirm which tended to manifest itself in issues of safety coordination across business units. Thus the Engineering Director was unable to demonstrate that the safety function was being governed – despite the presence of very many mature practices at the 'shop floor'.

The consequences of an inability to demonstrate appropriate governance were made worse when the organisation wanted to change safety systems, change its internal structure, or bid for work in an innovative manner.

- With regards to technical change, it was not always possible to determine which systems should be invoked and whether any gaps might be introduced.
- With regards to organisational change, it was not easily possible to determine how safety responsibilities were affected by business area changes.
- With regards to bidding for work with innovative solutions, it was not possible to be clear which safety management practices were essential (no matter what the commercial context) and which could be replaced by alternative means.

5 Case Study 2: Operational Infrastructure

5.1 Background

Case Study 2 concerned a large and complex part of the national infrastructure. The infrastructure was supported by a multi-disciplinary developmental safety case which itself was the subject of a complex regulatory regime. Traditionally the organisation had relied on the use of the developmental safety case to manage the operational system. However senior management had determined that it was appropriate to consider how to migrate from a developmental safety case to an operational safety case as the basis for on-going safety management. The client required an independent view of how such a transition could be safely managed.

5.2 Challenge – Identification of Operational Indicators

The significant challenge related to the determination of the actual operational safety indicators. At the time of writing, the developmental safety case contained detailed analyses to cover a number of novel risks. After a number of years of operation, these novel risks had become better understood and simpler ways of representing and controlling the actual risk had been determined.

In practical terms, the reason for this was that often the most significant risks during development are controlled adequately through design and procedure design, leaving the balance of the residual risks elsewhere. This leads to a requirement for different indicators to be used for effective on-going safety management. However, the basic tools for understanding the on-going risk of the infrastructure were based upon the developmental safety case. The migration away from these more sophisticated developmental models was therefore a significant challenge for the organisation.

The challenge from a governance perspective is to ensure that the operational safety case has identified the correct safety indicators and that these are being monitored sufficiently. Such identification is unlikely to be straightforward at the point of introduction of the equipment (or in this case infrastructure). The governance controls must therefore be sufficiently flexible to support the transition into operation and react as necessary to the indicator data.

6 Case Study 3: Operation of Legacy Platforms

6.1 Background

The client was operating and maintaining a fleet of over one hundred legacy platforms of a variety of types. The oldest platforms had been designed fifty years ago. All the platforms had been subject to upgrades, ranging from simple trials fits through to major modifications. The operator was responsible for the safe provision of platforms to the users of those platforms and for coordinating safety information amongst a number of organisations who collectively provided the design authority for the platforms.

The client wanted an independent view on the overall approach to safety management. The client was particularly concerned about the challenges brought about by the legacy nature of the platforms, and of modifying platforms using current standards when the evidence available to judge the modifications was produced using earlier less mature standards. The client had an ongoing safety management improvement programme.

A number of specific areas for investigation were identified.

- Relationship with subcontractors
- Safety Management System
- Safety Management transition
- Safety decision making – both tactical and strategic
- Competence management

6.2 Challenge – Demonstrating Legacy Safety

The key challenge concerned the manner in which safety could be demonstrated. Essentially this stemmed from the legacy nature of the platforms. Due to the age of the platform, safety evidence of the type expected today was not available. This meant that safety decisions were always being made on a case-by-case basis, involving the examination of evidence as it could be determined – rather than by recourse to some underlying whole platform analysis. The consequence of this is that whilst individual safety decisions could be assured as sound, there was a risk that whole fleet safety margins were being eroded.

A related issue concerned more significant upgrades to subsystems containing programmable systems where the gap in past and current standards was perhaps at its greatest. Surprisingly here the challenge was to have the confidence of avoiding doing too much. Where subsystems were being upgraded, there was a temptation to apply current standards in their entirety. Such an approach would have cost far more than an approach based upon careful application of the new standard in combination with evidence drawn from the significant operational experience that was available.

The final aspect of the challenge related to the complicated nature of the design authority for the platforms. The concept of design authority is important because it provides a central point for discussions on how safety issues are to be resolved.

The challenge with a long-standing platform is that the competence of the original equipment manufacturer can have eroded and other organisations (eg other subsystem maintainers) can have begun to play at least an implicit part in the actual design authority for the platform.
The challenge from a governance perspective was to be confident that the underlying safety requirements were being met through a safety argument based on available data, much of which may be 'non-standard' legacy data and some of which may be based on updated thinking and experience of use. The key issues being coverage/completeness and relevance of legacy data.

7 Case Study 4: Procurement

7.1 Background

The final case study concerned an organisation that carried out significant procurement of safety critical systems. The procurement typically involved multi-disciplinary systems. The client was concerned with what were perceived as escalating costs of safety management. In particular, the client wished to understand whether projects were becoming more risk averse, or whether more funding was required to drive down unacceptable safety risk.

7.2 Challenge – Competent Safety Decision Making

The key challenge related to competent safety decision making. In particular the challenge of ensuring that the correct competencies are available at the right stage of the life-cycle. At the end of the design life-cycle, decisions surrounding safety are typically much easier to make than at the beginning of the life-cycle. The product is well-understood, the safety challenges are well-understood, and the correct decision is usually brought into sharp focus if for no other reason than time is short. The challenge is to bring this thinking into the earlier parts of the life-cycle when the product is not so well understood, the safety concerns are not so well understood, and the time/cost imperative is not as strong.

From a management perspective, the challenge is one of clearly understanding risks early on in the feasibility and requirements phase and of then setting sensible risk targets for monitoring against during development. If appropriate targets can be set, development and acceptance can be against a clear set of criteria. From a governance perspective, the challenge is to monitor and ensure that the correct competencies are actually being deployed across the phases and to be able to demonstrate that this is so.

8 Safety Management Vulnerabilities

All of the organisations that were assessed as part of these case studies were mature organisations used to working with safety critical systems. Many had external accreditations and all had well-established procedures, safety policies, management systems, etc. and strong safety cultures. Despite this, each organisation was faced with a number of vulnerabilities that left the organisations facing increased exposure and risk. Table 1 summarises some of the manifestations of the vulnerabilities discussed.

Area of Vulnerability	How Vulnerability Manifested
Organisational requirements	• Inability to easily demonstrate to a third party that safety is being managed effectively across the organisation from top to bottom • Difficulty in changing aspects of the overall safety management system • Difficulty in auditing or demonstrating completeness of the organisation's overall approach to safety management • Difficulty in identifying corporate safety responsibilities when bidding with partners or to suppliers in an innovative manner
Identification of operational indicators	• Lack of explicit consideration of the issues within an operational safety case, as compared to a developmental safety case • A non-adaptive governance system that is unable to support the transition from novel introduction to mature practice
Demonstrating legacy safety	• Difficulty in knowing when to stop safety reduction • Difficulty in co-ordinating safety management across multiple organisations • Difficulty in knowing when not to apply modern safety standards
Competent safety decision making	• Difficulty in knowing which competencies to make available at which life-cycle stage to support safety decision making • Difficulty in knowing when to stop safety reduction • A non-adaptive governance system that is unable to control the changing role of decision making through the procurement life-cycle

Table 1. Manifestations of Vulnerabilities

It is interesting to note that these vulnerabilities all arose despite the various positive characteristics of the organisations involved. The conclusion that the

author has drawn from carrying out each of these studies is that the safety management policies put in place had been appropriate for the organisations at the point of definition, but that the organisations' circumstance had changed over time and the safety management policies and systems had not kept step. This indicates the importance of the supervision of these systems and of adapting these systems in response to the changing nature of the business – or more simply, of the importance of corporate governance of safety management.

In this next section, a number of principles of governance for safety management are proposed.

9 Principles of Safety Management Governance

In this Section, five governance principles are presented that together provide a framework for addressing the vulnerabilities identified above. The implementation of these principles provides a tool for senior management to influence how safety management is carried out within an organisation. Senior management must be able to influence the way in which the business implements its safety policies and it can do this through governance.

Principle 1: You can't govern everything.

Senior management has insufficient resource to govern everything. In addition, not all aspects of a business are equally critical at all times. This implies that governance controls should be turned on and off, and that such decisions should be active and deliberate depending on current views of criticality. Areas of criticality depend on business circumstance and what the business is trying to achieve at any particular time.

Principle 2: Set corporate objectives for safety.

A business needs to be trying to achieve something. In terms of safety management this may often simply be of demonstrating effective coverage and of maintaining the current status quo – assuming the current status quo is demonstrably satisfactory. However other objectives are possible, for instance objectives may be concerned with reducing costs, increasing safety margins, reducing incidents, etc. Whatever the objective is, it needs to be one that can be measured.

Principle 3: Set key performance indicators for each corporate objective.

The identification of key performance indicators for a corporate objective is likely to be challenging. Indicators can be quantitative, for instance in the case of reducing the number of incidents, but in other cases they may be qualitative, for instance in demonstrating regulatory compliance. The vulnerabilities presented in Section 8 provide an indication of the range of issues that could be monitored –

depending upon the objective of interest. Whatever the type, they must be chosen in order to provide a clear indication of how well a corporate objective is being met.

Principle 4: Ensure systems exist to provide key performance indicator data.

In order to gather the key performance indicator data, it is necessary to ensure that systems are in place to generate that data. Some form of gap analysis may be useful for complex large scale organisations and a change programme may be required to provide the remediation. However at the completion of this activity, there should be a static top-down governance system that links business objective through the means for testing progress against that objective into the systems that actually enable the business to function. Such a governance system must be exercised however and checked for ongoing sufficiency.

Principle 5: Regularly review key performance indicator data and systems.

It is this final principle that completes the circle, and that is the regular review of the data and the on-going assessment of the efficiency of the underlying systems.

The principles have been derived from the vulnerabilities as follows in Table 2.

Vulnerabilities / Principles	Organisational Requirements	Identification of Operational Indicators	Demonstrating Legacy Safety	Competent Safety Decision Making
You can't govern everything	Be clear on what the organisation has to do, so you can choose what to govern	Choose what needs to be monitored to ensure operational safety margins are maintained		
Set corporate objectives		Be clear on how good you need to be for the phase that your projects or organisation is in		
Set indicators		Know how effective you are being		
Ensure systems exist	Management systems must exist for everything that you want to do			
Regularly review	Ensure that the organisation is still correctly set up to manage safety in this context	Keep checking to see if the safety decision making is being as effective as you need it to be, given the context of the business		

Table 2. Mapping Principles to Vulnerabilities

10 Implementation –Requirements Engineering

It is the distinction between business objectives, governance requirements, and the measurement of management systems that is a key enabler to implementing the principles proposed in this paper. In each of the studies outlined above the author

has made use of requirements engineering principles to drive the implementation of either the assessments, the analysis, or the remediation.

The use of requirements engineering, an approach founded in product systems engineering, is relevant because the governance of safety management is in essence a problem like any other systems problem. Issues of context, underlying need, and means of meeting the need are common to both governance and product systems engineering. With products, the issues are perhaps more concrete, whereas with governance the problems relate to information provision, but the principles of problem solving are common.

The approach used by the author is that embodied within the REVEAL approach to requirements engineering (an example of its application is detailed in (Hammond, Rawlings, Hall 2001)). REVEAL is based upon a set of principles proposed by Michael Jackson (Jackson 95). Key building blocks in this work are:

- Domain Statements: Properties of the application domain that must be relied upon for the System to bring about the desired change, eg the various companies that together form a platform design authority
- Requirements: The desired change in the real world, independent of the system that will help bring about that change, eg safety margins on a legacy platform are maintained
- Specification: An interface-level definition of the system, eg the key features of the safety management system that together coordinate safety across a distributed organisation.

It is a combination of the specification with the domain that will bring about the requirements. Such an approach is particularly relevant given the importance that has been stressed in this paper of the need for governance controls to ensure that the approach to safety management adapts to the changing context of the business. Requirements engineering can therefore provide a framework for mitigation and control of risks in this area.

11 Conclusions

Large complex organisations face real risk around discharging safety responsibilities and in particular in demonstrably discharging the Health and Safety at Work Act etc. It is the nature of these types of organisations, and the environments within which they operate, that make the simply expressed duties of the Act challenging. The discharge of such responsibilities is usually carried out by the setting of policies, introduction of systems, rollout of training etc. These systems will typically be set in response to the state and context of the business at that time.

The introduction of safety policies, management systems etc. is a necessary but insufficient means of discharging these responsibilities. The challenge in ongoing safety management is that the context of the business changes – corporate risks change, products change, business models change, - and so whilst the systems may

have been appropriate at the time of introduction, their relevance can be eroded over time thus exposing an organisation to risk.

Areas where this erosion can manifest include the following:

- Over-engineering, and therefore increased cost, brought about by inappropriate application of standards that have evolved.
- Inability to provide a top-to-bottom picture of how safety is managed across a large complex organisation.
- Overlaps in safety management, and therefore increased cost, brought about by changes in a supply chain or organisational set-up.
- Gaps in safety management, and therefore increased risk, brought about by changes in a supply chain or organisational set-up.

A key mitigation to this risk is effective governance of safety management. Governance is the tool because it is at the level of governance, and of senior management, that the various contextual issues of the business are brought together. Senior management therefore play a key role, through governance, in ensuring that an organisation's safety management systems remain current and cost-effective. It is the distinction between business objectives, governance requirements, and management systems that is a key enabler.

12 Acknowledgements

The author would like to acknowledge the support of Praxis High Integrity Systems for the provision of time to write this paper.

Colleagues who have worked with the author on assignments to help solve client problems in the area of safety management and who have therefore helped to formulate the author's thinking in this area include: Keith Williams, David Bradley, Matt Barron, Richard Adams, Samantha Lautieri, Trevor Cockram, Ian Spalding, David Dickerson, and Jason Glew. The author is grateful to John Harvey for providing review comments on earlier drafts of this paper and for Felix Redmill for suggesting that such a paper may be of interest.

The author is also grateful to the anonymous clients who have presented such challenging problems.

This paper is in part based upon material presented by the author at the 2005 IEE/BCS Joint Working Group Conference on Independent Safety Assessment (Vickers 2005).

References

Hammond J A R, Rawlings R, Hall J A (2001). Will it Work?, RE'01, in the Proceedings of the 5th IEEE International Symposium on Requirements Engineering, August 2001

H&S@W etc Act (1974). Health and Safety at Work etc Act, Her Majesty's Stationary Office 1974, Statutes in Force, ISBN 0105437743

IoD (2004). Corporate Governance, practical advice for directors on today's most important boardroom issue, A Director's Guide, Institute of Directors

Jackson (1995). The World and the Machine, Michael Jackson, ICSE'95, in IEEE Proceedings of the 15th International Conference on Software Engineering.

Vickers (2005). Assessing the Corporate Governance of Safety Management, IEE/BCS Joint Working Group Conference on Independent Safety Assessment

Understanding the Risks Posed by Management

Felix Redmill
Redmill Consultancy
22 @ N10 3JU
UK
Felix.Redmill@ncl.ac.uk

Abstract

The risks posed by management are neither addressed by risk analysis nor included in safety cases. Yet they have been shown to be significant contributors to accidents. This paper argues for more attention to be paid to them and for the development of a risk-analysis method to address them. The paper examines the aspects of management risk that it might cover and offers a set of proposals for its design.

1 Introduction

Traditionally, risk analyses have addressed equipment failure, using processes and techniques derived from reliability theory. More recently, it has been recognised that the human components of systems also contribute, sometimes substantially, to functional risks, and an increasing number of analysts now attempt to address, to some degree, the hazards introduced by human operators. However, authoritative guidance has not kept up with awareness, and there is a lack of information on how to include human factors in risk analyses. For example, the meta-standard, IEC 61508 (IEC 2000), devotes lengthy parts (2 and 3) to the ways in which hardware and software (respectively) should be addressed, but offers no equivalent advice on analysing the risks posed by humans. The safety-critical systems industry now requires greater involvement of engineers in human factors issues, a determined focus on the dissemination of knowledge and experience in the field, and the development of guidelines on the inclusion of human factors in risk analyses (Redmill 2002).

Lagging even further behind is any attempt to address the risks posed by management, particularly senior management. Yet, judging by the results of numerous inquiries into major accidents, such risks can, in many cases, outweigh those thrown up by the failure of system components. The policies and strategies defined by senior management, the decisions that they make, and the cultures created by them, by design or default,

predispose accidents to occur or not to occur. When the predisposition is to accident, the final triggering event is relegated merely to the activation of 'an accident waiting to happen'.

In her examination of the origins of the 1986 *Challenger* space shuttle disaster, in which seven astronauts died, Vaughan (1996) points to mistake and disaster being 'socially organised and systematically produced by social structures' – due to management's acquiescence or negligence. She says that the cause of the disaster was 'a mistake embedded in the banality of organisational life' and she shows how 'deviance in organisations is transformed into acceptable behaviour'.

In his investigation of the same incident, Feynman (1989) found that engineers at NASA (National Aeronautics and Space Administration) considered the chance of a shuttle failure to be about 1 in 200 launches and, at best, 1 in 1000. But he found that NASA management took the figure to be 1 in 100,000 launches – which, Feynman pointed out, would mean that a shuttle could be launched every day with an average of almost 300 years between accidents. Historical data showed the engineers' estimate to be accurate, but organisational decision-making was implicitly carried out on the basis of management's estimate.

Risk analysts expend effort, often at considerable expense, to determine the likelihood of the final triggers of hazardous events. They address equipment failure and sometimes operator error; they address the hazards arising from unintended interactions of system components, even when no failure occurs; but they do not address the failure of management systems and, in general, the influence of management on functional risk. With respect to safety, the resulting analyses must be optimistic.

For safety cases truly to demonstrate the achieved safety of a system, they must cover all relevant risks. It is therefore time for them to include management risks. But this is not a trivial requirement. First, the junior- and middle-level staff who carry out risk analyses are, typically, not experienced in the higher-level issues, such as company policy, strategic plans, management style, organisational culture, and safety management systems, and are therefore not competent to analyse them for the risks that they might pose. Second, there is no generally accepted process of modelling and estimating such risks. Who should conduct the risk analyses, and how? Research is required.

The purposes of this paper are to raise awareness of the need and to propose a method for including management risks in risk analyses and safety cases. Section 2 briefly examines what might be done about management risks, Section 3 considers management and its risks from different perspectives, Section 4 makes proposals for a method of analysing the risks posed by management, and Section 5 offers a discussion of the proposals.

2 Options in Addressing Management Risk

Given that the risks posed by management can have significant effects on the functional safety of systems that are developed, operated, or disposed

of by or within a company, and that they are currently not included in risk analyses, what might be done about them? Four possibilities are considered.

2.1 Do Nothing

Bringing management risks within the ambit of risk analyses is likely to be a difficult and even controversial business, so the easy route would be not to 'rock the boat'. However, given that the purpose of this paper is to challenge the status quo, this is not an option to be examined further.

2.2 Improve Management's Risk Awareness

One option is to focus on reducing risk rather than analysing it. And one way of doing this is by improving management's awareness of safety risk, and of risk issues in general – with the added exhortation to manage the risks. Certainly, the raising of awareness is an essential starting point, no matter what is to follow. Happily, this step has already been taken, and in a manner that is visible to most companies in the UK.

In its guidance on Internal Control (ICAEW 1999), the Institute of Chartered Accountants in England and Wales requires boards of company directors to identify and analyse 'the significant risks faced by the company' and to 'disclose that there is an ongoing process for identifying, evaluating and managing' them. The Institute also invites directors to provide information in their annual reports 'to assist understanding of the company's risk management processes and system of internal control'. Thus, boards of directors are enjoined not only to be aware of their risks, including safety risks, but also to analyse, understand and manage them, and, further, to demonstrate to shareholders and other interested parties that they are doing so effectively. The Institute also calls on companies' boards of directors to adopt 'a risk-based approach to establishing a sound system of internal control', which is a requirement for boardroom-led systems based on risk management. For those companies that develop, operate or dispose of systems, the functional risks posed by those systems are risks 'faced by the company' and require to be managed within the company's system of risk-based internal control.

Thus, boardroom management is already required to be aware of its significant risks. More than that, it is required to accept responsibility for managing them and for demonstrating that it is doing so effectively. However, compliance with even legal requirements cannot be guaranteed, and where it exists it is certain to be inconsistent across companies and industry sectors (Ramsay and Hoad 1997), so the Institute has published advice for directors (Jones and Sutherland 1999) on the processes necessary for meeting the requirements. Moreover, taking a risk-based approach at the top of a company, and ensuring that the same is done at all lower levels, requires not merely an awareness of what is required but also a change of culture in senior management (Elliott et al 2000). For the benefit of companies for which significant risks are the functional risks of their systems, the Health and Safety Commission has issued advice to directors,

urging them to include health and safety issues in their annual reports (HSE 2001).

Many companies have introduced risk-based systems of internal control (Page and Spira 2005), but it is not known to what extent a risk-based way of thinking has led managers to examine the risks that they themselves pose - in their policies, decisions, and the cultures implicit in their leadership. Thus, general awareness is not enough to ensure that one of the major sources of safety risk is understood and managed. Nor is it sufficient to appeal to phrases like 'significant risks facing the company', for managers new to the discipline of risk management are unlikely to recognise such risks as potentially arising from their own decisions, actions and negligence. Additionally, it is necessary to create a process of getting to grips with the risks that are of interest in the present context.

2.3 Focus on Improving Safety Culture

Another way of reducing safety risk, without carrying out risk analysis, is, at least in theory, by improving an organisation's safety culture. This is expressed by the attitude and behaviour of staff, and should be defined, developed and nurtured by management. If this is to be done systematically, according to a plan to develop a 'good' culture as well as a 'strong' one (Levene 1997), it must necessarily include the raising of management's awareness, as discussed in the previous sub-section. Thus, improving culture is taking a step beyond the mere raising of awareness.

There has been a great deal of research into the subject of safety culture, with literature reviews being carried out, for example, by Guldenmund (2000) and the Health & Safety Laboratory (2002). Both the terms 'safety culture' and 'safety climate' are used, and, while some authors make a point of distinguishing between them, others use them interchangeably (Health & Safety Laboratory 2002). Universal agreement on definitions is therefore lacking. Indeed, Guldenmund (2000) points out that, although safety culture and climate are generally acknowledged to be important concepts, not much consensus has been reached on their cause, content and consequences. He further states that there is a lack of models specifying the relationship of the two concepts either with safety and risk management or with safety performance.

On the assumption that good culture is a good thing, and a way of attempting to improve safety, industry as well as academe has invested in it. The nuclear industry was perhaps the first to address the issue of safety culture (International Nuclear Safety Advisory Group 1991), and the same industry has prepared practical guidelines for the development and maintenance of such a culture (International Nuclear Safety Advisory Group 2001). Guidelines with the same intent have been produced in other large safety-related industry sectors, such as the railways and off-shore oil and gas exploration, and, more generally, for the Health and Safety Executive (2002). There has also been an attempt to define the development of a 'safety culture maturity model' (The Keil Centre 2001).

Thus, there is already a continuing attempt to define, improve and measure safety culture. Yet, even with increased awareness and improved

safety culture, and even if these do lead to improved safety, how can the adequacy of safety, with respect to risks posed by management, be demonstrated? Pointing out that awareness is high and culture good is not sufficient. Completeness also requires the inclusion of such risks in risk analyses, which may then inform safety cases.

2.4 Include Management Risks in Risk Analyses

If management risks are to be demonstrated in a safety case to be tolerable, or to have been reduced to a tolerable level, they must be managed, and to be managed they must be understood. The accepted way of arriving at an understanding of risks is to identify the hazards that could give rise to them and to analyse those hazards so as to acquire the knowledge necessary for the required understanding. It is therefore necessary to subject the risks posed by managers to risk analysis. As already observed, this may be a difficult task. Yet, if it could be done, the results would provide a basis for a number of activities, including assessing tolerability, raising management's awareness of their own risks, determining where it would be useful to propose changes to management behaviour, and identifying appropriate points for inserting risk-reducing barriers. A method designed to address the analysis of the risks posed by management would, potentially, be an asset. The remainder of this paper presents proposals for the design of such a method.

3 Inquiry into Management

A necessary prerequisite to determining how to bring management risks within the scope of risk analysis is to decide what 'management' means. In order to develop a method of addressing risks, there must be an understanding of the types of risk to be dealt with and the nature of the field of exploration. This section identifies a number of perspectives on management risk and considers their implications for addressing risk.

3.1 Levels of Management

In general, three levels of management in an organisation may be assumed - senior, middle and junior.

Typically, juniors constitute the greatest number of managers, their responsibilities are operational, and their influence is local. In operation, failure of their control is likely to lead to a single incident - though, in manufacture, it could introduce a systematic fault into many systems.

Middle managers are fewer and the influence of their decisions and actions extends over the lower level as well as their own. They are charged with ensuring that business objectives are met, so the ways in which they do this can introduce systematic faults into the ways in which junior managers and staff function.

Senior managers are less constrained by protocol than middle and

junior managers, and their decision-making is more by judgement, and even intuition, than according to rules and procedures. Their decisions and actions have strategic importance and their influence encompasses not only their own but also both lower levels. They are responsible for defining an organisation's policies and for approving the strategies for meeting them. Importantly, they are responsible for providing the leadership that defines and nurtures culture, including safety culture. Thus, whether explicitly or implicitly, they define not only the organisation's objectives but also the ways in which the staff attempt to meet them.

Having defined three typical levels of management, it should be pointed out that the Institute of Chartered Accountants in England and Wales distinguishes between management and directors, saying, 'It is the role of management to implement board policies on risk and control. In fulfilling its responsibilities, management should identify and evaluate the risks faced by the company for consideration by the board and design, operate and monitor a suitable system of internal control which implements the policies adopted by the board.' (ICAEW 1999) A company's board is therefore a fourth level.

In seeking a method of analysing management risk, it would be easiest to limit the task to the junior management level. An obvious starting point is to attempt to include the junior manager within the boundary or the system that is the source of hazard, and to fashion a method from the risk-analysis techniques already in use, the developing field of human error modelling, and one or more human reliability assessment (HRA) methods.

The scope of middle management may be expected to extend beyond a system boundary, and identifying and analysing the hazards at this level is likely to require innovation beyond the mere application of existing methods.

The higher the management level, the more difficult it would be to identify and analyse the hazards, and the more likely that every identified hazard would, from some point of view, be perceived as having a safety-related outcome. Yet, the higher the level, the greater the influence of decisions and actions and, therefore, the more worthwhile it would be to study and understand the risks. At the senior and board levels, risks include those of not adequately defining and installing appropriate risk-management systems for analysing and assessing the organisation's significant risks.

Thus, in setting out to devise a method of analysing management risks, decisions must be taken as to where the focus - at least, the initial focus - should be directed. It is likely that, in creating a risk-analysis model, the assumptions that would need to be made at any one level would differ from those at any other. Care would be necessary in devising a method that is applicable at all levels.

3.2 Management Systems

The concept of a 'quality management system' is familiar. Such a system (for example, ISO 9000) defines the roles, responsibilities and procedures necessary for achieving quality in meeting an organisation's objectives.

Similarly, a safety management system may be defined for the achievement and maintenance of safety in an organisation's activities.

At the lowest level in the organisation, staff are, typically, expected to adhere rigidly to the system's procedures. The higher the level, the more discretion a manager is expected to use. Indeed, senior management is expected to put the system in place and middle management to ensure that it functions both efficiently and effectively. From the perspective of a safety management system, management failure can be seen to differ qualitatively from level to level in the organisation.

As with high-level policies, strategies and decisions, the contents of a management system have a predisposing effect on safety. The system is intended to impose constraints on acts that could be unsafe and to place barriers in causal chains that could lead to accidents, so failure to build them into the system, or to introduce checks to ensure that they are being observed, could lead to unsafe outcomes. Similarly, failure to police conformity with a system, particularly when its rigour might cause staff to employ violations, can have the same effect. Thus, instead of addressing the levels of management, from whatever perspective, another option is to consider the safety management system itself. If its function is to define the ways in which safety-related work is carried out, and the barriers that should ensure safety, a method may be devised to determine its correctness, adequacy, and operational integrity.

Although humans, including managers, are integral parts of management systems, senior managers should also be identified as existing outside of the systems - because they are responsible for defining them, putting them in place, and monitoring them. System failures may extend back to these senior managers.

3.3 Organisational Culture

A management system promotes safety and defines the route to it. But it is the culture of staff that determines whether or not the route is systematically taken. Methods of 'measuring' an organisation's safety climate or safety culture, based on questionnaires that test the attitudes of members of the organisation, have been developed (e.g. Cooper and Phillips 1994, The Keil Centre 2001). It could be possible to reflect the results of such measurements as levels of risk, and research could be conducted into ways of doing so. This, however, is not within the objectives of this paper and will not be discussed further.

3.4 Policies and Strategies

In some cases a policy or strategy may be implied, but it is more usual – indeed, in a safety- or quality-conscious company it is normal – for them to be defined and documented. Given this, it is possible, in theory at least, to subject a policy or strategy to risk analysis. One option, therefore, is to seek to devise a method to achieve this. It is likely that a method that is appropriate to analysing a safety management system would also be appropriate to analysing policies and strategies, and this will be explored

further in this paper.

3.5 Decision Making

A key feature of management, particularly at higher levels, is decision making. Behind every management action and instruction lies a decision, whether or not it is consciously taken. Some risks may lie in the decision making itself, for example when the decision maker's mental model of the problem to be resolved, or its environment, does not match reality and the decision leads to an unsafe outcome. Others may arise from the translation of decisions into actions or instructions, or in the misinterpretation of instructions by subordinates. As managers set the scene for safe or unsafe actions with their decisions and resulting instructions, it would seem that an analysis of management risk should include the decision-making.

Yet, the number of decisions that a manager makes is necessarily huge, and each one could lead to a variety of outcomes, many of them not easily foreseeable. It is therefore not apparent how all management decisions could cost-effectively (or even usefully) be subjected to risk analysis. However, whether or not decisions are analysed, it may be possible for a management system to include processes that cause the introduction of barriers that would prevent certain undesirable outcomes to result from management decisions. And it may be possible to create a method that a manager could use to subject selected important decisions to risk analysis. This possibility will be explored below.

4 An Initial Proposal for a Method

The various aspects of management risk discussed above are different in kind. Taking the perspective of any one of them, it is not immediately clear that a single risk-analysis method would embrace them all. It is therefore worth starting from a different point, that of the need for a tangible representation of the 'system' to be studied. Thorough and methodical risk-analysis must be carried out on a model of the object of study. This section commences by addressing the need for an appropriate representational model and continues by considering other aspects of an intended risk-analysis method that is appropriate to the risks posed by management.

4.1 A Representation of Management

A common feature of management activities is that they may be defined in terms of a set of processes. This is clearly the case for a safety (or quality) management system. Thus, an initial attempt to create a method could usefully be aimed at the risk analysis of such a system, using a process model.

A process translates one or more inputs into an output. To do so, it employs resources, including humans, and it relies on assumptions about its external environment. All of these features can be included in a simple

model, which may be created from block diagrams, flow charts, the unified modelling language (UML), or a number of other representations. Figure 1 shows a simple process in which three activities, A, B and C, acting sequentially, transform an input into an output. The boundary of the process is defined (around the three activities); the input, as well as resources and data, are derived from outside of the boundary, across which the output is transferred.

A policy or strategy is usually expressed textually, as an intention or instruction, but its implications, including the way in which it would be applied and its likely or potential consequences, may be determined and laid out as a set of processes. Similarly, a significant decision and its implications may also be laid out as a process, though with a greater variety of possible paths and perhaps with less certainty.

A method designed for the analysis of processes would not depend on the pre-existence of a suitable representation of the management issue under consideration, for an expression in terms of a process could be created for the purpose of analysis.

Representing the processes defined or implied in the creation and implementation of management systems, decisions, plans, policies, and proposals for change would not only allow their analysis for safety and other risks in advance, but also allow management to assess the mechanisms and effects of their implementation. It would indicate where improvement is necessary and offer guidance to auditors on where to concentrate their efforts most effectively.

Thus, an initial proposal is to represent management as a set of processes and to design a method for their risk analysis.

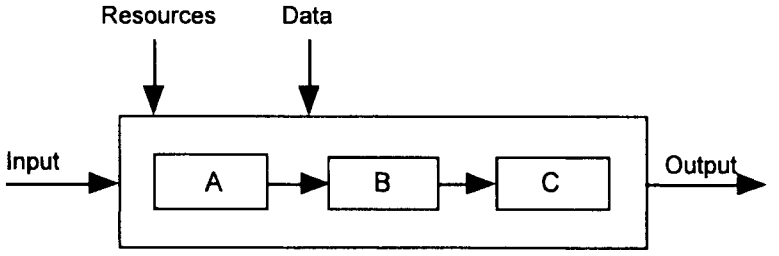


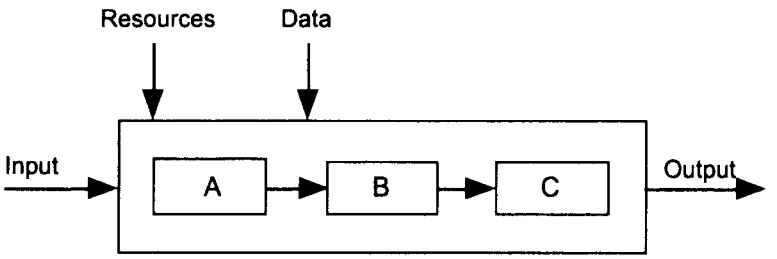
Figure 1: A simple process

4.2 Testing Assumptions

No activity is free of assumptions, and every assumption introduces risk. Assumptions are often made for good reasons, such as when information necessary for a better-informed decision is unavailable. In such cases the assumptions are known and should be recorded. Assumptions that are initially valid may become invalid with time (and often do) and they should be monitored. Many assumptions are implicit, particularly when

dependencies are involved. For example, in Figure 2, Activity B may assume correct input from Activity A, but this may not be the case if Activity A has been subject to staff shortage, competence deficiency, or the loss of a crucial item of equipment. The notes below the figure suggest other assumptions that may be implicit in the process.

A thorough risk analysis should identify assumptions, check their validity, test the confidence that they can reasonably attract, and determine their risks. In the intended method for conducting management-risk analysis, rules will ensure that assumptions are searched for, made explicit, and analysed.



Assumptions (examples only):
Correctness of input
Integrity of data
Availability of resources
Appropriateness of resources

Figure 2: Some implicit assumptions

4.3 Creativity versus Rules

A tool intended for use by a range of personnel should be easily and rapidly employed and its results should be repeatable and auditable. These criteria suggest that its operation should be mechanistic and based on rules. Yet, the management issues to be analysed by the intended method can have subtle, unexpected and unintended effects, which suggests that their successful exploration requires a creative approach. The method's design must therefore provide procedure-based operation, in which the procedures demand, in appropriate places, creativity. This needs to be designed for.

4.4 Integration into Management Systems

A method that proves its worth would, of course, be employed by an organisation for *ad hoc* use. It may even be defined as the standard tool for appropriate risk analyses. But it could also be integrated into management systems and defined as being essential when any new process is introduced or any change made to an existing process. Then, management

systems, whether safety or quality, would be subjected to risk analysis systematically, resulting in adjustments that reduce or eliminate risks rather than fixes after losses have been sustained.

4.5 A Tool for Individual Use

The intended easy-to-use method would be applicable by individual managers for the analysis of their policies, strategies, plans and decisions. In some cases an individual might carry out the analysis alone. But in most cases it would be preferable for the cooption of another person who would provide an independent perspective. The result would be instructions that have been tested for unintended and risky effects before being brought into use. At all levels of management, plans (for projects, project phases, or for the deployment of systems) could be subjected to analysis before being applied.

4.6 A Compliment to Audit

A management-risk analysis method would not only inform safety cases, it would also be complimentary to an organisation's audit function. For example, a checklist to inform audits could be made of processes or activities that are deemed particularly risky or that rely on assumptions in which there is limited confidence. In addition, the frequency or thoroughness of audits and the focus of safety assessments may also be increased for parts of a management system that are considered similarly risky, or whose failures could lead to particularly severe consequences. In these ways, both the efficiency and the effectiveness of audits and safety assessments could be improved.

4.7 Confidence Levels

Being concerned with the future, the results of risk analysis must contain uncertainty. Their accuracy, or reliability for the purpose in hand, must be expressed in terms of the level of confidence placed in them, for their level of correctness cannot be known. Yet statements of confidence seldom accompany risk analyses.

Confidence in a risk analysis depends on the completeness and accuracy of the information on which it is based, which in turn depend on the representativeness and pedigree of the sources of information. It also depends on other factors, such as the means of interpretation of the past information into predictions of the future and the assumptions involved. Analysts should understand these matters sufficiently well to determine the confidence levels that they can reasonably place on their results, and rules in the intended method will require them to do so.

There is also the problem of consistency in the determination of confidence levels. What confidence can there be that two analysts, given the same information, would claim the same confidence level? Or that the same standards would be employed by different analysts to arrive at their

confidence levels? The intended method should not only require statements of confidence but also provide guidelines for their derivation. The nature of these should be a part of the research into the method.

4.8 Coverage

Risk analysis may be described as a four-stage process. Risk mitigation adds a fifth. It is anticipated that the intended method will address all of them.

- Scope definition. The objectives or the analysis, and the constraints on it, such as time, are defined, as are other prerequisites such as the system to be analysed – including its boundary and the manner in which it is represented (e.g. as a data-flow diagram).
- Hazard identification. The things that could go wrong and their possible causes and potential consequences are identified, and it is determined whether they fall within the terms of reference of the analysis.
- Hazard analysis. The relevant identified hazards are analysed in order to determine values for the likelihood of their maturing into incidents and the severity of the consequences if they did so. Thus, risk values, either quantitative or qualitative, are derived.
- Risk assessment. The risks are assessed (evaluated) against predefined criteria to determine their degree of tolerability – from which, the appropriate risk management actions are derived.
- Risk mitigation and monitoring. Risk management actions are taken and monitoring of the risks put in train.

The method will require the essential prerequisite work of scope definition to be carried out. It will necessarily address the next three stages of risk analysis. Then, it will provide guidance on how the output of the risk assessment stage may be used to suggest options for risk management, for example by informing the placement of safety barriers. Further, the method will be appropriate to re-analysis of the improved system and may include guidance on this.

4.9 Composition

It is intended to base the method on an amalgam of established techniques, with the addition of administrative and operational rules appropriate to the method's goals (such as identifying and assessing assumptions and determining confidence levels). The exact composition is subject to research, but consideration has already been given to deriving the use of guidewords and disciplined teamwork from HAZOP (hazard and operability studies), the examination of failure modes from FMEA (failure modes and effects analysis), and to the need to explore chains of cause and effect. Starting with HAZOP and FMEA is justified by the fact that both of these techniques are not dependent for their efficacy on the type of system being explored, and both have been employed on several types of system representation, including textual representations.

5 Discussion

Although risks posed by management have been shown to be significant contributors to accidents, they are not normally included in risk analyses or safety cases – with the result that assessments must be optimistic. If management issues were addressed, not only would there be truer representations of risk, there would also be a basis for the assessment and improvement of an organisation's corporate governance.

This paper proposes a method for conducting the analysis of risks posed by management and points out the research issues that need to be tackled. These include the composition of the method itself, the types of representation of the management issues to be addressed by it, the rules to be built into the method and the guidelines to accompany it. For example, what rules are necessary for the effective exploration of assumptions? The ways in which the method will operate also need to be explored. For example, within the process in Figure 1 there are boxes, which enclose activities, and arrowed lines, which indicate the transmission of output. Do there need to be differences in the ways in which these two entities are analysed? Ease and repeatability of use by many people requires procedure-based operation, but thorough risk analysis demands creativity and, therefore, discernment in use. Ways in which the rules can embrace these apparently conflicting requirements need to be examined.

The paper also points to the potential usefulness of the intended method. It would be appropriate for systematic use on management-system processes, policies and strategies. Indeed, it could be integrated into management systems so as to enforce the investigation of the risks during the production or change of any such documents. Doing this would have the added advantage of ensuring the exploration of the ways in which policies, strategies and the clauses of management systems might be implemented, and this could lead to the early recognition of unsafe or otherwise risky approaches and to the definition of preferred procedures – the use of which could then be monitored.

It would be suitable for selective use in management decision-making, and managers could easily be trained to use it for that purpose. It could inform audits and safety assurance, and it would provide input to safety cases.

Indeed, such a method could be used systematically to provide input to a generic safety case for the corporate governance and safety management of an organisation. Such a document would be dynamically alive, being updated regularly, and would form the basis of input to the organisation's system safety cases.

Vickers (20006) shows that even the most safety-conscious companies suffer from organisational vulnerabilities when it comes to safety, and that these are manifested in a number of ways. For example, by the inability to demonstrate effective safety management, difficulty in changing aspects of their safety management systems, difficulty in auditing or demonstrating completeness of the approach to safety management, and difficulty in identifying corporate safety responsibilities. The proposals defined in this paper show that the intended method for the analysis of management risks

would make significant contributions to overcoming all of these problems.

Although the initial conception of the method was for the analysis of safety risk, it is apparent that it would be equally applicable to other types (e.g. financial, security, reputational, organisational risks and other unintended consequences) arising out of management activities. It would be suitable for analysing not only safety and quality management systems, but also management decisions, plans, policies, and proposals for change of any kind. Such a method, with appropriate rules and guidance, would enhance not only the safety management but, indeed, the overall quality of management of an organisation.

6 References

Cooper M D and Phillips R A (1994). Validation of a Safety Climate Measure. *Occupational Psychology Conference of the British Psychological Society*, 3-5 January, Birmingham, UK

Elliott D, Letza S, McGuinness M and Smallman C (2000). Governance, control and operational risk: the Turnbull effect. *Risk Management: An International Journal*, 2, 3

Feynman R P (1989). *What Do You Care What Other People Think?* Unwin Hyman, UK

Guldenmund F W (2000). The Nature of Safety Culture: A review of theory and research. *Safety Science*, Vol. 34, Issues 1-3, February, pp 215 - 257

Health & Safety Laboratory (2002). *Safety Culture: A review of the literature.* Report No. HSL/2002/25.

HSE (2001). *Health and Safety in Annual Reports.* Health and Safety Executive, http://www.hse.gov.uk/revitalising/annual.htm

Health and Safety Executive (2002). *Strategies to Promote Safe Behaviour as Part of a Health and Safety Management System.* Her Majesty's Stationary Office

ICAEW (1999). *Internal Control - Guidance for Directors on the Combined Code.* The Institute of Chartered Accountants in England and Wales, London

IEC (2000). *International Standard IEC 61508: Functional Safety of Electrical/Electronic/Programmable Electronic Systems.* International Electrotechnical Commission, Geneva

International Nuclear Safety Advisory Group (1991). *INSAG-4: Safety Culture.* International Atomic Energy Agency, Vienna

International Nuclear Safety Advisory Group (2001). *INSAG-15: Key Practical Issues in Strengthening Safety Culture.* International Atomic Energy Agency, Vienna

Jones, Martyn E and Sutherland, Gillian (1999). *Implementing Turnbull: a boardroom briefing.* The Institute of Chartered Accountants in England and Wales, London

Levene T (1997). Getting the Culture Right. In Redmill F and Dale C (Eds.): *Life Cycle Management for Dependability.* Springer-Verlag, London

Page, Michael and Spira, Laura F (2005). *The Turnbull Report, Internal Control and Risk Management: The Developing Role of Internal Audit.* The Institute of Chartered Accountants of Scotland

Ramsay, Ian M and Hoad, Richard (1997). *Disclosure of corporate governance practices by Australian companies.* Centre for Corporate Law and Securities Regulation, University of Melbourne

Redmill F (2002). Human Factors in Risk Analysis. *Engineering Management Journal,* Vol. 12, No. 4, August

The Keil Centre (2001). *Safety Culture Maturity Model.* Prepared by The Keil Centre for the Health and Safety Exeutive. The Keil Centre

Vaughan, Diane (1996). *The Challenger Launch Decision.* University of Chicago Press, Chicago

Vickers A (2006). Governing Safety Management. In Felix Redmill & Tom Anderson (Eds.): *Developments in Risk-based Approaches to Safety*: Proceedings of the Fourteenth Safety-critical Systems Symposium, Bristol, UK, 7-9 February 2006

Common Law Safety Case Approaches to Safety Critical Systems Assurance

Kevin J Anderson
Kevin Anderson and Associates Pty Ltd
Melbourne, Australia

Abstract

Robinson and Anderson (2005) outline seven different risk paradigms in support of the Rule of Law tests of causation, foreseeability, preventability and reasonableness. This paper describes the application of the "top down" paradigm using an example set on an offshore platform.
Keywords: system safety assurance, causation, foreseeability, preventability, reasonableness.

1 Introduction

The Rule of Law tests of causation, foreseeability, preventability and reasonableness are time tested - Courts and Enquiries. This paper identifies seven different paradigms or methods of satisfying legal arguments in the event of an unwanted event actually happening and escalating to cause harm to humans, the environment or property.

The rub is to demonstrate that a reasonable person (in the eyes of the court and with the advantage of 20:20 hindsight) in the same position would have undertaken certain procedures and processes to ensure that whatever it was that did happen, on the balance of probabilities, should not have occurred.

The scenario adopted in this paper involves people living on an offshore platform with a worst case loss of 500 lives if a storm destroys the platform and evacuation strategies fail.

Asking lawyers which paradigm is applicable to ensure 'due diligence' usually elicits the response 'all of them', that is, if any one paradigm would have identified sensible precautions to take against a credible threat, then those precautions ought to have been taken.

2 The Rule of Law

2.1 Tests of Negligence

In the common law tests of negligence, four keywords are used: *causation, foreseeability, preventability and reasonableness*. These can be broadly equated to

risk management concepts as follows:

(a) WHAT are we taking about? *causation*
(Define concept & scope of a matter about which a person goes to court.)

(b) What could go WRONG? *foreseeability*
(Identify hazards and risks)

(c) Explain WHY this will not happen. *preventability*
(Implement necessary risk reduction)

(d) But WHAT IF it does happen! *reasonableness*
(Accept residual risk)

This Rule of Law underpins the *ALARP* principle that risks must be reduced ..'*As Low As Reasonably Practicable*'. The four tests also provide a focus for other risk management principles including '*not less safe*', '*continuous improvement*' and '*good practice*'.

2.2 Seven Paradigms

The seven paradigms identified by Robinson and Anderson (2005) are:

(i) Threat and Vulnerability top-down techniques such as Strength-Weakness-Opportunity-Threat (SWOT), Dependence Diagram (DD), Fault Tree Analysis (FTA), and Markov Analysis (MA).

(ii) Asset-based bottom-up approaches such as Hazard and Operability Study (HAZOP) and Failure Mode Effects and Criticality Analysis (FMECA).

(iii) Historical evidence of loss expectancy/loss prevention through insurance.

(iv) Demonstration of good practice through cause-consequence modelling to show that sensible precautions are in place to cover all credible threats.

(v) Consideration of risk taking where there is a prospect of gain as opposed to pure unwanted events.

(vi) Human factors and risk culture considerations using Root Cause Analysis.

(vii) Sensitivity testing through scenario simulation modelling.

Note: The order of presentation of paradigms is changed here from that presented in Robinson and Anderson (2005) so as to focus on measures and techniques relevant to the various phases of the system safety assurance process.

This paper concentrates on the techniques listed under (i) above so as to provide an overall framework for the more detailed "evidence" that the other paradigms would need to provide.

2.3 Model Example

This section describes a hypothetical universe postulated to be subject to extreme weather events that require total evacuation of offshore platforms from time to time. Travel between platforms would be accomplished though a high-speed seatube system laid on the ocean bed with low speed risers to move up and down from a platform to the ocean bed.

An underwater interface would connect the high-speed horizontal travel and low speed ascent /descent sub-systems. Commencing from the viewpoint of traveller/s in a vehicle moving on the high speed system, the arrival interface would have four subsystems:

- firstly the vehicle has to leave the mainline onto a siding(de-merge), and
- secondly slow down to a low speed, and
- thirdly ascend to the offshore platform, and
- lastly, disembark (get off) the system

Departing from a platform would occur in reverse:

- firstly," get on" to the system, and
- secondly, descend to the seabed, and
- thirdly "speed up" to match the high speed system, and
- lastly, merge into a vacant slot on the high speed system.

3 Causation

The first of the four tests *causation* asks whether harm could occur because of some unsafe matter on which a charge of negligence could be based. Top-down techniques are appropriate at this phase of a functional safety assessment. The worst case scenario of destruction of a fully occupied platform would be 500 fatalities.

3.1 Measures and Techniques

This section describes the application of a number of top-down methodologies

(a) Vulnerability)

Vulnerability methods indicate general areas of strategic concern rather than

solutions to particular problems. First make a list of assets such as staff, equipment, operability, reputation and then consider what threats there are e.g. natural events, technical, financial, political, community. In this case study, natural events in the form of cyclones necessitate evacuation strategies.

The intersection of a threat and an asset is termed a 'vulnerability' and such a matrix (as in Table 1) can be rated in terms of criticality as to risk of loss or value added - the risk of gain.

	Assets			
Threats	Staff	Equipment	Operability	Reputation
Natural Event	xxx	xx	xxx	x
Technical	x	xxx	xx	x
Financial	x	xx	xxx	xx
Political	x	-	-	x
Community	xx	-	-	-

Table 1 Vulnerability Matrix

Scores:

xxx	Critical potential vulnerability that must be addressed
xx	Moderate potential vulnerability
x	Minor potential vulnerability
-	No noticeable vulnerability

(b) Dependence Diagram (DD)

A Dependence Diagram [see Figure 1] represents a chain of sub-systems failures arranged in series (logic OR with respect to failure) or parallel (logic AND with respect to failure) situations. In this case study, high-speed travel is assumed to be a single high-reliability technology, while other components may require duplication to achieve high reliability, especially with regard to evacuation.

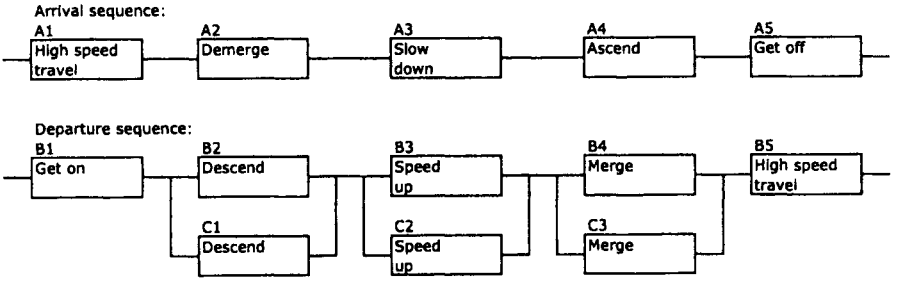


Figure 1 Dependence Diagram

Note that A1 follows B5 and B1 follows A5 in a circular fashion.

In this example, given approach of a threatening storm, failure to depart is a credible threat and the failure of any sub-system in the B series could lead to a dangerous failure. Necessary risk reduction is provided by systems C with parallel logic of B2-C1, B3-C2 AND B4-C3. The method is useful for high level summary of analyses and the various 'cut sets' leading to failure are identified at sub-system level as: B1 OR (B2 AND C1) OR (B3 AND C2) OR (B4 AND C3) OR B5.

(c) Fault Tree Analysis (FTA) and Event Tree Analysis (ETA)

Fault Tree Analysis FTA and Event Tree Analysis (ETA) (Figure 2) are alternative methods of depiction of causes of top events and lends themselves better at both expansion and quantification.

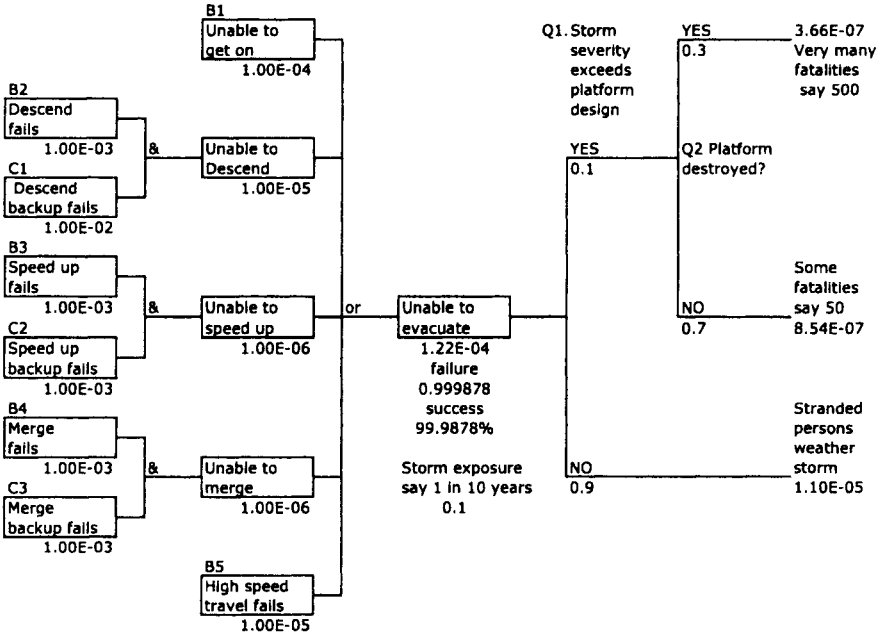


Figure 2 Fault Tree Analysis and Event Tree

In this example, failure of B1 contributes 80% of the risk of being unable to evacuate a platform in the teeth of a predicted storm. Beyond this loss of control point, event tree analysis in employed to assess the probability of the worst case outcome. There are two event sequences, each with a balance of probability, firstly as to whether the storm severity exceeds platform design parameters and lastly whether or not the platform is destroyed. The inclusion of an event tree analysis following a fault tree top event is termed a cause-consequence model.

The unmitigated risk is "undesirable" on a scale with a target level of safety of 10^{-8} (1.00 E-8) being at the top end of the ALARP region. Such risks are only tolerated if further risk reduction is impracticable or the cost of risk reduction

is grossly disproportionate to the benefit.

(d) Markov Analysis (MA)

Markov Analysis (Figure 3) depicts input/output transitions leading to system failure:

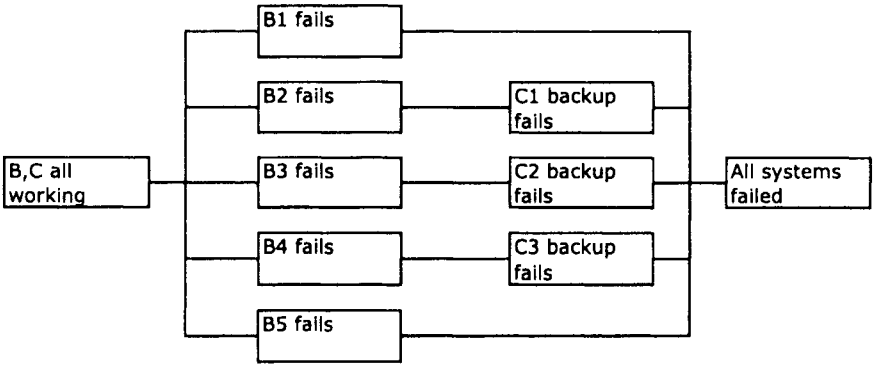


Figure 3 Markov Analysis

The Markov Analysis shows the various states and transitions between them. In particular, the significance of unrevealed failures in backup systems is brought to the fore. The various cut sets are:

Hazardous situation	Loss of control	Exposure	Storm (p.a)	Structure	Frequency
B1	1.00E-04	0.1	0.1	0.3	3.00E-07
B2*C1	1.00E-05	0.1	0.1	0.3	3.00E-08
B3*C2	1.00E-06	0.1	0.1	0.3	3.00E-09
B4*C3	1.00E-06	0.1	0.1	0.3	3.00E-09
B5	1.00E-05	0.1	0.1	0.3	3.00E-08
			Result for 500 fatalities		3.66E-07

Table 2 Cut Sets

3.2. Evidential Requirements

IEC 61508 (2000) sets out requirements for the documentation of Concept:

A thorough familiarity with the activities and their physical environment	The scenario of a "waterworld" necessitates people living their lives on platforms.
Identification of likely sources of hazards (human error, software etc.)	The storm scenario in the worst case could result in very many fatalities.
Information about the identified hazards (energy damage)	The platform structural design must take into account the risk of a destroyed platform.
Information about current national and international standards and regulations	A target level of safety of say 1.00E-8 must be achieved (this is not yet met with 3.66 E-7 postulated)
Consideration of hazards due to interactions (communications etc.)	The merger with the high speed system may require priorities for evacuation to override normal inter-platform travel
Information about relevant social /political issues	Immigrants to "waterworld" require confidence that the platform evacuation system must work.

Table 3 Concept Requirements

IEC 61508 (2000) then sets out requirements for the documentation of scope:

Definition of physical equipment including equipment-under-control, control systems and procedures	The normal transport system has both traffic control systems and plant control systems, including ventilation in undersea tunnels
Identification of external events	External events are high severity storms
Identification of sub-systems associated with hazards	The evacuation requirements are focussed on departure sub-systems are per the above DD, FTA and Markov analyses. Arrival systems would be shut down
Consideration of the types of accidents and their initiating events	The initiating event is a hyperstorm and the types of accidents relate to avoiding stranded persons remaining behind vulnerable to platform destruction.

Table 4 Scope Requirements

4 Foreseeability

This section deals with *foreseeability*: Did you know? or Ought you to have known? about a potential source of physical injury or damage to the health of people either directly or indirectly as a result of short-term or long-term damage to property or the environment IEC 61508 (2000), part 4 #3.1.

There is a tradeoff in the Case Study between evacuation strategies and structural hardening of the design. The latter is most likely uneconomic by comparison with the former

Asset based bottom-up approaches such as Hazard and Operability Study (HAZOP) and Failure Mode Effects and Criticality Analysis (FMECA) are relevant, but provide no guarantee of completeness and do not consider unwanted synergies and multiple failure leading to catastrophic outcomes.

For example, common mode failures and the need for additional layers of protection have not been addressed thus far in this Case Study. The Markov Analysis shows the various states and transitions between them. In particular, the significance of unrevealed failures in backup systems is brought to the fore.

4.1. Evidential Requirements

Identification of hazards is included in all modes of operation and all reasonably foreseeable circumstances including fault conditions and misuse. Considerations include:

All relevant human factor issues	Event B1 "Unable to get on" constitutes the highest probability single event leading to a loss of control failure to evacuate. Multiple warning and advisory systems are indicated including regular drills and enforcement.
Event sequences leading to hazards	Sub-systems must be analysed further to support respective reliability claims.
Frequencies (or probabilities) of hazardous events	Inability to evacuate has been advised as the "top event" and several cut sets (B1, B2*C1 and B5) need to be reduced to meet the target level of safety
Potential consequences associated with hazards	Depending on structural resistance to storm severity, a range of consequences from major to catastrophic is possible

Table 5 Foreseeability Requirements

The standard imposes twelve requirements for hazard and risk analysis, as in Table 6. (Note that tasks 8 and 9 are logically prior to tasks 2 to 7.)

1.	*Take into account information from scope definition phase.*	*A target failure rate for evacuation has been set at 3E-6. This equates to 99.9997% availability*
8.	*Requirements 1 to 7 can be met by the application of either qualitative or quantitative hazard and risk analysis techniques as per part 5 of the Standard.*	*The example to date has applied a quantitative analysis employing a number of techniques. In terms of qualitative assessment, an "unlikely" failure but with "catastrophic" outcome represents an extreme risk necessitating "necessary risk reduction" and application of ALARP (as low as reasonably practicable), good practice and continuous risk reduction principles*
9.	*Select appropriate techniques and the extent to which they need to be applied depending on a number of factors, including:*	
- the specific hazards and the consequences		*It is imperative that evacuation systems work at extremely high levels of availability.*
- the application sector and its accepted good practices		*Road and rail tunnel risk criteria are sufficiently established to provide a basis for good practice. (99.995% availability is a typical control system requirement)/*
- the legal and safety regulatory requirements		*The methodology described here is predicated on common law tests of negligence*
- the risk of the "equipment under control" (EUC).		*The evaluation to date suggests 1.22 E-4 chance of loss of evacuation capability, but the target is 3 E-6*
- the availability of accurate data upon which the hazard and risk analysis is to be based		*There is good data both on the storm threat and the sub-system, assembly and component failure rates and repair tactics*
2.	*Consider elimination of the hazards.*	*There is some speculation that storm activity and severity is increasing, so elimination is not an option.*
3.	*Determine the hazards and hazardous events of the EUC and the EUC control system under all reasonably foreseeable circumstances (including fault conditions and misuse). This shall include all relevant human factor issues, and shall give particular attention to abnormal or*	*A hazard is an incipient condition and a hazardous situation occurs only when the condition is manifested AND the various layers of protection are breached. The accident itself (the hazardous situation) with a balance of probability of escalating to a major or catastrophic outcome. The storm scenario presented here is infrequent,*

	infrequent modes of operation of the EUC.	*but nevertheless a credible threat.*
4.	*Determine the event sequences leading to the hazardous events.*	*Five cut sets are indicated on the Dependence diagram with "Failure" to get on the system as the most significant being a single event*
5.	*Evaluate the likelihood of the hazardous events for the conditions specified in 3*	*Taking credit for likely frequency of a storm and the structural design of the platform unlikely, destruction of a platform is assessed as "rare"*
6.	*Determine the potential consequences associated with the hazardous events.*	*However, the potential consequence of 500 fatalities is catastrophic.*
7.	*Evaluate or estimate the EUC risk for each determined hazardous event.*	*The combination of "rare" likelihood and catastrophic consequence represents an extreme risk*
10.	*Consider the following:*	
-	*each determined hazardous event and the components that contribute to it*	*The worst case "extreme" risk is defined at item 7. above.*
-	*the consequence and likelihood of the event sequences with which each hazardous event is associated*	*Extreme risks are intolerable except in extraordinary circumstances.*
-	*the necessary risk reduction for each hazardous event*	*Necessary Risk Reduction is advised not only for the B1 scenario, but also for B2*C1 and B5*
-	*the measures taken to reduce or remove hazards and risks*	*For B1, redundancy is advised. For B2*C1 and B5, an order of magnitude improvement in reliability must be attained.*
-	*the assumptions made during the analysis of the risks, including the estimated demand rates and equipment failure rates; any credit taken for operational constraints or human intervention shall be detailed*	*Whilst Dependence Diagrams and Markov Analyses provide high level views, the Fault Tree /Event Tree approach lends itself to expansion using Parts Count techniques for mechanical and electrical equipment failure rates together with claims for human factors, particularly as regards scenario B1.*
-	*refer to key information in project lifecycle documentation*	*For example, scenario B1 must be expanded, say B1A 'mechanical failure" B1B "electrical failure" B1C "human factors"*
11.	*Document the information and results which constitute the*	*For the first cut of system analysis, the Vee lifecycle and waterfall design*

	hazard and risk analysis.	*model is appropriate.*
12.	*Maintain this information throughout the overall safety lifecycle*	*However many designs require iteration an a spiral lifecycle model is indicated*

Table 6 Hazard and Risk Analysis Requirements

Historical evidence of loss expectancy /loss prevention through insurance is required to support the analysis and claims for dangerous failure rates must be supported through:

(i) actual operating experience in a similar application;

(ii) a reliability analysis carried out to a recognised procedure;

(iii) an industry database of reliability of generic equipment.

Parts Count Analysis (ii) using Generic data (iii) provides a starting point for verifying claims through method (i).

5. Preventability

The *preventability* question asks whether there is a practicable way or alternative to how things would be done which would have responded to a hazard actually happening and exposing persons to harm IEC 61508 (2000).

From the risk analysis, the tolerability of the risk from each hazard must be assessed and measures proposed to reduce or remove intolerable hazards. Suitable risk criteria must be used to evaluate the acceptability of risks calculated.

IEC 61508 (2000) provides examples in part 5 of qualitative and quantitative risk analysis. Concepts of ALARP, the risk 'triangle' and the risk matrix and the risk graph approaches are detailed.

For safety-critical systems and software, the concept of Safety Integrity levels (SIL) is a source of great debate. The table below sets out limit claims that can be made for a given SIL level.

For many systems, the author favours NOT designating an entire system as safety-related, but noting that any claims as to the likelihood of a given hazard actually happening must be supported by one or more of methods (i), (ii) and (iii) in section 4.1 above.

Given a frequency of 1 in 10 years for destructive storms, the necessary risk reduction would be assessed using the low demand figures in Table 1 above. If 90% to 99% risk reduction is sufficient, SIL1 will suffice. Typically, a Commercial-off-the-shelf (COTS) application provides 95% success (5% failure). The exact figure would depend on evidence as to failure rate assumptions for that particular item.

The provision of redundant hardware and diverse software theoretically gains significant risk reduction (5% x 5% = .05* .05 = 0.0025 failure probability (99.75% success)) However, care must be taken to identify common mode failures such as power supplies, fire etc. Generally, a good SIL 2 system claim can be defended using two SIL 1 systems.

The analysis to date (Refer Table 2 above) suggests must do improvements in three scenarios B1 (3.00 E-7), B2*C1 (3.00 E-8) and B5 (3.00 E-8) if a residual risk target of 1.00E-8 is to be met.

Safety Integrity level (SIL)	Mode of Operation	Greater or equal	to less than
No SIL	Continuous	Supported claim	1.00 E -5
SIL 1	Continuous	1.00 E -5	1.00 E -6
SIL 2	Continuous	1.00 E -6	1.00 E -7
SIL 3	Continuous	1.00 E -7	1.00 E -8
SIL 4	Continuous	1.00 E -8	1.00 E -9
No SIL	Low Demand	Supported claim	1.00 E -1
SIL 1	Low Demand	1.00 E -1	1.00 E -2
SIL 2	Low Demand	1.00 E -2	1.00 E -3
SIL 3	Low Demand	1.00 E -3	1.00 E -4
SIL 4	Low Demand	1.00 E -4	1.00 E -5

Table 7: Safety Integrity Levels

Given a frequency of 1 in 10 years for destructive storms, the necessary risk reduction would be assessed using the low demand figures in Table 1 above. If 90% to 99% risk reduction is sufficient, SIL1 will suffice. Typically, a Commercial-off-the-shelf (COTS) application provides 95% success (5% failure). The exact figure would depend on evidence as to failure rate assumptions for that particular item.

The provision of redundant hardware and diverse software theoretically gains significant risk reduction (5% x 5% = .05* .05 = 0.0025 failure probability (99.75% success)) However, care must be taken to identify common mode failures such as power supplies, fire etc. Generally, a good SIL 2 system claim can be defended using two SIL 1 systems.

The analysis to date (Refer Table 2 above) suggests must do improvements in three scenarios B1 (3.00 E-7), B2*C1 (3.00 E-8) and B5 (3.00 E-8) if a residual risk target of 1.00E-8 is to be met.

6 Reasonableness

Reasonableness requires a judgement as to the balance of the significance of the risk versus the effort required to reduce risk to an acceptable level. The standard

IEC 61508 (2000) part 1 #8 provides for a Functional Safety Assessor (FSA) to independently investigate and arrive at a judgment as to the level of safety required. Discussion of this rule-of-law test is outside of the intent of the paper.

The paper to date has dealt mainly with use of the top-down paradigm to establish a safety argument. The other paradigms would follow:

- HAZOP and FMECA techniques systematically assess the sub-systems down to assembly and component parts down to lowest replaceable unit (LRU)
- Historical evidence must be used to justify assumptions of both EUC and risk reduction items.
- Parts 2 and 3 of IEC 61508 (2000) provide a wealth of detail about good practice as it was some years ago but nevertheless provides a basis for questionnaires to update the measures and techniques to current good practice.
- There is an element of risk taking in the establishment of offshore platforms in the first place. However, once designed and constructed, the effort must be focussed on controlling unwanted events.
- Depending on the level of functionality of emergency operations and control systems, human factors may be the weak link in the safety argument.
- Sensitivity testing of assumptions and scenario simulation modelling of storm /structural interactions will provide guidance as to tradeoffs inherent in an evacuation strategy.

7 Conclusion

In conclusion, the need to put in place sensible precautions against ALL credible threats means that no one measure or technique will suffice to provide all of the 'evidence' of system safety assurance. The top-down methodologies related to the four rule-of-law tests provide a first cut of the overall safety argument, with the other paradigms adding to the evidence.

8 Acknowledgement

The Author acknowledges Risk & Reliability Associates Pty Ltd for quotations from Robinson and Anderson (2005).

References

IEC 61508 (2000) *'Functional safety of electrical /electronic / programmable electronic safety-related System's*. Commission Electronique Internationale, 1998-2000.

Robinson and Anderson (2005) *'Risk & Reliability - An Introductory Text'*, Risk & Reliability Associates Pty Ltd, Fifth Edition, 2005. Melbourne.

SOFTWARE SAFETY

Ada 2005 for High-Integrity Systems

José F. Ruiz

AdaCore
8 rue de Milan
75009 Paris, France

ruiz@adacore.com

Abstract

The forthcoming Ada 2005 standard has been enhanced to better address the needs of the real-time and high-integrity communities. This new standard introduces new restriction identifiers that can be used to define highly efficient, simple, and predictable run-time profiles. Among others, this language revision will standardize the Ravenscar profile, new scheduling policies, and will include execution time clocks and timers. Flexible object-oriented features are also supported without compromising performance or safety.

1 Introduction

For the development of safety-critical software, the choice of programming language makes a significant difference in meeting the requirements of exacting safety standards and, ultimately, high-reliability applications.

The Ada language was first introduced in 1983 (ISO 1983). Used primarily for large-scale safety and security critical projects, and embedded systems in particular, where reliability and efficiency are essential, Ada experienced its last major revision in 1995 (ISO 1995), making it the first internationally standardized object-oriented language. The latest revision (Ada 2005) responds to requests for features in the areas of multiple interface inheritance, real-time profiles, flexible task-dispatching policies, and a unification of concurrency and object-oriented features.

One of the most important achievements of Ada 2005 is the standardization of the Ravenscar restricted tasking profile. This profile defines a subset of the tasking features of Ada which is amenable to static analysis for high integrity system certification, and that can be supported by a small, reliable run-time system. This profile is founded on state-of-the-art, deterministic concurrency constructs that are adequate for constructing most types of real-time software.

Measuring and limiting the execution time of tasks is also possible in Ada 2005 by using execution time clocks and timers. This functionality is equivalent to the

execution time monitoring existing in the real-time extension to POSIX (IEEE 2003), allowing the implementation of flexible real-time scheduling algorithms, such as the sporadic server in fixed priority systems, or the constant bandwidth server in dynamic priority systems.

Timing events are also provided as an effective and efficient to execute user-defined time-triggered procedures without the need to use a task or a delay statement.

There have been major improvements to the scheduling and task dispatching mechanisms with the addition of further standard pragmas, policies, and packages which facilitate many different mechanisms such as non-preemption within priorities, timeslicing, and dynamic priority dispatching. Moreover, it is possible to mix different policies according to priority ranges within a partition.

The following sections will describe the advantages of using Ada for developing embedded real-time high-integrity systems, paying special attention to the new features that will be available in the forthcoming Ada 2005 standard.

2 Software engineering with Ada

The general design philosophy of the language promotes sound software engineering techniques basing on its considerable expressive power and high abstraction level features.

The original Ada 83 design introduced the package construct, a feature that supports encapsulation (information hiding) and modularization, and that allows the developer to control the namespace that is accessible within a given compilation unit, hence reducing data coupling. Ada 95 introduced the concept of child units, adding considerably flexibility and easing the design of very large systems. Packages provide strict separation of specification from implementation, and allow the structuring of code into a hierarchical set of components with strict control over visibility of encapsulated state data and methods.

One important capability of the child unit mechanism is that it allows developers to write test programs that can access encapsulated state data that is inaccessible to normal client code. This simplifies the job of meeting coverage analysis requirements from safety standards such as DO-178B (RTCA 1992), without compromising the need to have state data hidden.

Generics are a powerful mechanism for constructing large-scale programs through the parameterization of program units. The use of generics enhances program reliability by means of facilitating reuse, easing maintenance, reducing source code size, and helping avoid human replication error.

Ada 95 introduced direct support for object-oriented programming: encapsulation (as just noted), objects (entities that have state and operations), classes (abstractions of objects), inheritance, polymorphism, and dynamic binding.

Ada tasking provides a natural and powerful abstraction mechanism for decoupling application activities, including the functionality for sharing resources, communicating, and synchronizing.

3 Ada for embedded applications

Ada was designed with embedded applications in mind from the start. For example, the use of representation clauses, which have been extended and made more powerful in Ada 2005, allows close mapping of data structures to the hardware, and the built in concurrency can be used to map handling of multi-tasking at the hardware level. Additionally, many embedded applications require high reliability or are safety-critical, which is where a language designed for maximum safety really shines.

The Ada standard includes a normative annex which specifies additional capabilities provided for low-level programming (ISO 1995, Annex C). It allows access to hardware-specific features, such as:

- Insertion of assembly and intrinsic subprograms. Intrinsic subprograms are built-in to the compiler provided for convenient access to any machine operations that provide special capabilities or efficiency and that are not otherwise available through the language constructs. Examples of such instructions include atomic read-modify-write operations, standard numeric functions, string manipulation operations, vector operations, direct operations on I/O ports, etc.
- Representation clauses for specifying the desired address, size, alignment, and layout of data in memory.
- Shared variable control. Read and update operations can be forced to either be performed directly to memory, or in a indivisible (atomic) manner.
- Interrupt support. There is a language-defined model for hardware interrupts which includes the mechanisms for handling interrupts.
- Storage management. Specific storage pools can be specified with user-defined managers that may be placed in specific memory regions. They may be suitable for real-time systems because they can be made predictable.

Another important feature of Ada is that its functionality, notably its tasking capabilities, maps very well to the typical embedded operating systems used in many applications.

4 Ada for high-integrity applications

Ada is the language of choice for many high-integrity systems due to its careful design and the existence of clear guidelines for building high integrity systems (ISO 2000, Burns, Dobbing & Vardanega 2003).

Fitting its commitment to safety and reliability, a formal validation process exists based on an ISO (International Standards Organization) standard (ISO 1999). Ada is the only language for which such a validation standard exists. An Ada Conformity Assessment Test Suite (ACATS) (ACAA 2005) has been developed for this conformity testing, which exercises both the compiler and the run-time system.

The use of a standardized language (ISO 1999) ensures that your program will behave as you want (as it is designed to) even when changing target platforms or compilers. The effect of a program can be predicted from the language definition with few implementation dependencies of interactions among language features. The semantics of Ada programs are well defined even in error situations. The Ada standard includes a normative annex which specifies additional capabilities provided for systems that are safety critical or have security constraints (ISO 1995, Annex H).

When writing high reliability software, the full Ada language is inappropriate since the generality and flexibility may interfere with traceability and certification requirements. Ada addresses this issue by supplying configuration directives (that may restrict individual features or define a complete set of restrictions) that allows you to constrain the language features to a well-defined subset that facilitate analysis and safety, and avoids error prone or hard to analyze features. The ISO 15942 technical report (ISO 2000) contains a detailed analysis of the different Ada features with respect to their suitability for different verification techniques. The use of restricted profiles and restrictions also allows the compiler to remove unnecessary run-time support, simplifying the certification process and preventing the inclusion of inactive code in the final application.

One of the most interesting subsets for high-integrity systems is the Ravenscar profile, a collection of concurrency features that are powerful enough for real-time programming but simple enough to make certification practical. Another notable example is SPARK (Barnes 2003) that includes Ada constructs regarded as essential for the construction of complex software, but removes all the features that may jeopardize the requirements of verifiability, bounded space and time, and minimal run-time system.

Apart for the advantages derived from the high abstraction level provided by the language (encapsulation, data abstraction, reusability, tasking, etc.), there are many others features in the language that promote safety and reliability. Ada code is very readable, making code maintenance easier and simplifying certification steps, including peer review and walkthroughs. Strong typing ensure that most errors are detected statically at compile time, and many remaining errors are automatically detected at execution.

Access types in Ada have been designed in a way to prevent the occurrence of dangling references because they can never designate objects that have gone out of scope. Users can also further restrict the use of allocators and deallocators through appropriate restrictions.

Ada provides an exception mechanism for detecting and responding to exceptional run-time conditions in a controlled manner, providing well-defined semantics even under error conditions. It allows residual errors to be detected and handled, so the exception features are potentially a key part of a language for high-integrity applications (Motet, Marpinard & Geffroy 1996). Its use makes verification more difficult, unless restrictive strategies (ISO 2000) are used which simplify the verification process.

Ada 2005 contains determinism and hazard mitigation issues relating to task activation and interrupt handler execution semantics, in response to certification con-

cerns about potential race conditions that could occur due to tasks being activated and interrupt handlers being executed prior to completion of the library-level elaboration code. A new configuration pragma has been added (ARG 2005*d*) for guaranteeing the atomicity of program elaboration, that is, no interrupts are delivered and task activations are deferred until the completion of all library-level elaboration code. This eliminates all hazards that relate to tasks and interrupt handlers accessing global data prior to it having been elaborated, without having to resort to potentially complex elaboration order control.

Another major hazard in high-integrity systems, tasks terminating silently, has been addressed in Ada 2005 with a new mechanism for setting user-defined handlers which are executed when tasks are about to terminate. These procedures are invoked when tasks are about to terminate (either normally, as a result of an unhandled exception, or due to abort), allowing controlled responses at run time and also logging these events for post-mortem analysis.

5 Ada for real-time applications

Concurrency is a core part of the language, and there is a normative annex intended for real-time systems software (ISO 1995, Annex D) that supports sound real-time development techniques, such as Rate Monotonic Analysis (Liu & Layland 1973), Response Time Analysis (RTA) (Joseph & Pandya 1986), and some others introduced in the Ada 2005 revision that will be described later.

Ada provides well-defined semantics for scheduling, avoiding the disadvantages associated with the use of low-level constructions for task handling and synchronization. Task cooperate using synchronous message passing (rendezvous) and safe and efficient data-oriented communication and synchronization through protected objects.

Asynchronous capabilities are also very important for some real-time applications, and they are supported with the following mechanisms:

- Asynchronous Transfer of Control (ATC) is a mechanism that allows the execution of an abortable part to be cancelled by a triggering event (time event or another task), in which case an optional sequence of code can be executed after the abortable part is left.
- Preemptive task abortion can trigger asynchronously the termination of one or more target tasks.
- Asynchronous task control is a simple and efficient capability to suspend and resume the execution of another task.
- Asynchronous external events are modelled by interrupts, a language-defined class of events that are detected by the hardware or the system software.

A high-resolution monotonic clock together with support for both absolute and relative delays are also part of the Ada standard, which defines minimum requirements in terms of range and accuracy.

6 The Ravenscar profile

As the functionality and complexity of embedded software increases, more attention is being devoted to high level, abstract development methods. The Ada tasking model provides concurrency as a means of decoupling application activities, and hence making software easier to design and test (Vardanega & van Katwijk 1999).

The tasking model in Ada 95 is extremely powerful, but it has always been recognized that, in the case of high-integrity systems, it is appropriate to choose a subset of these facilities because accurate timing analysis is difficult to achieve. Advances in real-time systems timing analysis methods have paved the way to reliable tasking in Ada. Accurate analysis of real-time behavior is possible given a careful choice of scheduling/dispatching method together with suitable restrictions on the interactions allowed between tasks.

The Ravenscar profile (ARG 2005*f*) is a subset of Ada tasking that provides the basis for the implementation of deterministic and time analyzable applications. This subset is amenable to static analysis for high integrity system certification, and can be supported by a small, reliable run-time system. This profile is founded on state-of-the-art, deterministic concurrency constructs that are adequate for constructing most types of real-time software (Burns et al. 2003). Major benefits of this model are:

- Improved memory and execution time efficiency, by removing high overhead or complex features.
- Increased reliability and predictability, by removing non-deterministic and non analyzable features.
- Reduced certification cost by removing complex features of the language, thus simplifying the generation of proof of predictability, reliability, and safety.

The profile is based on a computation model similar to the one proposed by Vardanega (Vardanega 1998), which is based on the HRT-HOOD method (Burns & Wellings 1995), that includes the following features:

- A single processor.
- A fixed number of tasks.
- A single invocation event per task (either time-triggered or event-triggered tasks).
- Task interaction only by means of shared data (protected objects) with mutually exclusive access.

Constructions that are difficult to analyze, such as dynamic tasks and protected objects, task entries, dynamic priorities, select statements, asynchronous transfer of control, relative delays, or calendar clock, are forbidden. It allows memory usage and execution to be deterministic.

The concurrency model promoted by the Ravenscar Profile is consistent with the use of tools that allow the static properties of programs to be verified. Potential verification techniques include information flow analysis, schedulability analysis, execution-order analysis and model checking.

The Ravenscar profile will be part of the Ada 2005 standard, so compiler vendors must implement it. The intention is that not only will they support it, but in appropriate environments (notably embedded environments), efficient implementations of the Ravenscar tasking model will also be supplied.

7 Scheduling and dispatching policies

An important area of increased flexibility in Ada 2005 is that of task dispatching policies. In Ada 95, the only predefined policy is fixed-priority preemptive scheduling, although other policies are permitted. Ada 2005 provides further pragmas, policies, and packages which facilitate many different mechanisms such as non-preemption within priorities (ARG 2005*c*), round robin using timeslicing (ARG 2005*e*), and Earliest Deadline First (EDF) policy (ARG 2005*g*). Moreover, it is possible to mix different policies according to priority levels within a partition.

Time sharing the processor using round robin scheduling is adequate for non-real-time systems, and also in some soft real-time systems requiring a level of fairness. Many operating systems, including those compliant with the POSIX real-time scheduling model, support this scheduling policy that ensures that if there are multiple tasks at the same priority one of them will not monopolize the processor.

In order to reduce non-determinism and to increase the effectiveness of testing, non-preemptive execution is sometimes desirable (Burns 2001). The standard way of implementing many high-integrity applications is with a cyclic executive (Baker & Shaw 1989). Using this technique a sequence of procedures is called within a defined time interval. Each procedure runs to completion and there is no concept of preemption. Data is passed from one procedure to another via shared variables and no synchronization constraints are needed, since the procedures never run concurrently. The major disadvantage with non-preemption is that it will usually (although not always) lead to reduced schedulability.

Ada 2005 supports the notion of deadlines (the most important concept in real-time systems) via a predefined task attribute. The deadline of a task is an indication of the urgency of the task. EDF scheduling allocate the processor to the task with the earliest deadline. EDF has the advantage that higher levels of resource utilization are possible, although it is less predictable, compared to fixed-priority scheduling, in case of overload situations.

8 Execution time monitoring and control

Monitoring and control execution time is important for many real-time systems. Ada 2005 provides an additional timing mechanism (ARG 2005*a*, ARG 2005*b*) which allows for:

- monitoring execution time of individual tasks,
- defining and enabling timers and establishing a handler which is called by the run-time system when the execution time of the task reaches a given value, and
- defining a execution budget to be shared among several tasks, providing means whereby action can be taken when the budget expires.

This functionality is easily supported on top of operating systems compliant to the real-time extensions to POSIX (IEEE 2003), that has recently incorporated support for execution time monitoring and budgeting.

Monitoring CPU usage of individual tasks can be used to detect at run time an excessive consumption of computational resources, which are usually caused by either software errors or errors made in the computation of worst-case execution times.

Schedulability analysis are based on the assumption that the execution time of each task can be accurately estimated. Measurement is always difficult, because, with effects like cache misses, pipelined and superscalar processor architectures, the execution time is highly unpredictable. Run-time monitoring of processor usage permits detecting and responding to wrong estimations in a controlled manner.

CPU clocks and timers are also a key requirement for implementing some modern real-time scheduling policies which need to perform scheduling actions when a certain amount of execution time has been consumed. Providing common CPU budgets to groups of tasks is the basic support for implementing aperiodic servers, such as sporadic servers and deferrable servers (Sprunt, Sha & Lehoczky 1989) in fixed priority systems, or the constant bandwidth server (Ghazalie & Baker 1995) in EDF-scheduled systems.

9 Timing events

Timing events (ARG 2005*h*) allow for a handler to be executed at a future point in time in a efficient way, as it is a stand-alone timer which is execute directly in the context of the interrupt handler (it does not need a server task).

The use of timing events may reduce the number of tasks in a program, and hence reduce the overheads of context switching. It provides an effective solution for programming short time-triggered procedures, and for implementing some specific scheduling algorithms, such as those used for imprecise computation (Liu, Lin, Shih, Chuang-Shi, Chung & Zhao 1991). Imprecise computation increase the utilization and effectiveness of real-time applications by means of structuring tasks into two phases (one mandatory and one optional). Scheduling algorithms that try to maximize the likelihood that optional parts are completed typically require changing asynchronously the priority of a task, which can be implemented elegant and efficiently with timing events.

10 Object-oriented programming

Object-oriented programming is a term that covers a broad spectrum of ideas and features. At one end we have traditional object-oriented design (in which a problem is modeled as a set of objects with message passing). Such designs can be programmed in languages with no object-oriented features, and do not necessarily raise any special issues in the safety-critical arena. At the other end, we have the features that traditionally appear in what are known as object-oriented languages, namely type extension, inheritance and dynamic dispatching.

Programmers writing high-integrity systems want to take advantage of the powerful notions of object-oriented programming, and work is being done in the direction of providing guidelines for certifying object-oriented applications (FAA 2004). Ada 2005 is ideally suited as the vehicle for exploiting what is safe in this area, while avoiding what is dangerous.

Given Ada's emphasis on high-integrity applications, Ada 2005 directly addresses the use of object-oriented methods within the constraints of these kinds of systems. Type extension and inheritance do not cause any problems, but dynamic dispatching is worrisome, and there is no general agreement on how to handle dynamic dispatching, where the actual flow of controls is not known statically but at run time, from a certification perspective (based on knowing the flow of control statically so it can be tested). One conservative approach is to allow type extension and inheritance, but to avoid dynamic dispatching. Ada 2005 facilitates this approach in a number of ways. First there is a sharp distinction between inheritance (tagged types) and dynamic dispatching (their associated class-wide types). In Ada, methods are statically bound by default. If class-wide types are avoided, then dynamic dispatching never occurs, and it is still possible to make full use of inheritance and type extension, thus facilitating code reuse. Second, this can be enforced by use of a language defined restriction (*No_Dispatch*). Finally, Ada 2005 offers very fine-grained control over inheritance by allowing each operation to declare explicitly whether it is intended to inherit, and the compiler checks that the intention is met (this avoids accidentally confusing *Initialize* and *Initialise* for example, a well known hazard in object-oriented languages).

A conscious decision was made in the design of Ada 95 to not implement general multiple inheritance, because the complexities introduced to the language appeared to overwhelm the benefits. Idiomatic usage of Ada 95 object-oriented facilities still provided the ability to implement multiple inheritance at the application level through such features as access discriminants and generic units with class-wide formal parameters. But more recently, the notion of interfaces (or roles) has been developed as an effective alternative that gives the power of interfacing to multiple abstractions without the additional complexity of full multiple inheritance. Java introduced the idea of interfaces, and Ada 2005 builds on the concept to create a new and powerful form of the interface abstraction, which also extends to the unique Ada notions of task and concurrent object, maintaining the important design principle that concurrency is a first class citizen.

11 Conclusions

Ada's reliability has been field-proven for decades, even as the language evolves through real world innovation. The latest Ada 2005 responds to requests for features in the areas of multiple interface inheritance, real-time profiles, flexible task-dispatching policies, and a unification of concurrency and object-oriented features.

Safe tasking is promoted by the Ravenscar profile, which defines a deterministic and certifiable tasking subset, providing the high-level abstraction and expressive power needed for making software easy to design and test. Major hazards related to tasks terminating silently and potential race conditions at elaboration time have been addressed by new mechanisms added to Ada 2005.

The new language revision constitutes also the reference framework for high-integrity object-oriented programming, supporting powerful and flexible object-oriented features while avoiding those that jeopardize system certification.

Ada continues to be the reference language for high-integrity systems, providing high-level abstractions without compromising performance or safety.

References

ACAA (2005), *Ada Conformity Assessment Test Suite (ACATS)*, ACAA. Available at `http://www.ada-auth.org/acats.html`.

ARG (2005*a*), Execution-time clocks, Technical report, ISO/IEC/JTC1/SC22/WG9. Available at `http://www.ada-auth.org/cgi-bin/cvsweb.cgi/AIs/AI-00307.TXT`.

ARG (2005*b*), Group execution-time budgets, Technical report, ISO/IEC/JTC1/SC22/WG9. Available at `http://www.ada-auth.org/cgi-bin/cvsweb.cgi/AIs/AI-00354.TXT`.

ARG (2005*c*), Non-preemptive dispatching, Technical report, ISO/IEC/JTC1/SC22/WG9. Available at `http://www.ada-auth.org/cgi-bin/cvsweb.cgi/AIs/AI-00298.TXT`.

ARG (2005*d*), Partition elaboration policy for high-integrity systems, Technical report, ISO/IEC/JTC1/SC22/WG9. Available at `http://www.ada-auth.org/cgi-bin/cvsweb.cgi/AIs/AI-00265.TXT`.

ARG (2005*e*), Priority specific dispatching including round robin, Technical report, ISO/IEC/JTC1/SC22/WG9. Available at `http://www.ada-auth.org/cgi-bin/cvsweb.cgi/AIs/AI-00355.TXT`.

ARG (2005*f*), Ravenscar profile for high-integrity systems, Technical report, ISO/IEC/JTC1/SC22/WG9. Available at `http://www.ada-auth.org/cgi-bin/cvsweb.cgi/AIs/AI-00249.TXT`.

ARG (2005*g*), Support for deadlines and earliest deadline first scheduling, Technical report, ISO/IEC/JTC1/SC22/WG9. Available at `http://www.ada-auth.org/cgi-bin/cvsweb.cgi/AIs/AI-00357.TXT`.

ARG (2005*h*), Timing events, Technical report, ISO/IEC/JTC1/SC22/WG9. Available at `http://www.ada-auth.org/cgi-bin/cvsweb.cgi/AIs/AI-00297.TXT`.

Baker, T. & Shaw, A. (1989), 'The cyclic executive model and Ada', *Real-Time Systems*.

Barnes, J. (2003), *High Integrity Software. The SPARK Approach to Safety and Security*, Addison Wesley.

Burns, A. (2001), Defining new non-preemptive dispatching and locking policies for Ada, *in* D. Craeynest & A. Strohmeier, eds, 'Reliable Software Technologies — Ada-Europe 2001', number 2043 *in* 'Lecture Notes in Computer Science', Springer-Verlag, pp. 328–336.

Burns, A. & Wellings, A. (1995), *HRT-HOOD(TM): A Structured Design Method for Hard Real-Time Ada Systems*, North-Holland, Amsterdam.

Burns, A., Dobbing, B. & Vardanega, T. (2003), Guide for the use of the Ada Ravenscar Profile in high integrity systems, Technical Report YCS-2003-348, University of York. Available at `http://www.cs.york.ac.uk/ftpdir/reports/YCS-2003-348.pdf`.

FAA (2004), *Handbook for Object-Oriented Technology in Aviation (OOTiA)*. Available at `http://www.faa.gov/certification/aircraft/av-info/software/OOT.htm`.

Ghazalie, T. M. & Baker, T. P. (1995), 'Aperiodic servers in a deadline scheduling environment', *Real-Time Systems* **9**(1), 31–67.

IEEE (2003), *1003.13-2003 IEEE Standard for Information Technology - Standardization Application Environment Profile- POSIX Realtime and Embedded Application Support (AEP)*.

ISO (1983), *Reference Manual for the Ada Programming Language*. ANSI/MIL-STD-1815A-1983; ISO/8652-1987.

ISO (1995), *Ada 95 Reference Manual: Language and Standard Libraries. International Standard ANSI/ISO/IEC-8652:1995*. Available from Springer-Verlag, LNCS no. 1246.

ISO (1999), *Ada: Conformity assessment of a language processor*. ISO/IEC 18009:1999.

ISO (2000), *Guidance for the use of the Ada Programming Language in High Integrity Systems*. ISO/IEC TR 15942:2000.

Joseph, M. & Pandya, P. (1986), 'Finding response times in real-time systems', *BCS Computer Journal* **29**(5), 390–395.

Liu, C. & Layland, J. (1973), 'Scheduling algorithms for multiprogramming in a hard-real-time environment', *Journal of the ACM.*

Liu, J. W., Lin, K. J., Shih, W. K., Chuang-Shi, A., Chung, J. Y. & Zhao, W. (1991), 'Algorithms for Scheduling Imprecise Computations', *IEEE Computer* **24(5)**, 58–68.

Motet, G., Marpinard, A. & Geffroy, J. (1996), *Design of Dependable Ada Software*, Prentice Hall.

RTCA (1992), *RTCA/DO-178B: Software Considerations in Airborne Systems and Equipment Certification*, RTCA.

Sprunt, B., Sha, L. & Lehoczky, J. (1989), 'Aperiodic task scheduling for hard real-time systems', *Real-Time Systems.*

Vardanega, T. (1998), Development of On-Board Embedded Real-Time Systems: An Engineering Approach, PhD thesis, TU Delft. Also available as ESA STR-260.

Vardanega, T. & van Katwijk, J. (1999), 'A software process for the construction of predictable on-board embedded real-time systems', *Software Practice and Experience* **29**(3), 1–32.

Safety Aspects of a Landing Gear System

Dewi Daniels
Silver Software,
Malmesbury, United Kingdom

Abstract

This paper describes Silver Software's experience in carrying out the software development for the landing gear system for a major new airliner. We hope this paper will be of interest, for a number of reasons:

- The landing gear is one of the most safety-critical systems on the aircraft
- This aircraft uses Integrated Modular Avionics (IMA)
- Much of the work was carried out at our software development centre in Bangalore, India (in particular, development of the Level C software and verification of the Level A software)

1 Introduction

Integrated Modular Avionics (IMA) is an important new development in airborne software design. IMA systems allow systems integrators to share computing resources between applications, making more efficient use of the resources and providing a greater degree of fault tolerance. At the same time, they give applications a degree of hardware independence, so that software modules can be reused from one aircraft platform to another. The F-22 and the Boeing 777 were the first military and civil aircraft, respectively, to use IMA. The next generation of airliners now in flight-testing or on the drawing board all make extensive use of IMA.

This paper describes Silver Software's experience in developing the software for the landing gear system for such a next generation airliner.

2 Abbreviations

The abbreviations used in this paper are defined in Table 1 below.

AFDX	Avionics Full Duplex Switched Ethernet
APEX	Application Executive
API	Application Programming Interface

ARINC	Aeronautical Radio, Inc.
BLG	Body Landing Gear
BSP	Board Support Package
CMFDU	Colour Multi Function Display Unit
CPU	Central/Core Processing Unit
CSCI	Computer Software Configuration Item
FIFO	First In/First Out
IMA	Integrated Modular Avionics
MMU	Memory Management Unit
NLG	Nose Landing Gear
POSIX	Portable Operating System Interface
SEU	Single Event Upset
SIL	Safety Integrity Level
UK	United Kingdom
WCET	Worst Case Execution Time
WLG	Wing Landing Gear

Table 1. List of abbreviations.

3 The Landing Gear System

The landing gear consists of three sets of wheels:

1. The Nose Landing Gear (NLG)
2. The Wing Landing Gear (WLG)
3. The Body Landing Gear (BLG)

The basic functions of the landing gear system are:

- Opening the doors and extending the gear
- Retracting the gear and closing the doors
- Pitch trimming to position the wing and body gear bogies for extension and retraction properly
- Opening the doors on the ground to allow the maintenance crew access to the landing gear bays
- Interfacing to the flight crew for system command and indication
- Interfacing to the maintenance crew for test and data retrieval
- Providing data (such as Weight on Wheels) to other systems
- System monitoring and built-in test

The landing gear system has the following modes of operation:

- Normal operation

- Emergency operation (in case normal mode is unavailable)
- Ground door operation (for maintenance access)

Emergency mode is implemented by an independent, mechanical system.

An additional mode is planned to allow extension and retraction of the landing gear following partial loss of the hydraulic systems, without having to resort to the emergency mode.

The flight crew request extension and retraction of the landing gear using the Landing Gear Control Lever. The Landing Gear Control Lever has two positions: Down for extension and Up for retraction.

The Landing Gear state is indicated to the flight crew by two separate displays:

1. A Colour Multi-Function Display Unit (CMFDU)
2. A dedicated landing gear panel

The position of the nose, wing and body landing gears and doors may be displayed on a CMFDU.

A dedicated landing gear panel, positioned on the central instrument panel in the cockpit, also displays the unlock/downlock state for each gear, independently of the CMFDU. Each gear has its own indication light.

3.1 Safety Issues

The main aircraft-level hazards to which the landing gear system can contribute are:

- Uncommanded retraction of the landing gear (catastrophic, particularly on the ground)
- Inability to extend the landing gear (catastrophic)
- Deployment of the landing gear at high speed (hazardous)

3.2 System Architecture

The system architecture is shown in Figure 1. The landing gear software runs on four processors. These are PowerPC microprocessors running a real-time operating system. The processors are divided into two identical sides (or lanes). Only one side controls the landing gear at any time; the other side has the ability to assume control of the landing gear in the event of a failure. All four processors run the same software; the software is configured by pin programming.

Both processors in a side must agree before an output is asserted. Should a side fail, control is switched to the other side. In any case, control is switched from side to side whenever the Landing Gear Control Lever is operated, ensuring that both sides are exercised regularly so that dormant faults are detected.

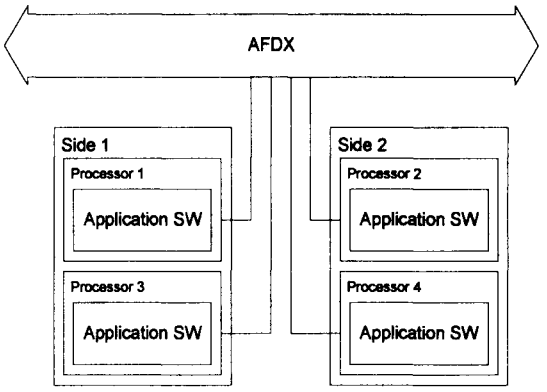


Figure 1. System architecture.

3.3 Software Architecture

The landing gear software consists of two Computer Software Configuration Items (CSCIs).

Tuning data is supplied via a separately loadable application database per partition. This allows the software to be tuned without having to update the object code, reducing re-verification overhead.

All the software was written in a subset of C, running under a real-time operating system that complies with ARINC 653 (ARINC 1997). ARINC 653 is described in section 4. The operating system also supports a number of extensions to ARINC 653.

The software architecture is shown in Figure 2.

3.4 Safety Features

As already discussed in section 3.2, the landing gear system uses hardware redundancy to mitigate some of the hazards associated with a landing gear control system. The system architecture uses four processors.

The landing gear design also mitigates against Single Event Upsets (SEUs). We carried out an analysis of the design to confirm that it provides adequate protection. The use of redundant processors provides protection against SEUs. The main memory is error-correcting memory. As far as possible, all decisions are refreshed every cycle so that an SEU can only affect a single cycle. We protect persistent data from SEUs by maintaining multiple copies of the data. We also maintain a count of the number of SEUs that we have observed.

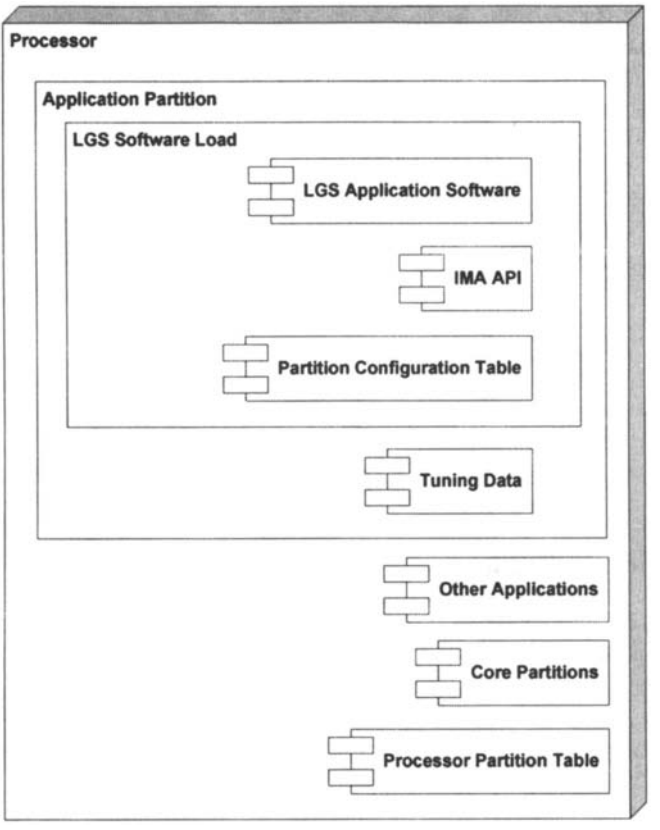


Figure 2. Software architecture.

Some of the application software was developed to DO-178B Level A; the remainder was developed to DO-178B Level C. All safety requirements were clearly labelled as such and traced to the design, implementation and test procedures.

If the landing gear system were to fail so that normal landing gear extension is not possible, the flight crew can still lower the landing gear using a separate mechanical backup system.

4 ARINC 653

The landing gear software runs under a bespoke operating system. This embedded real-time operating system extends the ARINC 653 standard and has been qualified to DO-178B Level A for this aircraft programme.

ARINC 653 is a very important new standard for IMA applications. ARINC 653 specifies the baseline operating-environment for application software used within an IMA system. It defines a general-purpose APEX (Application Executive) interface between the operating system and the application software. The APEX interface is language and hardware independent. This allows application software developed for one aircraft to be ported to other aircraft types with minimal recertification effort. Unlike general-purpose standards such as POSIX, ARINC 653 is designed to support the specific needs of safety-critical, hard-real-time avionic applications.

One of the key features of ARINC 653 is partitioning. Partitioning is fundamental to the IMA concept as it allows several software applications, potentially of differing Safety Integrity Levels (SILs), to execute on the same processor, with complete spatial and temporal isolation between them. Partitioning guarantees that an application running in one partition cannot have an adverse effect on an application running in another partition, either by overwriting memory used by the other application or by stealing processor cycles from the other application.

4.1 Temporal Partitioning

Partitions within an IMA system are scheduled on a fixed, cyclic basis. An ARINC 653 scheduler uses the concept of a Major Frame. A Major Frame is a repeating, fixed-length period during which each partition is executed at least once. Each partition is allocated to one or more slices of a Major Frame, each slice being defined by its offset from the start of the Major Frame and its expected duration. The order in which the partitions are scheduled is defined at configuration time using configuration tables (see later in section 4.7). This provides a deterministic scheduling scheme in which each partition is allocated a predetermined amount of time in which to run. Temporal partitioning guarantees each partition uninterrupted access to the processor for its allotted time. For example, a partition that is scheduled to run for 2 ms every 20 ms will run at precisely that interval and is guaranteed not to be pre-empted during its 2 ms slice. Figure 3 illustrates how partitions are scheduled in ARINC 653.

The operating system also allowed each Major Frame to be further subdivided into Minor Frames. This is an extension to ARINC 653, which we will not discuss any further in this paper.

A partition may contain a single process, or several processes. Processes are the smallest threads of control handled by the operating system. Pre-emptive, priority-based scheduling is used to schedule processes within a partition.

4.2 Spatial Partitioning

ARINC 653 uses the processor's Memory Management Unit (MMU) to ensure that applications running in different partitions cannot overwrite each other's memory space.

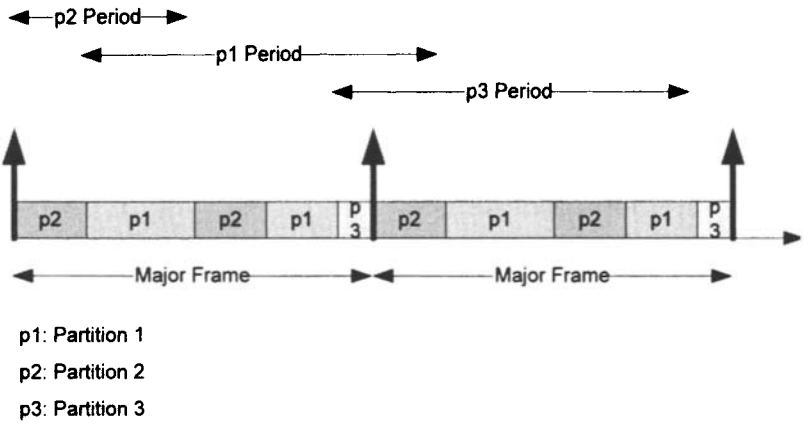


Figure 3. ARINC 653 scheduling.

4.3 Inter-Partition Communication

All inter-partition communication is done by messages. Partitions communicate with each other via communication ports provided by the operating system API. To send data, a partition writes a message to an API output port. The mechanism used by the operating system to send the message depends on whether the message is to be sent to another partition running on the same processor, a partition running on another processor or to an interface device. It is a very useful feature of ARINC 653 that the same interface is used in all these cases, making it relatively easy to move applications between processors and to substitute software simulations of hardware devices for testing.

Each port may be configured to work in either sampling mode or queuing mode. In sampling mode, successive messages carry identical, but updated data. No queuing is performed this mode. If a second message is sent before the previous message has been read, the previous message is overwritten. This makes sampling mode suitable for data that is refreshed regularly, such as sensor inputs.

In queuing mode, incoming messages are queued rather than overwriting the previous message. This makes queuing mode suitable for general-purpose message transmission.

Our landing gear software uses both queuing and sampling ports.

4.4 Intra-Partition Communication

Messages are addressed to partitions, not to individual processes within a partition. Processes within a partition can communicate with each other using the following mechanisms, avoiding the overhead of the inter-partition message-passing scheme:

- Buffers
- Blackboards
- Semaphores
- Events

Buffers are used to store multiple messages in message queues. Messages are stored in First In/First Out (FIFO) order.

Blackboards do not use queues. A message written to a blackboard remains there until either it is cleared or it is overwritten by a later message.

ARINC 653 semaphores are counting semaphores. They are commonly used to control access to shared partition resources. A process waits on a semaphore to gain access to a resource and signals the semaphore when it is done.

ARINC 653 events are binary semaphores. An event is a variable that can be in one of two states, "up" or "down".

Our landing gear software did not use any of the intra-partition communication mechanisms.

4.5 Inter-Processor Synchronisation

The fixed-frame scheduling scheme in ARINC 653, whilst it has many advantages, does make it difficult to retain tight synchronisation between applications running on multiple processors. If the processor clocks drift apart, the applications may become unable to synchronise within their allotted time slice. This problem is illustrated in Figure 4.

The operating system provided a mechanism to overcome this problem by synchronising multiple processors using a synchronisation signal transmitted over a dedicated RS485 line. This mechanism ensures that the frames (actually, the minor frames) on each processor start at the same time, but of course, introduces a potential common point of failure.

Fortunately, the landing gear moves very slowly in relation to the software cycle time. The four application partitions only needed to be very loosely synchronised, so no special synchronisation mechanism was needed.

4.6 Health Monitoring

The part of the operating system responsible for monitoring and reporting hardware, software and operating system faults is called the Health Monitor. The Health Monitor helps to isolate faults and to prevent failures from propagating.

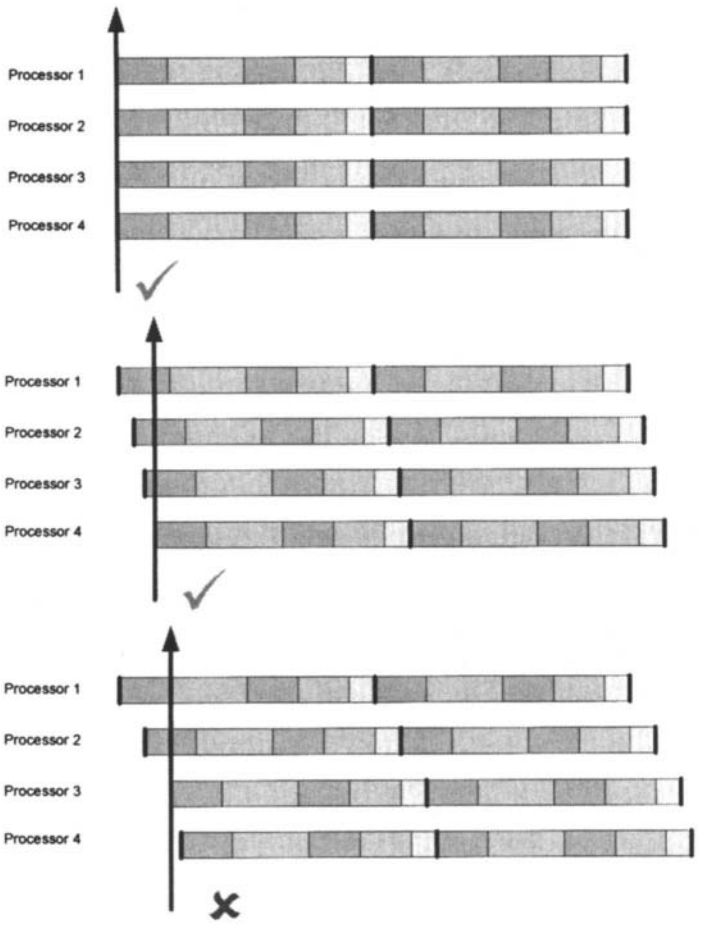


Figure 4. As the processor clocks drift apart, the applications become unable to synchronise within their allocated time slice.

Errors may occur at the processor, partition or process level. A processor-level error affects all the partitions running on that processor. A partition-level error affects only one partition. A process-level error may affect just one process, or it may affect all processes running in the same partition.

When the software detects an error, it reports the error to the Health Monitor, which logs the error and performs a recovery action. Configuration tables are used to determine the recovery action that the Health Monitor takes for each error (see section 4.7).

4.7 Configuration Tables

Configuration tables are static data areas that are used by, but are separate from, the operating system. Configuration tables are used in ARINC 653 for the following purposes:

1. System initialisation. A configuration table defines the partitions that are resident on the module, the memory required by each partition and the number of ports required by each partition.
2. Inter-partition communication. When a partition writes a message to a port, the operating system uses a configuration table to determine the destination of the message. The destination could be a partition running in the same module, or in another module.
3. Health Monitor. The Health Monitor uses configuration tables to determine how to handle each error. For example, one such table defines the recovery action to take (e.g. shut down the module, reset the module).

4.8 ARINC 653 Contrasted with Ada Tasking

ARINC 653 was developed with the Ada programming language in mind, although it also allows applications to be written in other languages, such as C. ARINC 653 does not support Ada tasking, but introduces specific API calls to support inter-process communication. This was because, when ARINC 653 was published in 1997, Ada tasking did not provide the predictable, deterministic scheduling required for safety-critical, hard-real-time programming.

The ARINC 653 specification states, "Since Ada is the preferred development language for airline avionics, it is possible to use the Ada tasking construct to define each process. However, this is an undesirable approach since the tasking construct as defined by the language itself is not sufficiently deterministic, nor specific enough, for the requirements of IMA. Therefore, the model specified herein defines a process with nearly identical characteristics of an Ada task without using the Ada tasking construct".

Since the ARINC 653 specification was published, the Ravenscar profile (Burns, Dobbing and Romanski 1998) has been produced for Ada tasking. The Ravenscar Profile is a subset of the Ada tasking model that is deterministic, schedulable and memory-bounded. It was designed specifically to support the development of safety-critical, hard-real-time Ada programs.

It is therefore instructive to compare ARINC 653 with the Ravenscar profile.

4.8.1 Advantages of ARINC 653

ARINC 653 allows several applications to run on the same processor, each in its own partition. These applications may be from multiple vendors and may or may

not communicate with each other. The Ravenscar profile is concerned with writing a single Ada program, divided into a number of tasks.

ARINC 653 provides temporal partitioning. The operating system prevents a partition from exceeding its time slice, ensuring that the next partition is always scheduled at the expected time. A partition can query the operating system to determine whether it attempted to exceed its time slice the last time it was scheduled, and take corrective action if necessary. This means that each partition can be assured of satisfying its real-time constraints, regardless of the behaviour of the other partitions executing on the same processor. This allows applications of differing SILs or DO-178B Software Levels to run on the same processor. The Ravenscar profile depends upon the worst-case execution times having been calculated correctly and all potential deadlocks having been identified and removed, which is admittedly not a problem when all the applications running on a processor are of a high SIL.

ARINC 653 provides spatial partitioning. An ARINC 653-compliant operating system uses the processor's MMU to ensure that an application running in a partition cannot inadvertently overwrite memory used by another partition or by the operating system. Again, this allows applications of differing SILs to run on the same processor. There is no memory protection between tasks in an Ada program using the Ravenscar profile. If the program were written in SPARK (Amey and Chapman 2005), then static analysis could be used to demonstrate non-interference between tasks, although this still does not mitigate against Single Event Upsets and other random failures.

ARINC 653 is independent of the programming language used. Although ARINC 653 was developed with Ada specifically in mind, it also allows the use of other programming languages such as C. The Ravenscar profile assumes that the entire program is written in Ada.

ARINC 653 allows an application to be split across multiple partitions, each containing software of a different SIL. For example, the landing gear application consists of four DO-178B Level A partitions and two DO-178B Level C partitions. This feature of ARINC 653 reduced the software development effort because the alternative would have been to have written the entire application as a single DO-178B Level A program.

ARINC 653 provides great flexibility in system design. Multiple applications (of mixed SILs) can be run on the same processor. Partitions can be moved between processors without altering the source code. The API and strict isolation from the hardware means that there is a natural boundary at which hardware-software integration takes place.

4.8.2 Disadvantages of ARINC 653

Fixed-frame scheduling uses the CPU very inefficiently. The fixed time slice allocated to each partition has to be long enough to accommodate the Worst Case Execution Time (WCET). If a partition completes in less than the worst case time, then the unused CPU time cannot be used by another real-time partition (only by

the idle process) and the processor idle waits until the start of the next slice. Fixed frame schedulers typically achieve about 50% CPU utilisation.

ARINC 653 requires applications to be written as a number of separate programs (running in separate partitions), each of which run to completion each Major Frame. In the author's opinion, this is less intuitive than the Ada tasking model and can distort the software design.

Fixed-frame scheduling makes it difficult to maintain tight synchronisation between applications running on multiple processors.

4.8.3 Advantages of Ravenscar tasking

Ravenscar programs are amenable to static code analysis. A Ravenscar program can be written in SPARK and analysed as a single program. While the programs running in individual ARINC 653 partitions could also be written in SPARK, they could only be analysed as individual programs. The author is not aware of any static code analysis tools that would allow all the partitions making up an ARINC 653 application to be analysed as a single entity.

4.9 Experience of Using ARINC 653

We used a large number of processes in our original design. We soon simplified the design to use fewer processes.

Error handling was very similar across partitions. There is scope for a standard library to be developed that could be tuned using configuration tables.

The software planning stage needs to take into account the effort required to develop the ARINC 653 configuration tables. New tools, skills and methods were involved in developing these tables and a number of people were involved in their production.

ARINC 653 allows an application to be divided into a number of partitions of differing DO-178B software levels, reducing the amount of code that needs to be developed to DO-178B Level A. When developing the landing gear software, we were able to keep the Level C code separate from the Level A code, which was certainly advantageous. However, we found that ARINC 653 partitions are quite a blunt instrument for this purpose. While ARINC 653 allows multiple processes within a partition, the processes do not enjoy the temporal and spatial independence enjoyed by separate partitions, so all the processes within a partition have to be developed to the same software level. A new technology that allows software of differing software levels to reside within the same partition is the extension to SPARK described in (Amey, Chapman and White 2005). Moving a piece of code into a separate partition is a significant design step and not one to be taken lightly. In addition, one needs to be careful with data coupling between partitions. For example, one needs to ensure that a Level A partition cannot exhibit unsafe behaviour even when it is passed incorrect data by a Level C partition.

ARINC 653 certainly provides flexibility in system design. It allowed the landing gear application to be divided easily into two CSCIs running in six

partitions on four separate processors, which were able to communicate easily with each other using message passing. This gave us two advantages:

1. We were able to develop the DO-178B Level A software separately from the DO-178B Level C software, reducing software development and certification costs.
2. We were able to take advantage of the four processors to provide tolerance of hardware faults and of single event upsets.

However, the large number of stakeholders on an aircraft programme reduces flexibility as the development progresses.

The bespoke operating system supported application databases. We should have paid more consideration to these during the initial design to identify tuning and data that could have been isolated to reduce regression costs further. We did not exploit this useful feature as fully as we might have done.

ARINC 653 also allowed the application to be integrated and tested very easily by simulating the hardware devices and the other partitions. System testing was performed in two stages:

1. the landing gear software running stand-alone
2. the landing gear software integrated with other applications running on the shared hardware

The tool chain is larger than that used on traditional, federated developments. This requires more training and familiarisation. The tool chain evolved during the development, which also required careful attention.

The use of ARINC 653 meant that we were able to concentrate on the application logic. The platform supplier provided all the low-level software, including the Board Support Package (BSP) and the device drivers.

ARINC 653 provided a measure of isolation from the hardware platform, increasing portability.

5 Multi-National Working

We carried out the software development activity at three locations, on two continents:

1. The landing gear system supplier's premises, in the UK
2. Silver Software, Malmesbury, UK
3. Silver Software, Bangalore, India

The work was allocated between sites according to the degree of interaction required with our customer's engineers:

1. Software requirements analysis was carried out jointly with the systems supplier.

2. Design and implementation of the Level A CSCI was carried out at Silver Software's offices in Malmesbury.
3. Design and implementation of the Level C CSCI was carried out at Silver Software's offices in Bangalore.
4. Code scrutiny and unit test were largely carried out in Bangalore.
5. Software integration was carried out in the UK and in India.
6. Hardware/software integration testing and system testing were carried out at the system supplier's premises.

The IMA partitioning facilitated the work split because each team developed an encapsulated, cohesive application that was loosely coupled to the other partitions via the operating system. This, in turn, helped the programme schedule as the separate teams could work in parallel.

We purposely rotated software engineers between the system supplier's premises, Malmesbury and Bangalore to ensure that the engineers working on the project came to know each other well, shared a common working culture and understood the needs of the client.

This split of work worked very well and combined the low cost of software engineering in India with the need to work closely with the client to ensure that his needs were satisfied.

6 Conclusions

This was one of the first implementations of IMA on a civil aircraft. It is also one of the first aircraft to use the new ARINC 653 operating system standard.

Most airframe manufacturers are now adopting ARINC 653. For example, both Airbus A380 and Boeing 787 Dreamliner use ARINC 653. There is a strong trend towards adopting IMA throughout the aerospace industry.

Silver Software found the use of IMA and of ARINC 653 to be of great assistance in developing such a complex system while ensuring safety, fault tolerance, maintainability and portability. There is a lack of real-world experience of ARINC 653 and of design patterns for IMA. We compensated by keeping things simple. We could have made more use of ARINC 653 facilities, but we were cautious.

We carried out the software development at three locations, in the UK and in India. This split of work worked very well and resulted in a significant cost saving for our client.

References

Amey P and Chapman R (2005). SPARK 95 — The SPARK Ada 95 Kernel (including RavenSPARK). Praxis High Integrity Systems, Bath, 2005.
Amey P, Chapman R and White N (2005). Smart Certification of Mixed Criticality Systems. In: Vardanega T and Wellings A (ed) Reliable Software Technology —

Ada-Europe 2005. Springer-Verlag Heidelberg, 2005, pp144–155 (Lecture Notes in Computer Science, Volume 3555).
ARINC (1997). Avionics Application Software Standard Interface. ARINC 653. Aeronautical Radio, Inc., Annapolis, 1997.
Burns A, Dobbing B, Romanski G (1998). The Ravenscar Tasking Profile for High Integrity Real-Time Programs. In: Asplund L (ed) Reliable Software Technologies — Ada-Europe '98. Springer-Verlag, Heidelberg, 1998, p263 (Lecture Notes in Computer Science, Volume 1411).

NEW TECHNOLOGIES IN SAFETY-CRITICAL SYSTEMS

Optimising Data-Driven Safety Related Systems

Richard Everson, Jonathan Fieldsend, Trevor Bailey, Wojtek Krzanowski, Derek Partridge, Vitaly Schetinin[1] and Adolfo Hernandez

School of Engineering, Computer Science and Mathematics, University of Exeter, Exeter, UK.

Abstract

The operation of many safety related systems is dependent upon a number of interacting parameters. Frequently these parameters must be 'tuned' to the particular operating environment to provide the best possible performance. We focus on the Short Term Conflict Alert (STCA) system, which warns of airspace infractions between aircraft, as an example of a safety related system that must raise an alert to dangerous situations, but should not raise false alarms. Current practice is to 'tune' by hand the many parameters governing the system in order to optimise the operating point in terms of the true positive and false positive rates, which are frequently associated with highly imbalanced costs.

We regard the tuning of safety related systems as a multi-objective optimisation problem. We show how a region of the optimal receiver operating characteristic (ROC) curve may be obtained, permitting the system operators to select the operating point. We apply this methodology to the STCA system, showing that we can improve upon the current hand-tuned operating point, as well as providing the salient ROC curve describing the true positive versus false positive trade-off. We also address the robustness of the optimal ROC curve to perturbations of the data used to learn it. Bootstrap resampling is used to evaluate the uncertainty in the optimal operating curve and show how the probability of a particular operating point can be estimated.

1 Introduction

The operation of many safety critical and safety related systems depends upon a number of parameters that determine the system behaviour. Although appropriate values for some of these parameters may be known from first principles or from measurement, others must be inferred from data. The particular example with which this work is concerned is the Short Term Conflict Alert (STCA) system in operation in the United Kingdom and elsewhere. STCA monitors aircraft locations from ground radar and provides advisory alerts

[1] Present address: Department of Computing, University of Luton, Luton, UK.

to air traffic controllers if a pair of aircraft are likely to become dangerously close. The STCA system is designed to raise a warning to air traffic controllers if there is a developing conflict between aircraft, giving them time to redirect the aircraft. As we describe below, the system in operation for the airspace above London, handling at least 2500 aircraft per day, has approximately 1550 parameters which may be adjusted to affect the circumstances in which STCA raises an alert. Typical of these are the 'vertical closing rate threshold' (1600 to 3000 ft/min) and 'total reaction time before lateral manoeuvre' for which approximate ranges are respectively 1600 to 3000 ft/min and 22 to 55 seconds. However, precise values are not *a priori* known.

Determining appropriate parameter values is made more difficult by the fact that a balance must be struck between the true positive alert rate and the false positive alert rate. If the false positive rate is allowed to become too great air traffic controllers will be disinclined to respond to genuine alerts raised by the system; on the other hand the STCA system should raise alerts when a pair of aircraft are in genuine danger of becoming too close.

Current practise at the National Air Traffic Services (NATS, the principal civil air traffic control service for the United Kingdom) is to incrementally adjust the STCA system parameters manually in order to reduce the false positive rate, while maintaining the true positive rate. This tuning is performed by staff on the basis of a large (170 000) database of track pairs containing recent and historical aircraft encounters. However the 1500 adjustable parameters make tuning a highly skilled and laborious procedure.

Here we report on a method for automatically determining the optimal trade-off curve between true and false positive rates. This optimal *Receiver Operating Characteristic* (ROC) curve has not been previously available and knowledge of it permits a principled choice of operating point to be made. It is clearly important to be confident of the STCA system operation and we present methods to assess the variability of the true and false positive rates.

We first describe the STCA system and its parameterisation in more detail. In section 2 we cast the STCA system as a classifier and describe how multi-objective optimisation may be used to locate the optimal ROC curve describing the trade-off between true and false positive rates; we illustrate the procedure on data from the London airspace. It is important to assess how robust the optimised solutions are to changes in the data, and we address this problem in section 3 before the concluding with a discussion.

The STCA system comprises a complex and proprietary algorithm. However, its main components are readily understood. Signals from ground radars track the aircraft in an airspace and on each cycle of the STCA system (every 4 seconds) *track pairs* for each pair of aircraft being monitored by the system are created. A *coarse* filter discards all those pairs for which the aircraft are simply too distant to conceivably constitute a potential hazard. The remaining pairs in potential conflict are passed to three *fine filters*: a current proximity filter; a linear prediction filter; and a manoeuvre hazard filter. These check for whether the aircraft are already too close, whether they will lose separation

if they continue on their current headings at their current speeds, and for potential loss of separation if either aircraft is turning. The combinations of the binary classifications from the fine filters is performed by an alert confirmation module, which alerts the air traffic controller if there is a potential conflict signalled by the fine filters for a number of successive cycles.

There are 96 adjustable parameters which control the operation of the fine filters. However, the complexity of the system is magnified by the fact that the airspace is divided into different region types, for example, *en route* or *stack*. Since aircraft in the different region types tend to have different flight behaviour, separate parameter sets are used for aircraft in each region type; there are additional rules determine the relevant parameter sets if the aircraft comprising a track pair have different region types. The fact that aircraft forming a pair may be have different region types means that the parameters for each region type cannot be adjusted independently of the others. There are 16 separate region types for the STCA system monitoring the London airspace, leading to approximately 1550 parameters which may be adjusted to control the STCA system behaviour. In fact, we tune only the approximately two-thirds of the available parameters that are routinely adjusted by NATS staff.

The STCA system is in operation in the four UK air traffic control centres and at other air traffic control centres in Europe, so appropriate parameter setting must be chosen for each particular locale. Moreover, changes in the volume of air traffic, changes in local air traffic operational procedures and changes in the regulatory environment mean that the STCA operational parameters must be reviewed and updated in order to prevent the system becoming out of date. In the UK all serious near-miss encounters are reviewed under the auspices of the Airprox Board (see for example [1]). In addition NATS regularly assesses the efficacy of the STCA system by running an off-line version with a database comprised of recent general traffic encounters together with historical serious encounters. Track pairs in this database are manually annotated into five severity categories: in this work we group these into those for which an alert should be raised (a genuine alert) and those for which an alert should not be raised (a nuisance alert). The STCA system may thus be regarded as a binary classifier, which should discriminate between track pairs for which an alert should be raised and those which no alert should be raised. Viewing the STCA as a classifier allows it to be analysed and optimised using Receiver Characteristic analysis as we now describe.

2 ROC analysis & Pareto optimality

In general we consider a classifier $g(\mathbf{x};\boldsymbol{\theta})$ which gives an estimate of the probability that a feature vector $\mathbf{x}$ belongs to one of two classes. We assume that the classifier depends upon a vector of adjustable parameters $\boldsymbol{\theta}$, and we denote

by $T(\boldsymbol{\theta})$ the classifier's true positive classification rate, while the false positive rate is denoted by $F(\boldsymbol{\theta})$.

If the costs of an incorrect classification were known it would be straightforward to calculate the expected cost for any particular parameter and data set [7]. It would then be possible to adjust the parameters to minimise the expected cost. However, this procedure requires accurate specification of the misclassification costs which are seldom known accurately. Only the ratio of misclassification costs is important and the ROC curve displays the trade-off between true and false positive rates as this ratio is varied for fixed parameters (see [9] for a recent review of ROC methods). As the cost ratio is varied a non-decreasing ROC curve in the (F, T) plane is obtained for any particular fixed set of parameters, and different ROC curves are obtained for different parameters. With the ROC curves on hand the user can select the operating point with a full knowledge of the possible trade-offs involved.

Of course, all these measures based upon the ROC curve require knowledge of the ROC curve, which hitherto has been unavailable for the STCA system. In this section we show how multi-objective evolutionary algorithms may be used to derive the ROC curve for the STCA system optimised over all possible parameter values. That is, we seek to discover the set of parameters that simultaneously minimise $F(\boldsymbol{\theta})$ and maximise $T(\boldsymbol{\theta})$.

A general multi-objective optimisation problem seeks to simultaneously extremise D objectives:

$$y_i = f_i(\boldsymbol{\theta}), \qquad i = 1, \ldots, D \tag{1}$$

where each objective depends upon a vector $\boldsymbol{\theta}$ of P parameters. It is convenient to assume that all the objectives are to be minimised, so for the STCA system we minimise the pair of objectives $(-T(\boldsymbol{\theta}), F(\boldsymbol{\theta}))$. The parameters may also be subject to the J constraints:

$$e_j(\boldsymbol{\theta}) \geq 0, \qquad j = 1, \ldots J \tag{2}$$

so that the multi-objective optimisation problem may be expressed as:

$$\text{minimise} \qquad \mathbf{y} = \mathbf{f}(\boldsymbol{\theta}) = (f_1(\boldsymbol{\theta}), \ldots, f_D(\boldsymbol{\theta})) \tag{3}$$

$$\text{subject to} \quad \mathbf{e}(\boldsymbol{\theta}) = (e_1(\boldsymbol{\theta}), \ldots, e_J(\boldsymbol{\theta})) \geq 0 \tag{4}$$

where $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_P)$ and $\mathbf{y} = (y_1, \ldots, y_D)$.

When faced with only a single objective an optimal solution is one which minimises the objective given the constraints. However, when there is more than one objective to be minimised solutions may exist for which performance on one objective cannot be improved without sacrificing performance on at least one other. Such solutions are said to be *Pareto optimal* [4, 14] and the set of all Pareto optimal solutions is said to form the Pareto front.

The notion of *dominance* may be used to make Pareto optimality clearer. A decision vector $\boldsymbol{\theta}$ is said to *strictly dominate* another $\boldsymbol{\phi}$ (denoted $\boldsymbol{\theta} \prec \boldsymbol{\phi}$) iff

Algorithm 1 Multi-objective optimisation of STCA.

```
1:  A := initialise()
2:  for n := 1 : N                          Loop for N generations
3:      θ := select(A)                      Select parent to perturb
4:      θ' := perturb(θ)                    Perturb parameters
5:      (T(θ'), F(θ')) := STCA(θ')          Evaluate classification rates
6:      if θ' ⊀ φ ∀φ ∈ A
7:          A := {φ ∈ A | φ ⊀ θ'}           Remove dominated elements
8:          A := A ∪ θ'                     Insert θ'
9:      end
10: end
```

$$\begin{aligned} f_i(\boldsymbol{\theta}) \leq f_i(\boldsymbol{\phi}) \quad \forall i = 1, \ldots, D \quad \text{and} \\ f_i(\boldsymbol{\theta}) < f_i(\boldsymbol{\phi}) \quad \text{for some } i. \end{aligned} \tag{5}$$

Less stringently, $\boldsymbol{\theta}$ *weakly dominates* $\boldsymbol{\phi}$ (denoted $\boldsymbol{\theta} \preceq \boldsymbol{\phi}$) iff

$$f_i(\boldsymbol{\theta}) \leq f_i(\boldsymbol{\phi}) \quad \forall i = 1, \ldots, D. \tag{6}$$

A set A of decision vectors is said to be a *non-dominated set* if no member of the set is dominated by any other member:

$$\boldsymbol{\theta} \not\prec \boldsymbol{\phi} \quad \forall \boldsymbol{\theta}, \boldsymbol{\phi} \in A. \tag{7}$$

A solution to the minimisation problem (3) is thus *Pareto optimal* if it is not dominated by any other feasible solution, and the non-dominated set of all Pareto optimal solutions is the Pareto front. Recent years have seen the development of a number of evolutionary techniques based on dominance measures for locating the Pareto front; see [4, 6, 16] for recent reviews.

2.1 ROC optimisation

Anastasio & Kupinski [13] and Anastasio, Kupinski & Nishikawa [3] introduced the use of multi-objective evolutionary algorithms to optimise ROC curves, illustrating the method on a synthetic data and for medical imaging problems.

The multi-objective evolutionary algorithm used here is a stochastic search algorithm, based on a simple $(1 + 1)$-evolution strategy (ES), similar to that introduced in [12]. In outline, the procedure for locating the Pareto front/ROC curve, operates by maintaining an archive, A, of mutually non-dominating solutions, $\boldsymbol{\theta}$, which is the current approximation to the Pareto front/ROC curve. At each stage of the algorithm some solutions in A are copied and perturbed. Those perturbed solutions that are dominated by members of A are discarded, while the others are added to A and any dominated solutions in A are removed.

In this way the estimated Pareto front A can only advance towards the true Pareto front. This algorithm, unlike earlier versions [12], maintains an archive which is unrestricted in size, permitting better convergence [11].

Algorithm 1 describes in more detail the algorithm as applied to the optimisation of the STCA system. Following the current operating practise of NATS, we choose to optimise only 912 of the $\approx$ 1550 available parameters; these parameters are those parameters which have different values in different regions after tuning by NATS. Furthermore we restrict these parameters to the ranges over which they are adjusted by NATS.

The archive or frontal set A is initialised by drawing parameters for the STCA system uniformly from their feasible ranges; in addition the current 'best' parameter set from manual tuning $\boldsymbol{\theta}^\star$ is added to A. Of course many of these randomly selected parameter vectors are dominated by other parameter vectors and these dominated parameters are deleted from A so that A is a non-dominated set (7). In fact, in the work reported here, we found that of 100 randomly initialised parameters only $\boldsymbol{\theta}^\star$ and one other parameter vector remained in A after dominated parameter vectors were removed.

Following initialisation, the loop on lines 2–10 of Algorithm 1 is repeated for N iterations. At each iteration a single parameter vector $\boldsymbol{\theta}$ is selected from A; selection may be uniformly random, but partitioned quasi-random selection (PQRS) [11] was used here to promote exploration of the front. The selected parent vector is perturbed to generate a single *child* (line 4). Each individual parameter in the parent vector is perturbed with equal probability (0.2 here, selected following a small empirical study); the perturbations themselves are made by adding a random number to the parent parameter value. Yao *et al.* [15] have shown that perturbations drawn from heavy-tailed distributions facilitate convergence by promoting exploration and escape from local minima. We therefore draw perturbations from a Laplacian density, $p(x) \propto e^{-|x/w|}$, whose width is set equal to one tenth the feasible range of the parameter being perturbed; perturbations that lie outside the feasible range are resampled.

The true $T(\boldsymbol{\theta}')$ and false $F(\boldsymbol{\theta}')$ positive rates for the perturbed vector are evaluated by running the STCA system with parameters $\boldsymbol{\theta}'$ on the test database of track pairs. If the child $\boldsymbol{\theta}'$ is not dominated by any of the parameter vectors in A, any parameter vectors in A that $\boldsymbol{\theta}'$ dominates are deleted from the archive (line 7) and $\boldsymbol{\theta}'$ is added to A (line 8). These two steps ensure that A is always a non-dominated set whose members dominate any other solution encountered thus far in the search.

2.2 Results

We present a conservative application of the evolutionary scheme to STCA optimisation. It is conservative in that the ranges of parameters to be varied are limited by the current ranges of that parameter across the 16 region types within the current STCA parameterisation used by NATS so that the parameters are confined to regions of decision space with which personnel at

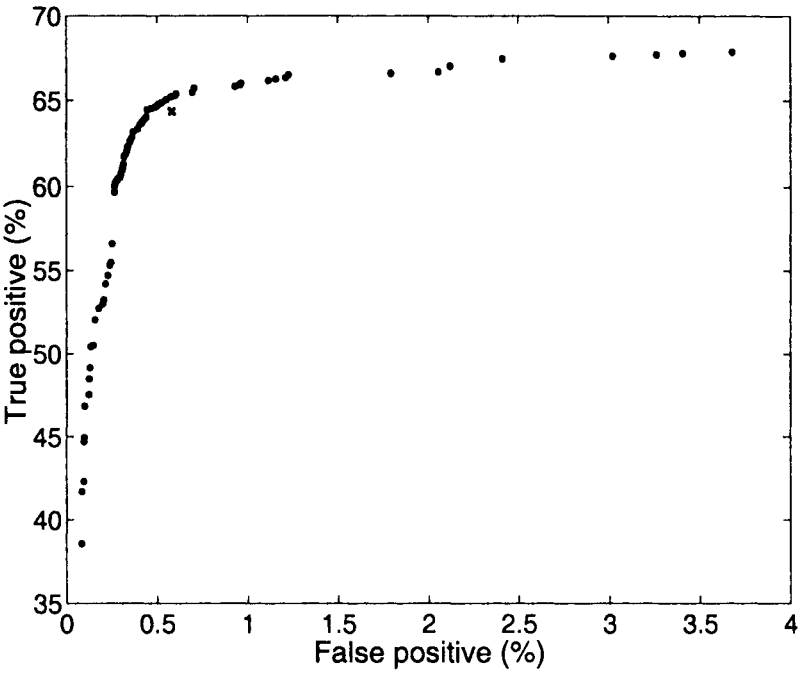


Fig. 1. Dots show estimates of the Pareto optimal ROC curve for STCA obtained after 6000 evaluations of the (1+1)-ES multi-objective optimiser. The cross indicates the manually tuned operating point $\boldsymbol{\theta}^\star$.

NATS have considerable experience. Although we could adjust more parameters and adjust parameters over a greater range, the strategy adopted here provides an assurance that the optimised system is still operating within the usual parameter ranges.

We optimised the true and false positive rates for a database comprised of manually and semi-automatically categorised encounters. The database included historical track pairs leading to serious or potentially serious encounters together with general traffic track pairs from two weeks in 2001.

Even this conservative optimisation approach produces some striking results. Figure 1 shows the estimates of the Pareto optimal ROC curve obtained using the multi-objective optimiser after $N = 6000$ evaluations (approximately 12 days computation). The current NATS operating point is also plotted as a cross. The optimisation has located an ROC curve consisting of 76 points ranging from 38.5% to 67.9% true positive and 0.1% to 3.7% false positive. In addition the manually tuned STCA operating point $\boldsymbol{\theta}^\star$ lies *behind* (is dominated by) several operating points on the estimated ROC curve. Although the improvement over $\boldsymbol{\theta}^\star$ is relatively small in percentage terms, the quantity of track pairs processed by the STCA system means that a significant

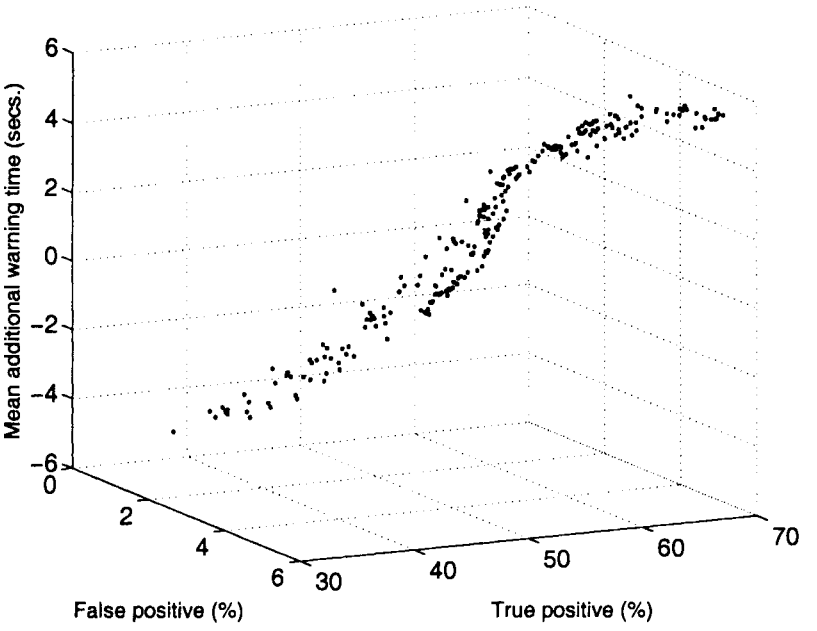


Fig. 2. Estimated Pareto front optimising warning time together with true and false positive rates.

reduction in the *number* of false alerts could be achieved while maintaining the current genuine alert rate. We regard as more important, however, the production of the ROC curve itself, because it reveals the true positive versus false positive trade-off, permitting the operating point to be chosen. In fact it may be observed that the current operating point $\boldsymbol{\theta}^\star$ is close to the corner of the Pareto optimal curve. Choosing an operating point to the left of the corner would result in a rapidly diminishing genuine alert rate for little gain in the nuisance alert rate; whereas operating points to the right of the corner provide small increases in the true positive rate at the expense of relatively large increases in the false positive rate.

2.3 Warning time optimisation

In addition to the trade-off between correct alerts and incorrect alerts, it is desirable to increase the warning time of genuine alerts given to air traffic controllers. Current practise is to compare a new parameter set with the current operating point by calculating the mean increase or decrease in warning times over the coincident genuine warnings of the two parameter sets. Using the same method we can compare all our frontal operating points with the current operating point. Furthermore we can use this extra objective to create a three-objective optimisation problem in which we seek to maximise the

mean warning time and true positive rate, while minimising the false positive rate.

Again we use a $(1+1)$-ES, with the same parameters as the previous experiment. We initialise the algorithm with the frontal points discovered in the previous optimisation (which by definition also form an estimated Pareto front in the 3 objective case). The front located after 5000 generations looks like a twisted ribbon, as shown in Figure 2. As before the current operating point $\boldsymbol{\theta}^\star$ lies behind the discovered front. We remark that this required approximately 10 days computation as evaluation of the warning time is negligible in comparison with the alert rate computation. However, close examination of the three dimensional front shows that significant gains in warning time can only be achieved if the false positive rate is substantially increased.

3 Robustness of optimised solutions

As we described above, the location of the Pareto front is based upon evaluating the STCA system on a representative sample of encounters, and although approximately 170 000 encounters were used, it is important to discover the sensitivity of any putative operating point to the data sample. Indeed, it is especially important not to over-train the system to one particular set of data. Ideally one would optimise the entire STCA system on several independent data sets collected at different times. This, however, is impractical both because of the expense in collecting and annotating the data and because of the computational expense of multiple optimisations (although this cost might be reduced by initialising new optimisations from fronts obtained in earlier optimisation runs). A further consideration is that serious encounters are (fortunately) rare, so that although independent sets of general traffic may be obtained, the serious encounters would have to be reused. For these reasons we employ a bootstrapping technique [8, 5] in order to estimate the variability in error rates around the front.

We were also provided with a second set of general traffic for 5th-20th September 2000. Here we analyse the effect of using these general traffic data instead of the general traffic data for 1st-14th July 2001, but keeping the historical serious traffic data unchanged. Using the solutions obtained for the three-objective optimisation on the original data but evaluating the true and false positive rates and warning time on the second general traffic and historical serious encounters data shows that the solutions have very similar alert rates and warning times, providing some reassurance of the robustness of the front. However, the historical serious encounters were identical in both evaluations. Since collecting additional serious encounters is impractical and to gain better estimates of the variability in the front with data we employ resampling methods.

The bootstrap evaluates the error rate on a number of surrogate data sets constructed by sampling the original data set. Suppose that the original

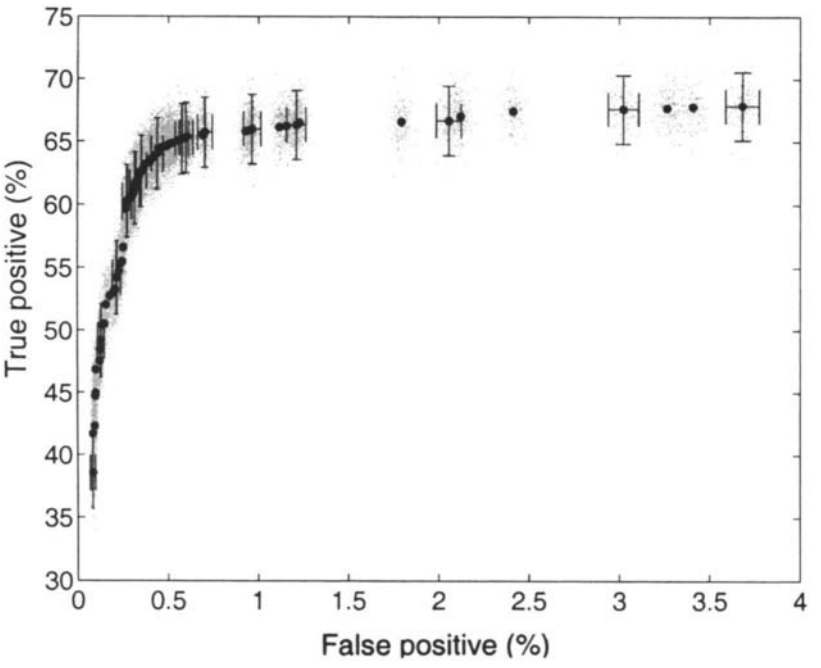


Fig. 3. Uncertainty of points on the estimated Pareto optimal ROC curve evaluated using bootstrapping. Each point in a cloud around a heavy dot on the mean front indicates the true positive and false positive rates for a bootstrap sample. Error bars indicate two-standard deviation intervals for a few representative points.

data set comprises $N = N_D + N_B$ examples, where N_D is the number of examples in the *dangerous* class and N_B is the number of *benign* examples. A bootstrap sample is constructed by drawing at random with replacement N examples from the original sample. Note that some examples in the original data will be included in a particular bootstrap surrogate more than once, while others will be excluded entirely. The classification rate averaged over a number of bootstrap replications is just the classification rate evaluated on the original data set, but an estimate of the variability in the classification rate may be obtained from the variation in the classification rates over the bootstrap replications.

Figure 3 shows the true and false positive rates obtained by evaluating the STCA system on 500 bootstrap replications for parameters on the front. While there is considerable spread about each location on the front, these scatter diagrams provide an estimate of the robustness of the parameter set to the data and indicate the range of true and false positive rates that may be expected at a particular operating point. Plots and statistics such as these permit the decision maker to accurately assess the probability of the true

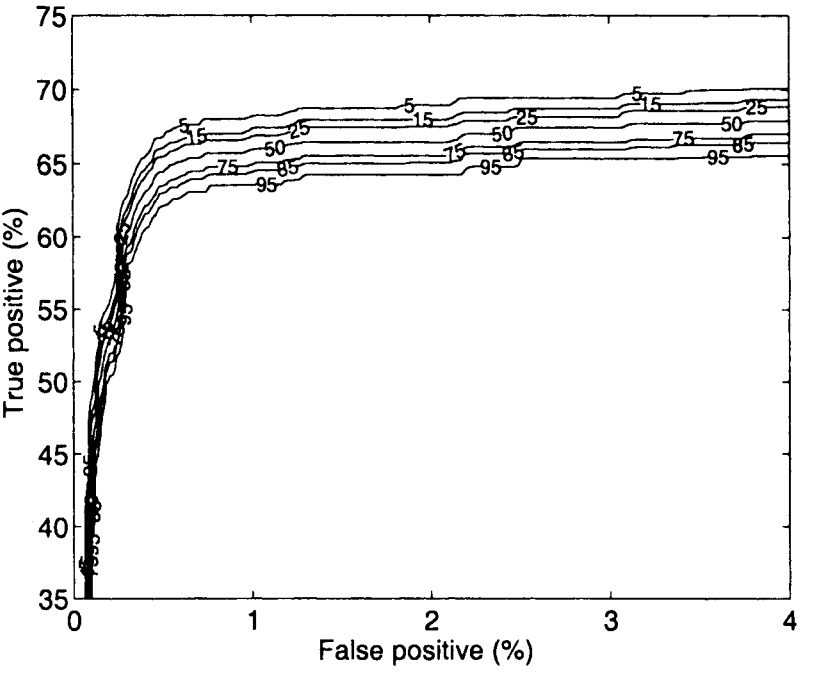


Fig. 4. Contours of the probability that a point on the Pareto front dominates a true-false positive operating point.

or false positive rate exceeding a given threshold; they might, for example, with a knowledge of the spread choose a more conservative operating point with respect to one or more of the objectives than would be chosen with only the mean front. However, like the choice of a particular operating point, the assessment of whether the front is robust *enough* depends upon the costs that the user places upon the objectives and what spread can be tolerated.

In fact, the variability in rates may be obtained without recourse to numerical sampling. Focusing on the true positive rate, $T(\boldsymbol{\theta})$, the distribution of true positive rates over the bootstrap samples is described by a binomial distribution, which is well approximated for even moderate amounts of data by a normal density with mean $T(\boldsymbol{\theta})$ and variance

$$\sigma_T^2 = \frac{T(1-T)}{N_D}. \tag{8}$$

The error bars plotted in Figure 3 were calculated using this expression.

The bootstrap samples may be used to assess the probability of achieving a particular operating point. Figure 4 displays contours of the probability, estimated from the bootstrap samples, that a point on the Pareto front dominates a particular operating point. Clearly, choosing the parameters at which

to operate based on the 50% contour may be over-optimistic in light of the day-to-day fluctuations in traffic characteristics. Likewise, one would have to be 'lucky' to achieve true-false positive rate combinations on, say, the 10% contour. A more conservative assessment would report the rates for the 90% or 95% contour along with the probability that the rates will be dominated.

4 Discussion

Many safety critical and safety related systems monitor a process and attempt to separate dangerous events from the more usual benign situations. Here we have focused on the STCA system, which is a component of the NATS 'safety net' providing *advisory* alerts to air traffic controllers of potential airspace proximity violations. Its importance is highlighted by the fact that it is thought that one of the factors contributing to the midair collision over the border between Germany and Switzerland in July 2002 was that parts of the STCA system in the relevant Swiss control station were switched off for maintenance [2]. In common with many safety critical and safety related systems, it has a large number of parameters which must be adjusted to ensure optimal performance in response to changing operational conditions. Here we have presented a straightforward multi-objective optimisation scheme for locating the parameter sets describing the optimal ROC curve for the STCA system as an example of general safety critical systems. This permits the operating point for the system to be set with explicit knowledge of the trade-off.

We emphasise that during the optimisation process the STCA system is treated purely as a subroutine of our evolutionary algorithm. Indeed in our implementation, the STCA programs run on a separate computer. This 'wrapping' of the system to be optimised is important for two reasons. First, it shows that the technique is applicable to any critical system whose operating point is dependent on parameters that must be tuned and whose performance can be automatically evaluated. Second, and more importantly for safety-related systems, the wrapped system has not been modified in any way, thus preserving its integrity and the integrity of any safety case constructed for it.

The idea of dominance is essential to the simultaneous optimisation of both true and false positive alert rates and it is interesting to note that the manually tuned operating point is dominated by several of the solutions found by multi-objective optimisation. However, despite these, relatively small, improvements we view the major contribution of this work to be the production of the optimal ROC curve which permits selection of the operating point with a full knowledge of the available alternatives. We remark that the London airspace which we study here is subject to frequent review and manual tuning by NATS and therefore may be expected to be well optimised, however, the methods presented here can be applied without alteration to any other less highly tuned airspace. In addition we have simultaneously optimised the warning time given

for genuine alerts, although we find that significant gains in warning time can only be achieved if the nuisance alert rate is substantially increased.

Bootstrapping of the test dataset around the optimised front provides an indication of the robustness of the optimised operating point. While these bootstrap estimates quantify the uncertainty in the optimised front, we remark that it would be beneficial to update a 'probabilistic front' so that new entrants were guaranteed with, say 90%, certainty not to be dominated by other elements of the front [10].

Finally we remark that the majority of the parameters in the STCA filters have direct physical or mechanical interpretation, and that the transparency of the classification process is an important component in assuring the safety case for STCA. However, whether tuned by hand or optimised by a machine algorithm, the operational parameters are inferred from data. An alternative to direct physical modelling is to employ purely statistical classifiers, for example k-nearest neighbour classifiers or neural networks, for which there is no ready interpretation of the parameters. Nonetheless, these methods are highly effective in other areas and the machine optimisation of STCA parameters blurs the distinction between physical models on one hand and statistical 'black boxes' on the other. We look forward to the construction of safety cases for purely statistical classifiers whose operational parameters are inferred from data and which have no ready physical interpretation.

Acknowledgements

We would like to express our thanks to Rod Bacon, Hellen Finney, Katherine Marren and Dan Roberts from the National Air Traffic Service Operational Analysis & Support group. We are pleased to acknowledge support under the Critical-Systems Programme from the Engineering and Physical Sciences Research Council of the UK.

References

[1] Analysis of Airprox in UK Airspace (July 2002 to December 2002). United Kingdom Airprox Board, 2003. Available from `http://www.caa.co.uk/ukab`.

[2] Investigation Report, AX001-1-2/02. Bundesstelle für Flugunfalluntersuchung, Hermann-Blenk-Strasse 16, 38108 Braunschweig, Germany, May 2004. Available from `http://www.bfu-web.de`.

[3] M. Anastasio, M. Kupinski, and R. Nishikawa:. Optimization and FROC analysis of rule-based detection schemes using a multiobjective approach. *IEEE Transactions on Medical Imaging*, 17:1089–1093, 1998.

[4] C.A.C Coello. A Comprehensive Survey of Evolutionary-Based Multiobjective Optimization Techniques. *Knowledge and Information Systems. An International Journal*, 1(3):269–308, 1999.

[5] A.C. Davison and D.V. Hinkley. *Bootstrap methods and their applications.* Cambridge University Press, 1997.

[6] K. Deb. *Multi-Objective Optimization Using Evolutionary Algorithms.* Wiley, Chichester, 2001.

[7] R.O. Duda and P.E. Hart. *Pattern Classification and Scene Analysis.* Wiley, New York, 1973.

[8] B. Efron and R.J. Tibshirani. *An introduction to the Bootstrap.* Number 57 in Monographs on Statistics and Probability. Chapman & Hall, New York, 1993.

[9] T. Fawcett. ROC Graphs: Notes and Practical Considerations for Researchers. *Machine Learning*, 2004. (Submitted).

[10] J.E. Fieldsend and R.M. Everson. Multi-objective optimisation in the presence of uncertainty. In *IEEE Congress on Evolutionary Computation 2005, (CEC'05)*, 2005. (Submitted).

[11] J.E. Fieldsend, R.M. Everson, and S. Singh. Using Unconstrained Elite Archives for Multi-Objective Optimisation. *IEEE Transactions on Evolutionary Computation*, 7(3):305–323, 2003.

[12] J. Knowles and D. Corne. The Pareto Archived Evolution Strategy: A new baseline algorithm for Pareto multiobjective optimisation. In *Proceedings of the 1999 Congress on Evolutionary Computation*, pages 98–105, 1999.

[13] M.A. Kupinski and M.A. Anastasio. Multiobjective Genetic Optimization of Diagnostic Classifiers with Implications for Generating Receiver Operating Characterisitic Curves. *IEEE Transactions on Medical Imaging*, 18(8):675–685, 1999.

[14] D. Van Veldhuizen and G. Lamont. Multiobjective Evolutionary Algorithms: Analyzing the State-of-the-Art. *Evolutionary Computation*, 8(2): 125–147, 2000.

[15] X. Yao, Y. Liu, and G. Lin. Evolutionary Programming Made Faster. *IEEE Transactions on Evolutionary Computation*, 3(2):82–102, 1999.

[16] E. Zitzler, L. Thiele, M. Laumanns, C.M. Fonseca, and V. Grunert da Fonseca. Performance Assessment of Multiobjective Optimizers: An Analysis and Review. *IEEE Transactions on Evolutionary Computation*, 7(2):117–132, 2003.

Classification with Confidence for Critical Systems

D. Partridge*, T.C. Bailey†, R.M. Everson*, J.E. Fieldsend*, A. Hernandez†, W.J. Krzanowski† and V. Schetinin#

*Department of Computer Science, School of Engineering, Computer Science and Mathematics, University of Exeter, Exeter EX4 4QF, UK.

† Department of Mathematical Sciences, School of Engineering, Computer Science and Mathematics, University of Exeter, Exeter EX4 4QF, UK.

#Department of Computing, University of Luton, Park Square, Luton, Beds. LU1 3JU, UK.

Abstract

In this paper we demonstrate an application of data-driven software development in a Bayesian framework such that every computed result arises from within a context and so can be associated with a `confidence' estimate whose validity is underpinned by Bayesian principles. This technique, which induces software modules from data samples (e.g., training a neural network), can be contrasted with more traditional, abstract specification driven, software development that has tended to compute a result and then added secondary computation to produce an associated `confidence' measure.

We demonstrate this approach applied to classification tasks --- i.e., the challenge is to construct a software module that aims to classify its input vector as one of a number of potential target classes. Thus a series of features extracted from a mammogram (an input vector) might need to be classified as either *tumour* or *non-tumour*, in this case just two target classes.

The set of classification probability estimates, which are fundamental to the Bayesian approach and constitute the `context' of any classification result, are generated by means of massive, but systematic, recomputation of results. We use state-of-the-art Reversible-Jump Markov Chain Monte Carlo (RJMCMC) methods to simulate the otherwise intractable integrals that emerge in applications of Bayes' Theorem.

The focus of this paper is on `confidence' estimates as an integral part of classification software and on the role of such estimates in critical systems rather than on the recomputation techniques employed to get the results.

1 Introduction

The nature of reality in computational systems is that some measure of uncertainty with respect to correctness is associated with every computed result. In the particularly demanding context of critical systems, this uncertainty is minimized by a variety of means (rigorous software development regimes, extensive testing, safety-case analysis, etc.). The residual uncertainty, which must

be deemed acceptable within the intended application of each particular system, is viewed as a global property of the software module, or system, as a whole. Thus an accepted system may be believed to generate an erroneous classification in, say, less than 1 in 10,000 executions.

An alternative approach is to compute an uncertainty, or a 'confidence', estimate with every system result. Uncertainty, or confidence, in the system's performance then becomes a result-specific quantity --- some results will be delivered with high confidence and others with low confidence. Dependent upon the intended application of the software, a significant proportion of low confidence results might be acceptable if, say, there are sufficient high confidence results and the low confidence results can be ignored or subjected to an appropriate remedial procedure. It is this approach to result-specific system uncertainty that we will demonstrate and explore. Note, this is not viewed as an alternative in the sense of a total replacement for the tried and tested practices, but as an alternative viewpoint, one that will need to be added to and integrated with current best practice. A major goal of this paper is to begin to explore exactly what such integration might involve in terms of modifications to current procedures for acceptance of such inductively-generated modules within the demands of critical-systems technology.

A further reality for many desired systems is that the task is fundamentally data defined: data samples of the input-output behaviour of the desired system exist, or can be generated, whilst an abstract specification (i.e., which implies the existence of a theory of how inputs can be transformed into outputs) may be problematic, or worse. Thus expert radiologists can accurately label mammograms as belonging to the class *tumour* or *non-tumour*, but the details of how to compute these classifications (i.e., the basis of a useful specification) are largely a mystery. The full arguments for and against data-driven, or inductive, software development have been explored previously (e.g., Partridge 1997). They will not be revisited in this paper except insofar as the issues surrounding data validity that arise in the critical-systems context.

2 Bayesian Computation of Classification Results

Within the example to be described, an input generates not a single specific classification, but a systematically determined set of probabilities that the particular input vector belongs to each of the target classes. Thus in a two-class situation, such as the *tumour/non-tumour* example, an input vector might be determined, on one occasion, to belong with 0.8 probability to the *tumour* class and with 0.2 probability to the *non-tumour* class. This single pair of results will be recomputed with a systematically chosen set of different (see below for explanation of difference) classifiers. The resultant set of distributions constitutes a well-defined approximation to the classification of the input data, which may be further interpreted to yield a discrete classification outcome together with a 'confidence' estimate of the uncertainty in the specific result selected. Although the computational details underlying the generation of such a result is not the focus of this paper (and has been amply dealt with elsewhere, e.g. Bailey et al. 2005), we provide a general overview for completeness.

One way to view data-driven development of classification software is that it involves the fitting of a mathematical model to the data, i.e., training, and different models give rise to different classifier systems (e.g. k nearest neighbours, knn, or neural networks). Each model contains adjustable parameters (e.g. the number of nearest neighbours and association weight, in the knn model, and the connection weights in a neural network model). Thus the fitting of a model involves techniques for optimizing the particular model parameter values from the information contained in (sub)set of the available data --- the training data. Having set the parameter values, we have a classifier for this data such that for any valid input we obtain a classification result.

The example to be described uses Bayes' theorem. Thus the output of our computation is $p(y \mid x, D, M)$, the probability that the input data ($x_n = (X_1, \cdots, X_n)$ a vector of n features) can be classified as target class, y given a set of training data, D, and a classifier model $M(\theta)$ parameterized by the vector of parameters,θ. Bayes' rule gives the posterior density over the model parameters θ, $p(\theta \mid D, M)$ as $p(\theta \mid D, M) = \dfrac{p(D \mid \theta, M)\, p(\theta \mid M)}{p(D \mid M)}$ for which we need to define prior probabilities over the parameters, $p(\theta \mid M)$. With the posterior density over θ on hand we can integrate out the dependence upon the model parameters $p(y \mid x, D, M) = \int p(y \mid x, \theta, \mathrm{M}) \mathrm{p}(\theta \mid \mathrm{D}, \mathrm{M}) \mathrm{d}\theta$, and so obtain a probability distribution for the input vector (conditioned on both the training data and the classifier model type) for each of the target classes, y where $y \in Q$ the set of target classes. The above integral delivers a classification result that is not dependent upon any particular parameter setting.

Typically, the required integrals are analytically insoluble. Consequently, Markov Chain Monte Carlo (MCMC) methods are used to sample distributions in a way that focuses the sampling in areas of high probability thus providing a means of efficient approximation to the desired integrals. This is the set of different classifiers that delivers the set of classification probabilities.

A recent theoretical extension, called Reversible-Jump MCMC (RJMCMC) due to (Green 1995), enables this sampling procedure to encompass with different numbers of parameters, and indeed even models with different parametrisations. In the example, we use a limited RJMCMC procedure following those of (Dennison et al., 2002) and (Holmes and Adams 2002).

The well-founded bases of both Bayes' theorem and MCMC methods underwrites the validity of the classification probability distributions generated, and the distributions provide a basis for the `confidence' associated with each classification result. The outcome is that MCMC methods permit us to draw samples $\theta^{(i)}$ from $p(\theta \mid D, M)$ so that $p(x \mid y, D, M)$ is approximated as $p(y \mid x, D, M) \approx \dfrac{1}{N} \sum_{i=1}^{N} p(y \mid x, \theta^{(i)}, M)$. Each such accepted sample (and

there are detailed rules to determine acceptance or not), a new setting of the model's parameters, constitutes a new classifier model. Thus the N samples amount to N potentially different classifier models (although typically the `good' parameter settings, as judged by training set performance, will be repeatedly selected). In general, these different models will generate different classification probabilities. It is this set of N individual results on the same input that constitute the histograms illustrated in Figure 1.

An MCMC-based simulation of a Bayesian process is computationally expensive, and it is only recent hardware advances that have moved such strategies into the realms of practicality. The sampling is massive (typically 10,000), as it must be to generate accurate approximations to the theoretical continua, but the reward is the extra information contained in such soft solutions as opposed to the poverty of information and the brittleness of a single categorical result (or indeed a set of such results when generated by an ad hoc collection of different classifiers).

3 Extracting a `Confidence' with Every Result

Using the above-outlined computational procedures, the classifier system developed generates from a novel input a probability density histogram for each of the target classes.

Figures 1 illustrates the probability histograms generated (when the number of samples and hence recomputations, N=10,000) for an input from the UCI Image data (aha@ics.uci.edu). The Machine-Learning database maintained at UC Irvine provides a good set of test problems because it contains wide variation in difficulty, and, being publicly available, there are both previously published results (to give benchmarks on performance levels), and the opportunity for future investigators to access the same data for comparative experiments. This particular problem has 7 output classes, and the result for each class is illustrated, classes 1 to 7, top to bottom. Each horizontal axis indicates probability on a zero to one scale, the probability that the input vector, x, is an observation from class C_i (i=1,...,7) given the available training data and a specific family of classifier models used (in this case, an augmented probabilistic k-nearest-neighbour model, based on that of Holmes and Adams 2002). The vertical axes show the quantity of evidence for each probability estimate. From this set of histograms, we can clearly see the distributions of evidence for each of the seven alternative outcomes. As can also be seen, no single class has accrued high probability in relation to its alternatives, with classes 1 and 3 ahead of the rest. These seven histograms, if reduced to a categorical result, which might be 'Class 1 or 3', would be an uncertain, or low confidence, result. However, the computer would know that its predicted result in this particular case is uncertain, and an appropriate caution could be issued.

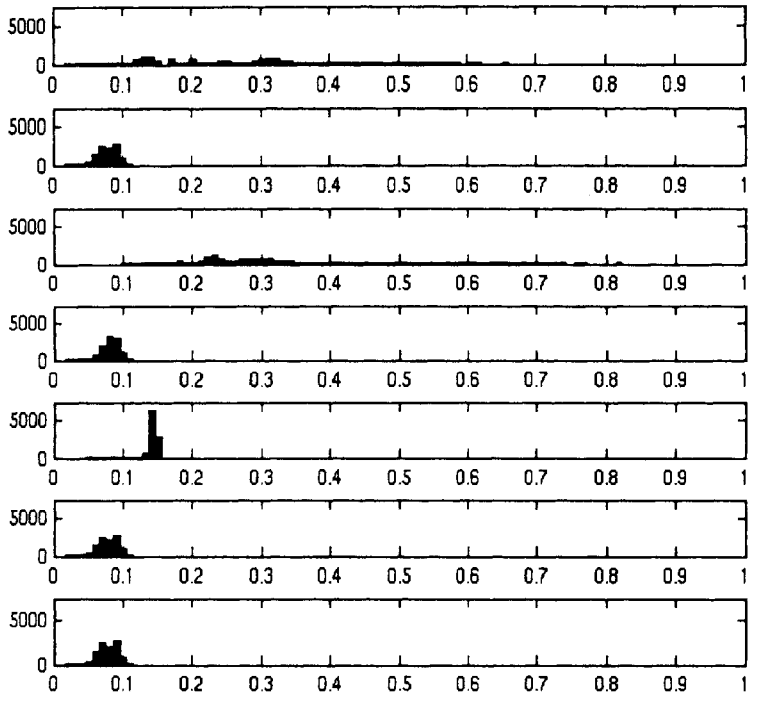


Fig.ure 1: Seven probability histograms that are the result of computing the classification outcome for one input vector from the UCI Image database, recomputed 10,000 times (target class 1 at top through to class 7 at bottom).

With the wealth of information that this approach generates (i.e., the 10,000 classification estimates in every target class for each input), the options for deriving discrete outputs and confidence estimates from a set of such histograms are many, and remain to be thoroughly explored. It is, however, clear that such an approach does provide much more information than more traditional approaches to classification systems, and it is, moreover, information that we would expect to be germane to the production of meaningful uncertainty estimates for a categorical result derived from this same information. It is only this generalized potential that we wish to establish in order to provide a basis for discussion in the context of critical systems.

Purely for the purpose of providing further specific examples, we will briefly described one strategy for generating a categorical result and an associated confidence estimate from these sets of histograms. After the MCMC process has converged, i.e. it has settled around 'good' parameter values (as determined by classification performance on the training data), the subsequent sequence of samples (each a classifier determined by specific parameter settings) is used to compute a classification result on the new input data. So when N=10,000, there are 10,000 individual predictions of the probability that the input vector under

consideration belongs to each of the target classes. It is these predictions that comprise the histograms illustrated in Figure 1.

We have experimented with a coarse, two-level, confidence estimate --- *SURE* and *UNSURE* --- that is determined by taking the most probable prediction provided by a sample classifier as the predicted class, and then simply counting the number of times each class is predicted within the set of N predictions.

If within the set of N classifiers operating on this data point, >1% classify this point as an alternative to the majority class q,
then class q is an *UNSURE* classification of this data point
else class q is a *SURE* classification.

Simply put: if 99% or more of the sample classifiers agree on the target classification then that is a certain result, otherwise it is an uncertain result. Note that our label *SURE* is merely a label denoting high confidence, not certainty, of course.

4 Illustrative Results

					Bayesian system test results		
UCI data set	Q	D	Test set size	Correct (%) **A**	*SURE(%)* Correct **B**	*UNSURE* (%) **C**	*SURE(%)* Incorrect **D**
Wisconsin	2	455	228	99.1	88.6	11.4	0.0
Ionosphere	2	200	151	94.0	58.9	41.1	0.0
Votes	2	391	44	95.5	81.8	15.9	2.3
Sonar	2	138	70	88.6	20.0	80.0	0.0
Vehicle	4	564	282	67.7	47.5	42.6	9.9
Image	7	210	2100	14.3	0.0	100.0	0.0

Table 1: Classification results on various publicly available test problems from the UCI database using a probabilistic k-nearest-neighbours classifier.

In Table 1 we present some illustrative results taken from a previous publication (Bailey et al. 2005) where full details can be found. Suffice it to say that these are publicly available data sets of varying difficulty and selected at random. They do however illustrate a number of possible outcomes for the computational approach proposed, and so provide us with a basis for discussion (as well as an 'existence proof' that our approach is practically feasible). The test results are found in the columns labelled A through D; the other columns give details of the databases used.

The first observation is that, without exception, for every data set, the number of test case classifications that were correct (column **A**) is greater than the number of correct classifications generated as *SURE* (column **B**). This not surprising as our computational approach was designed primarily to attach an accurate confidence estimate to each result, not to maximize correct results per se.

This being the case, it is column **D** that provides a maximum of information. This column shows the percentage (and when combined with the "Test set size" column, we can get the number) of test cases that were confident predictions, i.e. results labelled *SURE*, that were in fact incorrect. The values in this column should really be zero, and although most are, for two data sets a number of test cases generated *SURE* predictions that were wrong --- i.e., the system was confident of its answer, but it was incorrect! This highly unsatisfactory outcome can, of course, be addressed from several directions, for example, setting a higher threshold than 99% or, more radically, changing the classifier model. It may also be indicative of corrupted data, or data than is insufficiently complete for the classification task required. A non-zero column **D** result focuses attention on a `problematic' subset of the data. By way of contrast, non-zero results in column **C** are acceptable. This localization of uncertainty may not solve all our problems but it does provide a useful focus of attention.

The Wisconsin data set behaviour is indicative of 'normal expectation' in that although it produced only 88.6% confident predictions, which were all correct, as against 99.1% correct that a traditional classifier might generate, all of its incorrect predictions were generated with a low confidence. In other words, the loss in correct classification performance is compensated by the fact that its confident predictions were never wrong. Indeed the 99.1% result in column A is highly misleading because all 100% predictions of the test cases were generated equally as correct classifications. It was only subsequent comparison with the known, correct test results that determined that some 0.9% were, in fact, incorrect. Whereas the *SURE/UNSURE* labelled test results specified exactly which of the test cases had been correctly classified. There is a world of difference between knowing that 2 of 228 (i.e., 0.9%) results are likely to be wrong without knowing exactly which they are, and knowing exactly which 26(11.4%) results are not to be trusted. And, one might argue, this difference is particularly important in the context of a critical system.

A last observation concerns the Sonar and the Image data sets. Given that the former posts only 20% correct classifications, and the latter 0%, the classifier systems developed might be described as weak and useless, respectively. But as both produced 0% in the crucial column D, in the context of a critical system they both might be viewed as good systems as neither one generated any *SURE* results that were incorrect. But clearly the unsatisfactory percentages of correct classification results would need to be addressed. This might be done through data set enhancement, or new classifier models, or both. The major point being that system behaviour was safe despite very poor performance.

5 Ramifications for Critical Systems

It does seem that the switch from overall assessment of software uncertainty to localization of uncertainty for specific system results would be a beneficial change in most software applications, not just for critical systems. In fact, for critical systems knowing exactly which system results may be untrustworthy would be a significant advance in dealing with the inevitable uncertainties of software systems. However, this last gain can only be realized if all of the system

uncertainty can be localized within the outputs classed as uncertain. Although a number of the results in Table 1 might suggest that this essential localization is possible, the results are: firstly, just small, single-test results with unknown repeatability, and secondly, they are test results rather than a verified performance characteristic. In general, it would seem that we must develop and utilize the uncertainty associated with the *SURE*-but-incorrect performance characteristic of any system, and this observation appears to be pointing towards an infinite regress of uncertainty of uncertainties. At the very least, this difficulty appears to require a return to estimation and acceptance of a general system uncertainty, but one that will be coupled to specific performance uncertainties which does seem to be a significant step towards focusing our knowledge of uncertainty on the specific computations where it primarily resides.

A second issue that arises, one that is specific to inductive software development rather than associating uncertainty with every system output, concerns the scope of the software, the domain of the function induced --- i.e., which new inputs are valid and which are not. In the traditional, specification-driven approach to this question of input data validity is, or should be, pre-specified and thus not an issue (although it often is a problem in reality due to specification incompleteness or ambiguity). But without delving into the many problems associated with software specification, there is still a clear difference to address. The scope of an induced software module is a function of both the induction algorithm used and, crucially, the data set used for the induction. The software is data-defined, but defined by a subset of its operational domain. The difficulty is to determine exactly how the induction procedure has generalized over the specific set of training data, and hence the scope of the induced module. Research into this problem has looked at techniques such as interpolation between data points versus extrapolation beyond, at potential data-set circumscribing techniques such convex hulls, and at detecting invalid data inputs as 'novel'. However, this remains an issue on which progress is needed. We note, however, that it is also an issue in classical critical systems technology. In a multi-module software system, with perhaps different levels of safety associated with different parts of the system, the data-validity concern arises when data output from one module becomes input for another.

Another data-related concern is data that is difficult data to obtain or to generate (which would also impact adversely on the testing phase of traditional software development technologies). This situation adds extra difficulty, or uncertainty, to inductive software development, but we make no claim that inductive methods are always possible, desirable or preferable to traditional methods. Our general claim is that in some situations inductive techniques may be preferable, or the only possibility, and when they are used their algorithmic nature opens new possibilities for dealing with software uncertainty.

6 Conclusions

Without pretending that the problems of computing results with a meaningful confidence estimate associated with each have been solved by our suggested approach, we would claim that we have done enough to show that it will be

practically feasible, in some cases at least. This being so, the question arises of what does this mean in a critical-systems context. In critical systems it is not so much high levels of uncertainty that are unacceptable as the fact that there is also uncertainty as to where the uncertainty lies. A move to computation-specific uncertainty can change the system-acceptability requirements provided that the localized high uncertainty reduces global uncertainty, and doesn't just add to it. Hopefully, this paper has begun a useful discussion in this context, and has concentrated interest on understanding data in terms of the validity/invalidity of a new data item with respect to a previously used data set --- a training or test set in the context of inductively generated software, or just a test set in the traditional software development context.

Acknowledgements

This research was conducted with support under the Critical-Systems Programme of the Engineering and Physical Sciences Research Council of the UK (grant no. GR/R24357/01).

References

Denison DGT, Holmes CC, Mallick BK and Smith AFM (2002) *Bayesian Methods for Nonlinear Classification and Regression.* John Wiley & Sons, Ltd., Chichester.

Bailey TC, Everson RM, Fieldsend JE, Krzanowski WJ, Partridge, D and Schetinin V (2005) Representing classifier confidence in the safety-critical domain, Neural Computing and Applications (in press).

Green PJ (1995) Reversible jump Markov Chain Monte Carlo computation and Bayesian model determination, *Biometrika,* **82**: 711-732.

Holmes CC and Adams NM (2002) A probabilistic nearest neighbour method for statistical pattern recognition. Journal Royal Statistical Society B, 64: 1-12.

Partridge D (1997) The case for inductive software engineering. IEEE Computer 30(1):36-41.

Use of Graphical Probabilistic Models to build SIL claims based on software safety standards such as IEC61508-3

Mario Brito[1], John May[1], Julio Gallardo[1] and Ed Fergus[2]
[1] Safety Systems Research Centre, Department of Civil Engineering, University of Bristol, Queen's Building , Bristol BS8 1TR - United Kingdom.
[2] Electrical and Control Systems Group, Health and Safety Executive, Magdalen House, Stanley Precinct, Bootle, Merseyside, L20 3QZ - United Kingdom.

Abstract

Software reliability assessment is 'different' from traditional reliability techniques and requires a different process. The use of development standards is common in current good practice. Software safety standards recommend processes to design and assure the integrity of safety-related software. However the reasoning on the validity of these processes is complex and opaque. In this paper an attempt is made to use Graphical Probability Models (GPMs) to formalise the reasoning that underpins the construction of a Safety Integrity Level (SIL) claim based upon a safety standard such as IEC61508 Part 3. There are three major benefits: the reasoning becomes compact and easy to comprehend, facilitating its scrutiny, and making it easier for experts to develop a consensus using a common formal framework; the task of the regulator is supported because to some degree the subjective reasoning which underpins the expert consensus on compliance is captured in the structure of the GPM; the users will benefit from software tools that support implementation of IEC61508, such tools even have the potential to allow cost-benefit analysis of alternative safety assurance techniques.

This report and the work it describes were funded by the Health and Safety Executive. The opinions or conclusions expressed are those of the authors alone and do not necessarily represent the views of the Health and Safety Executive.

1 Introduction

Safety Critical software development processes are based on software safety standards such as IEC61605-3, DEF-0055 or DO-178. [IEC61508, 1998-2000] is a well established standard [Brown, 2000] in the civil industry. The standard is

intended to serve as basis for the preparation of more sector specific standards such as IEC61511 [Black, 2000] or for stand-alone use where no more specific sector standard or industry code of practice exists. IEC61508 provides requirements for, and guidance on, developing programmable systems for protection and safety-related control that implement safety functions of sufficient integrity to reduce to an acceptable level the risk arising from identified hazards. Examples of protection and safety-related control systems include: a nuclear power station protection system; a railway traffic management system; a machinery control system; an offline advisory system whose output contributes to a safety decision. A safety function implemented by a safety-related system can comprise a software element, a hardware element or both. The integrity of a safety function is a measure of the confidence that the safety-related system will satisfactorily perform the specified safety function when required. The integrity is expressed as a Safety Integrity Level (or SIL) in the range 1-4 where 4 is the most demanding. In determining safety integrity, all causes of unsafe failures are taken into account: random hardware failures, and systematic failures of both hardware and software. Some types of failure such as random hardware failures may be quantified as failure rates, while factors in systematic failure often cannot be accurately quantified but can only be considered qualitatively. A qualitative SIL requirement is interpreted as the degree of rigour with which recommended system development techniques should be applied in order to achieve the required confidence in the correct operation of the associated safety function.

The IEC61508 safety standard consists of 7 parts. Parts 1 and 2 are concerned with the system and hardware development whereas Part 3 addresses the software development. The remaining Parts 4-7 provide definitions of terms and guidance on the use of IEC61508. The work presented in this paper addresses exclusively Part 3 of IEC61508 (software requirements), where SIL requirements are usually qualitative and where reliability analysis is less mature. The principles of developing safety-related software are complex and contain a high degree of subjectivity (engineering judgement). The purpose of the research presented in this paper is to develop a method of analysing the largely qualitative reasoning contained within the parts of standards that address software safety, initially with respect to IEC61508-3 but potentially also applicable to other standards for software safety. Furthermore, it is hoped to capture this reasoning in a form that will subsequently allow the application of these standards to benefit from tool support. This report assesses the feasibility of using the BBN formalism for exposing and formalising the subjective factors in the assessment of software integrity. BBNs were chosen for their ability to capture subjective arguments and because commercial tool support already exists to support BBN modelling.

The use of IEC61508-3 to build software SIL claims is a complex process, involving the choice of appropriate development and assurance procedures (we will use the blanket term 'methods') at all stages of the safety software development lifecycle. The central concept in the standard is that satisfactory application of appropriate methods will result in software that is likely to meet its safety integrity target. The justification of a SIL claim from development/assurance methods is a probabilistic and uncertain argument. Uncertainty arises from, for example, the effectiveness of the method in addressing the problem characteristics

of the safety application; the training and competence of the people applying the methods; and the extent of tool support for the method. IEC61508-3 does not seek to model this uncertainty. Bayesian Belief networks (BBNs) are a form of GPM that has proven to be a very powerful technique for reasoning under uncertainty. This paper proposes a particular BBNs model of the uncertain reasoning in safety standards.

In this paper we shall first discuss the reasons why support for a software safety standard such as contained in IEC61508-3 would be useful to the stakeholders in the process of developing and assuring safety-related software. We then discuss some other research work related to the model in this paper, followed by an overview of BBNs. In section 4 we introduce our model prototype and in section 5 we provide some examples as to how our model can be used to support SIL claims within IEC61508-3. Section 6 provides the conclusions.

2 Support for IEC61508

This section discusses some of the areas in which the development and application of IEC61508-3 can be supported by models of uncertain reasoning.

Standardisation, transparency, and future development of the standard.
The reasoning used to justify SIL claims within IEC61508 has been developed by consultation, but the reasoning process itself is not explicit. Indeed IEC61508 does not provide any evidence so as to why the product integrity should be inferred from processes [McDermid, Pumfrey, 2001]. This is a significant drawback, since the reasoning in IEC61508-3 relies heavily on subjective judgment, and so the ability of experts to understand its basis and then build a consensus, is central to the purpose of the standard. To this end, it would be useful to capture the reasoning in a more compact and understandable form.

Tool support for guidance and feedback when applying IEC61508
Assessing compliance with a standard such as IEC61508-3 is not a trivial task. Companies would benefit from a tool that provided quick informative feedback during the process of application of IEC61508-3. In its most straightforward form this tool could answer the question 'Have we done enough to comply?' It would be even more attractive to provide help with pro-active decisions earlier in the safety lifecycle, such as 'So far, I have used methods X, Y, Z..., what should I do next to achieve compliance?'

Compliance with IEC61508-3 can be achieved by adopting the specific recommended methods for software development. However, IEC61508-3 also recognises that a large number of factors affect system and software safety integrity, and that with our current understanding of software and system technology it is not generally possible to give an algorithm for selecting a package of software development methods that will guarantee to achieve the required integrity in any given safety application. IEC61508-3 therefore also permits compliance through the use of alternative development methods that depart from

the specific recommendations, and requires these departures to be justified by some rationale. However, this is also a potential source of combinatorial explosion of possibilities due to the number of factors involved (types of application, different development methods, different verification methods, different integrity targets). Consequently, the expression of alternative routes to compliance using static tables in IEC61508-3 may be complex and difficult to follow.

A related issue is that compliance based on static tables of methods may encourage over-prescriptive use of the standard, whereas tool support that effectively assists in the evaluation of selected methods (whether recommended or alternative) may result in a more transparent and convincing argument that the software achieves its required SIL.

Cost/Benefit analysis

Ideally, it would be possible to go beyond simple listing of alternative sets of methods, as described above. Suppose a company has applied a set of procedures that are judged to not quite achieve compliance. The question they most want answered is 'which procedures should we apply next, in order to achieve compliance in the most efficient way?' There are two aspects to this question. Firstly, which procedures best fit the 'gaps' in their compliance case? Secondly, what costs result from applying the procedures, which depend on the procedures themselves but also on the existing expertise within the company?

It may also be desirable to attempt more general questions e.g. 'What is the most effective compliance process our company could develop given the types of application we develop?'

Need for further discussion surrounding rigour of methods

Different procedures are not of equal effectiveness, and it is necessary to state explicit beliefs about this if we are to reason effectively about the alternative procedures sets. Effectiveness is not just a matter of the inherent power of procedures. For example, it is tempting to state that the use of formal methods *provides more effective assurance* than, say, the use of traditional code inspection. However, before this statement can be made it is necessary to know how *intensively* the formal method has been applied (e.g. full proof of code, or just a few key system properties?), how many people inspected the code, how long were they given, and their experience/training etc. To give another example, the statement 'we have applied dynamic testing' is not informative in itself. A further qualification of the actual test techniques used is needed together with the level and depth that were achieved (– some measure of the amount and power of testing.)

The concept of intensity of application of methods is also a factor in cost-benefit analysis, since it affects both the effectiveness of a method and its costs. It would be plausible to assume a law of diminishing returns applies to effectiveness i.e. a greater intensity of application results in improved assurance, but the rate of improvement achieved decreases with increasing intensity.

3 Related work

Use of Bayesian belief networks to predict software quality has been proposed previously. [Hall, May, et. al, 1992] working on the FASGEP project used BBNs to measure confidence in the level of integrity of software design process, based on prediction of fault numbers. Fenton in [Fenton, Neil, 1999] present a critique of existing defect prediction models such as Multivariate approach and size and complexity metrics and also conclude that BBNs offer some attractive benefits compared to the existing software metrics techniques.

A significant amount of effort has been put in the development of a graphical tool called Goal Structuring Notation (GSN). The Computer sciences department at York University have been developing this graphical tool to support safety cases in the aerospace and railway industry as it is presented in [Weaver, 2002] and [Weaver, Despotou, et. al, 2005]. Whilst there are many benefits, the tool does not provide a quantitative assessment as to how much processes influence one another.

The BBN structure resulted from the SERENE project [Fenton, Neil, 2004] was designed to support software assurance. More recently, in [Fenton, Neil, 2005a] and [Fenton, Neil, et. al, 2005b]. Bayesian belief network structures were developed to predict the quantity of unknown defects in a software development. Although these BBNs estimate quantities relating to, for example, the coders' performance and the number of faults in the code, they do not predict the SIL that can be claimed.

Although neither the BBN or GSN approaches above attempt to encapsulate how conformance can be achieved in a software safety standard such as IEC61508-3, a major benefit of both approaches is that they provide a strong visual aid that improves transparency in an assessment process.

The first application of BBN in the specific context of a software safety standard was presented in [Gran, 2002]. The standard in question was DO-178.

4 Background

The theory supporting Bayesian Belief networks rests on a rich tradition of probability theory, and statistical decision theory and it is supported by excellent axiomatic and behavioural arguments [Pearl, 1998]. A Bayesian belief network for a set of variables $\mathbf{X} = \{X_1, X_2, \ldots, X_n\}$consists of a) a directed network structure that encodes a set of conditional independence assertions about variables in X and b) a set P of 'local' probability distributions associated with each variable, describing the distribution of the variable conditioned on its parent variables. The nodes in the network structure are in one-to-one correspondence with the variables in the probabilistic model.

Typically there are two main tasks in the overall design of BBNs: structure design and parameter elicitation. Structure design involves the task of deciding "what depends on what?" and encoding that using the conditional independence semantics of the network (directed acyclic graph) structure. It uses qualitative information, background knowledge and empirical experience. In some cases, it

can be obtained by learning where there is a large body of experimental data. Where this is not the case, the key problem is how to acquire knowledge from domain experts. The task of parameter elicitation, on the other hand, is to fill out the conditional probability tables (CPTs) or node probability tables (NPTs) for every random variable. These can be obtained from either quantitative data or subjective estimation by domain experts.

In previous work on graph structure, much emphasis is placed on the need to achieve a correct representation of causality in the underlying DAG. Whilst important, this task depends on an earlier, arguably more fundamental and difficult, problem of defining relevant variables whose values are measurable (or at least 'plausibly estimable,') since it clear that some quantities are harder for people to estimate than others. This is usually a difficult problem due to the huge number of different variables and networks that can suggest themselves when modelling a problem with BBN. Measurable/estimable variables are important for two reasons. Firstly, the standard BBN updating procedure, that performs probabilistic inference based on BBN parameters, uses evidence entered into a BBN in the form of statements about the values of the variables at the BBN nodes. Inaccuracies in these statements will produce inaccurate inference. Secondly, BBNs can learn (or update) their local probability tables from data, and this will not be possible if data values for variables are not measurable or estimable.

5 Proposed network structures

The suggested BBN structures presented in this section are a result of discussions held with safety experts and project managers. They represent an attempt to capture the reasoning in software safety standards, but are no means claimed to be fully accurate representations. The structure of the networks, and their conditional probability tables, must be further evolved in a process of expert consensus building.

5.1 Structure of IEC61508-3 activities

IEC61508 system development is structured into three safety lifecycles: the overall, the E/E/PES, and the software. Only the software safety lifecycle is addressed in this paper. Figure 1 illustrates the software safety lifecycle as it is presented in IEC61508-3.

The activities of the software safety lifecycle are organised into a number of "phases" including: safety requirements specification; architecture design; selection of support tools and translators; detailed software design and coding; module and integration testing; In each phase there is a verification exercise that aims to find errors introduced in the development process.

The software safety lifecycle phases are ordered according to the well known V diagram for software development, see Figure 2 for more detail.

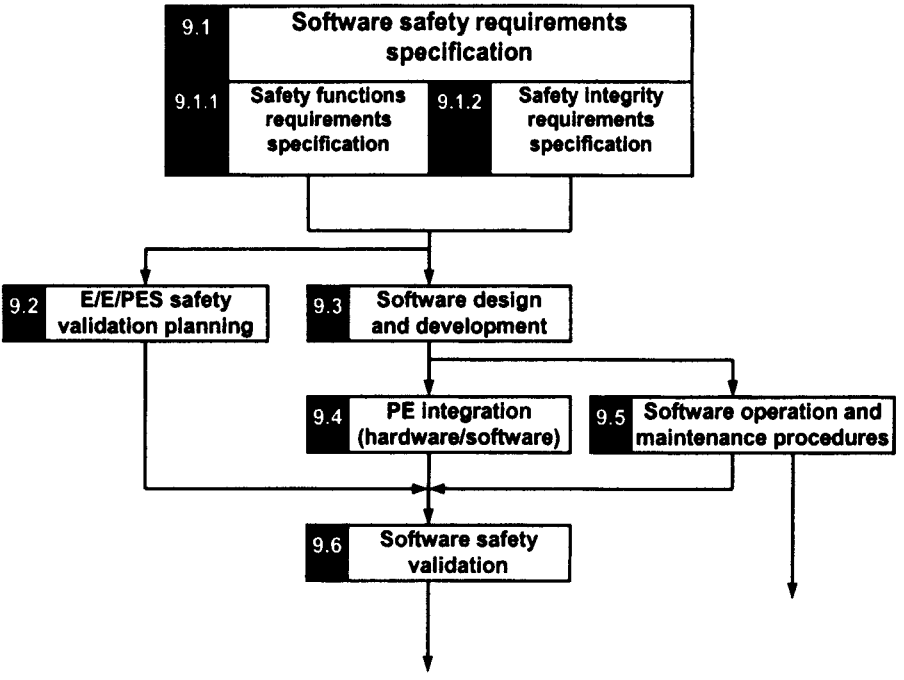


Figure 1 – Software safety lifecycle

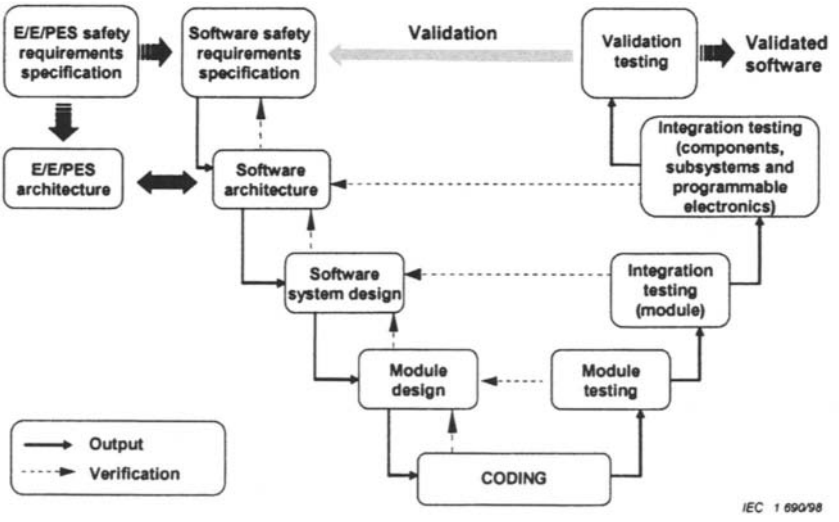


Figure 2 – The V diagram

The methods used in each phase (specification, design, testing, etc) of the software development may be selected according to the specific IEC61508-3

recommendations for the required SIL, or they may be alternative methods for which a BBN-based justification is being developed.

5.2 Single-phase BBN prototype

This part of the problem involves prediction of software integrity from the character of the methods used in a single phase of a safety software life-cycle presented in a standard such as IEC61508-3. There is some previous work that is relevant to this problem as summarised briefly in section 3. The network structure in Figure 3 is proposed.

The main purpose of the BBN is to estimate the significance of the outstanding errors remaining in the system at the end of the phase. This clearly depends on a wide variety of factors, many of which are usually implicit but are exposed in the BBN. One important example is that the reliability of a piece of software will depend on its operational profile of inputs during use. It is assumed (as it is in some standards) that such information is implicitly factored in to the assessment i.e. the verification methods used are focused on the proposed usage of the system, so that faults that cause large numbers of failures are found quickly. There is some evidence that if this is true, software reliability is predictable provided latent fault numbers can be predicted [Bishop, Bloomfield, 1996].

In each phase the BBN divides methods into two types: build methods (e.g. in a specification phase, these are the methods used to construct the specification) and verification methods (e.g. methods used within the phase to check that the produced specification is satisfactory).

In the proposed BBN, as a general principle, we model the rigour of application of any method (its effectiveness), in terms of two subsidiary concepts: the inherent power to do the job ('power of build/verification method i' nodes) and the intensity of its application ('intensity at which build/verification method i was applied' nodes). The multiple node notation presented in Figure 3 indicates that every phase of the IEC61508 safety software development lifecycle has one or more build and verification methods, the precise number of nodes depends on:-

1. The number of build and verification methods applied in each phase, and
2. The position of the phase within the whole process.

In the generic single phase BBN shown in Figure 3, phase 'i' is shown with three build methods and three verification methods.

The 'Quality of the development process at phase i' is there to capture the quality of implementation of the build methods. If in phase 'i' the build methods were poorly implemented, one expects phase i to have implemented development processes of poor quality. The quality of the development process has a causal effect on the number of faults introduced in the system.

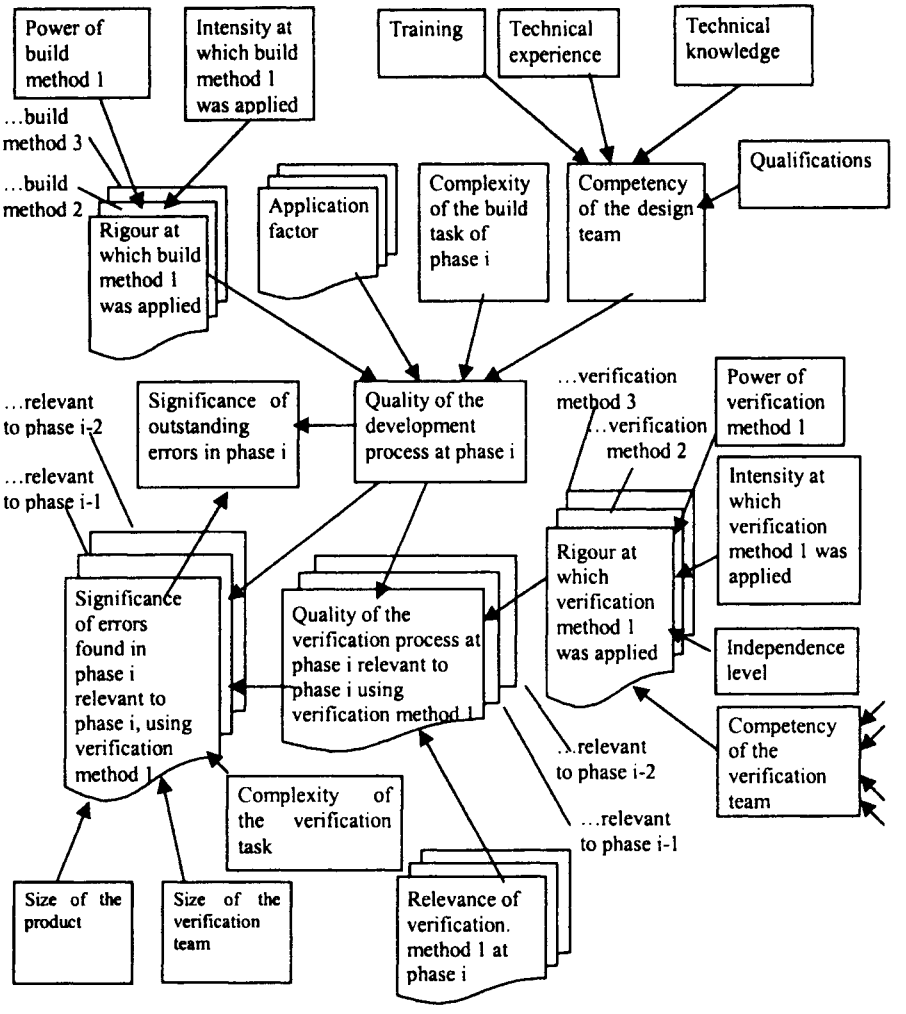


Figure 3 – Generic BBN 'Flat'structure for one phase of the safety software development lifecycle.

The 'significance of errors found ...' nodes measures the criticality of errors found during the verification process. Similarly, the 'significance of outstanding errors...' node refers to the faults that remain undiscovered. The latter node will ultimately feed in to the computation of probable integrity levels (see section 5.3). The computation of the probability distribution for this node from its parents is one of the most contentious aspects of the reasoning in safety standards. It may be that there are other, more comprehensive network structures that can capture the underlying reasoning more accurately. However, it should be possible to capture such reasoning in BBN form, and thus to expose it to expert scrutiny. The central point of each phase is to estimate the significance of the outstanding errors. This

has a direct causal effect on the estimate for the SIL that one can claim to comply with for phase i.

The 'complexity of the build task...' nodes capture the inherent difficulty of the tasks being undertaken in a phase. To see why this is an important factor, suppose a phase was modelled in two ways. Firstly, as a single 'meta-phase'. Secondly, as split into two smaller phases. Further, suppose that the same methods were used in all two phases with the same intensity. Without the complexity node, the estimated quality loss in each sub-phase would be equal to the quality loss for the original meta-phase. Depending on how integrity measures from separate phases are composed, the BBN model could be incoherent (self-contradictory) e.g. one model would be that the integrity of the process consisting of two smaller phases is less than the integrity of each single phase in that process.

The 'Quality of the verification process...' nodes take values from a discrete set of values such as {very poor, poor, medium, good, very good}. Estimations of their values are made, based on the rigour at which verification methods were applied and their relevance. The 'relevance of the verification method j for phase i' node effectively 'selects' the verification processes that are relevant to previous phases of the software development lifecycle.

Finally, the 'Application factor' nodes model the effect that different industrial sectors have different perceptions as to the degree of rigour at which build methods should be applied. Thus, in a sector where a particular method is perceived to be low rigour, that method will make only a small contribution to the quality of the development process.

5.3 Multi-Phase BBN prototype

In order to model the software safety development lifecycle a larger BBN is needed to combine estimations from individual phases. We propose that this larger network should feed forward the quality of the development process of each single phase, since the subsequent development work will depend on that quality. The network also should have a feedback connection so that errors found in later phases have an impact on the contribution to the estimated SIL from a previous phase.

This view is more complex than previous process-based attempts to capture software reliability using BBNs. It takes the view that a phase is associated with a set of processes (e.g. a requirements specification), and these processes are subject to active updating for the duration of the project due to work in later phases. This approach allows us to capture intricate influences between 'phases'. The generic BBN structure shown in Figure 4 presents a sub-net for each phase of the safety software life-cycle and a net for interaction among phases. The interaction net aggregates integrity estimates from multiple phases.

In each phase of the safety software development life cycle, there is a verification exercise that aims to find errors introduced in the development process. This verification exercise, or process, aims to find errors that are relevant to particular phase at which is being applied. The verification process of a particular phase can clearly also find errors made in previous phases of the safety software life cycle.

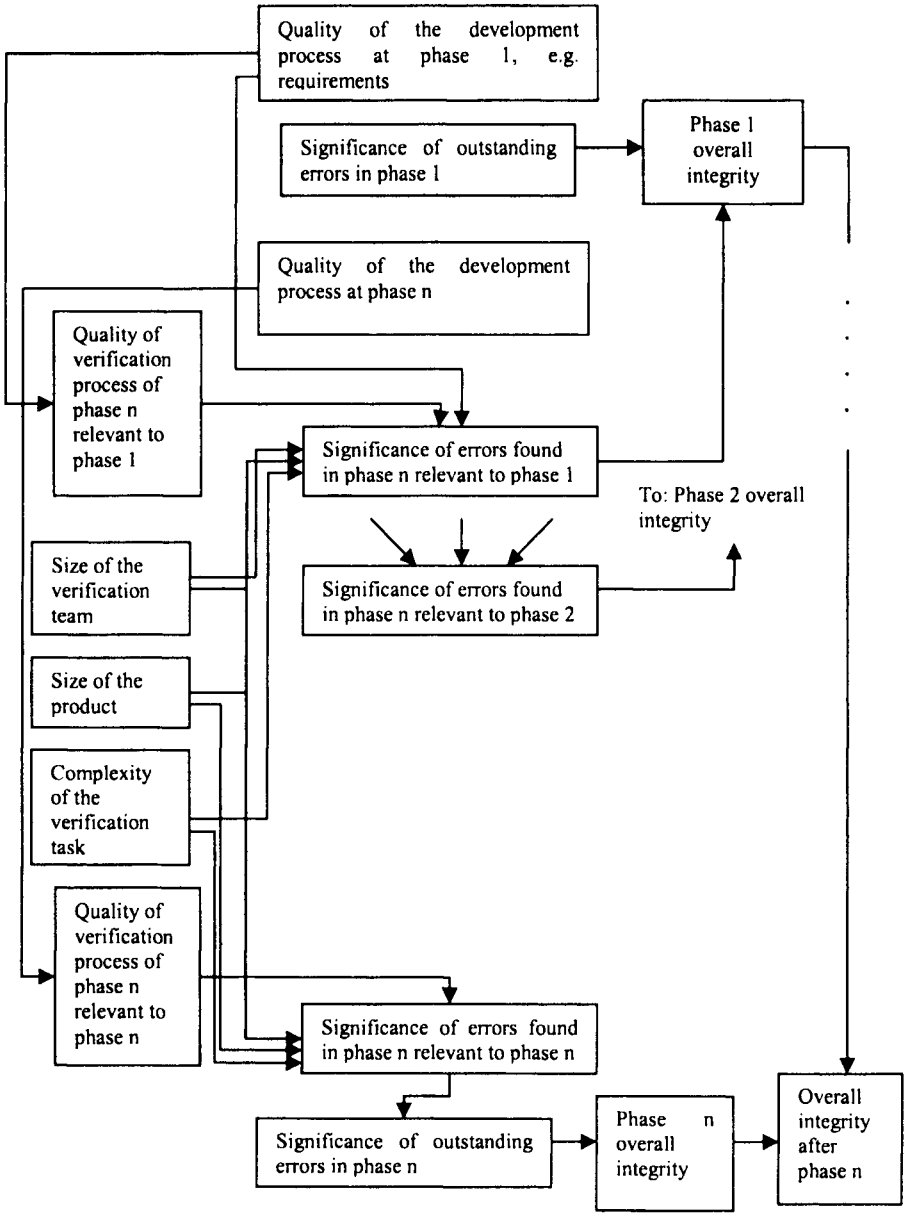


Figure 4 – Generic BBN Multi-Levelled structure for several phases of the safety software development lifecycle.

Errors found in later phases are deemed to be corrected resulting in a gain in integrity level achieved at the end of the previous phase. An example is for

instance, errors found whilst testing the software. Some of these errors will be relevant to the software implementation phase, however some of these errors may have been made in the development process of the software functional specification phase.

We implemented a particularly simple rule, that the overall integrity after any phase is the minimum level of integrity achieved for all previous phases including its own. For instance, if one claims SIL 1 for phase 1 and SIL 3 for phase 2, the overall integrity that one can claim after phase 2 is SIL 1. Clearly, this is a candidate for debate, and there is a strong case for additive models (described briefly in section 5.2 in the context of the 'complexity...' nodes).

6 Examples of GPM

This section gives some examples of the use of BBNs to estimate integrity levels based on the style of reasoning found in safety standards.

6.1 Example 1: Predicting the criticality of outstanding errors in a phase of the software development

Figure 5 is a concrete example of Figure 3, showing a phase with one development method and one verification method. It concerns an arbitrary phase of a safety lifecycle, called phase 1. In the following example, all nodes without parents have been given hard evidence. This is highlighted in dark grey on Figure 5, and means the variables have been instantiated with a value: a measurement of the quantity being modelled by the variable e.g. the 'Training' node has value 'satisfactory'. This evidence is then propagated through the network updating the belief in the states of the nodes that were not given values. In probability terms, a probability distribution is calculated for each of the latter nodes, conditioned on all of the hard evidence.

Below it is presented how the criticality of outstanding errors can be estimated based upon measurements of other nodes.
Let's observe the following scenarios:

- Assuming that there was overwhelming evidence for the following statements: 'Power of build method X' is 'poor'; 'Intensity at which build method X was applied' is 'very low'; 'complexity of task' is 'fair'; 'Application factor' is 'low'; "complexity of the verification task" is 'fair', "application factor" is 'medium'. In addition parent nodes of the competency of design staff as well as the verification staff were set with evidence according to the information shown in figure 5. Finally lets consider that the independence level is 'high', the relevance of verification method Y is high;

and that the size of the product, size of the verification team is 'medium' and that the complexity of the verification task is 'fair'.

- If the power of verification method Y was 'very poor' and if it was applied at 'medium' intensity, the following distribution for the "significance of errors found using verification method Y" would be obtained: {9.80, 1.12, 4.42, 84.66}. Consequently the distribution for the 'significance of outstanding errors in phase 1' would be as follows: {29.70, 43.79, 18.64, 7.86}. Hence one could say with belief (or 'confidence') 44% that the significance of the outstanding errors is at 'tolerable' level, and belief 74% that the level is tolerable or better.

- If the power of the verification method Y was 'medium' the following distribution would be obtained for the 'significance of errors found during the verification process': {8.52, 1.02, 4.08, 86.38}. For the same conditions the following distribution would be obtained for the 'significance of outstanding errors in phase 1' :{ 30.08, 44.58, 18.43, 7.01}. Hence the confidence that the criticality of outstanding errors in phase 1 were tolerable increased slightly to 45% (75% tolerable or better).

- Moreover if evidence was provided supporting the fact the "power of verification method Y" was 'very good'. The following distribution would be obtained for the 'significance of errors found during the verification process': {7.54, 0.94, 3.83, 87.69}. Again the new estimated belief on the state of the 'significance of errors found during the verification process' would propagate through the network updating the estimated belief on the state of the 'significance of the outstanding errors'. The following distribution would be obtained for the 'significance of the outstanding errors in phase 1': {30.36, 45, 18.29, 6.36}, a negligible improvement.

- However, if evidence was provided supporting the fact the "intensity at which verification method Y was applied" was 'very high', the following distribution would be obtained for the 'significance of errors found during the verification process': {0.07, 0.33, 1.70, 97.90}. For the same conditions the following distribution would be obtained for the 'significance of the outstanding errors in phase 1': {32.55, 48.99, 17.09, 1.37}. Consequently the estimated belief that the 'significance of outstanding errors' was tolerable increased to 49% (82% tolerable or better). This scenario is illustrated in figure 5. The figures are notional. They are proposed as a basis for future discussion and nothing more. They do however, illustrate the power of the BBN to capture uncertain argumentation of the type needed in standards.

- What is encapsulated here whether you agree with or not is that using a more powerful method will not necessarily increase integrity unless it is applied diligently.

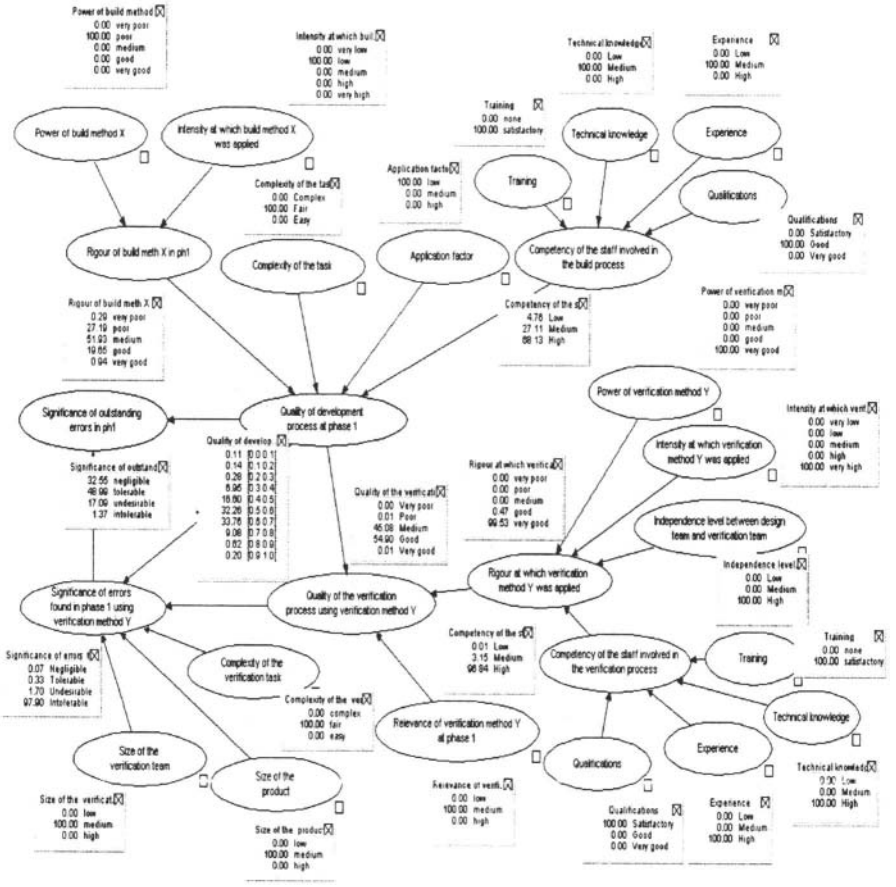


Figure 5 – Reliability estimation for phase 1 of the safety software lifecycle.

The behaviour of the prototype BBN suggests that the Single-Phase BBN model may be usefully capture the reasoning in standards such as IEC61508.

6.2 Example 2: Estimating phase_1 overall integrity

This scenario illustrates how evidence gathered regarding the significance of errors found during the verification process of phases 2 and 3 influence the overall integrity of phase one. Figure 6 shows a concrete example of the general model in Figure 4, modelling 3 phases of a safety software development lifecycle. Each phase is represented with an 'instance node', which is a sub-model containing all information relevant to a sub-network (in this case, a single phase of the safety software development life-cycle). The instance node for phase one has two outputs and these are the "Significance of outstanding errors in phase 1" and the "Quality of the development of phase 1". The latter measures the safety integrity that one

can claim for phase 1 and has the following states: {SIL1, SIL2, SIL3, SIL4}. As shown in Figure 6 the overall integrity that one can claim to comply with for phase 1 also depends on the significance of errors found in later phases. The instance node representing phase two has one input node and three output nodes. The instance node representing phase three has two input nodes and three output nodes. A key feature of the integration BBN is the computation by which SILs from individual phases are composed into a software lifecycle SIL.

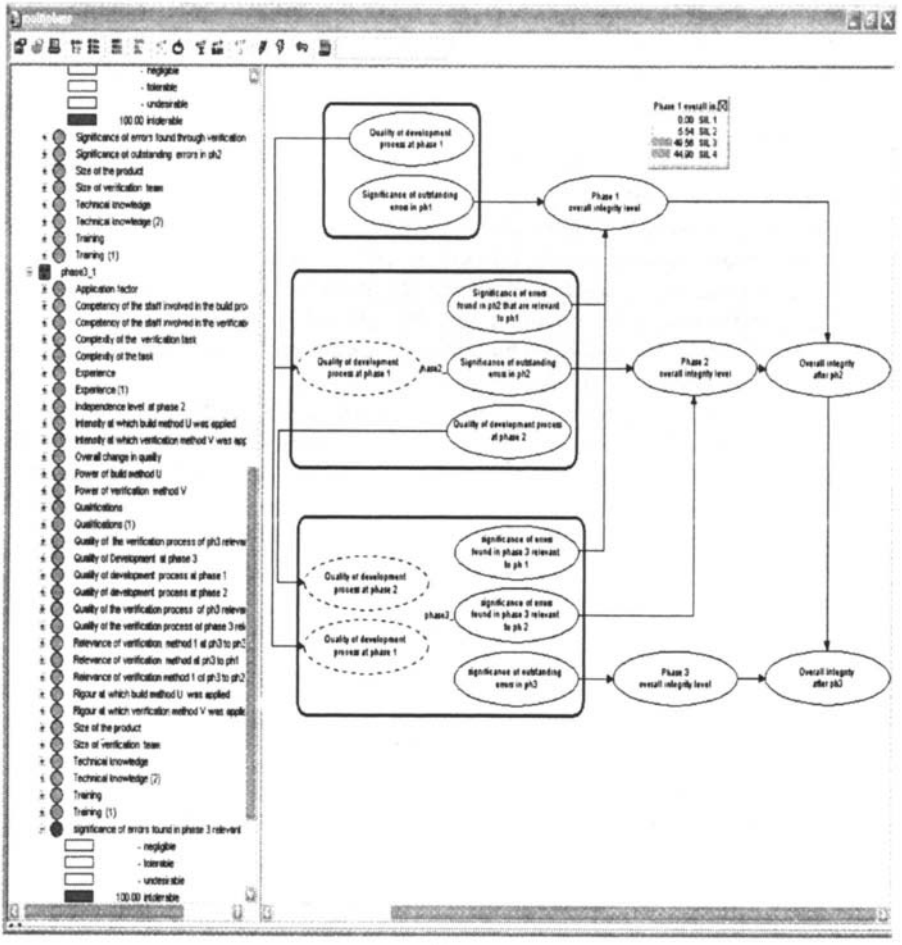


Figure 6- Bayesian belief network model of three phases of the safety software life-cycle.

Figure 6 shows the Hugin interface output for this example, which uses the following scenarios:

- An initial estimation for the 'Phase 1 overall integrity level' node was obtained assuming that evidence was provided to support the following

statements: the 'power of build method X' was 'poor', the 'intensity at which build method X was applied' was 'very low', the 'power of verification method Y' was 'good', the 'intensity at which verification method Y was applied' was 'very high'. In addition it was also considered that the staff involved in the development and testing had the same amount of experience, training, technical knowledge and qualifications as used in the previous example. Finally, the independence level between the verification team and the development team is high and that the relevance of verification method Y was 'high'. For the conditions presented the following distribution was obtained for the 'significance of outstanding errors in phase 1': {23.02, 48.23, 25, 3.75}. Thus one can say with 71% confidence that the significance of outstanding errors is tolerable or better. For the same conditions the following distribution was obtained for the 'Phase 1 overall integrity' node: {4.90, 16.62, 39.45, 39.03}. Thus, there is 78% confidence that phase 1 can claim to comply with SIL 3.

- Now consider that during the verification process of phase 2 one finds further errors created during phase 1 (there is 100% confidence that errors found were intolerable). Once this evidence is entered in the model the following distribution would be obtained for the "Phase 1 overall integrity" node: {2.47, 9.77, 42.59, 45.18}. Thus, there is 88% confidence that phase 1 can comply to claim with SIL 3.
- Finally, consider that errors were also found in the verification process of phase 3. Once this evidence is entered into the model and propagated through the network. The following distribution would be obtained for the 'Phase 1 overall integrity' node: {0, 5.54, 49.56, 44.90}. Hence one can say with 94% confidence that the development process of phase 1 complies with SIL 3.

The results present the concept of reliability growth in software development processes. This feature is currently not addressed in IEC61508.

6.3 Example 3: Estimating SIL for the overall product lifecycle

In this example all three phases were populated with evidence, see Figure 7 for more detail. The example was simplified in some ways, for example assuming that the same members of staff were involved in the development of the product in all phases. In addition the application factor was considered to be low, the complexity of the development and verification tasks were considered to be fair. Furthermore it was assumed that all three phases have one build method and one verification method. In addition the following assumptions were made for phase 1: the 'power of build method X' was 'very poor', the 'intensity at which build method X' was applied was 'very low', the 'power of verification method Y' was 'good' and the 'intensity at which verification method Y was applied' was 'very high'. For phase 2, the 'power of build method Z' was 'poor', the 'intensity at which build method Z was applied' was 'medium', the 'power of verification method W' was 'medium' and the 'intensity at which verification method W was applied' was 'high'. For phase 3, the 'power of build method U' was 'good', the 'intensity at which build

method U was applied' was 'medium', the 'power of verification method V' was 'good' and the 'intensity at which verification method V was applied' was 'high'. The following estimate would be obtained for the overall integrity that one can claim to comply with after phase 3 :{ 21.06, 43.76, 33.44, 1.74}. Thus there would be 35% confidence that the overall development process of phase 1 complies with SIL 3 and 79% confidence that it can comply with SIL 2.

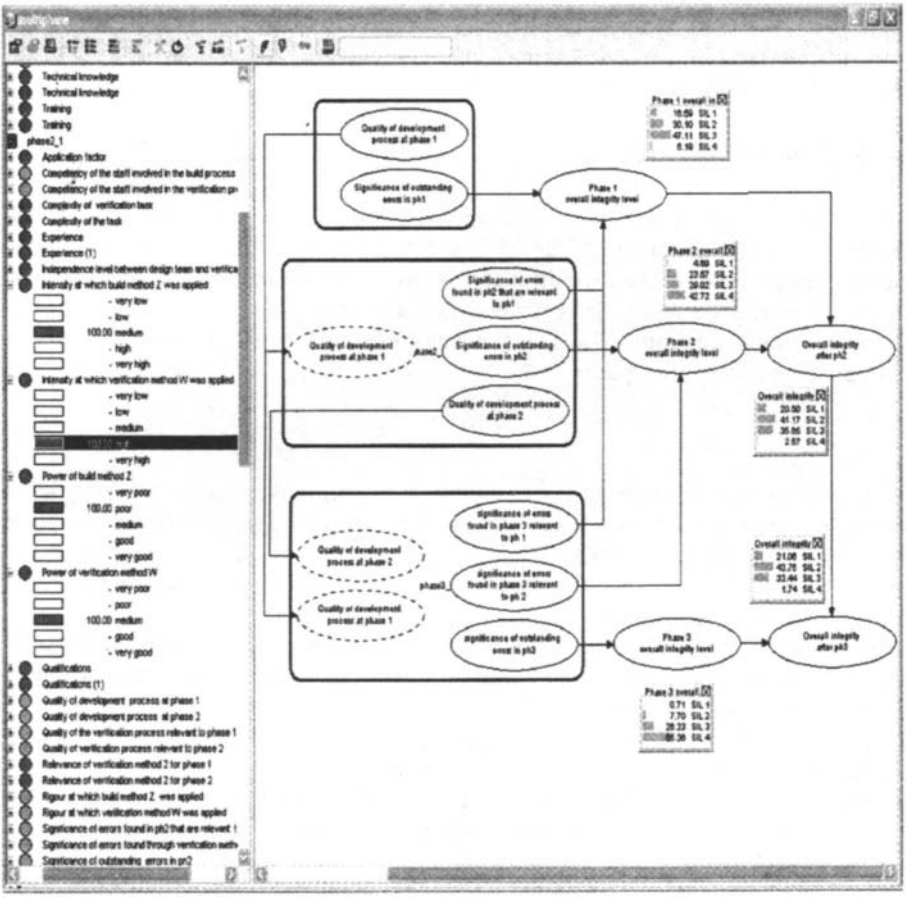


Figure 7- Results of phase 1 integrity estimation when taking into account errors found in phases 2 and 3.

7 Conclusions

A Bayesian Belief Network model has been proposed to predict software safety integrity according to a safety standard such as IEC61508-3. There is a strong argument for formalising the underlying reasoning in such safety standards using BBNs. The development and application of a software safety standard such as IEC61508-3 are both highly complex processes. Probabilistic reasoning can

provide a sound framework within which to perform them. The examples in this paper give an initial indication of the promise of BBNs in this respect, although they are quite limited, and there may be elements of safety standards that cannot be captured in this way.

The proposed BBN structures and local probability tables represent an attempt to capture the reasoning in safety standards, but are no means claimed to be fully accurate representations. The BBNs must be evolved further in a process of consensus building amongst domain experts. The proposed prototype BBN structure introduces a novel way to capture the effects that interactions between phases of a standard have on integrity claims.

8 Acknowledgments

We would like to acknowledge our sponsors the Health and Safety Executive (under contract number 6013/R38.039) and Stirling Dynamics Ltd for providing the necessary funding. We would also like to thank the rest of the SSRC group Marcelin Fortes da Cruz, Silke Kuball and Lorenzo Wijk for their advice and interest. Finally we would also like to thank the editor for his useful comments to the earlier version of this paper.

References

Bishop P G and Bloomfield R E (1996). A conservative theory for long term reliability growth prediction. The Seventh International Symposium on Software Reliability Engineering (ISSRE '96), pp 308.

Black W S (2000).IEC 61508 – What doesn't tell you. Computing and Control Engineering Journal, February 2000.

Brown S (2000). Overview of IEC61508- design of electric/electronic/ programmable electronic safety related systems. Computing and Control Engineering Journal, February 2000.

Fenton N E and Neil M (2005a). Improved Software Defect Prediction. Tenth Annual European SEPG, London 13-16 June 2005.

Fenton N E, Neil M, Marsh W, Krause P and Mishra R (2005b). Predicting Software Defects in Varying Development Lifecycles using Bayesian Nets, submitted to ESEC 2005.

Fenton N E and Neil M (2004). Combining evidence in risk analysis using Bayesian Networks. Safety Critical Systems Club Newsletter 13 (4) September 2004.

Fenton N E, Krause P and Neil M (2002). Software Measurement: Uncertainty and Causal Modelling. IEEE Software 10(4), 116-122, 2002.

Fenton N E, Krause P and Neil M (2001a). Probabilistic Modelling for Software Quality Control. Sixth European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty September 19-21, Toulouse, France , 2001.

Fenton N E and Neil M (2001b). Making Decisions: Using Bayesian Nets and MCDA. Knowledge-Based Systems vol. 14, pp. 307-325, 2001.

Neil M, Fenton N E, Forey S and Harris R (2001c). Using Bayesian Belief networks to predict the reliability of military vehicles. Computing and Control Engineering Journal. February 2001, vol. 12 issue 1, pp 11-20.

Fenton N E and Ohlsson N (2000). Quantitative Analysis of Faults and Failures in a Complex Software System.IEEE Transactions on Software Engineering, 26(8), 797-814, 2000.

Fenton N E and Neil M (1999). A Critique of Software Defect Prediction Models', 25(5) IEEE Transactions on Software Engineering, 675-689,1999.

Gran B A (2002). Assessment of programmable systems using Bayesian Belief nets. Safety Science 40 pp 797-812. 2002.

Hall P, May J, Nichol D, Csachur K and Kinch B (1992). Integrity Prediction during Software Development. Safety of Computer Control Systems. (SAFECOMP'92), Computer Systems in Safety-Critical Applications, Proceedings of the IFAC Symposium, Zurich, Switzerland, 28-30 Oct 1992, 1992.

IEC61508 (1998 - 2000). IEC61508 functional safety of electrical/ electronic/ programmable electronic safety-related systems parts 1-7. 1998-2000. Published by the International Electrotechnical Commission (IEC), Geneva, Switzerland.

McDermid J and Pumfrey D J (2001). Software safety: Why is there no Consensus?. Proceedings of the 19th International System Safety Conference, Huntsville, AL, System Safety Society, P.O. Box 70, Unionville, VA 22567-0070

Pearl J (1998). Probabilistic reasoning in intelligent systems. Morgan Kaufmann, San Mateo 1988 (revised 1997) 0934613737.

Weaver R A, McDermid J A and Kelly T P (2002). Software Safety Arguments - Towards a Systematic Categorisation of Evidence. Twentieth International System Safety Conference, Denver, Colorado, USA, August 2002.

Weaver R A, Despotou G, Kelly T P, McDermind J A (2005). Combining Software Evidence – Arguments and Assurance. Twenty seventh International Conference on Software Engineering (ICSE): Workshop on Realising Evidence-based Software Engineering. St. Louis Missouri, USA, May 2005.

ADDING DIMENSIONS TO SAFETY CASES

Safety arguments for use with data-driven safety systems

Alastair Faulkner
CSE International Ltd
Flixborough, North Lincolnshire UK

Abstract

Well-managed data is fundamental to the dependability and operational integrity of a system. Many systems are not only reliant on data, but also the integrity of data. Therefore data should be addressed as part of the system safety case in common with other elements of the system. The system safety argument(s) should address the use of data and the influence of data errors on the system behaviour. However responsibility for data and its associated data integrity is often poorly defined. This lack of clarity allows vendors to abdicate responsibility for data, and its integrity to the client.

This paper discusses arguments that might be used to justify the use of data within safety systems.

1 Introduction

Large-scale systems tend to be complex, we only become aware of many of these systems through failures reported in the press. The sheer size and complexity of these computer-based systems is often an obstacle to comprehension. A major constraint to comprehension is difficulty of visualisation founded in the variability of often hybrid architectures. This variability leads to a requirement for standardisation, not least in the visualisation of such systems. This problem also extends beyond visualisation, into the realisation of the system and its justification in terms of the system safety case. Without adequate comprehension a desire to facilitate system administration and maintenance often results in unmitigated single point failure, such as the routine application of patches or during operating system upgrade [OGC 2000]. With hindsight such failures appear as gross oversights on the part of the designer, implementer and administrator.

Safety is a property of the operational system and therefore safety arguments should address all the operational domain including the normal and degraded modes of the system, its maintenance, data updates and data passed across the boundary of the operational system. System hazard identification and subsequent hazard analysis should consider all system hazards including those that arise from data errors. If data is not considered within the hazard analysis stage, no data-related hazards will be identified, suggesting that the data has no specific safety

requirements. This is perhaps the reason why no specific safety integrity requirements are normally assigned to the data. Experience shows that data is often poorly structured, making data errors more likely and harder to detect. Hardware or software elements requiring high integrity will often make use of fault detection or fault tolerant techniques to overcome faults. Since data usually has no integrity requirements, such techniques would not normally be considered. Since data often has no specific safety requirements and no safety integrity requirements, it is common not to verify the correctness of the data. When the completed system is validated this will clearly provide some validation of the data used in *this* implementation of the system, but will give little confidence in the safety of other installations that use *different* data sets.

1.1 Integrity requirements based upon numerical targets for failure rates

Integrity requirements are commonly expressed as numerical targets for failure rates for each function of the system or sub-system. To attain the required integrity level the system should not only be developed using the techniques and measures recommended by standards such as IEC 61508, but this integrity level should also be demonstrated by the system whilst in service. This demonstration will be through the achievement of the appropriate target failure rates while in service, when supported by appropriate maintenance procedures.

In basing the integrity requirements upon minimum failure rate targets, standards such as IEC 61508, take no account of the scale or size of the system. These numerical targets present difficulties for the implementation of a large-scale system; either each component of the system attains the highest possible integrity, so that when these components are combined, they result in the desired system integrity or the system components are low integrity and in the limit the system attains little or no integrity. The rule of thumb becomes, the 'larger the system to be developed, the greater the amount of effort required to achieve the minimum failure rate'. Therefore the development of larger systems requires improvements in both process and design (architecture) to achieve these minimum failure rates.

1.2 Apportionment of the system safety requirements

Integrity requirements expressed in the form of an allowable failure rate can be considered as a failure budget for the system. The definition of the system architecture plays a large part in the determination of the integrity requirements and is used in the apportionment of the failure budget between the system components.

While integrity requirements are more than just a set of target failure rates, these targets *are* of importance. In an arrangement consisting of several component parts in a series (in other words, all of which are necessary for the correct functioning of the system), the overall number of system failures will be

equal to the sum of the system failures produced by each component. Therefore, if a system consists of hardware, software and data [Storey 2002] elements, the overall target failure rate may be apportioned to provide separate failure rates for each element. This implies that in a data-intensive system of a particular SIL, one aspect of the data integrity requirements should be that data errors should not produce system failures at a rate greater than that allocated to the data component [Storey 2002]. It also implies that the data will require a target failure rate that is *lower* than the figure given for the corresponding SIL.

2 The characteristics of *configurable* systems

It is common for systems to be configured, adapted to the specific instance of its application. This adaptation might include a description of the environment in which the application is to operate. However the configuration may not be achieved by data alone. In many cases this configuration is achieved through the use of a mixture of software and data.

2.1 Systems configured by *software*

Where software is used to configure a system, this software may be classified in terms of its functionality. One such software classification is described by Duggan [Duggan 2003, Duggan 2004] using the axis of configurability and functionality giving rise to a box model of six categories (see Fig 1).

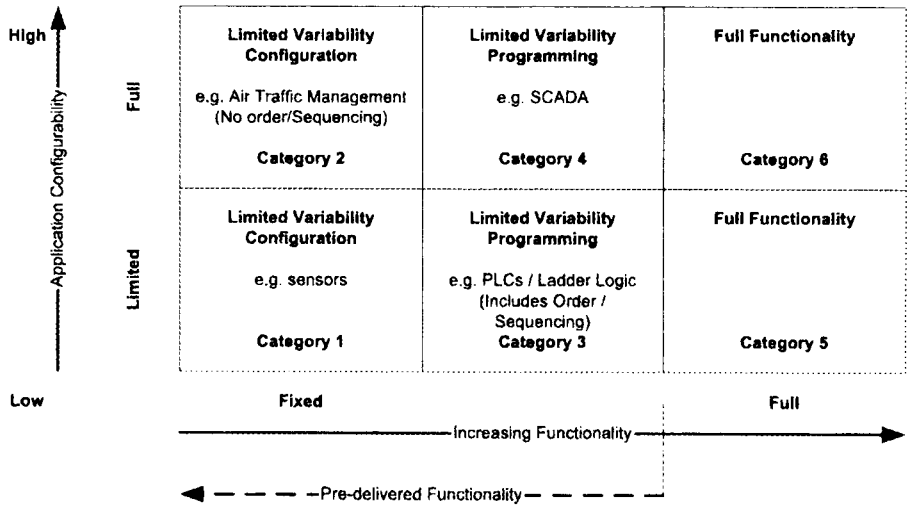


Figure 1: – A classification of software used to configure safety systems

There are many different types of systems, each may be configured to a greater or lesser extent. In many of these systems the question of *is it data or is it software* [Faulkner 2001] is complex and dependant upon each system, its application environment and the tools available to produce and maintain the configuration. In general software should be treated as required by the existing techniques and measures described in standards such as IEC 61508 [IEC 61508 2000].

2.2 Systems configured by *data*

Where data is used to configure a system, the characteristics of this data and its use by the system might be a significant influence on the integrity of the system. Therefore the safety analysis, safety management and safety argument ought to address the integrity of the data and the influence of data errors on the integrity of the overall system.

Safety analysis, safety management and safety arguments should recognise that as well as being configured by data the system may also consume and produce data. The system may also be part of a hierarchy of systems. Where such a hierarchy has safety responsibilities, the production of data by one system and its consumption by another system, may provide a means for the propagation of data and data errors across the hierarchy. Therefore a safety management system should take into account data exchanged across the system boundary and between layers within the hierarchy.

As data-driven systems become larger they tend to make more extensive use of data, and the identification and management of data integrity becomes a significant factor in the demonstration of system integrity. Larger systems often form part of a hierarchy of computer systems that share data. The various elements of this hierarchy will invariably use the data in different ways, and will impose different requirements on it. Where data-intensive systems are linked to distributed information systems, the same data may be used by a range of machines for very different purposes. Under these circumstances the requirements of the data (including the integrity requirements) will vary between these machines. For example, in a railway system, data that represents the current position of the trains is used by signalling control systems (where its use is safety-related) and also by passenger information systems (where it is not).

Implicit in the development or implementation of a data-driven system is a description of the data model and the data requirements. The data model, in common with other system components, should be developed to the same integrity as the overall system. Unfortunately, experience and anecdotal evidence suggests that this is not commonly the case. Development of the data model is complicated by the fact that many systems interface with peer, subordinate and supervisory systems [Faulkner 2002, Storey 2003].

2.3 The use (and re-use) of data by safety related systems

Within a systems hierarchy [Faulkner 2002, Storey 2003], data is exchanged across external interfaces, and across internal interfaces. In such a hierarchy, data may be used within several layers. Each system component may use this common data in a different context and hence re-used data may be also attributed a subtly different meaning for each usage. The layered model [Faulkner 2002] allows the visualisation of the extent of the influence of these data structures, data elements or data items. This visualisation may identify a requirement for validation at an interface to preserve the integrity of one or more safety functions.

2.4 Rules for exchanging data between systems

The rules for data exchange between systems are derived from practices common to many safety-related standards. The requirements presented below are adapted and extend from the DISC report [Petersen 1998]. Data should only be shared amongst systems when the data integrity requirements of each consuming system are fully satisfied.

The requirements are that:

1. The data integrity requirements of all sub-systems or applications within the system are documented;
2. Data may be passed from a higher integrity system to a lower integrity system (provided that the data from the higher integrity system exceeds the data integrity requirements of the lower integrity system for each of the data elements passed across the interface, including error rates and error modes);
3. Data may be passed between systems of the same integrity requirements if and only if these data integrity requirements are compatible, including error rates and error modes;
4. Data may not be passed (without verification) from a lower integrity system to a higher integrity system unless data integrity requirements are compatible, as this low integrity data, by definition, may contain a data error rate greater than that required by the high integrity system; and
5. The hardware and software components of these systems meet the integrity requirements for each system.

3 Safety Analysis

The design, development and safety analysis of computer-based systems is well treated within both standards and the literature within the computer science domain. Many of these standards are based upon the requirements of large customers such as military or government projects. These projects range in size from small self-contained systems to large projects including submarines or social

security systems. All these systems are developed to a set of requirements, which describe the desired properties of the system. These safety analysis techniques must be adapted if data-driven safety-related system technology is to be effectively and safely exploited. In addition it is essential that those responsible for reasoning about the safety of systems have sufficient guidance on the safety aspects of the use of, and reliance upon, data by safety-related systems for safe operation.

The safety analysis of a configurable system should take account of the nature of the configuration, the tools used to create and manage that configuration and additional features such as in service changes to the generic part or the configurable parts of the system. Perhaps the pertinent question would be where does the information come from that will give rise to the configuration of the system?

One example of a data-driven application are Air Traffic Control (ATC) systems that use data in several forms, a static description of the airspace and the aircraft (including its capabilities and capacities), dynamic data to represent the instantaneous position of the aircraft in flight, a command schedule (a set of flight plans) to describe the intended use of the system and data representing the current operational conditions (such as weather conditions) which may constrain the use of the system.

The safety analysis should establish the nature and role of the data within the system and as a consequence, document the influence of data errors on operation of the system. This safety analysis should note that the lifetime of the data may exceed the specified working life of several generations of the implementation of the system [Needle 2003].

3.1 Documenting the safety management strategy

The documentation of the safety management strategy should address the application, its configuration and the provision of data including any requirement to upgrade the application hardware, software or data.

This safety management strategy should at a minimum address:

1. The determination and documentation of the method to be used for hazard identification and safety analysis in determination of the system integrity requirements and their subsequent apportionment to the system components, hardware software and data.
2. The party (or parties) responsible for the provision of data of the required integrity.
3. The data architecture including the application, system and enterprise data architectures as the same data may be used (and re-used) by several systems within the systems hierarchy. Each system may interpret this data and attribute to it subtly different meanings;
4. The scope and extent of safety functions, particularly those safety functions that might be influenced by data (and data errors). This data may be part of the configuration of the system or passed across the system boundary. Where data is exchanged between system the safety management strategy

should document rules for the exchange of data between systems of differing integrity.

5. The controls to be applied to the possible propagation of data errors across the system hierarchy. These controls might include specific requirements for the verification of data at the system boundary.
6. The processes, procedures tools or transformations used to provide data for the system.
7. The processes to be used to update or wholly or partially replace data in the in service system.

3.2 Establishing the system boundary

Establishing the system boundary will require consideration of the mix of system components. These components (hardware, software, data, process and procedure) should be balanced so that no one aspect of the system attracts a high integrity requirement that may be difficult to realise. In addition where the system is required to attain high reliability, availability and dependability factors such as diversity and fall back should also be assessed. Often a system will provide adequate performance during normal operation. Inadequate consideration of degraded modes of operation often result in some system components attracting additional integrity requirements. An inspection of the system hierarchy framework identifies a number of requirements that might be a significant influence upon the arguments structure to demonstrate that the system integrity (and data integrity) requirements have been attained:

1. The boundary of the safety-function should be limited to the first four layers of the data framework (*plant, plant interface, reflex* and *supervisory* layers [Faulkner 2002]);
2. External data systems should not be assumed to use a single representation of the system described in a compatible set of data models. Indeed many of these external systems have been created over a substantial time period;
3. Data passing through these external systems will be subject to a number of adaptations and transformations; and
4. The integrity of the safety function is therefore dependent upon the integrity of the data not only from static data, but also on dynamic data passed to the system through its interfaces with external systems.

3.3 Determination of system (and data) integrity requirements

The determination of the system (and data) integrity requirements depends upon the identified system boundary, taking account of the system architecture.

A suitable classification of data should also be proposed. The author proposes a data classification of *static configuration, operational, dynamic status data* and the *schedule* [Faulkner 2002]. This data classification may be used as the basis of an

analysis to establish the data integrity requirement by consideration of the types of failure that each of these forms of data might exhibit. The choice of analysis method should be appropriate to the system and its context. For the purposes of illustration this paper will use Functional Failure Analysis (FFA).

FFA is based upon the functions or active components of the system. For each function the effects of failure are considered using each of the guidewords. These guidewords are prompts and may require interpretation in the context of the system, but this interpretation should be consistently applied for the system under study.

The basic FFA process is to:

1. Identify the functions;
2. For each identified function, suggest (data) failure modes, using the keywords;
3. For each failure mode, consider the effects of the failure on the system (this may require the development of a number of operational scenarios); and
4. Identify and record any actions necessary to improve the design.

A set of guidewords is used to prompt the consideration of possible failure modes. The guidewords used could be the classical FFA guidewords; *not provided when required, provided when not required* and *incorrect operation*. Pumfrey [Pumfrey 1999] identifies a five-guideword set in the SHARD method. In this method, the guidewords are *omission, commission, early, late* and *value failure*. Harrison and Pierce [Harrison 2000] adapt SHARD and identify guidewords to describe the faults in the data used to describe the railway static infrastructure of *omission, spurious data (commission), positioning, topological, addressing, type, labelling* and *value* (scalar). Faulkner [Faulkner 2002] extends these guidewords to consider operational, dynamic and schedule data. Dynamic data may be considered to comprise two distinct components that status data derived from equipment directly connected to the control system within the control area, and that data presented to the control system through interfaces to external information systems.

The FFA process should be applied at the highest level of the system to allow the apportionment of data integrity requirements between elements of the system at each level of decomposition of the system design.

4 Safety arguments for configurable systems

The system safety case will contain a series of arguments supporting the overall assertion that the system is tolerably safe to be used in the specified environment in all of its operational modes. Configurable systems require additional safety arguments to address the generic application, its application instance and specific arguments addressing the configurable components the tools used to create and manage the configuration including the data component.

Systems that make extensive use of data will require additional safety arguments to address the use made of data, the verification of the data and the means used to control the propagation of data errors across the systems hierarchy. The use of data presents additional safety requirements where data may be updated or passed across the system boundary during its normal operation. One such requirement is the verification of data at the system boundary.

A key feature of these safety arguments is the responsibility for the integrity of the configured part of the system. This statement of responsibility should also address data consumed by such systems. Safety arguments for systems configured by software will address the nature of this software and the tools used in its production. For a data-driven system the author asserts that the consuming system will to be responsible for the data it consumes, including its integrity, as this policy is aligned with the establishment of the system boundary and balances the need to describe how the system is tolerably safe.

4.1 Safety arguments for the *System*

A computer-based system may contain several applications and computational environments. The system level safety argument should address the system, its behaviour, its process and procedures as well as the training and competence of the human operators.

Large-scale systems may implement safety functions across several application and/or computational platforms. These extended safety functions, particularly where they are coupled through the use of data passed between applications, present an opportunity for this data (and the data errors that the data may contain) to influence the integrity of the safety function.

The system level safety arguments should address the use of the system within the system hierarchy model, the coupling between systems and the interaction of one or more systems under a range of operational conditions. These operational conditions should include the degraded modes of one or more of the applications.

Within the system hierarchy data may be shared by many systems. The system level safety arguments should address the System Architecture and in particular the data architecture at the level of the *Enterprise*, *System*, and *Application*. These system level safety arguments should identify any data management and data verification tools. In particular the system level safety arguments should address the planning and implementation of data validation. In addition the system level safety arguments should address verification at the system boundary.

The system level safety arguments should address how would the system recover from exceptional events (fire, flood etc). If an interim service planned what would a limited service be; how would it be co-ordinated; will it be tolerably safe. Exceptional events although rare are likely to be high consequence. An example of an exceptional event is the closure of American Airspace immediately after 9/11.

4.2 Safety arguments for the generic application

In this paper the generic application is taken to be the core product that may be used in several installations. The safety arguments for the generic application address the safety of this application in a sterile but representative environment. This is typically the development environment and a limited number of field-trial installations.

4.3 Safety arguments addressing the application instance

A specific application environment of the generic system will contain hazards already addressed in the generic application safety arguments and may also contain additional hazards. In addition the data used to describe this application environment may contain a mix of application features or combinations of data elements that require specific safety arguments. In particular safety arguments will be required to address rules developed to verify data at the application boundary and any modification of the tools used to create and manage the system configuration.

4.4 Safety arguments for the data update process

The use of a configurable system presents specific requirements for additional safety arguments. The author asserts, that in addition, a safety case is required for the process of undertaking the data update process. Safety arguments within this process safety case would address the hazards presented in the creation and management of the datasets, the verification of the data, the update process itself and the provision of a fall back process to re-install the previous data set should the update be unsuccessful.

A desirable feature of the data design would be a tag that indicated the version of the data architecture to facilitate feature or version locking between the dataset and the application. This feature or version locking is also desirable when the application is upgraded, as the new application might not be compatible with the existing dataset. This versioning feature should also be extended to the tools used to create and manage the dataset to provide audit trail as one means to aid investigation into data error.

4.5 Safety arguments addressing data provision

The use of a data configurable system requires the safety management system to address the *provision* of data. Data provision is the term used to describe the logistics of transportation, collation, transformation and preparation of the dataset used for the data update. The overall data integrity requirements will be apportioned between the data origin the processes and logistics used to create the

dataset and the dataset itself. The logistics of data provision are described as a *data supply chain.*

Safety arguments for data provision should address the data origination, the tools used in the data supply chain and in the creation of the dataset. The use of a data supply chain requires that changes to the data architecture are rolled out across the data supply chain, the tools it employs and the criteria used to verify the data.

4.6 Safety argument addressing backward compatibility

A common feature of many configurable systems is the use of version and feature locking. In particular feature locking may be used to restrict the features of the application of the system. One use of feature locking is the ability to add a new feature to the application and/or its data structures, to have these features released in the executable code. Only at some future time would these additional features be enabled without replacement of the existing executables at that future time.

The use of feature locking requires safety arguments to address non-interference of the disabled features with the operation of the system and to provide a means for the use of roll back to some known stable (and tolerably safe) state. These safety arguments might require one or more generations of the application and its associated datasets to be held as a backup to be restored in the event of failure. This position is made more complex where the system contains many installations each at a different of version.

5 Discussion

The use of configurable systems, particularly data-driven system requires an extensive series of safety arguments. These safety argument need to address the application and its context within the systems hierarchy paying particular attention to data passed between systems of differing integrity and across the systems boundary.

A key feature of these safety arguments is the responsibility for the integrity of the data consumed by such systems. The safety management strategy should contain a statement that the consuming system will to be responsible for the data it consumes, including its integrity, as this policy is aligned with the establishment of the system boundary and balances the need to describe how the system is tolerably safe.

The selection and identification of the system boundary is a key feature of the safety analysis. This safety analysis may used demonstrate that elements of the system attract high integrity requirements. A reformulation of the system design and its system architecture may yield elements of the system that attract lesser integrity requirements. This process of analysis and reformulation is also applicable to the data architecture and its constituent parts. The formulation of the system data architectures (enterprise, system and application) addresses the

structural design of the data. Additional consideration is required the provision of data and for data *content* errors.

Safety arguments are also required to address data provision and its data supply chain. A dataset presented at the system boundary should be subject to verification. This dataset should be compliant with the specified integrity requirements. Establishing the system boundary also implies that this boundary should be coincident with an organisational boundary. The question of liability for data content errors is left as the subject of a future paper.

The updating and upgrading of configurable systems require safety arguments to address both feature and version locking. The upgrade or update process should be reversible. That is should these processes fail the system should be restored to its former tolerably safe state. Any structural change to data to a system that uses external data provision, particularly those that employ a data supply chain will require process based safety arguments that extend down data supply chain to the data origin. The logistics of these arguments may be extensive where data drawn from many sources and may be provided too many consuming systems.

6 Conclusion

Data used by a safety-related system should be classified based upon the uses made of the data and the way in which the data influences the behaviour of the system. The nature and influences of data faults will also vary with the form and use of the data within a system. Data integrity requirements are essential if the suitability of data models is to be assessed. This paper has presented a process by which these data integrity requirements may be established. Additional design analysis may identify that the structure and composition of the data set or that data from the real world cannot be obtained in either the quantity, nor of the requisite quality. These data integrity requirements may also be used to identify verification and validation requirements for the system.

The key to successful data management lies in the use of well-designed data structures that permit and ease verification. Good design practice requires the design of components, which have low coupling. Such components are usually modular, with interfaces that are resilient to changes in design. Isolation of data modules is also important since this can dramatically reduce the effort required for system validation.

This paper has presented a series of requirements for the safety management strategy of configurable systems focused primarily on data-driven system. In addition this paper has also outlined a number of possible safety arguments addressing the system, its context within the systems hierarchy and the safety arguments required to support data provision.

References

[Duggan 2003], P Duggan, Presentation "*Configuration of Data Driven Systems, an Assurance Perspective*", April 2003
http://www.csr.ncl.ac.uk/calendar/csrEventView.php?targetId=160

[Duggan 2004], P Duggan, "*Data Driven Systems and their configuration, Safety Systems*", the newsletter of the Safety Critical Systems Club, Vol 16 No 1; 28:127-162

[Faulkner 2001], A. Faulkner and R. H. Pierce, "*Is it Software or Is it data*"; Proceedings of the 19th International Safety System Conference 2001, Huntsville. Alabama, USA. pp 323-329.

[Faulkner 2002], A. Faulkner: *Safer Data: The use of data in the context of a railway control system*", Proc. 10th Safety-critical Systems Symposium, pp 217-230 ISBN: 1-85233-561-0, Southampton, UK (2002)

[Harrison 2000], A. Harrison and R. H. Pierce. *Data Management Safety Requirements Derivation.* Railtrack: West Coast Route Modernisation Internal report. June 2000. RAILTRACK PLC, London 2000

IEC 61508 (2000), International Electrotechnical Commission; IEC 6150 *Functional Safety of electrical / electronic / programmable electronic safety-related systems*": 2000 Definitions and abbreviations. Geneva 2000.

[Needle 2003], B. Needle, Presentation "*Data is for life: data attributes should support the proposed life of a system, particularly validation*", April 2003 http://www.csr.ncl.ac.uk/calendar/csrEventView.php?targetId=160

[OGC 2000] Office of Government Commerce, 2000, "*A Review of Major Government IT Projects*", available at: http:// www.ogc.gov.uk/.

[Petersen 1998], DISC Consortium, '*Research into waterborne transport area, Demonstration of ISC – DISC: Final report*', Erik Styhr Petersen, SCL DISC Project Manager, Ref D101.00.01.047.003C pp 33-35

[Pumfrey 1999], D. J. Pumfrey; "*The principled design of computer system safety analysis*", DPhil Thesis; Department of Computer Science, University of York; 1999

[Storey 2002], N. Storey and A. Faulkner: "*Data Management in Data-Driven Safety-Related Systems*"; Proceedings of the 20th International Safety System Conference 2002, Denver. Colorado USA. pp 466-475 ISBN 0-9721385-1-X.

[Storey 2003], N. Storey and A. Faulkner: "*The Characteristics of Data in Data-Intensive Safety-Related Systems*"; Lecture notes in computer science - Proceedings of the 22nd International Conference SafeComp 2003, pp 396-409, ISBN 3-540-20126-2.

Gaining Confidence in Goal-based Safety Cases

Rob Weaver and Tim Kelly
Department of Computer Science, University of York,
York, UK

Paul Mayo
Silver Software Consultants Ltd
Malmesbury, UK

Abstract

Goal-based safety standards are now a reality. As well as confidently satisfying the mandatory requirements laid down in these standards, there are a number of other secondary factors that influence the confidence a regulator or assessor has in any safety case produced. These factors include novelty of argument approach, experience of stakeholders and scale of the system and safety case. Currently, the certainty with which requirements are satisfied and the consideration of the other confidence factors often remains implicit within the certification process. In a goal-based safety case regime, users and regulators require intelligent customers who are aware of these issues and can explicitly consider them during the production of their safety case to increase the confidence of all stakeholders involved. Standards, guidance and other publications have covered in detail the structure and content of safety cases and this paper does not intend to repeat this information. Instead, this paper brings together and discusses the other confidence factors and approaches to managing them within the safety case development process.

1 Introduction

Within a regulatory process using goal-based standards (such as DS 00-56 (MoD 2004a) and SW01 (CAA 1999)) the acceptance of a safety case requires the assessors to be confident that the safety case meets the requirements laid down in the standard. However, both the developers and the assessors can sometimes be uncertain that the safety case has high assurance. An example of this type of problem is where standards require "sufficient" evidence. Determining, what amount or what types of evidence are sufficient can be extremely difficult. If we are uncertain about the sufficiency of the evidence, then our confidence in the safety case is reduced.

As well as this aspect of safety case assurance, which is affected by confidence in requirements satisfaction, uncertainty can also be caused by other factors currently outside the scope of the "technical" safety requirements. An example of this might be a

situation where the safety argument is developed by external consultants. In this circumstance, there may be concern about the ownership of the safety case by the developers. If a developer relies on someone else to construct the safety argument, an assessor may be less confident that the argument addresses all the safety issues of the system (about which the external consultants may have a less detailed domain knowledge). Such factors are often only considered implicitly within the safety case development and acceptance processes. They are used to determine the level of belief in the safety case, but are often not explicitly addressed within the safety case. Exploring these concepts gives both developers and regulators an opportunity to assess their impact on confidence and, more importantly, begin to find approaches to increasing assurance.

1.1 Types of Uncertainty

Uncertainty can be defined as "lacking complete confidence or assurance". During safety case assessment confidence can be affected by epistemic uncertainty, which relates to a lack of knowledge (Thunnissen 2005). Safety case assessment is primarily concerned with determining that the level of knowledge about the acceptable safety of the system is sufficient. Epistemic uncertainty within safety case assessment can be classified in two ways: Information Uncertainty and Inadequate Understanding (Lipshitz & Strauss 1997)[1].

Information uncertainty can be caused by completely lacking, partially lacking or unreliable information. Where safety arguments are built upon identifiably incomplete evidence, assessors will be less confident in the assurance provided by the safety case. We discuss the issues surrounding this type of uncertainty in Section 2.

Inadequate understanding can be caused by factors including equivocality, instability, tractability or novelty. With this type of uncertainty, it is not that the information is not present (and could be); instead, it is that the information cannot be known due to a lack of basic understanding of the subject. For example, with a novel technology the knowledge often does not exist to construct (and certify) the safety argument with confidence. Due to the fact that we have not previously built similar systems, the understanding of how to argue that the system is safe may be lacking. We discuss the issues surrounding this type of uncertainty in Section 3.

In broad terms, where understanding already exists within the safety community, reducing uncertainty is focussed on presenting the correct information at the correct time (i.e. reduction of information uncertainty). Where understanding is lacking within the safety community, there are more fundamental issues that must be addressed to increase confidence (i.e. reduction of inadequate understanding).

Confidence in a safety case can be affected by not putting into practice understanding that currently exists, or by not having the fundamental understanding. In this paper we attempt to address confidence in safety cases by considering these two different types of uncertainty. In the following subsection, two requirements from Defence Standard 00-56 are presented as examples which demonstrate the difference between the areas where we currently have a suitable level of understanding and problems arise due to what is actually presented (i.e. information uncertainty), and areas

[1] A third category - conflict - exists where uncertainty is caused by equally plausible positive or negative conclusions or incompatible role requirements. This is not relevant to safety case assessment.

where we lack the basic understanding to be able to improve uncertainty (i.e. inadequate understanding).

1.2 Addressing Uncertainty through Requirements

1.2.1 Information Uncertainty

Many of the requirements laid down in standards aim to increase confidence. For example requirement 9.1 from Defence Standard 00-56 (issue 3) states:

> "A Safety Case is a structured argument, supported by a body of evidence, that provides a compelling, comprehensible and valid case that a system is safe for a given application in a given environment."

Due to the experience that already exists in the safety critical community with respect to argument construction, addressing this requirement with confidence is primarily concerned with reducing information uncertainty. A large amount of research and discussion around the construction and presentation of arguments already exists to aid satisfaction of this requirement and many papers and guidance have been published, including (MoD 2004b, Kelly 1998, Bishop et al. 1998, Weaver 2003). In simple terms, developers have the basic knowledge available to satisfy this requirement. We know how to build "compelling, comprehensible and valid" cases, and so the assessor's confidence will only be affected by how this knowledge is put into practice (i.e. whether it is ignored or not).

1.2.2 Inadequate Understanding

However, there are numerous other requirements that aim to reduce uncertainty which have been considered in less detail and thus we lack the knowledge or understanding as to how to increase confidence. An example is requirement 9.5 from Defence Standard 00-56 (issue 3):

> "The Safety Case shall contain a structured argument demonstrating that the evidence contained therein is sufficient to show that the system is safe. The argument shall be commensurate with the potential risk posed by the system and the complexity of the system."

To increase confidence, knowledge is required with regards to determining *sufficiency of evidence* and arguments that are *commensurate with the potential risk and complexity of the system.* From an information perspective, an assessor can ask for more (or more reliable) evidence to be presented. However, for the developer and assessor to determine (in abstract terms) what is sufficient and thus reduce uncertainty is, presently, far more difficult.

The second sentence in part addresses the complexity of the system. As the complexity of a system increases, the uncertainty about the safety case can increase due to an inadequate understanding of the relationships and dependencies between parts of the system. This intractability is not due to the fact that we lack information about the dependencies and relationships in a more complex system. It is more due to the fact that we lack the understanding about the *implications* of the dependencies.

Section 2 discusses factors that are known to help to gain confidence, but are often not considered. Section 3, discusses more complex and fundamental issues which affect uncertainty and about which we have less understanding regarding their resolution. This discussion aims to prompt developers and assessors to think more explicitly about consideration of these areas in the products and processes of safety case development. It also aims to promote areas where further research is needed.

2 Information Uncertainty in Safety Case Development

As described above, confidence can be affected by not putting into practice knowledge about how to reduce uncertainty, which already exists. For the majority of concerns that fall into this category (e.g. how to gain clarity and structure in the presentation of safety arguments), there is a large amount of material in the form of publications and guidance, which can be used to help remove this uncertainty. However there are some areas in which knowledge and experience exists, but remains implicit or is poorly observed in practice. These areas are discussed in more detail in the following subsections.

2.1 Time and Speed of Argument Production

A safety argument that is produced quickly and late on in the safety lifecycle may inspire less confidence than an argument produced upfront and developed continuously. Underlying this lack of confidence, is typically scepticism that acceptable safety arguments can be produced post design completion where there has been little or no consideration of the required safety arguments during system development. When the creation of a safety argument is delayed whilst influence on a system design is limited, it remains possible to generate further forms of evidence. In extreme cases, this can mean that evidence is 'tweaked' until an acceptable position is established, or that a bulk of evidence is used to disguise an inadequate design. Safety cases built late in the lifecycle also tend to rely on incorporating mitigation strategies such as safety devices, warnings and procedures rather than redesigning to eliminate or reduce the likelihood of a hazard. It is widely known that development of the safety argument should be initiated early on in a project and should continue from requirements definition through to commissioning and beyond. Where safety argument design is left until the final stages of design, there is potential for large amounts of redesign late on in the lifecycle. In the worst cases, systems may have to be discarded or redeveloped. The loss of safety rationale for design decisions due to a lag in argument production makes the process far more difficult. In essence it can be difficult to remember why things were done after the event, so other reasoning has to be found to demonstrate safety. Inherently it becomes more difficult to create an argument the later the process is started in the lifecycle.

As well as time of production, speed of production can affect confidence in the safety case. An argument that is produced quickly may, from the assessor's point of view, have a greater potential to be incomplete or inaccurate. A more continuous process, which involves regular communication between designers and safety case developers, allows the development of a more considered, and higher confidence safety case.

A phased safety case production process allows the developers and assessors to have an increasing confidence in the final product. One of the best ways to reduce uncertainty is to reduce it over time. The staged production and delivery of safety case reports allows discussion between stakeholders about the merits of a particular argument approach. This goes some way to addressing the sufficiency issue - agreement can be reached before evidence production that the evidence set is sufficient. Common intervals for safety case report production are:

- Preliminary Safety Case Report
- Interim Safety Case Report
- Pre-Operational Safety Case Report
- Operational Safety Case Report
- Decommissioning / Disposal Safety Case Report

Even where this approach is not mandated, there can be significant benefits for safety case developers obtaining feedback from regulators early in the lifecycle concerning the proposed argument approach.

2.2 Acceptability of Assumptions

Assumptions are an integral and inevitable element of any safety case. Assumptions impact on the scope and nature of the arguments and evidence presented. For example, assumptions made concerning the lifetime and maintenance of a system can affect the details of probabilistic risk assessment. Similarly, assumptions made regarding the independence of system functions will determine whether function interactions are explicitly addressed in the safety case.

Assumptions can be considered legitimate and acceptable where there is a genuine lack of information or lack of understanding that cannot easily be resolved at the time the safety case is presented. For example, an assumption made regarding system maintenance procedures may be considered acceptable if the procedures have not been fully defined at the time of safety case production, and responsibility for their production lies outside of the safety case developer's duty.

There can be many assumptions in a safety case considered unacceptable by the above criterion. For example, unjustified assumptions (often implicit) can be made regarding the quality and completeness of evidence presented, the competency of development and assessment personnel, or the role and extent of involvement of Independent Safety Assessors. If the information and intellectual capacity exists to address these issues, and yet has not been presented, confidence in the safety case can be undermined. To mitigate this problem, it is essential to regularly consider and reveal the assumptions underlying any safety case (e.g. through exposing the case to regular peer review). Any assumptions associated with the safety case should be explicitly documented. Once documented, assumptions should be challenged: Does the information and understanding exist to turn an assumption into a supported claim? If the arguments and evidence don't yet exist to support an assumption, it should be identified at what point in the lifetime of the system this information may become available.

2.3 Experience

When it comes to safety-critical systems, engineering competence is basic and essential; however, there are weaknesses with regard to providing a measure of a company's engineering competence that need to be addressed. There are also issues over how safety work should be outsourced when the need arises (or even *if* it should).

Technical competence (MODISA 2004) needs to be understood and measurable. The type of technical competence needs to be determined; is it technical competence in Safety Engineering practices perhaps with formal qualifications? Or technical competence in the domain (or with similar technology) developed through experience? How can technical competence be determined and how much is enough?

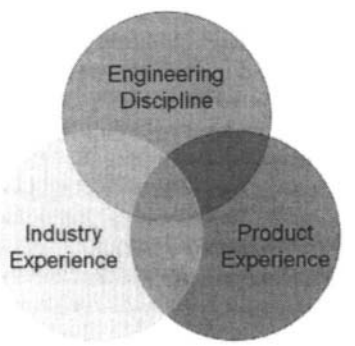
Engineering companies trade on their reputation, which can serve as a measure of their technical competence. However, past performance needs to be balanced against turnover of staff (either permanent or contract) as key knowledge that may have been underpinning past success will leave with the staff that move on.

Safety relies heavily on knowledge of the domain or technology to ensure that hazards are discovered and subsequently mitigated. It also relies on a *safety culture* within the developing organization. This paper has already discussed the need for the safety argument to be initiated early. Clearly safety engineering needs to be interwoven within a company's processes and procedures.

2.3.1 Experience – 'What' and 'How Much' is Enough?

The area of experience is complex. However, a question mark over the experience of a supplier could suggest that information uncertainty exists. Therefore, the safety case needs to convince the reader that the developing organization has sufficient relevant experience – this is particularly necessary when engineering judgement is being relied upon to support elements of the argument.

Whilst this is an area involving personal judgement, experience must be assessable at some level. The experience of the developing organization needs to be considered to be at least adequate by all stakeholders. In order to discuss experience further, a representation is required. The representation chosen is based upon typical engineering job advert requirements and is shown in Figure 1. The figure can be used to discuss both an organization developing safety critical artefacts and a safety engineering capability (both of which could be one person).



Key

Engineering Discipline – Background skill i.e. mechanical, electronic, software etc.

Industry Experience – Sector or area skill i.e. aerospace, rail, nuclear etc.

Product Experience – Particularly deep understanding of a system

Figure 1. Basic Elements of Experience

Proven ability within the Engineering Discipline would appear to be obvious; however, instances still occur where this fundamental and basic constraint is ignored and engineers without electronic and software knowledge are tasked with assessing those types of systems. It is unlikely in this type of situation that anything is presented in the safety report to indicate such an occurrence.

Industry Experience is more blurred. It may be considered acceptable for someone with aerospace avionics experience to work on elements of a rail signalling system and vice versa. However, the nuclear industry requires nuclear experience. An organization could break into a new industry by employing new staff with relevant experience and providing evidence of infrastructure, culture, processes and procedures (perhaps through professional accreditation schemes).

The final element – Product Experience – depends on the size of the system. In most cases it would be acceptable for an individual to initially lack product experience and gain it through an appropriate development programme. This may include courses, on the job training, coaching, mentoring etc.; however, the authors are aware of instances where no training was given and individuals were expected to "hit the ground running".

Clearly, the ideal would be for the developing company to be at the centre of the diagram. However, we know that companies positioned in other areas of the diagram can undertake satisfactory development of safety critical systems. There needs to be research and evaluation carried out to at least focus professional judgement in this key area.

2.3.2 Determining Levels of Domain Knowledge

Domain knowledge can be ascertained through the CVs and affiliations of the company's staff. Additionally, the company could use a competency framework. The Engineering Council UK's (ECUK) mission is:

> "to set and maintain realistic and internationally recognised standards of professional competence and ethics for engineers, technologists and technicians, and to license competent institutions to promote and uphold the standards."

In order to register as a Chartered Engineer, Incorporated Engineer or Engineering Technician, a candidate must meet or exceed the competence standards of ECUK and be a member of a licensed institution. They must have a satisfactory educational base, have undergone approved professional development and have demonstrated their professional competence. A fundamental weakness with ECUK registration is that once registered, there is no further need to demonstrate professional competence. Members are expected to undertake continuing professional development (CPD); however, there is no enforcement. Provided they continue to pay the yearly subscriptions, they remain registered.

A curriculum vitae (CV) can do far more than simply list previous jobs. It could be focused on how the different experiences in a person's career (or in some instances even outside their career) provide evidence of CPD. Thus a correctly structured CV can support ECUK registration to show current professional competency.

The IEE/BCS competency guidelines state that competence requires all practitioners to have qualifications, experience and qualities appropriate to their duties. The

guidelines provide a framework for the development of a competency scheme within an organization. By implementing the framework, a company can provide evidence of its staff levels of competence.

Ideally, in order to reduce the uncertainty in a safety case due to lack of information on experience, the safety case should link to evidence in the form of a combination of short CVs, professional affiliations and competency levels all supporting one another.

In an ideal world, the structure, style and content of a safety case would be enough to convince an approving body of the credibility of the developing organization. However, in reality credibility needs to be explicit and this could be achieved by including some or all of the evidence mentioned above. This is not widely understood and so situations can arise where a company may refuse to give any evidence of individual competence (or even the names of individuals who had been involved in the project). In worst-case scenarios, these types of problems may lead to projects failing to deliver.

2.3.3 Argument Construction by External/Independent Contractors

Great care and thought will need to be exercised if safety is to be outsourced. What level of outsourcing is required and how is it to be managed? If an organization is developing a safety critical product for the first time then it may need to outsource the safety work due to the fact that there is not the capability internally. Clearly in this instance the safety engineering sub-contractor would need to place engineers within the developing organization right from project initiation to ensure continuous communication and safety culture. At the other end of the scale, a developer may simply need to outsource some specialist analysis technique(s) in which case it may be more feasible to send the work to an organization with the appropriate skills.

Obviously the developing organization needs to have domain expertise; but how much domain knowledge is required by the Safety Engineer? Clearly the safety of a product relies on the developer having extensive understanding of the product; however, with the correct approach it may not be necessary for the safety engineer to have a deep understanding – thus allowing outsourcing of safety engineering to become viable. However, engineering discipline knowledge is paramount for ensuring that the correct approach is taken to the safety argument.

A Safety Engineer skilled at facilitating and focusing hazard analysis meetings attended by experts could provide an acceptable hazard analysis with only limited personal knowledge of the industry / product. Furthermore, if they were skilled at constructing Safety Cases they would be able to focus the project towards providing the correct evidence to support it. This, of course, relies upon the developing organization outsourcing the safety work to the right organization from project initiation. Unfortunately, this is often not achieved and leads to the further issues related to late Safety Case construction, which have already been discussed.

2.4 Conclusions on Information Uncertainty

While the concepts presented in this section may appear to be fairly obvious and, whilst (hopefully) many readers may find themselves agreeing with some or all the points made – they will also undoubtedly be aware of instances where they have not been considered – and the resultant effect on the strength of the argument within the Safety Case.

With regard to experience, the effects of not doing things correctly from the start of a project are going to be similar to those of starting the development of the safety case too late. The effect of poor consideration of assumptions also in the end has a similar effect – confidence in the safety case is undermined. The areas of safety case development process, consideration of assumptions and experience, therefore, call for greater understanding so that all stakeholders can reach agreement to effectively support and underwrite a safety critical development.

3 Inadequate Understanding in Safety Case Development

In this section we look at how inadequate understanding about and within the safety case development process effects confidence in the final product during certification. As discussed previously, in these areas confidence is not reduced because of poor or insufficient information in the safety case. Instead confidence is reduced because of a lack of understanding within the process of safety case production.

3.1 Development Process for Safety Cases

An argument that changes radically over the development process will be more concerning than an argument that shows methodical growth. In section 2.1 we talked about the benefits of phased safety case production; however, currently there is still a lack of understanding about how to track the development of safety cases over time. While safety case reports can be produced at important milestones, we do not have approaches to trace the construction of arguments and to learn from how arguments are developed. Guidance exists about the membership of teams which construct safety arguments, however there are still research questions surrounding the collaborative work processes of safety case production. The level and frequency of team interaction that maximises the development process is an area where limited research (specific to safety cases) has been conducted. How teams work across company and international boundaries to develop safety cases has also not been explored in great detail. If we wish to capitalise on the benefits of computer supported collaborative work (e.g. through the use of web-based safety case production, (Mir 2005)), greater understanding is needed about the processes within safety case development. This knowledge, achieved through research about the development process, will allow for more advanced guidance. In a similar way to the increased confidence provided by following guidance about structuring arguments, further guidance about collaborative work process of safety case development would increase confidence in the final product.

Another area of the development process where further research would be beneficial is the role of review. If we have to make a large number of updates to the safety argument based upon review comments, there are implications about the quality of the argument (even after review). It cannot be assumed that all errors in the safety case are captured by review, and it could be said that if a large number of errors are found during review then it is more likely that there are still more uncovered errors. These issues are not always addressed within the safety case, however they can impact on our confidence.

3.2 Use of Safety Cases beyond Certification

Understandably, a large amount of the focus of safety cases is towards the time of certification and the introduction of the system into operation. However, a safety case that generates a high level of assurance should include some plan for the use of the safety case during operation. The role of the safety case at this stage varies widely depending upon industry. To continue to benefit from the safety assurance provided by the safety case, it must be maintained throughout the life and decommissioning of the system. Current research exists which has examined the effect of incidents and accidents on safety cases (Greenwell 2004), however more consideration is needed of this phase of the safety case's life. Questions that must be posed include:

- How do we systematically assess the implications of incidents and accidents?
- How do we feedback in-service experience and compare it with predicted values for failure rates?
- What responsibility is there for addressing increased failure rates?
- How do we assess the impact of procedure changes for operation and maintenance?
- How do we address changes in assumptions and context during the systems operational life?
- Who has the responsibility for revisiting the safety case during system operation?
- Who assesses the extent to which safety cases are reviewed during operation?
- How are safety case changes during operation tracked and implications disseminated to the system users/owners and the owners of other safety cases?

To maintain the safety case we must feedback information into the safety case from operational experience. If the safety case becomes outdated and there is a loss of relevance to the system in operation, the argument upon which the safety of the system is assured loses its validity. The safety case must be used to ensure safety of the system throughout system life and (where necessary) decommissioning. At the time of certification, strategies and procedures should be included as part of the safety case to determine how this feedback process is to be conducted. If this is not done, we cannot say that the safety case will cannot give holistic assurance about the safety of the system throughout operational life.

3.3 Novelty

Clearly novelty has an impact on safety case confidence. An argument approach that hasn't been seen before is more difficult to assess than an approach that has been successfully used. The historical approach to safety engineering has always included an element of "fly and fix", whereby faults are removed once the system has been operated and accidents or incidents have occurred. The necessity for this type of approach (which is not ideal) is due to a lack of ability to assess novelty. Novelty occurs within safety cases due to the use of new technologies and new argument strategies.

New argument strategies might be imposed by the changes made in standards. There is no de facto safety standard that has remained fixed over a significant amount of time. Research and development of safety critical systems continues at pace and

similarly the standards are updated to reflect new understanding. Examples include the recently released Issue 3 of DS 00-56 and the current reworking of DO-178B by Special Committee 205/Working Group 71 (RTCA/EUROCAE). Following new (and hopefully improved) standards may include following a novel argument approach. While a large amount of thought goes into the development of new standards, care has to be taken to make sure that the new argument approaches used to satisfy the standard maintain or improve on the actual level of safety that systems developed to previous standards have achieved. In essence, how do we know that a new standard (which encapsulates a new argument approach) will lead to the production of a safe system? One approach to combating the uncertainty that comes with this form of novelty is to allow (where appropriate) previously existing and tested standards to be used to satisfy the requirements of new standards. One strategy for satisfying the new issue of DS 00-56 is to follow an "as-civil-as-possible, as-military-as-necessary" approach where other standards are used (McDermid 2005). Following this approach, existing standards such as DO-178B can be used to show satisfaction with only military specific hazards having to be addressed outside the scope of the civil software guidelines.

The approach of using tested arguments based on standards with a strong pedigree will not be possible for all systems satisfying new standards. In the majority this is due to the systems employing new technology to which the older standards are not applicable. The novel use of technology may be in the system under construction or in the development process. Novel technology within the system may be due to a change of domain application, or it may have not been previously used within safety critical systems. Current examples of this type of technology include Integrated Modular Avionics and Neural Networks. Inherently there is a lack of understanding with respect to how arguments can be built about these types of new technologies. As with expertise and competency of personnel, it is necessary to build up confidence about new technology gradually over time. Introduction of technology has to occur in small steps so that assurance in the safety of the technology can be built. There is a greater need to review the failures of new technology or technology used in a new context during operation than technology that has been proven to be acceptable safe. Unfortunately the problems of novel technology are exacerbated by the lengthy time scales involved in the development of safety critical systems. It can be difficult to introduce new technology in stages and review performance in operation when the process of development is so long.

Technology within the development process suffers from the same issues as technology used in the actual system. For example, if a new approach is adopted to the construction of software within a safety critical system (e.g. model based development, or the adoption of a software product lines oriented approach) the impact on the certification argument approach can be poorly understood. Development technology could be used for evidence production, system production and argument production. However, in some cases the introduction of technology within the process may not have as strict regulation as when introduced in the system. Again it is important to introduce this type of technology gradually and review when failures occur.

Capitalising upon and reusing existing successful argument approaches is a possible means of reducing uncertainty due to argument novelty. Over time, it is possible to develop mature argument strategies to address well-known safety issues (such as how to argue risks are ALARP). If such strategies are well defined and widely agreed, improved confidence can be gained in any safety case that employs them. However, there is also a potential downside to reusing existing argument approaches. A safety

case argument that is very similar to existing arguments may be viewed with scepticism. Whilst the argument may appear plausible, there can be uncertainty on the part of the assessor that the strategy has been fully understood and that it is truly appropriate for the system under development. To address this problem, information regarding the experience and competence of the safety engineers responsible for the production of the safety case must be presented. In addition, it can be necessary to present information regarding the processes employed in deliberating amongst, and selecting, argument strategies.

Another area where we may see novelty affecting confidence is where evidence is used in a novel manner. In this situation questions can arise about the relevance of the evidence. For example, there may be concerns about the use of historical evidence taken from a different or non-safety critical domain. At a basic level we have some understanding about concepts of relevance, trustworthiness and independence of evidence (Weaver 2003). We know that relevance can be affected by the directness or coverage of the evidence; we know that conceptual independence is more convincing than mechanistic independence; and we know that trustworthiness of evidence can be affected by a number of factors, including:

- "Buggy-ness" – how many "faults" there are in the evidence presented;
- The level of review undertaken of the evidence
- For tool-derived evidence: tool qualification and assurance;
- Experience and competence of the personnel.

However there are many questions that arise about these properties of evidence and in particular how we measure them either qualitatively or quantitatively. Currently, a large amount of the consideration of relevance occurs on a case-by-case basis using subjective judgement. Conceptual frameworks for assessing relevance, trustworthiness and independence methodically and the presentation of this information in the safety case would improve our understanding of the role the evidence plays. This problem is not specific to goal-based standards and is equally applicable to evidence generated to satisfy prescriptive standards.

3.4 Scale

The size and complexity of systems and their safety cases is growing continuously. A safety argument, set of hazards, or set of safety evidence that is too small (insufficient in breadth or depth of topics covered) will cause a reduction in confidence. However, if these items are too large, confidence can also be reduced. A safety argument that is not of a manageable size or grows to a proportion such that it becomes impossible to understand, while being factually correct will reduce confidence. It must be possible for the assessor to reach a final decision with respect to the safety case and this is only possible if the size of the safety argument is within their capacity. Modularisation allows for large safety arguments to be separated into manageable chunks (as opposed to a monolithic safety case), however this itself suffers from problems relating to confidence as discussed later in this section. As with safety arguments, if the set of hazards or set of evidence grows to a proportion that they become intractable it becomes more difficult to assure safety with confidence. A safety case must be succinct in its presentation of information, but questions remain about what is too little and what is too much. How do we determine the capacity for people to understand large and

complex safety cases? How do we determine sufficient information has been presented? We are not implying that a safety case should have a maximum or minimum page length, but how confident can we be in the upper or lower boundaries of the scale of a safety case? Currently these issues are resolved through experience and judgement. However, it would be beneficial for both developers and assessors if there was greater guidance about the sufficiency or level of granularity required in safety cases.

Modular safety cases, e.g. where separate but interrelated safety cases are produced for systems within a system of systems or are produced by different members of large development consortiums, require the definition of safety case interfaces. Safety cases have moved away from being monolithic documents specific to an independent system developed by a single organisation. While confidence has grown in developing single one off safety cases, many of the issues with current safety case production exist at the boundary of the safety case and its interactions with other safety cases. There is a need for understanding of the dependencies between systems and organisations in order that these dependencies can be reflected within inter-safety case contracts. As the complexity of safety case interrelationships grows, these dependencies will demand a significant proportion of the effort involved in safety case production. How these dependencies can be predicted and modelled without major restrictions on the functionality of the systems is an area of continuous research.

4 Conclusions

This paper has presented some of key issues which effect confidence in the acceptance of safety cases. The aim of this paper has been to promote discussion about what can reduce the assurance in a safety case and how we can consider these issues more explicitly. Confidence can be affected by both information uncertainty and inadequate understanding. Information uncertainty can be addressed by developers and assessors through greater consideration of the information that should be included within the safety case in order to increase confidence. Addressing inadequate understanding will require further research into the topics identified. By focussing attention on these issues, we hope that means of addressing them can start to be identified. We make no claim that this set of issues is complete, so we also hope that this discussion will provide a basis for identifying other issues that effect confidence in a safety case.

References

Bishop P, Bloomfield R, Emmet L, Jones C, Froome P (1998). Adelard Safety Case Development Manual, Adelard, 1998

CAA (1999). Regulatory Objective for Software Safety Assurance in Air Traffic Service Equipment SW01. Civil Aviation Authority, UK, 1999

Lipshitz R & Strauss O (1997). Coping with Uncertainty: A Naturalistic Decision-Making Analysis. Organizational Behaviour and Human Decision Processes 1997; 69-2:149-163

Greenwell W S, Strunk E A, Knight J C (2004). Failure Analysis and the Safety-Case Lifecycle. Proceedings of the 7th IFIP Working Conference on Human Error, Safety and System Development (HESSD), Toulouse, France, August 2004. Ed. C W Johnson and P Palanque. Boston: Kluwer, 2004

Kelly T P (1998). Arguing Safety – A Systematic Approach to Safety Case Management, DPhil Thesis YCST99-05, Department of Computer Science, University of York, UK, 1998

McDermid J, Kelly T, Weaver R (2005). Goal-Based Safety Standards: Opportunities and Challenges. Proceedings of the 23rd International System Safety Conference, San Diego, USA, August 2005

Mir N H (2005). Web Based Integrated Safety Management Groupware. MSc Thesis, Department of Computer Science, University of York, York, UK, 2005

MoD (2004a). Defence Standard 00-56, Safety Management Requirements for Defence Systems - Part 1 Requirements, Issue 3, UK Ministry of Defence, 2004

MoD (2004b). Defence Standard 00-56, Safety Management Requirements for Defence Systems - Part 2 Guidance on Establishing a Means of Complying with Part 1, Issue 3, UK Ministry of Defence, 2004

MODISA (2004). Guidance for Integrated Project Teams for Use in Contracting for Independent Safety Auditor (ISA) Services. STG/181/1/9/1, Safety Management Offices Group, Ministry of Defence, 2004

Thunnissen D P (2005). Propagating and Mitigating Uncertainty in the Design of Complex Multidisciplinary Systems. PhD Thesis, California Institute of Technology, Pasedena, California, 2005

Weaver R A (2003). The Safety of Software – Constructing and Assuring Arguments. PhD Thesis, Green Report YCST 2004/01, Department of Computer Science University of York, York UK, 2003

AUTHOR INDEX