

CONFERENCES IN RESEARCH AND PRACTICE IN
INFORMATION TECHNOLOGY

VOLUME 133

AUSTRALIAN SYSTEM SAFETY
CONFERENCE 2011



AUSTRALIAN
COMPUTER
SOCIETY



AUSTRALIAN SYSTEM SAFETY CONFERENCE 2011

Proceedings of the
Australian System Safety Conference (ASSC 2011),
Melbourne, Australia, 25th-27th May 2011

Tony Cant, Ed.

Volume 133 in the Conferences in Research and Practice in Information Technology Series.
Published by the Australian Computer Society Inc.



Published in association with the ACM Digital Library.

Australian System Safety Conference 2011. Proceedings of the Australian System Safety Conference (ASSC 2011), Melbourne, Australia, 25th-27th May 2011

Conferences in Research and Practice in Information Technology, Volume 133.

Copyright © 2012, Australian Computer Society. Reproduction for academic, not-for-profit purposes permitted provided the copyright text at the foot of the first page of each paper is included.

Editors:

Tony Cant
Defence Science and Technology Organisation
C3I Division, PO Box 1500, Edinburgh SA 5111, Australia
E-mail: Tony.Cant@dsto.defence.gov.au

Series Editors:

Vladimir Estivill-Castro, Griffith University, Queensland
Simeon Simoff, University of Western Sydney, NSW
crpit@infoeng.flinders.edu.au

Publisher: Australian Computer Society Inc.
PO Box Q534, QVB Post Office
Sydney 1230
New South Wales
Australia.

Conferences in Research and Practice in Information Technology, Volume 133
ISSN 1445-1336
ISBN 978-1-921770-13-5

Printed April 2012 by Griffith University, CD proceedings.

The *Conferences in Research and Practice in Information Technology* series aims to disseminate the results of peer-reviewed research in all areas of Information Technology. Further details can be found at <http://crpit.com/>.

Table of Contents

Proceedings of the Australian System Safety Conference (ASSC 2011), Melbourne, Australia, 25th-27th May 2011

Preface	vii
Programme Committee	ix

Research Papers

Maritime Safety Case in a Box	3
<i>Murray Bailes</i>	
Moving Towards Goal-Based Safety Management	19
<i>Holger Becht</i>	
Developing a methodology for the use of COTS operating systems with safety-related software	27
<i>Simon Connolly and Holger Becht</i>	
Urgent Operational Requirements: Impact on the Safety Case	37
<i>Tony Cant and Brendan Mahony</i>	
Establishing Safety Case Strategies for Mission Planning or Situational Awareness Systems	49
<i>Brett J. Martin and Derek W. Reinhardt</i>	
Managing Systems and Software Safety Risks in Emerging Technologies – A Surface Transport Perspective	67
<i>Len Neist</i>	
Safety Assurance: Fact or Fiction?	71
<i>Carl Sandom</i>	
System safety in hybrid and electric vehicles	79
<i>David D. Ward</i>	
The Language of System Safety Engineering: Loose Language Surrounding ALARP	85
<i>Tracy A. White</i>	
Author Index	95

Preface

The *Australian System Safety Conference 2011* was held at the Rendezvous Hotel, Melbourne, on 25-27 May, 2011. The conference, jointly sponsored by the Australian Safety Critical Systems Association (aSCSa) and the Australian Chapter of the System Safety Society, had the theme: “Managing Systems and Software Safety Risks in Emerging Technologies” and was attended by more than 100 participants. The conference program was greatly enhanced by four keynote speakers:

- Dr Jeffrey J. Joyce, President (Critical Systems Labs Inc, Canada)
- Len Neist, NSW Independent Transport Safety Regulator (Australia)
- Dr David Ward, General Manager for Functional Safety (MIRA Ltd, UK)
- Dr Carl Sandom, Director (iSys Integrity Ltd, UK)

Prior to the conference, Carl Sandom presented a tutorial entitled “Human Factors and Safety Engineering”. Full program details are available from www.asssc.org/conference. More information on the aSCSa can be found at www.safety-club.org.au.

The Organising Committee is very grateful to the authors for the trouble they have taken in preparing their work to be included in these conference proceedings. The papers were peer-reviewed for relevance and quality by the Program Committee. Note, however, that the views expressed in the papers are the authors’ own, and in no way represent the views of the editor, the Australian Safety Critical Systems Association, the System Safety Society, or the Australian Computer Society. The fact that the papers have been accepted for publication should not be interpreted as an endorsement of the views or methods they describe, and no responsibility or liability is accepted for the contents of the articles or their use.

The committee also wishes to thank the conference sponsors for their support: the Australian Computer Society; Ansaldo STS; Invensys Rail; University of Queensland; RGB Assurance; Nova Systems; Airservices Australia; and the Defence Materiel Organisation in the Australian Government Department of Defence. These organisations have all helped to make the conference a success.

I wish to thank all those involved in organising the conference (listed below). In particular, I would like to acknowledge the commitment and drive of my colleagues B.J. Martin, Holger Becht and Derek Reinhardt, who worked hard to make sure that the conference was a success.

We are also grateful to Ksenija Catic of the Melbourne Branch of the Australian Computer Society for her assistance. Finally, our thanks to the Computer Systems and Software Engineering Board of the ACS for ongoing support.

Tony Cant, Defence Science and Technology Organisation

Programme Committee

Programme Chairs

- Brett J. Martin (Chair)
- Holger Becht (Vice Chair)
- Onn Eng Lin (Advisor)
- Glenn Larsen (Publicity and Sponsorship Chair)
- Derek Reinhardt (Registration)
- Kevin Anderson (Facilities and Operations)
- Tariq Mahmood (Facilities and Operations)

Programme Committee

- Tony Cant (Chair)
- Holger Becht (Vice Chair)
- Simon Connelly (Member)
- Derek Reinhardt (Member)
- George Nikandros (Member)
- Clive Boughton (Member)
- Tariq Mahmoud (Member)
- Tim Kelly (Member)
- Paul Caseley (Member)
- Rob Weaver (Member)
- Brendan Mahony (Member)

Australian Safety-Critical Systems Association Committee

- Clive Boughton (Chair)
- George Nikandros (Immediate Past Chair)
- Kevin Anderson (Secretary)
- Chris Edwards (Treasurer)
- Brett J. Martin (Member)
- Tony Cant (Member)
- Tariq Mahmood (Member)
- Rob Weaver (Member)
- Derek Reinhardt (Member)

RESEARCH PAPERS

Maritime Safety Case in a Box

Murray Bailes

murraybailes@iinet.net.au

Abstract

The Maritime Safety Case in a Box is the result of applying the principles and techniques of Model Based Systems Engineering to Safety Engineering to establish a framework of models that support the definition of a generic safety case for a maritime combat system.

These models are currently constructed in CORE™, a MBSE application built by Vitech Corporation but could be ported to other modelling tools or a set of processes as required.

These models span:

- a) A set of DoDAF compliant Operational and System domain models of the combat system for a naval maritime platform;
- b) A set of Program Domain models that describe the Program Activities and their products to define a Safety Case that is compliant with the relevant statutory and regulatory requirements;
- c) A set of generic hazard, cause, control and accident assessments for physical hazards such as hazardous materials, slip, trip or fall, electricity, confined space, etc
- d) A set of generic functional hazard assessments based on analysis of the Operational and System Models described above.

The intersection of these models provides a solid framework to maximise the effectiveness of the safety engineering process while reducing the cost by providing a set of partially completed hazard assessments or patterns for the system under consideration that can be tailored or extended for each class of ship.¹

Keywords: Maritime, Safety Case, DoDAF, Model Based Systems Engineering, Safety Patterns.

1 Introduction

Safety assessment is one of the fundamental human activities that we all inherently perform on a continuous basis throughout our lives. In basic terms, we assess the safety risk before we do anything.

In the days of old we were responsible for our own safety. Our safety assessments were performed using our understanding of the environment based on knowledge passed down through the generations, prior experience and intuition.

In more recent times however the rapidly expanding human knowledge base has led to the introduction of many types of unfamiliar technology. Our lives and our operating environments are now full of increasingly complex processes in which the individual may only play a small role. These increasing complex operating processes and use of many unfamiliar man-made materials and new construction methods expose us to new, at times personally undetectable hazards, reducing the ability of the individual to adequately assess safety risk within their environment.

As a consequence of this there is a shift in the focus of the responsibility for modern safety assessment from the individuals who interact with our systems, to the owner of the systems or capabilities in which the individual operates. This is particularly true within the workplace.

The safety assessment process is now beyond our intuition or past experience and requires the knowledge of a range of experts with diverse specialities. Safety management requires the coordination of these specialised skills within the safety process to ensure that:

- a) The required information is available when needed to support each safety program activity.
- b) The outputs of each specialist activity are captured and integrated into a single body of Objective Quality Evidence (OQE).

For complex systems it is now the Safety Manager and the Safety Specialist that often perform the safety assessment on behalf of the owner or provider of a capability in accordance with the applicable statutory and regulatory requirements. These requirements define the safety assessment process and the safety program outputs. Due to the scale of the overall safety program for complex systems, hazard analysis is typically performed by a range of subject matter experts in a wide range of engineering specialities.

Differences in the way individuals approach this hazard assessment process introduce variations in the way each hazard, cause and accident are defined. I.e. One person's hazard is another person's cause. This results in a lack of consistency in the definition of like type hazards that adds considerable complexity to the hazard management and overall risk assessment

The Maritime Safety Case In a Box (SCIB) has evolved from applying model based systems engineering to the safety assessment process for a maritime combat system. By providing a set of safety assessment patterns or templates that are based around analysis of DoDAF compliant models it provides generic safety assessments that can be specialised for a particular maritime combat system.

Generic functional hazard assessments performed against the maritime combat system operational domain model

provide a set of functional hazard analysis templates that can be specialised for each class and instance of that capability type.

Generic physical hazard templates guide the form and content of the physical hazard analysis.

By using the hazard templates to guide the analysis a consistent form can be achieved providing uniformity in the management of like type hazards simplifying the overall risk assessment.

These templates provide a preliminary hazard assessment performed against a generic representation of the maritime combatant that can be taken into consideration in the design, guiding the subsequent safety assessment for each specific instance of a maritime combat system. This approach can result in a consistent treatment of same type hazards both within a single platform, a class of ships or even the entire RAN fleet.

2 Safety Engineering Verses Systems Engineering

Traditional Systems Engineering focuses on identifying and defining the user's needs as an input into developing an implementation that satisfies those needs. Through the design synthesis process the Systems Engineer creates a set of requirements and other design documentation that **positively define the required capability and implementation**. These positively defined requirements define an input into the system implementation, verification and eventual acceptance.

Conversely, Safety Engineering focuses on identifying and managing hazards, causative factors, accidents, preventative controls and mitigating controls to **limit the possibility of negative outcomes** for the users, stakeholders, the system itself or the environment to As Low As Reasonably Practicable (ALARP).

This fundamental difference between the Systems Engineering focus on positive validation of a **positively defined capability** as opposed to the safety focus of verifying the **limitation of negative outcomes** can create difficulties that are often not well understood or managed within project processes and by engineering management.

Safety risk cannot be closed early in the implementation process in the same way as program risk or technical risk. Managing safety risk requires the development of engineering or procedural based controls that by their very nature may remain open at least until completion of the system implementation and verification and at times throughout the entire lifecycle of the capability. Changes in the user's needs, system upgrade programs or changes in the regulatory and legislative environment necessitate reassessment of the safety baseline throughout the system lifecycle.

Furthermore, as elimination of negative outcomes can neither be fully achieved nor positively validated, demonstrating that safety risk has been reduced to ALARP the safety engineer must follow a set of best practice processes and peer review. That is not to say that best practice processes and peer review alone are adequate in achieving safety however they are very important in determining that the risk treatments, as determined by a panel of suitably qualified practitioners,

have been systematically incorporated into the system design until the remaining treatment cost is grossly disproportionate to the safety benefit gained. In more familiar terms, to determine that the safety risk ALARP.

The importance of integrating Safety within the Systems Engineering process to ensure that Safety is given its due consideration within the design and implementation cannot be overstated. It is, of course, not possible to separate Safety Engineering from mainstream Systems Engineering. Any attempt to do so fails to recognise the fundamental relationships between safety and other Systems Engineering disciplines. Safety has become its own system design speciality along with fitness for purpose, usability, reliability, maintainability, etc and is supported by other systems engineering practices such as configuration management, requirements management and test and evaluation.

Within the Safety Engineering process there is a tendency for individuals to assess similar hazards in different ways making an overall safety assessment difficult to determine. It is understandable that different people will perceive things differently, so in order to reach a consistent representation we need a standardised way of identifying and controlling like hazard types.

3 The Regulatory Environment

The OH&S Act 1991 and the OH&S Regulations 1994 [Commonwealth] require employers to establish both a set of Health & Safety Management Arrangements (HSMA) and a method for systematically identifying and controlling hazards. These two processes form part of the OH&S Management System (OHSMS) as described in AS/NZS 4801 Occupational health and safety management systems - Specification with guidance for use.

Proving the safety of a new or existing capability must describe how this capability will operate while meeting legislative requirements, and demonstrate from first principles that the safety and health hazards identified are indeed being effectively managed to ALARP.

Establishing the safety argument from the operational perspective satisfies this 'first principles' objective.

4 The Maritime Safety Case in a Box

The Maritime SCIB is a methodology that has evolved from applying Model Based Systems Engineering (MBSE) to the disciplines of Safety Engineering and Safety Management. It consists of a set of integrated Department of Defence Architecture Framework (DoDAF) compliant Operational, System, and Programmatic models for a maritime combat system safety program that defines the safety program activities while providing a roadmap for performing a generic safety assessment. This assessment is based on using operational patterns within the models to form the basis of the safety argument. Having said that, the Maritime SCIB is not intended to be prescriptive nor is it intended to provide complete insight into all aspects of the safety assessment and Safety Case development. It does not provide a ready made solution to fit every combat system instance, but rather is intended to provide a framework

that may be adopted and adapted for a particular platform as appropriate and necessary.

In MBSE speak; the Maritime SCIB is a “Safety” Integrated Decision Database (IDD) that supports safety engineering by defining the activities, processes and products needed to satisfy the relevant Commonwealth legislative and regulatory requirements.

It provides a framework to support both functional and physical hazard identification and assessment and the identification, definition, specification and verification of safety controls.

Engineering and procedural based controls are defined, implemented and verified through requirements that are traced and managed within the Safety IDD.

The operational and system models support the hazard templates to enable the identification and management of functional safety risk throughout the system lifecycle. The Safety IDD allows for the flagging of identified safety critical functions and components within a system, recording the reasoning behind decisions and providing traceability to the hazard treatments.

5 The Use of Frameworks and Patterns in Safety Engineering.

The use of safety engineering frameworks and patterns is starting to become commonplace in more mature safety engineering domains such as aviation.

The Eurocontrol Safety Assessment Methodology¹ (SAM) describes “a generic process for the safety assessment of Air Navigation Systems”. The methodology is supported by a number of publications including:

- European Organisation for the Safety of Air Navigation - Safety Case Development Manual² that “provides guidance on the development of Safety Cases as a means of structuring and documenting the demonstration of the safety of a an ATM service or new / modified system”
- European Organisation for the Safety of Air Navigation - Safety Assessment Made Easier Safety Principles and an Introduction to Safety Assessment³ that “is intended to describe the broad framework on to which the SAM-defined processes, and the associated safety, human-factors and system-engineering methods, tools and techniques, are mapped in order to explain their purpose and interrelationships.”

This document suggests “that success and failure approaches should be used together in the developing the Safety Requirements for a new ATM system (or change to an existing ATM system”) and “show that satisfaction of those Safety Requirements would result in an acceptable level of safety.”

Where the

- “the **success** approach – which seeks to assess the achieved level of safety when the ATM system in question is working as intended – ie in the absence of failure” and
- “the **failure** approach – which seeks to assess the effect, on the achieved level of safety, in the

event of failure (ie deviation from what is intended) internal to the ATM system.”

This document” explains, in straight forward terms, why the scope of ATM safety assessment needs to be broadened in order to encompass the success approach, including what in the past may have been thought of as “operational” (rather than safety) issues.”

Tim Kelly and John McDermid, from the University of York, assert that the use of patterns within safety case is not new. In a paper entitled Safety Case Patterns – Reusing successful arguments⁴ they state that:

“reuse of safety case arguments is already commonplace – i.e. using ‘largely the same’ arguments of safety as used on previous projects.”

However they point out that this:

“form of reuse often occurs through ‘Cut and Paste’ of the textual safety case documents between projects. However, there are a number of problems with such an approach:

- It can be difficult to identify opportunities for reuse (i.e. take full advantage of successful arguments).
- Reuse occurs in an ad-hoc fashion – in a way that cannot be predicted or depended upon for project management.
- Inappropriate reuse occurs. The context of a safety argument may not be exactly the same from one instance to another. Critical assumptions may be challenged.
- Lack of traceability. There is difficulty in knowing where arguments have been repeated. Problems can arise if ‘faulty’ arguments are propagated.
- Lack of consistency / process maturity – different (sometimes only subtly different) argument approaches may be unnecessarily used where reuse would improve consistency of approach and better support claims of a mature process.
- Loss of knowledge. There is no mechanism or medium for recording the essential ‘best practice’ of safety case development / safety argument construction.”

The SCIB offers a solution to achieving the objective of reuse of a safety argument while addressing these identified objectives and issues.

The operational models provide an operational context in which the safety assessment can be guided by the reuse of the patterns that support both the “Success approach” and the “Failure approach” from an operational perspective.

The safety IDD provides the mechanisms to avoid loss of knowledge and provide traceability and consistency across the safety argument.

By providing the foundations to identify and assess safety issues associated with a capability the SCIB provides a safety case reference architecture that defines a standard for either the development of a new Safety Case

or provides a benchmark for the review and update of existing Safety Cases.

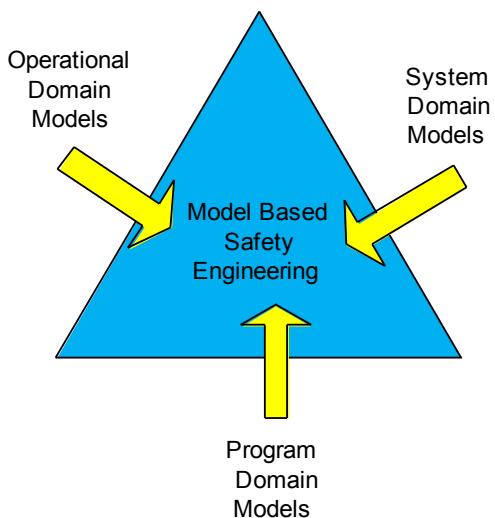


Figure 1: Model Based Systems Safety Domain Description

Figure 1: Model Based Systems Safety Domain Description provides a simplified view of how the three complementary domains defined within the Maritime

SCIB combine to provide an integrated view into System Safety Engineering.

By providing a top down approach to both the safety management process and the relationship between the operational and system domains to support the system safety assessment, the Maritime SCIB can minimise rework, realising efficiencies that reduce the overall cost of achieving safety while delivering improved OH&S by defining a standardised method for the assessment and management of same type hazards.

Figure 2: Royal Australian Navy Safety Program Overview describes how the Maritime SCIB supports the foundations of safety engineering within the Royal Australian Navy organisational and regulatory framework. It identifies the primary stakeholders, their program activities and the products they produce.

The program activity model functionally decomposes the activities and the products of the various safety stakeholders, such as the system safety team members, including their management, the regulators, platform and combat system designers, builders and testers subcontractors and COTS suppliers and ILS providers into a framework that maximises the integration of their disparate activities into a single consolidated effort.

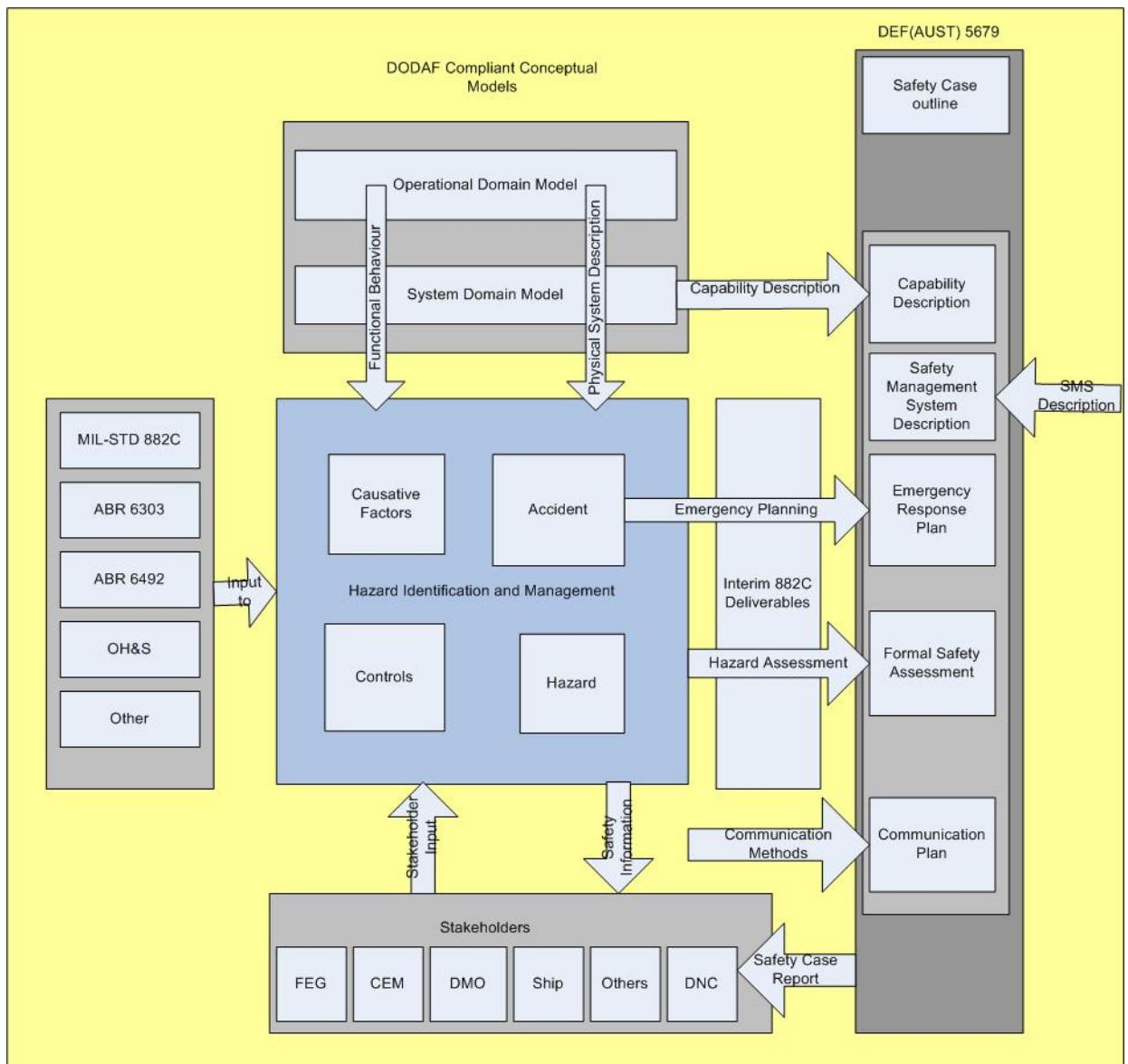


Figure 2: Royal Australian Navy Safety Program Overview

The Hazard Identification and Management module defines the implementation of the OH&S Program. It contains the underlying hazard management engine as well as a set of program activities that define a Safety Management System that produces the safety management plans, interim safety deliverables and the final Safety Case report.

Figures 3, 4, 5, 6, 7, 8 and 9 below present extracts from the Maritime SCIB Program Activity model to provide more detail of the Hazard Identification and Management block from this diagram. Due to size and space constraints of this paper only a selected set of the safety process diagrams are included however the

selection is intended to demonstrate how the Maritime SCIB is a well defined, documented, measurable and auditable safety management process that satisfies many of the objectives of AS/NZS 4801:2001 Occupational Health and Safety Management Systems and OHSHS 18001:2007 Occupational Health and Safety Management Systems Requirements.

The complete set of these safety process diagrams provide the content for the development of both the safety management plans interim safety deliverables and the final Safety Case report's Safety Management System Description.

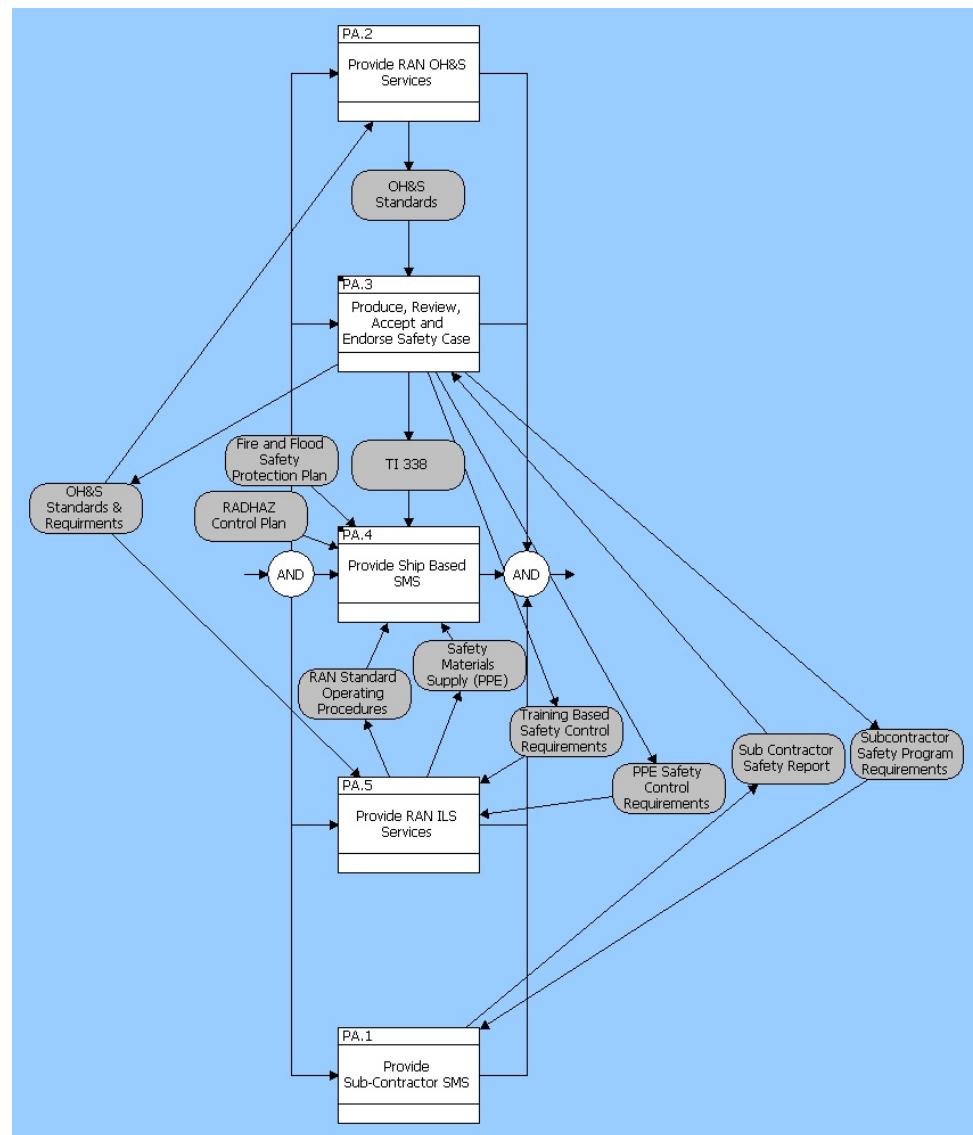


Figure 3 Provide Capability Development Safety Management System Context

Figure 3 Provide Capability Development Safety Management System Context describes the top level of the Program Activity Model. It places the activity to Produce, Review, Accept and Endorse Safety Case in the

context of the other RAN organisations and subcontractor (COTS suppliers or construction agency) SMS activities.

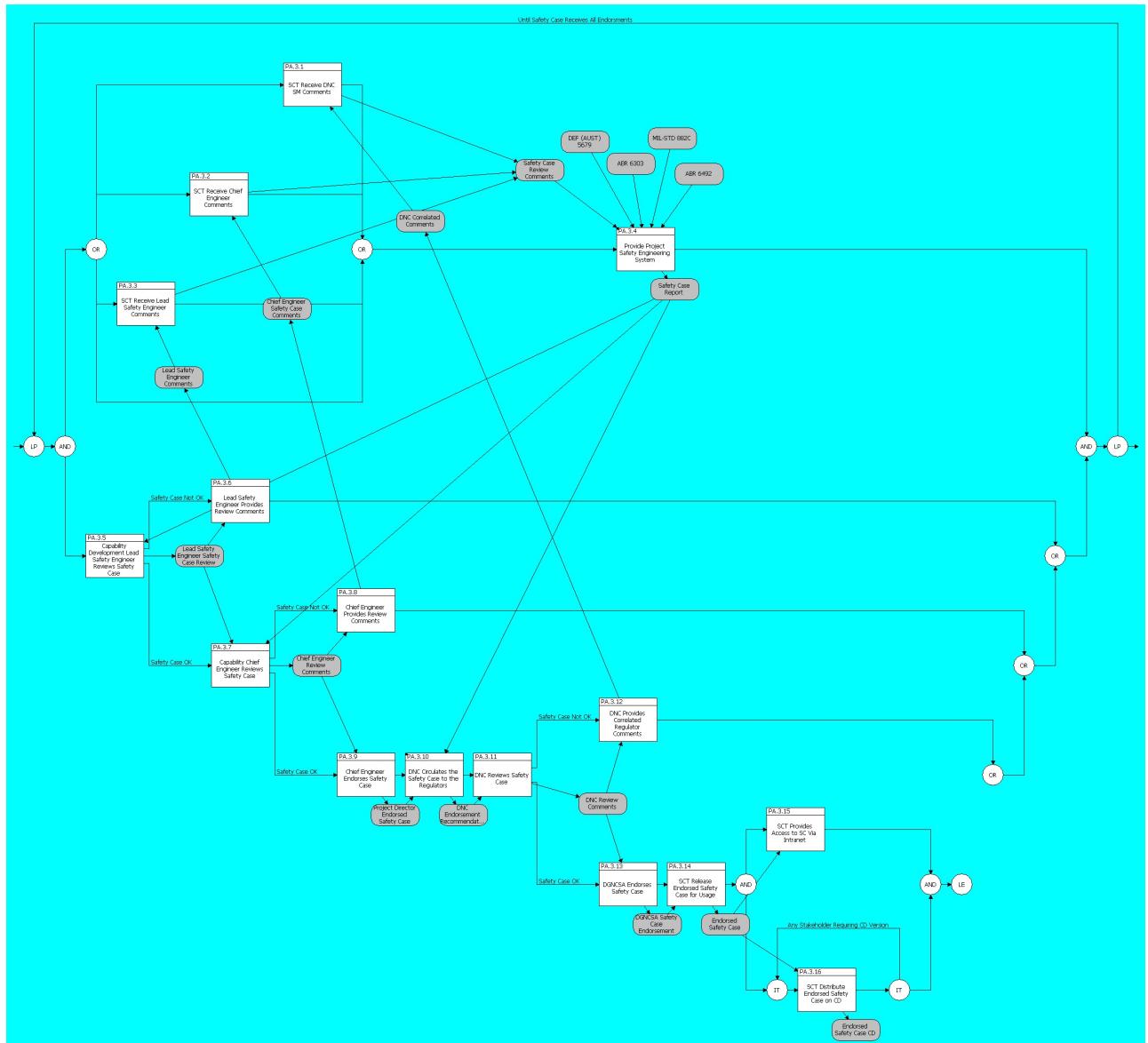


Figure 4: Produce, Review, Assess and Endorse Safety Case

Figure 4: Produce, Review, Accept and Endorse

Figure 4: Produce, Review, Accept and Endorse Safety Case Figure 4: describes the Safety Case production and its iteration through several levels of review and comment, until all review comments have been resolved to the satisfaction of the reviewers, at which time it receives DGNCSA endorsement

At a more abstract level this diagram can be considered as describing two basic top level activities represented by the two branches of the 'and' gate. These are:

1. To produce the Safety Case
 2. To review, accept and endorse the Safety Case

To achieve this, the various actors involved continuously iterate through the process of performing safety engineering activities and producing Safety Case documentation (see the lower level activity decomposition diagrams for more details of that process). The Safety Case is then submitted for comment through various levels of review until the Safety Case Team (SCT) has received and resolved all identified issues and

the Safety Case is endorsed as having reduced the safety risk to ALARP.

For simplicity, only the final Safety Case report is shown on this diagram however the process described is also applicable to the production and review of interim safety deliverables produced throughout the safety program activities. For interim deliveries, such as the various MIL_STD 882c defined hazard analysis reports, typically the lead safety engineer and chief engineer will perform the review function while selected interim documentation may be provided to the regulators for review and comment.

5.1 Safety Case Production

The SCT are required to produce a Safety Case report that satisfies the defined standards. The SCT typically consists of a project level safety team, a set of subsystem level safety teams that are supported by the component manufacturers and construction agency safety teams. It is their role to ensure that everything is done to identify hazards and reduce the safety risk in design and construction. This is a 'hands on' role that endeavours to

ensure that safety is adequately incorporated into the engineering process. They engage in continuous safety engineering and review function throughout the project. Their 'in process' progress is reported through the interim deliverables defined by MIL-STD 882C while the final Safety Case Report provides the final Safety Argument and supporting documentation.

Within a project, the internal safety management system interim documentation review is usually performed first at the peer level, then by the lead safety engineer and then finally by the chief design engineer.

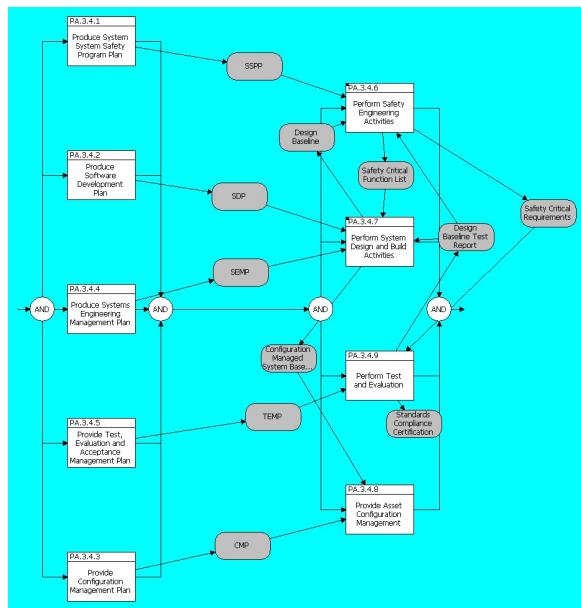


Figure 5: Provide Project Safety Management System

Figure 5: Provide Project Safety Management System details the definition of the set of safety related management plans for their subsequent use in guiding the performance of the project safety related activities. While system design and build, test and evaluation and configuration management are not exclusively related to safety, their contribution in providing a safe system implementation is of particular importance.

5.2 Safety Case Review and Endorsement

Although safety management plans, status reports and interim safety deliverables may be provided to the regulators for review and comment throughout a project lifecycle it is the set of final formal Safety Case deliverables that are of primary interest to the Director of Naval Certification (DNC) and its sub-agencies. It is on their review and advice that the Director General Navy Certification and Safety (DGNCSA) relies for the decision on endorsement before a capability can be put into service.

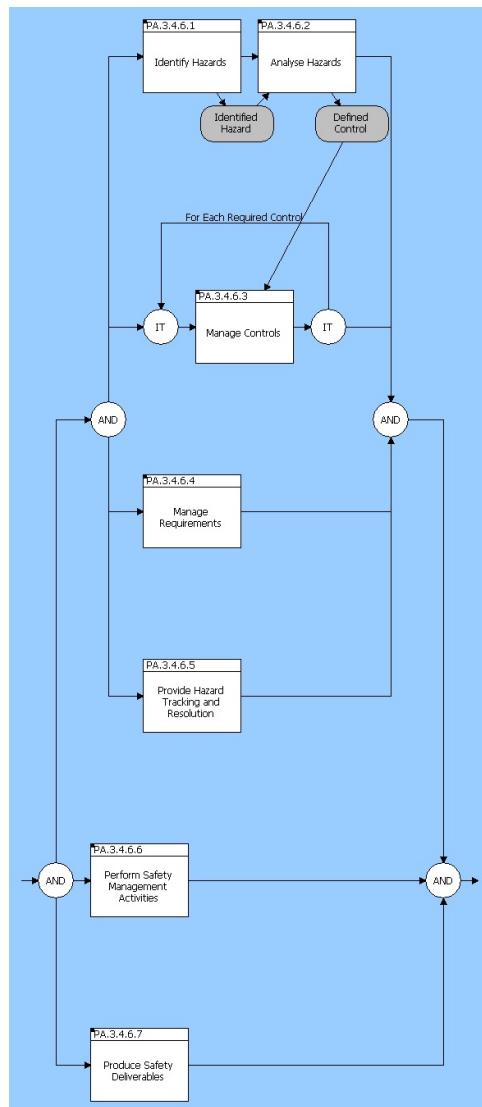


Figure 6: Perform Safety Engineering Activities

Figure 6: Perform Safety Engineering Activities describes the identification and analysis of hazards, the management of the defined controls and the tracking of all of these artefacts to resolution. It also includes the production of the safety case deliverables and the management of the safety management program. Similar to **Figure 5: Provide Project Safety Management System** managing requirements is not exclusively a "safety" activity but once again its contribution to achieving safety in the design process is very important.

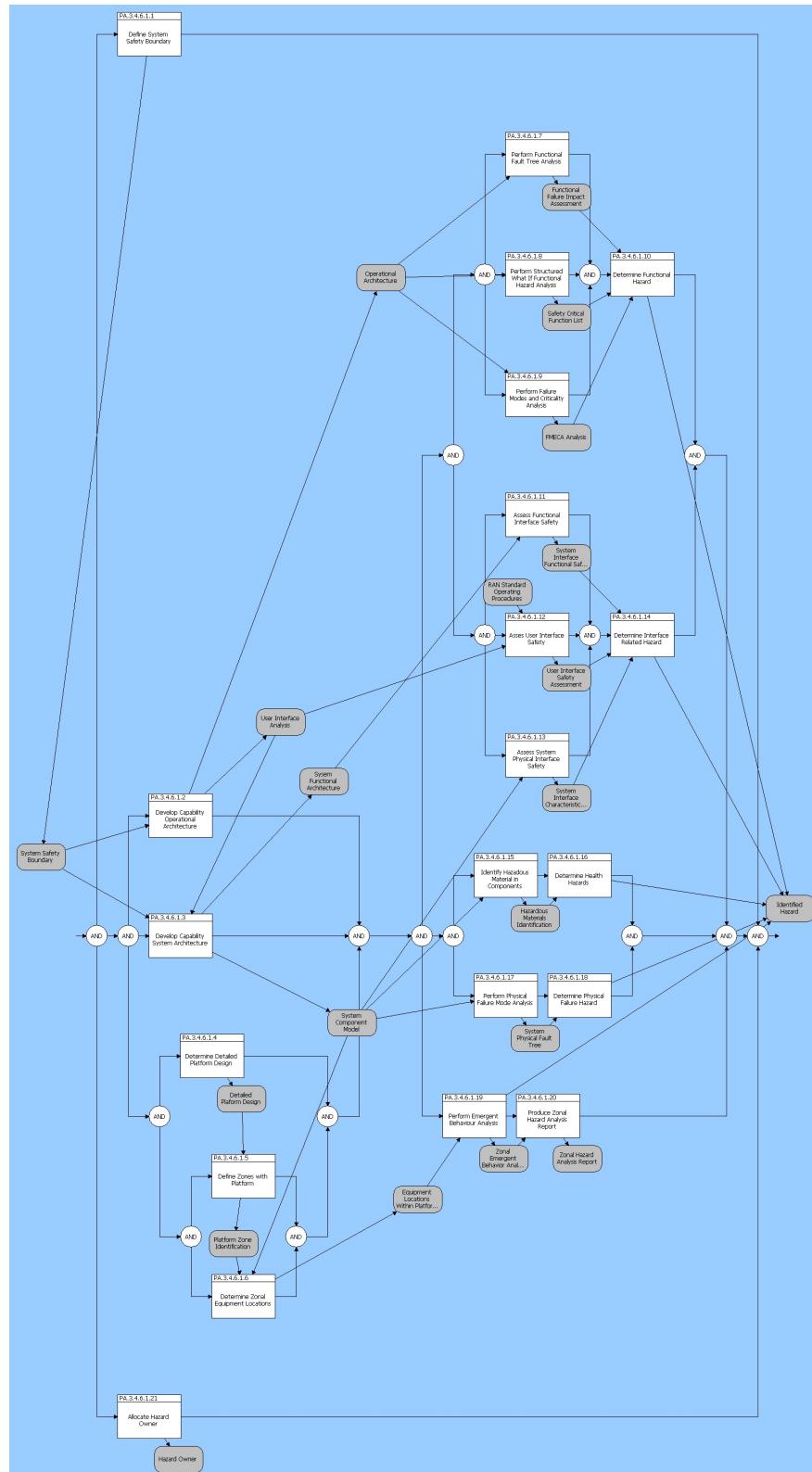


Figure 7 Identify Hazards

Figure 7 Identify Hazards describes the use of the various operational and system domain model elements as inputs into the hazard identification activity. The set of hazard identification techniques within this diagram is not intended to be exhaustive but rather to provide an overview of how the various models of the Maritime SCIB can be used to Identify Hazards. The concepts

contained within the Eurocontrol Safety Assessment Made Easy – Safety Principles and an Introduction to Safety³ are well supported by this approach. The SCIB operational model provides the foundation for the success approach analysis by providing a description of the operation of the combat system while also providing a solid basis of considering the implications of particular

system failures at the operational level for the failure approach.

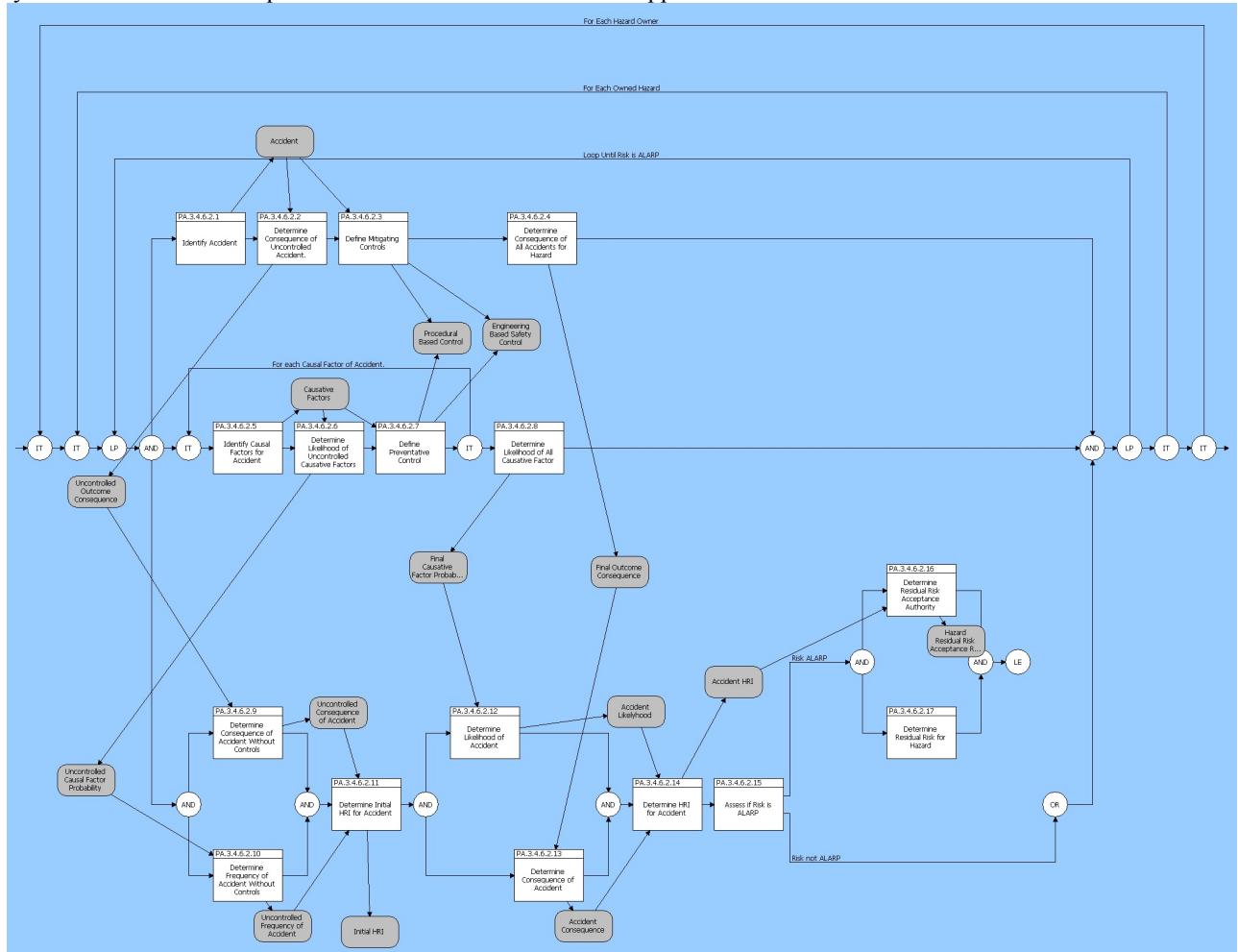


Figure 8 Analyse Hazards

Figure 8 Analyse Hazards describes, that for each hazard, the accidents and their causal factors must be identified then the initial consequence of each accident and the likelihood of each causal factor. From these the Initial Hazard Risk Index is calculated. For each accident the mitigative controls are defined to reduce the accident impact while for causal factors the preventative controls are defined to reduce the likelihood of its occurrence.

Following the identification of the planned hazard treatment the final likelihood and consequence are determined on which the Final HRI is based. Then the determination on whether the hazard risk has been managed to ALARP is made, and based on the final HRI, the level of authorisation required to accept the residual risk is assigned.

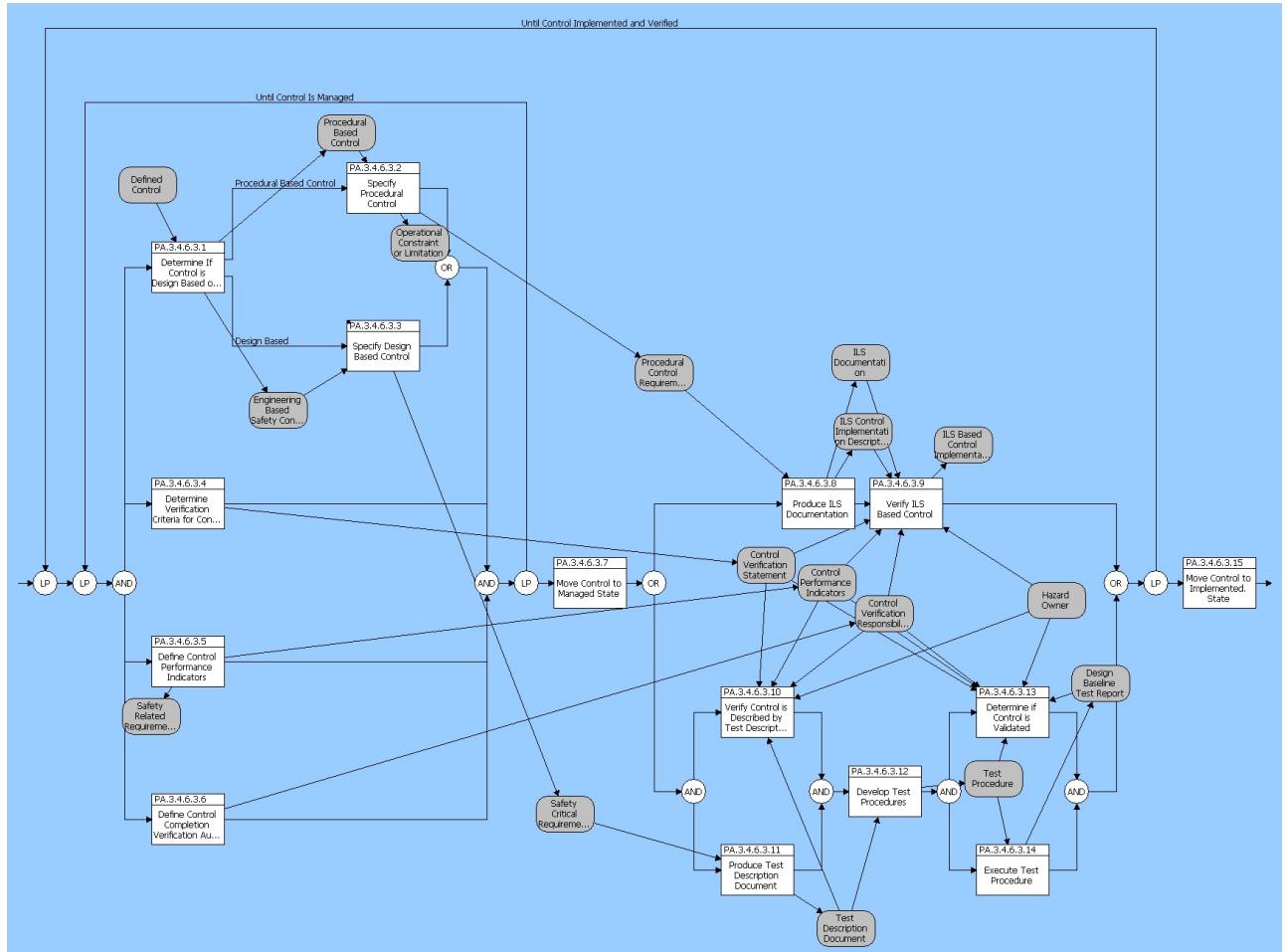


Figure 9: Manage Controls

Figure 9: Manage Control manages the controls that are identified during the Analyse Hazard activity. Controls may be either engineering based controls or procedural based controls. Engineering based controls are specified through requirements that are implemented and verified through the project systems engineering build and test and evaluation activities. Procedural based controls are implemented throughout the ILS program. When defining each control it is required to also define the verification criteria, performance indicators and a

control completion acceptance authority. Only after each of those control parameters has been defined can the control be moved to the managed state.

While the safety related requirements are verified through the normal project test and evaluation activities each control must also be validated by the defined control completion acceptance authority using the defined verification criteria. The ongoing performance of the control is assessed against the defined performance indicators.

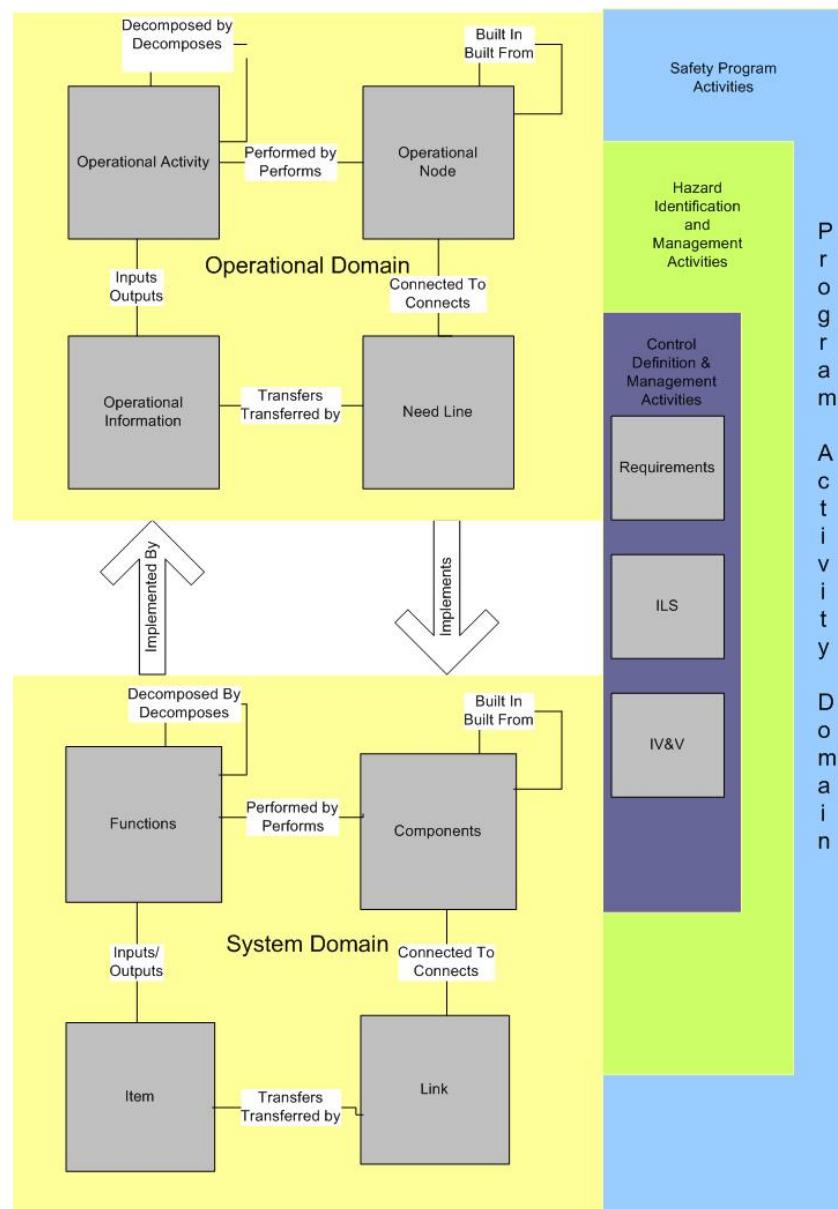


Figure 7: Model Based Operational Safety Risk Assessment

Figure 7 Model Based Operational Safety Risk Assessment describes the operational, system and program domain model schema. For a description of the information classes within the operational and system domains please refer to the Architecture Definition Guide DoDAF 2.0⁵ produced by the Vitech Corporation.

These operational and system models provide the basis of an operationally based hazard and risk assessment. Hazards identified through analysis of the operational activities and the operational information flow prior to the implementation. This allows for the identification of safety critical activities to be taken into consideration within the design and implementation. The identification and allocation of safety critical activities to the operational nodes (roles) provides an input into the development of the standard operating procedures for these roles.

The system domain model for a specific capability defines the components, functions, links and (data) items that describe the implementation. By establishing the

“Implemented by” linkages between the safeties critical operational domain elements and those within the system domain the safety assessment can accommodate the “success argument” referred to by the Eurocontrol methodology while the “failure argument” can incorporate a failure mode or change impact assessment that crosses the operational and systems domain boundaries.

The Maritime SCIB currently contains a full operational domain model of a maritime combat system. This model has its origins in work performed on the Collins Submarine RCS Project. It defines the operational activities and its information flow, operational nodes (Roles) and needlines (a need to communicate between Roles) for a maritime combat system.

This model covers the operational activity and operational node definitions for navigation and managing the tactical environment. It is planned to extend this model to include other non combat system activities

associated with the platform to support a full safety engineering assessment for a capability.

6 Using the Operational Model for Functional Safety Assessment

Figure 7 Identify Hazards describes how the operational and system models provide a direct input into identifying functional hazards using techniques such as structured what if analysis and failure mode analysis.

Analysis of the operational activities and their information flow identifies the relative safety criticality of some activities and their information flow over others. Preliminary Functional Hazard Assessment of the combat system operation is performed against the operational model by identifying the dependencies between safety critical operational activities.

As the system design evolves and matures the corresponding system model elements and their relationships to the identified critical operational domain elements provide the basis of evaluating the degree to how these preliminary hazards have either been removed or instantiated within the system design.

For example, analysis of the navigation operational activity model identifies that to navigate safely the navigation officer is dependent on determining ownship position, the position and dynamic behaviour of any fixed or mobile navigation hazards, depth below keel, ownship speed, ownship heading, ownship course and ownship depth (for a submarine).

Determining ownship position and the position and dynamic behaviour of any fixed or mobile navigation hazards is of critical importance to safe navigation. The reliability and accuracy of this information at the system level has a direct impact on the platforms ability to be navigated safely. By maintaining adequate separation between the ownship and any navigation hazards then the navigation function can be achieved safely.

Maintaining this adequate separation is the domain of the experienced navigation officer. The safety engineering objective is to ensure that the navigation officer is provided with reliable and accurate data on which to make his decisions.

For a surface combatant, depth below keel is of secondary importance but still requires adequate safety consideration. While depth below keel may assist the operator to identify and avoid navigation hazards it is only as a secondary information source and rarely contributes to ownship position determination. It is only capable of being measured directly under the ship. The navigation officer first relies on charts to determine depth ahead of the ownship position prior to navigating to another location.

The depth for a current location is only useful to validate chart depth predictions which may be considered as a safety function in particular applications. That is not to say that depth below keel does not require safety assessment, but rather, when performing that safety assessment the relative importance of depth is taken into consideration in the control definition and ALARP considerations.

For a subsurface combatant however the depth below keel measurement is more critical in determining ownship position. As the submarine must navigate within the water column, the checking of depth and depth below keel data against navigation charts is a primary means of providing submarines positional awareness.

For both surface and subsurface of platforms ownship speed, heading and course and environmental data are also secondary to safe navigation, as while this information assists the navigation officer in achieving the physical separation between the ownship and any navigation hazards it is not as important as knowing where those hazard are.

Managing the tactical environment can be seen to have similar safety critical activities and information dependencies. In simple terms, to manage the tactical environment safely requires complete and accurate information about the current operational environment. If you know the position, classification and the dynamic behaviour (bearing, range, course and speed) of all contacts in your environment (including Ownship) then you are in an optimal position to safely manage that environment. With the addition of accurate environmental data you are also in a position to safely deploy any weapons.

Table 1: Example Functional Hazard Assessment provides a functional hazard template for Collision.

Based on analysis of the critical information flow that can lead to a collision (see previous discussion) safe navigation is dependent on knowing where you are (ownship position) and knowing the position of all fixed and mobile navigation hazards.

The navigation officer requires accurate and complete information to perform his primary safety related duty of maintaining adequate separation between ownship and the identified hazards.

It is the role of the safety engineer to ensure that the system design provides sufficient reliability and accuracy for each safety critical function and information item based on its relative safety criticality. System design features such as functional redundancy of information sources and additional safety functions such as divergence checking between information sources can reduce the risk of failure to the required level (ALARP).

Hazard	Causative Factor	Preventative Control	Accident	Mitigating Control
Collision	<ul style="list-style-type: none"> Incorrect ownship position 	<ul style="list-style-type: none"> Provide ownship position data redundancy Provide divergence checking between redundant data sources 	<ul style="list-style-type: none"> Personal damage – Impact injury / drowning Equipment Damage – Loss of Platform Environmental damage – impact damage or pollution 	<ul style="list-style-type: none"> Provide life rafts / flotation devices. Provide collision response standard operating procedures Provide flood containment zones in platform
	<ul style="list-style-type: none"> Incorrect mobile navigation hazard position 	<ul style="list-style-type: none"> Provide redundant contact detection (e.g. Sonar, radar and visual) Provide divergence checking between data sources 		
	<ul style="list-style-type: none"> Incorrect fixed navigation hazard position 	<ul style="list-style-type: none"> Ensure most up to date navigation charts are used Provide depth sensors to ensure depth awareness. Provide redundant charting ability (digital and paper charts) 		
	<ul style="list-style-type: none"> Navigation officer human error 	<ul style="list-style-type: none"> Provide adequate training for navigation officer Perform HMI Design and usability analysis and testing. 		

Table 1: Example Functional Hazard Assessment

7 Using the Hazard Templates for the Physical Hazard Assessment

While the physical hazard templates can be used to guide design and construction activities at the preliminary hazard analysis the final physical hazard assessment must be made against the implementation. It is the system component models that provide input into the assessment of physical hazards. Components are evaluated against the set of physical hazard templates within the SCIB Hazard Management Engine. These templates provide a set of generic hazard, causative factor, preventative control, accident and mitigating control guidelines that the Safety engineer can consider against the system design. **Table 2 Example Physical Hazard Assessment Template** provides the template for Fire / Heat as an example. Other templates are defined for electrical shock / short circuit, electrical static discharge, electrical fire,

flammable chemical, mechanical, mechanical failure, mechanical vibration or chaffing failure, ionising radiation, non ionising radiation, over pressurisation explosion, electrical loss of power, strain and sprains, struck against, temperature extreme (hot or cold), hazardous substance, visibility, weather phenomena (snow/rain/wind/ice), struck by mass acceleration, fall, slip or trip, high intensity light, noise, excavation, dangerous substances and confined spaces.

While it is recognised that there is some minor overlap between these physical hazard categories, they are intended to be used by the safety engineer to trigger consideration of all hazard types in the analysis. Consideration of a particular hazard under more than one hazard category does not affect the underlying hazard identification, accident, causative factor or control definition.

Hazard	Causative Factor	Preventative Control	Accident	Mitigating Control
Fire/ Heat	<ul style="list-style-type: none"> Flammable material within design Unsafe work practices 	<ul style="list-style-type: none"> Identify / minimise flammable material within design or supplies Provide warning signage Provide safe work procedures 	<ul style="list-style-type: none"> Personal injury – Burn Equipment damage – Fire Environmental damage - Fire Environmental damage – Pollution 	<ul style="list-style-type: none"> Provide first aid facilities and procedures Provide fire detection and suppression Provide personal protective equipment
	<ul style="list-style-type: none"> Ignition sources within design Unsafe work practices 	<ul style="list-style-type: none"> Identify / minimise ignition sources within design or supplies Ensure adequate EMC/EMI treatment Provide warning signage Provide safe work procedures 		
	<ul style="list-style-type: none"> Equipment that is hot during normal operation 	<ul style="list-style-type: none"> Ensure adequate guarding around hot surfaces within design 		
	<ul style="list-style-type: none"> Electrical short circuit causing heating of equipment 	<ul style="list-style-type: none"> Ensure equipment design and construction is performed in accordance with defined standards Ensure electrical installation is performed in accordance with defined standards 		

Table 2: Example Physical Hazard Assessment Template

Table 2: Example Physical Hazard Assessment Template provides a physical hazard assessment for the Fire/ Heat hazard. This assessment is initially performed

during preliminary hazard assessment to guide the implementation and further refined as the design is evolved and synthesised.

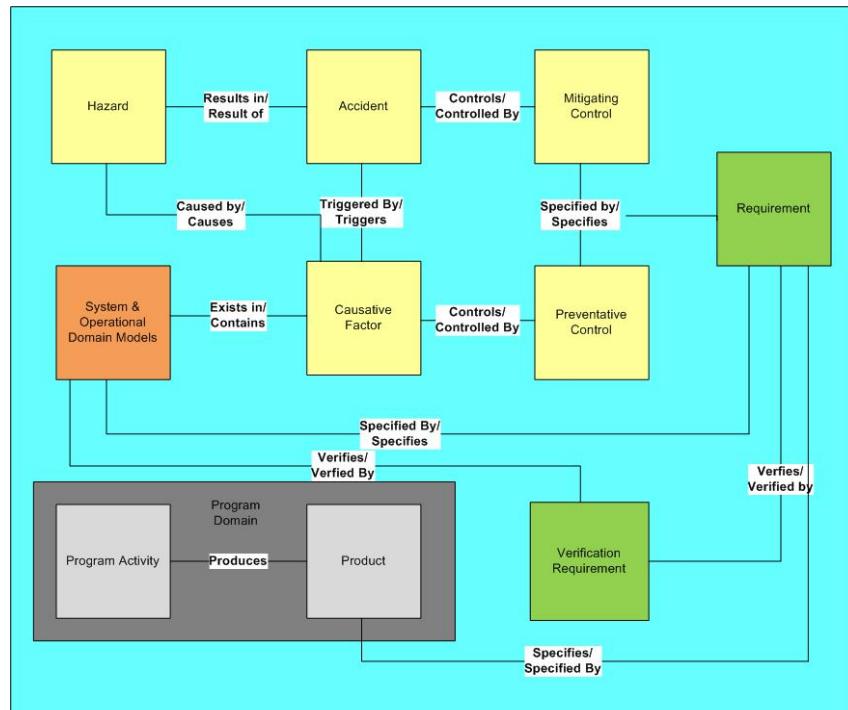


Figure 10: Hazard Management Schema

Figure 10: Hazard Management Schema describes a set of hazard, causative factor, preventative control and accident and mitigating control information classes

Within the Maritime SCIB these information classes have been populated with the set of hazard management templates that guide the risk assessment that are traced to and from the system and operational domain model elements.

8 Summary

The Maritime SCIB provides a defining standard for the management of safety with a framework upon which to manage safety risk. The hazard templates provide guidelines for the hazard definition and management that use the relationships between the operational and system domain models to support a top down operational based safety assessment. The Maritime SCIB program activities define a safety program that satisfies the applicable set of Royal Australian Navy (RAN) statutory and regulatory requirements. It is suitable for application on any maritime combat system safety program.

The approach is based on using a set of operational and system domain models for a maritime combat system to perform the required Preliminary Hazard Analysis.

It includes establishing and maintaining the Safety IDD as a knowledge base to input into the System Safety Engineering and OH&S programs of current and subsequent lifecycle phases. This continuous capture of information throughout the entire lifecycle of a naval combat system maximises the subsequent availability of that information for future use. This information not only supports safety but also the development of user documentation, standard operating procedures and training, but more importantly to support future system upgrade programs. This has the potential to considerably reduce the overall cost of safety to the RAN.

The basic hazard management concepts defined within the MCIB are transferrable to the development of virtually any other complex capability.

While the set of statutory and regulatory requirements may vary from one type of system to another, and hence the exact form of the deliverable documentation may vary, the same hazard management of applying an operational model of a capability to provide the safety argument structure and to guide the Preliminary Functional Hazard Assessment is applicable to virtually any domain.

The models described in this document have been developed within the Vitech Corporation's Model Based Systems Engineering tool, CORE using their base DoDAF schema extended to meet the specific needs of Safety. While the SCIB has been developed within CORE it is intended that the contents of these models could be used to define a set of processes, hazard assessment templates, interim and final safety deliverables templates and associated management plans that may not necessarily use CORE in any final implementation.

9 References

[1] Eurocontrol Safety Assessment Methodology web site. (2011):

www.eurocontrol.int/safety/public/standard_page/samtf.htm. Accessed April 4 2011

[2] Eurocontrol Safety Assessment Task Force. (2006): Safety Case Development Manual www.eurocontrol.int/cascade/gallery/content/public/documents/safetycasedevmanual.pdf. Accessed April 4 2011

[3] Eurocontrol Safety Assessment Task Force. (2010): Safety Assessment Made Easier Safety Principles and an Introduction to Safety Assessment www.eurocontrol.int/safety/gallery/content/public/library/SAME/Safety_Assessment_Made_Easier%20%9D_Part_1_v1_0_released.pdf. Accessed April 4 2011.

[4] Kelly, T., McDermid J.. (1998): Safety Case Patterns – Reusing Successful Arguments. in Proceedings of the IEE Colloquium on Understanding Patterns and Their Application to System Engineering (Digest No. 1998/308), Institute of Electrical Engineers.

[5] Vitech Corporation (2010): CORE Architecture Design Definition Guide DoDAF. http://www.vitechcorp.com/support/docs/70/Architecture_DefinitionGuideDoDAFv20.pdf Accessed 5 May 2011. Accessed 25 Apr 2011

Moving Towards Goal-Based Safety Management

Dr Holger Becht

Head of Signalling Systems Australia
RAMS Signalling Business Unit, Ansaldo STS
PO Box 1168, Eagle Farm 4009, Queensland

holger@asssc.org

Abstract

In virtually all safety-critical industries the operators of systems have to demonstrate a systematic and thorough consideration of safety. This is generally done through the application of safety standards as part of the development of safety critical systems.

Many safety assurance standards (like EN50126 (1999), IEC 61508 (1995), DEF (Aust) 5679) (1998) are very prescriptive. They require specific techniques, approaches or measures to be applied to achieve the safety objective without allowing the users to select a suite of techniques and measures best suited for their application and development environment. The application of prescriptive techniques can work well for some systems but can be a hindrance for emerging technologies.

There has therefore been an increasing trend in many industries to demonstrate safety by assuring certain goals have been achieved, rather than simply following prescriptive standards.

Goal-based safety standards are now a reality and applied in the medical industry and defence. This paper will describe the pros and cons of prescriptive and goal-based standards, and make recommendations for the evolution of future safety standards.

Keywords: Safety Goal-Based Standards, Safety Management

1 Introduction

In this paper we look at what benefits goal-based standards can provide to and if goal-based safety cases could be a valuable tool for reasoning about safety. We discuss opportunities and challenges for the development and use of goal-based safety cases. Finally we discuss the future of safety standards and investigate how this can become a reality for system safety management.

The structure of the paper is outlined as follows.

1. Why we need goal-based standards
2. What goal-based standards exist
3. Generic goal structures
4. Generic safety management goals
5. Generic safety development assurance goals

6. Generic sets of goals
7. The evidence required for assurance
8. The impact on industry safety standards

2 Background

The term “Assurance” inherently means a positive declaration intended to give confidence. It is a subjective determination of the strength of an inference. Safety assurance is the determination of the confidence that can be placed in the safety of a system. Assurance is a property of an argument’s conclusion and is based upon:

1. the likelihood that the claims are true (i.e. the assurance of the claims); and
2. the extent to which the claims entail the conclusion.

Safety Assurance is therefore a qualitative statement expressing the degree of confidence that a safety claim is true. The overall assurance of a system is equal to the assurance of the top-level goal.

A Safety Case is the primary means of communicating the goals, safety requirements, safety management environment and argument for assurance of critical systems. More specifically a safety case is a documented body of evidence that provides a convincing and valid argument that a system is adequately safe for a given application in a given environment.

Although safety cases are generally accepted, there are different ways of constructing an argument and providing the supporting evidence. The three main approaches can be characterised as shown in Figure 1.

1. Assurance via a set of evidence supported claims about the system’s safety behaviour.
2. The use of accepted industry “good” practices and guidelines.
3. An investigation of known potential vulnerabilities of the system.

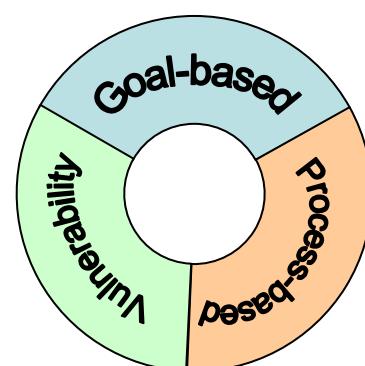


Figure 1: Safety case approaches

The first approach is goal-based – where specific safety goals for the systems are supported by arguments and evidence. The second approach is based on demonstrating compliance to a known industry accepted good practice (generally captured in a process-based safety standard). The final approach is a vulnerability-based argument where it is demonstrated that potential vulnerabilities within a system do not constitute a problem – this is essentially a “bottom-up” approach as opposed to the “top-down” approach used in goal-based methods.

These approaches are not mutually exclusive, and a combination can be used to support a safety argument, especially where the system consists of both off-the-shelf components of unknown pedigree and application-specific components.

In the past, safety arguments tended to be implicit and process-based. Compliance to accepted good practice was deemed to imply adequate safety; this is the general approach applied for most industries where compliance to standards is considered to imply adequate safety. This compliance approach works well in stable environments where good practice is supported by extensive experience, like railway signalling. However with fast moving, emerging technologies, a more pragmatic approach is required that can accommodate change and alternative strategies to achieve the same safety objective. This is why goal-based approaches are being advocated, particularly for systems with novel components and developmental systems.

3 Why Goal-Based Standards?

Historically many safety process standards have been prescriptive (i.e. tell people what to do) and/or proscriptive (i.e. tell people what to avoid doing). In contrast, goal-based standards tell people what they need to achieve (and allow alternative means to achieve this). The goal-based approach is a requirements based analysis and at a very high level, the goals are:

1. to establish safety requirements;
2. to design the system in compliance with the safety requirements; and
3. to show that the safety requirements have been fulfilled.

For example, in a goal-based approach there could be a goal to “Demonstrate completeness of the safety requirements”. In “prescriptive standard” the specific means of achieving compliance is mandated; “You shall perform a Functional Failure Analysis and Accident Sequence Analysis”.

Prescriptive process-based standards, like EN50128 (2001), IEC61508 (1995), DO-178B (1992), encode the good engineering practice at the time that they are written and rapidly become deficient as good practice is continuously changing with evolving technologies. In fact it is quite probable that prescriptive process eventually prevent the service provider from adopting current industry good practice.

Furthermore, technology changes rapidly and many projects find that cutting edge technology at the beginning of a project can be out-dated by the time it goes into service. The problem is that standards change

relatively slowly taking up to 10 years to be updated and released. This means that prescriptive standards will always be behind the technology curve.

Consequently there are clear benefits in adopting a goal-based approach as it gives greater freedom in developing technical solutions and accommodating different technical solutions. In order to adopt a goal-based approach, it is necessary to provide a coherent and convincing safety justification.

A goal-based approach can be applied at any level from the top-level system downwards. It is important that there are clear links between the top-level goals and the sub-goals. At each level, the acceptance authority requires explicit safety goals, convincing arguments to justify the goals are met, and adequate evidence to support the arguments. In practice the rigour of the arguments and the amount of evidence will depend on the safety significance of the individual system functions.

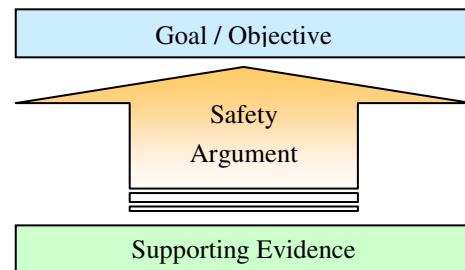


Figure 2: Goal-based Argument

The advantages, or opportunities, offered by a goal-based approach bring some attendant challenges, including:

1. Agreeing on appropriate means, and depth of evidence, for demonstrating safety, especially with emerging technology;
2. Contracting for a safety program where the set of safety activities and required evidence may not be determined “up front”.

It will also be challenging for certifying bodies to certify products to a goal-based standard. With prescriptive standards this is a relative mechanical process. The certifier would assess a product by using the prescriptive requirements in that standard as a checklist to confirm compliance. With goal-based standards this is not possible and there is much more responsibility placed on the certifier who will need to make a subjective judgement instead of an objective one. Certifiers in turn will most likely shift this responsibility onto the Independent Safety Assessor to make the judgement that a specific product or system is safe and fit-for-purpose.

This means that in order for goal-based standard to be effective some of the inherent subjectivity of this approach needs to be reduced to simplify the acceptance and certification process.

4 Goal-Based Standards

Despite the differences in detail, goal-based approaches are now being adopted in standards with the key premise that they are not to be technology specific.

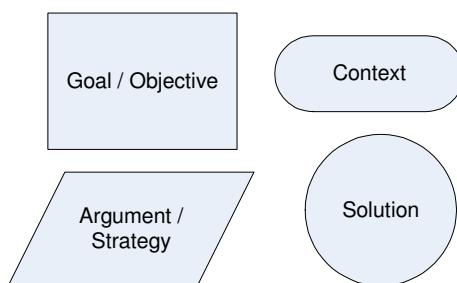
The UK Civil Aviation Authority software safety assurance standard, CAA SW01 (2002) identifies a standard set of top-level goals for software based systems which are generic (e.g. specification is valid, specification is correctly implemented, etc.).

The software part of Def Stan 00-56 (2007) requires goal-based safety justification and explicit safety arguments to support the safety claims made. Def Stan 00-56 (2007) may have taken the goal-based approach too far in an attempt to be completely flexible. The standard places the entire onus on the service provider to develop the system as they please and provide justification that the system is safe. It is clear from this standard that some structure and minimal processes need to be prescribed. In reality we see both approaches working in parallel. The Yellow Book is one of the few standards that provides high level goals and suggests several process-based standards to achieve each goal.

As stated, a combination of somewhat prescriptive safety management activities, generic goals, and process-based guidance must be captured in future standards for them to be effective and to allow a wide range of technologies to be certified. More specifically, it must be recognised that the prescriptive process-based standards are primarily a hindrance for the development and assurance of software, particularly for new and emerging technologies. It is this aspect of safety engineering that needs to be and that will gain the most benefit from a goal-based approach. The Safety Management approach should remain fairly prescriptive, structured and consistent in future safety standards. In fact it is already fairly consistent across existing safety standards from different countries and industries. The objectives and goals of safety management are investigated in more detail in a subsequent section but before this is done, we depict the generic top-level goals that would be applicable to most development projects and that should be reflected in future standards.

5 Generic Goal Structures

Although several standards have adopted goal-based approaches to safety assurance, there are differences in the way the safety argument is constructed and justified. The Goal Structuring Notation (GSN) is emerging as one of the preferred methods for constructing a goal-based



argument, and is defined in The Yellow Book (2007).

Figure 3: Elements of Goal Structured Notation

The GSN is a graphical notation that explicitly represents the individual elements of a safety argument

(requirements, claims, evidence and context) and, perhaps more significantly, the relationships that exist between these elements. That is the GSN depicts how individual requirements are supported by specific claims, how claims are supported by evidence and the assumed context that is defined for the argument. The principal symbols of the notation are shown in Figure 3.

Figure 4 provides an example of a goal structure of safety arguments, which is generally applicable to most applications.

6 Generic Safety Management Goals

As detailed above, the safety management approach should remain prescriptive and consistent amongst future safety standards. This section will expand goal G7 of Figure 4 to define the safety management goals that would enable the other goals to be achieved by ensuring that safety activities are planned, monitored against the plan, and effectively executed.

Practical experience in safety-related systems and research of existing safety standards (e.g. Def Stan 00-56 (2007), The Yellow Book (2007), IEC 61508 (1995), MIL-STD-882C (1996), and Def(Aust) 5679 (1998)) have identified the following key requirements for the development of safety systems.

1. It is essential to have a systematic approach to safety that incorporates techniques which are valid for hardware, operators and software.
2. System design must be inherently safe; issues raised during hazard analyses must be allowed to impact system design if necessary.
3. The use of integrity levels allows the application of techniques and measures which is appropriate to the criticality of a component. A practicable and sound approach is needed for the assessment of integrity levels for system components.
4. A well-defined set of appropriate techniques and measures must be applied to deliver assurance of safety.

It can be seen that these key requirements are reflected in the main safety argument S1 of Figure 4, and are based on:

1. Safety requirements are complete and correct (G2)
2. Safety requirements are satisfied (G3)
3. Appropriate standards applied (G4)

From the surveyed standards, the generic Safety Management goals indentified are:

1. Define Safety Scope: Describe the safety policy, collect information about the system and environment in which it will operate, establish the boundaries of the system and define the scope of the hazard analyses.
2. Define Safety Acceptability / Tolerability Criteria: This must be done in cooperation with the customer. It should be noted that different countries and different industries require the risk scale to be adaptable to suit the particular system

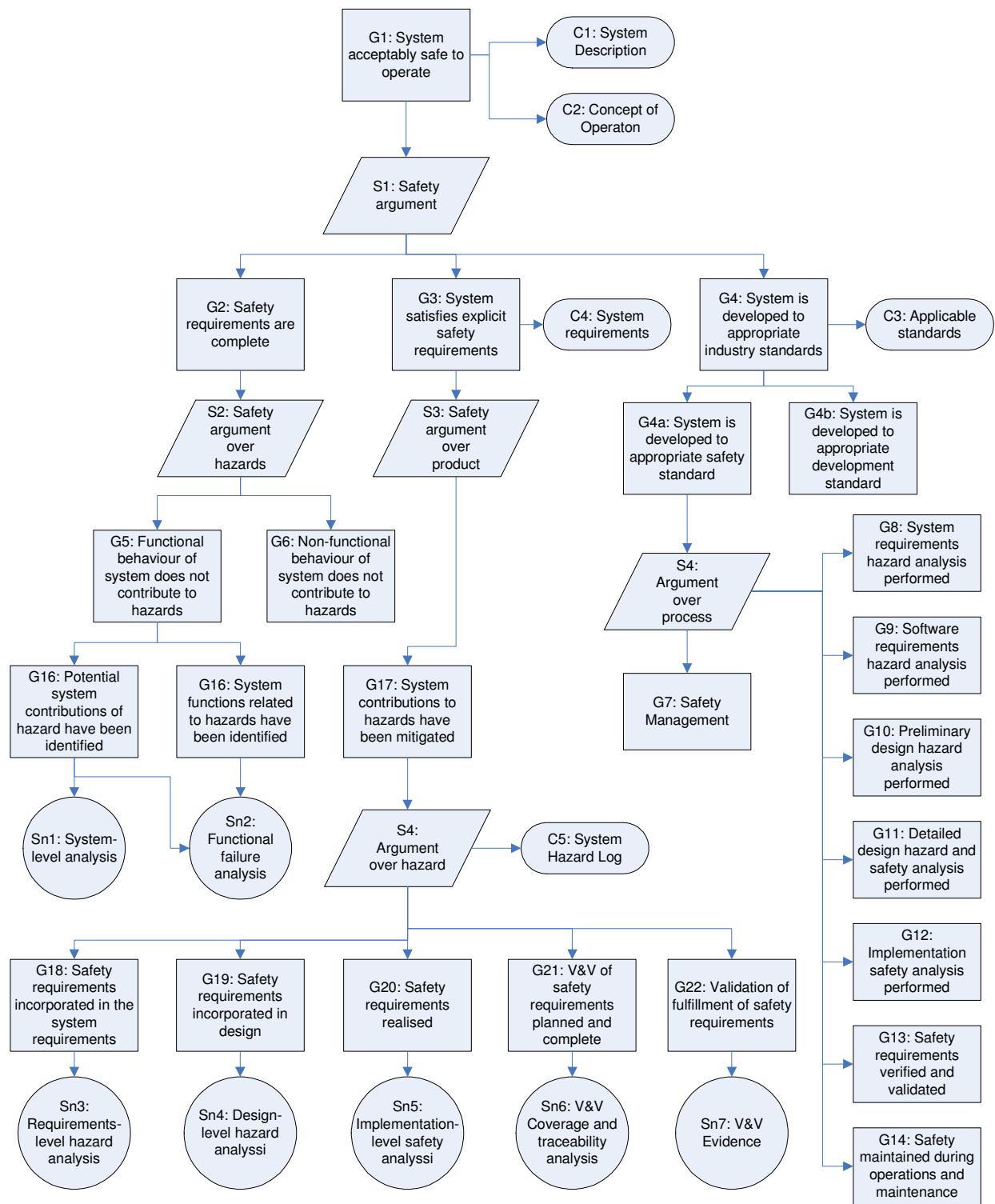


Figure 4: Example Generic Goal-Structured Safety Argument

- implementation depending on the operational profile of the relevant system.
3. **Define Safety Organisation:** Establish and maintain a safety organisation structure for the project, including specifying roles and duties of personnel and groups, providing reporting channels, and ensuring adequate levels of managerial and technical skills and independence.
 4. **Define Interface to Other Disciplines:** Define the interactions and data/information flow to and from other safety disciplines and other system engineering disciplines to ensure they effectively work together and do not duplicate work.
 5. **Define System Safety Management Plan:** Describe the activities for achieving functional safety, plan the safety analyses and assessments,

- and describe the means to develop and maintain the Safety Case.
6. **Define Hazard Tracking System:** Define a single closed-loop hazard tracking system to document hazards from identification to closure, detailing the risk assessment, risk reduction and verification evidence.
 7. **Establish Safety Management Group:** Set up a system safety management group (also referred to as system safety working group and safety committee by the surveyed standards) to oversee, review and endorse safety management and engineering activities.
 8. **Define Safety Development Assurance Tasks:** Define the process for demonstrating allocated integrity / assurance levels of components.
 9. **Independent Safety Assessment:** Plan for and assign an independent organisation to provide assurance that relevant legislations, standards and policies are complied with.
 10. **Define Safety Management System:** Provide a through life safety management plan to manage and maintain the system Safety Case during maintenance and modifications until decommissioning and disposal of the system.

Future safety standards should prescribe the abovementioned system safety management requirements. The main reason why this can be more prescriptive is because it is not technology specific.

The key benefit of the goal-based approach will however be more evident and obvious for the development assurance of software and, to a lesser extent, hardware which are technology dependent.

7 Safety Development Assurance

The primary objective of development assurance is to provide confidence that the system is free from systematic faults. The second objective is to demonstrate that the safety requirements have been correctly implemented. Development assurance is required for the development of software, hardware and configuration data, like application data for a generic product.

Functional safety requirements (generally) require the performance of certain functions to provide a level of hazard mitigation and risk reduction. The safety integrity (also referred to as Development Assurance Level) requirements define these performance requirements, which are directly proportional to the level of risk reduction required and claimed. The higher the level of risk reduction, the higher the level of integrity and confidence required that the component is functioning correctly.

Integrity requirements define the reliability and robustness required for the given safety requirements and can also be used to define the availability of the system to perform its functions.

Standards that use Safety Integrity Levels (e.g. IEC 61508 (1995)), or their equivalent concepts (Development Assurance Levels in SAE ARP 4761 (1996) or Safety Assurance Levels in DEF (Aust) 5679, (1998) explicitly or implicitly define good practice for each of the levels and therefore implicitly link

engineering methods and tools with risk and quantitative or pseudo-quantitative requirements. By dictating methods, a strategy for achieving the requisite confidence is imposed, which may work well for some applications but be a hindrance in others as already discussed.

This is why development assurance needs to provide flexibility to allow service providers to select the most appropriate set of techniques and practices for the system under development. We cannot get away from applying a set of techniques and measures to develop software and hardware. But unless the techniques and measures applied are considered to be industry good practice, it will be difficult to justify in the safety argument.

The Yellow Book provides the service provider some flexibility when it comes to development assurance by providing a list of prescriptive process standards (e.g. EN50128 (2001), IEC 61508 (1995)) that may be applied. What would be even more practical is to allow for the service provider to select, mix and match, techniques and measures from various development standards, or wherever current industry good practice is defined. It is acknowledged that this is easier said than done. For this mixing and matching of techniques and measures to be effective, there needs to be a link between development assurance goals and development processes defined within the standards.

The software assurance parts of development assurance standards, like EN50128 (2001), IEC 61508 (1995), DO-178B (1992), DO-278 (2002), DEF(Aust) 5679 (1998), SAE ARP 4761 (1996), need to eliminate prescriptive requirements, particularly those that are technology dependent. These standards need to provide a tailorabile safety assurance framework that links goals to a flexible development process. The derivation of the framework must focus on safe design concepts (i.e. goal-based) instead of good design practices (i.e. process-based), as design practices are generally tuned towards reliability and quality instead of safety as identified by McDermid (2001).

In addition, these standards need to provide sufficient guidance for alternative techniques and measures that can select in order to achieve these goals for the required integrity. This means a link needs to be provided between goals and development processes to make it easier for service providers to justify that a selected set of processes meets the development goal.

For example, when considering software safety development assurance, the good-practice techniques and measures mandated and/or suggested in the surveyed standards can be categorised into four key objectives or goals:

1. Providing a good design basis for development, customized for safety; expressed as a design and coding standard including selection of a suitable programming language or a safe subset of the programming language.
2. Ensuring that safety requirements are correct and complete; by the application of structured hazard and risk analyses.
3. Ensuring that safety requirements are adequately addressed in the design, and that the code implements only the allocated and derived

- requirements; by the provision of traceability and coverage.
- Providing evidence that each software component meets its allocated safety requirements; by the provision of design and coding verification & validation.

The key generic goals for the development of hardware would be very similar and cover:

- Quality and Reliability Assurance of Components.
- Completeness of safety requirements.
- Requirements traceability and coverage.
- Design and manufacturing verification and validation.

It is believed that defining generic sets of development goals, particularly for software, as detailed above, is what standards bodies need to focus on in order for the future safety standards to be practical and effective. Generic sets of development goals will most likely need to be defined and fine-tuned for different industries and different types of application to make it easier for the service provider to determine what evidence is required and easier to convince the acceptance authority. This will by no means be an easy activity as much effort and expertise is required to get this right.

8 Show me the Evidence!

The main problem and the question always asked with the goal-based approach, as mentioned already, is “What evidence is needed and how much evidence is enough?” Unfortunately there is no definitive answer to this question. Much effort is required by the service provider to define what evidence will be provided and then convince the acceptance authority. The reason that there is no definitive answer is intrinsic to the goal-based approach in that the evidence required is application specific and specific to the selected method of development. What is clearer is that the amount of evidence required significantly increases as the level of integrity required for (or associated with) the product increases.

Having generic sets of development goals defined, as detailed above, will help by providing a more structured breakdown of the type of evidence required. The service provider needs to break each goal down into manageable sub-goals which in turn make it easier to identify what evidence would support an argument to justify each sub-goal.

Def Stan 00-56 (2007) discusses the need for three types of evidence, and requires that a combination of these need to be provided to justify the overall safety argument; these are: process-based, product-based, and counter evidence based on vulnerability studies. It should be noted that these actually reflect the three approaches described in Figure 1, and are also evident in the generic goal structure shown in Figure 4.

Process-based evidence needs to provide confidence that industry “good” practice was applied for system development and safety management. Generally, product-based evidence is considered to be an output or result of following a particular process. Subsequently having the development processes identified should guide the

service provider in identifying the type of product-based evidence that is required for the system under development.

It is important to understand the purpose of the evidence, and what it will be used for. The evidence will be to support arguments about the behaviour of a system to gain confidence that the system is safe. The independent safety assessor will assess each piece of evidence subjectively against each argument by considering:

- Relevance
- Sufficiency
- Argument coverage
- Validity
- Independence

As already mentioned, the intrinsic subjectivity of the goal-based approach is the main drawback with this approach. This is why well-defined sets of generic development goals and a consistent safety management approach is so important for reducing some of the subjectivity.

Evidence needs to be placed under configuration management and associated with the system configuration that it allies to. Quality attributes that are associated with most engineering artefacts are likewise applicable to evidence. It must be possible to demonstrate the following properties for each piece of evidence.

- Existence
- Precision
- Completeness
- Correctness

These will be assessed objectively by the safety assessor.

9 Impact on Existing Safety Standards

So what does this mean for the current popular safety assurance standards (i.e. CENELEC standards EN50126, EN50128 and EN50129, DO-178B, IEC 61508, and The Yellow Book (2007))?

The suggested approach for future safety standards does not necessarily mean that this would be the end of existing standards. In fact, most standards would not require significant change, as large portions are not technology specific and define a relatively generic safety lifecycle and acceptance framework, along the lines of the generic safety argument in Figure 4.

One important change would be the decoupling between these standards, e.g. EN50129 should not prescribe the use of EN50126 (1999).

The Safety Cases approach needs to become goal-based which require the evidence supported safety arguments to be against the behaviour of the system instead of focusing on compliance against the the application of specific development techniques.

The biggest impact would be for the software assurance approach (e.g. EN50128 (2001), IEC 61508 Part 3, DO-178B), which must focus on safe design concepts, covering:

- Design and coding standard.
- Application of structured hazard and risk analyses.
- Safety requirements traceability and coverage.
- Design and coding verification & validation.

Out of the surveyed standards, the Yellow Book (2007) is the only standard that broadly complies to the concepts discussed in this paper, and hence would require the least change.

1. It is already goal-based and includes the goal structured notation.
2. It will need to allow for flexibility for the selection of development processes.
3. It must define generic sets of development goals, instead of listing prescriptive standards that should be applied.

Some acceptance authorities (e.g. RailCorp, TIDC) are already requiring service providers to provide more evidence to support the assurance argument and not just show compliance to standards and principles. Even without the use of goal-based standards, there will be much more effort required by the acceptance authorities in the future to justify the safety of a design and its implementation. However the goal-based approach will allow service providers to develop the system using techniques that best suit their needs.

10 Conclusions

It should be clear at this stage that prescriptive standards hampers the continual move forward in technology, while the goal-based approach leaves us without suitable advice or agreement on achieving assurance.

A goal-based approach, along the lines of that used in The Yellow Book (2007), has obvious benefits as it imposes fewer constraints on the implementation, both in terms of processes and in technical solutions. The goal-based approach is useful from a safety assurance perspective, as the questions focus on safety-related outcomes (e.g. "what evidence do you have to show that display updates occur within x seconds?").

In a goal-based approach, it is not sufficient to demonstrate compliance to a generic safety process (such as IEC 61508 (1995)). Convincing arguments have to be constructed that relate to the behaviour of the specific product and its safety properties and this can be difficult for service providers to adopt. There is a need to shift from documenting how hard people have tried to develop a system, to providing evidence and arguments about the behaviour of that system.

However, it has to be recognised that such an approach represents a significant shift from:

1. a process compliance approach to a product orientated, safety property approach
2. a tick-box mentality to argument-based mind-set

Safety program management should remain relatively prescriptive. Whereas the future of safety assurance standards needs to be goal-based as prescriptive standards cannot keep up with fast changing technology. For a goal-based approach to be effective and efficient:

1. The goals need to not be technologically specific and focus on safe design concepts.
2. There needs to be a well-defined (somewhat prescriptive) and structured process for safety management, as detailed in Figure 4.
3. Development assurance processes, particularly for software, need to be tailor able and flexible, with a clear link to goals.

4. A rich collection of generic sets of development goals needs to be defined and captured in standards.
5. Guidance needs to be provided for defining the goals and indentifying (and gaining agreement with the acceptance authority) on the type and amount of evidence required.

This shift towards goal-based assurance and arguments will by no means be easy and it will most likely take some time to get things right. A quite a mature industry with lots of experts is required, with the UK leading the way, particularly to develop the generic sets of goals for each industry.

The main challenge with the goal-based approach will be for the service provider and acceptance authority to agree on the goals and required evidence. It is also not clear if the goal-based approach would actually make it easier or more difficult for cross standard acceptance and certification, because of the more subjective nature of the goal-based approach. This requires further research and analysis.

11 References

- CENELEC EN 50126 (1999): Railway applications - The specification and demonstration of Reliability, Availability, Maintainability and Safety (RAMS).
- CENELEC EN 50128 (2001): Railway applications - Communications, signalling and processing systems - Software for railway control and protection systems.
- CENELEC EN 50129 (2003): Railway applications - Communication, signalling and processing systems - Safety related electronic systems for signalling.
- MIL-STD-882C (1996): System Safety Program Requirements. United States of America Department of Defense.
- Def Stan 00-56 (2007): Safety Management of Defence Systems. United Kingdom Ministry of Defence.
- Def (Aust) 5679 (1998): The Procurement Of Computer-based Safety Critical Systems. Defence Science Technology Organisation (DSTO).
- RTCA DO-178B (1992): Software Considerations in Airborne Systems and Equipment Certification. Radio Technical Commission for Aeronautics (RTCA).
- IEC 61508 (1995): Functional Safety: Safety Related Systems. International Electro-technical Commission (IEC).
- RTCA DO-278 (2002): Guidelines for Communications, Navigation, Surveillance, and Air Traffic Management. Radio Technical Commission for Aeronautics (RTCA).
- SAE ARP 4761 (1996): Guidelines and Methods for Conducting the Safety Assessment Process on Civil Airborne Systems and Equipments. Society of Automotive Engineers.
- The Yellow Book (2007): Engineering Safety Management, Volumes 1 and 2, Fundamentals and Guidance. Rail Safety and Standards Board. Issue 4.
- CAA SW01 (2002): Regulatory Objective for Software in Safety Related Air Traffic Services. Civil Aviation Authority, Safety Regulation Group, Air Traffic

Services Safety Requirement, Document CAP 670,
Section SW01.

McDermid, J.A. (2001): Software Safety: Where's the Evidence? *6th Australian Workshop on Industrial Experience with Safety Critical Systems and Software* (SCS'01).

Developing a methodology for the use of COTS operating systems with safety-related software

Simon Connelly Holger Becht

Ansaldo STS, PO Box 1168, Brisbane 4009, Queensland

{connelly.simon, becht.holger}@ansaldo-sts.com.au

Abstract

Conventional wisdom within the System Safety community has been that Commercial-Off-The-Shelf (COTS) Operating Systems (OS) with unknown pedigree are unsuitable for deployment in safety-related systems at anything other than the lowest integrity levels. Without assurance evidence for the OS it is difficult to gain confidence in safe behaviour of the functions provided. The typical solution therefore has been to either develop wholly embedded systems or use operating systems which have been certified to a particular standard.

Regulatory and societal expectations on software assurance is continually increasing, however ever-competitive market conditions are causing budgets to remain stable, if not decreased. As modern systems become more complex artefacts, the use of certified operating systems, or development of a bespoke embedded system, present challenges to system designers which are difficult to solve within these budgetary and schedule constraints. Consequently, the use of generic COTS OS is becoming more of a necessity.

Standards poorly define how to manage OS as far as COTS is concerned, allowing for either excessively restrictive or permissive definitions of what is required. This paper proposes a methodology to isolate the safety-related service or program from failures of the COTS OS through design and detection techniques.

The model argument presented, within the framework of the SIL based standards, justifies the use of Microsoft Windows OS (or equivalent) to enable safety-related functionality up to SIL 2.

Keywords: COTS, software safety, windows operating system.

1 Scope

The scope of this paper discusses safety-related applications (i.e. up to SIL 2, SW Level C) only and is not applicable to safety-critical (i.e. vital, SIL 3/4, SW Level A/B); the reason for this is discussed further towards the end of this paper. This paper expands previous work to discuss a it's applicability to a more complex example system, and the observed difficulties this presents.

Copyright © 2011, Australian Computer Society, Inc. This paper appeared at the Australian System Safety Conference (ASSC 2011), held in Melbourne 25-27 May, 2011. Conferences in Research and Practice in Information Technology (CRPIT), Vol. 133, Ed. Tony Cant. Reproduction for academic, not-for profit purposes permitted provided this text is included.

2 Introduction

The use of COTS software components within safety-related applications is a reality and has become increasingly more a necessity for service providers to remain competitive in a market that is driven by cost savings due to recent economic downturns. COTS software artefacts are continuing to increase in complexity, making the development of an assurance argument about systems utilising them increasingly difficult in the context of existing safety standards. Understanding the impact of COTS software failures with respect to system safety is a crucial and difficult step but key to the safety assurance of the overall system.

The term COTS, in this paper refers to software components which are readily available from commercial sources, for general application and not easily modified. Access to source code and development process is denied, or heavily restricted. Typical characteristics of COTS components are that a number of different configurations may be available, more functions than required are available, and upgrades may occur either during system development or while it is in service.

This paper is organized into the following sections.

1. A literature survey of software safety standards and how they address COTS.
2. Previous use of and assessment of COTS Operating Systems within safety-related applications.
3. Taking into consideration the literature survey and previous assessment, our approach to providing safety assurance for the use of COTS operating systems to enable a safety-related application up to SIL 2.
4. An example of this approach implemented as part of a train movement authority management system. This section expands on previous work [Connelly10] to discuss a more complex system.

3 Literature Survey

Most current safety standards require that a Safety Case (or similar) be developed to provide assurance evidence that the system is safe to operate and maintain. Assurance evidence for software components which perform one or more safety functions is required to demonstrate that sufficient rigour has been applied to the development process to meet the safety obligation. Generally this is executed through demonstration of compliance / achievement of a Safety Integrity Level (or something similar). The assurance evidence for software safety that is then required relies on a rigorous development process where the level of rigour and independence between teams is proportional to the SIL associated with the

functions provided by that software component. Because the provision of assurance evidence for software safety is through demonstration of compliance to a specific development process, this approach cannot be applied for COTS components as this information is generally not available. As such most safety standards provide guidance on how to manage COTS, with varying degrees of expected effort.

3.1 IEC61508

Requires a proven-in-use argument, and a previously developed subsystem shall only be regarded as proven in use when it has a clearly restricted functionality and when there is adequate documentary evidence, based on the previous use of a specific configuration of the subsystem (during which time all failures have been formally recorded), and which takes into account any additional analysis or testing, as required. A component or software module can be sufficiently trusted if it is already verified to the required safety integrity level, or if it fulfils the following criteria: unchanged specification; systems in different applications; at least one year of service history; - operating time according to the safety integrity level, e.g. 100,000 hours for SIL 2.

3.2 DO-178B

Requires a proven-in-use argument. That is if equivalent safety for the software can be demonstrated by the use of the software's product service history, some certification credit may be granted. The acceptability of this method is dependent on: Configuration management of the software; Effectiveness of problem reporting activity; Stability and maturity of the software; Relevance of product service history environment; Actual error rates and product service history; and Impact of modifications.

3.3 CENELEC EN50128

The use of COTS software shall be subject to the following restrictions for SIL 1 or 2; it shall be included in the software validation process.

3.4 Def(Aust) 5679

Allows for cross-standards acceptance up to SIL 2 only. It also requires that all the prescribed System modelling and verification activities are required for the COTS components.

3.5 UK DefStan 00-56

Requires a Safety Case for the COTS components, and requires "sufficient" evidence to be provided to argue for the safety of the component.

4 Use of COTS Operating Systems

HSE conducted a study to assess the safety and integrity of the Linux operating system [Pierce02]. The overall conclusion of the study was that Linux would be, in broad terms, suitable for use in many safety related applications with SIL 1 and SIL 2 integrity requirements, and that its certification to SIL 3 might be possible. However, it is not likely to be either suitable or certifiable for SIL 4 applications.

It was argued by Pierce that for an OS (or indeed any pre-existing software) to be suitable for use in safety related system, it must satisfy the following criteria with an argument provided in the Safety Case.

- C1. The behaviour must be known with sufficient exactness, in all relevant domains of behaviour, to provide adequate confidence that hazardous behaviour of the safety related application does not arise because of a mismatch between the belief of the application designer and the true behaviour of the operating system;
- C2. The behaviour must be appropriate for the characteristics of the safety related application, in all relevant domains of behaviour; and
- C3. It must be sufficiently reliable to allow the safety integrity requirements of the application to be met (when taken together with other system features). In other words, the likelihood of failures must be sufficiently low.
- C4. An analysis has been carried out to show that the OS is suitable for that application, and that suitable mitigation is in place for any hazards arising from OS failure.

As part of the analysis for C4, the following OS features were identified by Pierce which should be used as a minimum to assess the sufficiency or completeness of the safety requirements set on the OS:

1. Executive and scheduling – the process switching time and the employed scheduling policy of the operating system must meet all time-related application requirements;
2. Resource management (both internal to the operating system and provided to the application software) – the operating system's own internal use of resources must be predictable and bounded;
3. Internal communication – the operating system inter-process communication mechanisms must be robust and the risk of a corrupt message affecting safety adequately low;
4. External communication – the operating system communication mechanisms used for communication with either other computers in the network or some external system must be robust and the risk of a corrupt message affecting safety adequately low;
5. Internal liveness failures – the operating system must allow the application to meet its availability requirements;
6. Partitioning – if the operating system is used to partition functions of differing SILs, functions of lower SIL should not interfere with the correct operation of higher SIL functions;
7. Real-time – timing facilities and interrupt handling features must be sufficiently accurate to meet all application response time requirements;
8. Security – only if the operating system is used in a secure application;

9. User interface – when the operating system is used to provide a user interface, the risk of the interface corrupting the user input to the application or the output data of the application must be sufficiently low;
10. Robustness – the operating system must be able to detect and respond appropriately to the failure of the application processes and external interfaces;
11. Installation – installation procedures must include measures to protect against producing a faulty installation due to user error.

The Certification Authorities Software Team (CAST) produced a paper to argue for the use of a COTS OS in safety-related application (i.e. up to SIL 2 or DO-178B level C) [CAST02]. This paper argues that the maximum integrity level that can be claimed for a COTS OS (when the source code and design information are not available) is SIL 1 (or Level D). It then goes on to argue that the COTS OS can be used with SIL 2 application if a protection and partitioning analysis is performed in conjunction with the system safety assessment. It is the opinion of the authors that the intent is equivalent to the approach suggested in the Linux paper [Pierce02].

5 Solution

Guidance in standards is somewhat contradictory with widely varying requirements on assurance evidence for COTS. They offer little practical guidance on the development of safety assurance evidence for COTS software components. Research conducted into COTS OS has identified the broad requirements to achieve a satisfactory safety argument; however a specific strategy is not derived, nor is there evidence of this technique being applied. We therefore revert back to the fundamentals of safety assurance and focus on:

1. Analysing failure modes of the COTS components and mitigating these to eliminate unspecified/unexpected behaviours. When the COTS component is an OS, the analysis performed must respect the criteria C1 to C4 detailed above.
2. Verifying and validating the safety of the required behaviour in the required operational context.
3. Ensuring and maintaining safety during system upgrades and change.

Our approach is to utilize the functionality of low integrity COTS components within a high integrity design by restricting the influence of the component on the rest of the system. The way to restrict the influence of COTS components is by isolating them using encapsulation mechanisms such as wrappers [O'Halloran99]. To achieve this, a hazard analysis is conducted, at a level of design commensurate with the SIL of the application, to identify how failures in the operating system can cause or contribute to hazardous failure modes of the system.

To ensure the base platform remains invariant, the approach presented here assumes that OS upgrades will not be applied without conducting further analysis, and

sufficient regression testing conducted. Additionally, the OS will be minimised as far as is practicable by disabling unnecessary system services and removal \ restriction of third party applications (such as anti-virus programs). Protection against virus infections is considered to be outside the scope of this approach. This is considered acceptable as the systems under discussion are generally not utilised on an open network, or exposed to external media (e.g. USB drives) which has not been previously determined to be uninfected.

6 Putting it Into Practice

Hereafter we will relate the theory above to a practical SIL 2 software application within the framework of the railway safety standard EN 50128.

To comply with EN 50128 for SIL 2 one must at least demonstrate that the COTS products are included in the software validation process. System testing must be conducted in compliance with EN 50128 and requires that it be conducted on the system configured for its final application, including the hosting environment provided by the COTS OS, and all other COTS products. This testing will action only those features of the COTS products required to provide the functionality used for the product under development. Whilst the above approach satisfies the EN 50128 requirement for SIL 2 products, it is however acknowledged that conducting sufficient testing on the COTS products to ensure correct functionality in all circumstances is infeasible due to the complexity of these artefacts. As such additional precautions are required; guided by EN 50128 (as detailed above), specifically:

1. The possible failures of the OS (e.g. data corruption), which may affect safe functioning of the product, will be identified and assessed, and mitigations will be designed within the developed SIL 2 software.
2. System testing will test these mitigations as far as is practicable; and
3. The OS will be minimised, and all un-necessary services and products either removed or disabled.

Essentially, rather than assuring the COTS, we propose using the developed safety-related code to protect against failures which may impact upon the safety functions provided (the wrapper argument). In addition to providing protection to the safety functions, it is essential to protect safety-related input and output data to and from the SIL 2 software, because this safety-related data must pass through the untrusted COTS OS. To overcome this, guidance is drawn from the CENELEC vital communications standard EN 50159-2. It must also be noted that special consideration needs to be given to the Human-Machine interface (HMI), as often it will rely heavily on interaction with many libraries and un-trusted screen elements from the COTS OS. This is demonstrated in the analysis below where specific constraints are placed on the interaction sequence.

An added benefit of the wrapper / isolation approach is that, as the developed SIL 2 software code's interface is to the OS only, with no direct interface to the hardware;

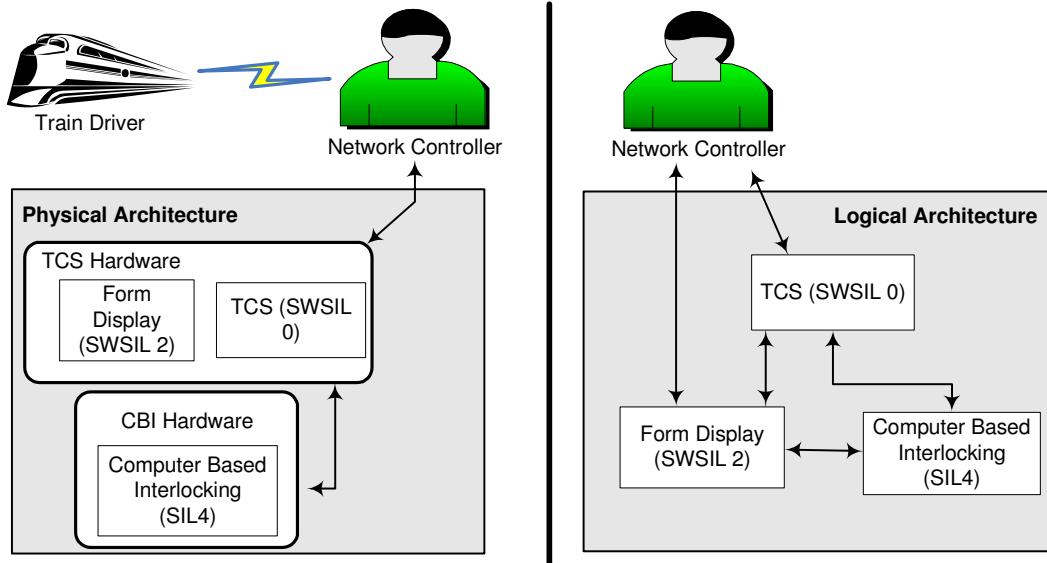


Figure 1: Train Movement Authority Management System

this simplifies and limits the need to assess hardware interactions.

6.1 Example: Train Order Management System

The example train movement authority system examined in the previous work [Connelly10] represented a centralised train control system of signalled track, with limited modification to the trackside infrastructure, the only change was the addition of track blocking and detection resets. This model has been expanded to examine management of non-signalled or “dark territory” through the use of limited trackside infrastructure with a similar concept. The analysis has been updated to take into account this operational context, and demonstration provided that the same safety requirements are appropriate in this context. A high level diagram of the example train order system is provided in Figure 1, where a SIL 0 and SIL 2 component are being executed on the same COTS OS. The logical interactions are also provided to demonstrate that the SIL rated components have separate logical communication channels.

The example system is configured as follows:

1. Safety-critical interface to trackside infrastructure is a SIL 4 interlocking, which manages validation of authorities for issuance to rail traffic and ensures points are set appropriately prior to issuing the authority to TCS for delivery; the connected infrastructure is limited to overswitch track circuits and points machines; and
2. An interface is provided to a central Train Control System (TCS), which allows the network controller to request an authority for a train, and receive a validated form for delivery to a train driver.
3. A form is delivered to a train driver via a voice communication channel. The driver is required to record each form field on a local paper copy of the form. A form is only considered “issued” and valid for execution when the network

controller confirms a correct readback from the train driver.

4. The network controller confirms successful readback of the form via the TCS to the interlocking. The TCS runs on a Microsoft Windows XP PC.

As identified in the previous analysis, traditional signalling systems rely on an interlocking design such that all controls from TCS are validated and confirmed as safe prior to modifying track status. Such is not possible in train order working, as whilst the interlocking is capable of validating an authority is safe for issuance, based on the safeworking rules, it cannot determine the current location of the train being issued an authority, or of any conflicting trains. Additionally the interlocking system cannot issue an authority directly to the train driver (as opposed to clearing signals along a route). The TCS is therefore required to allow for a network controller to confirm train location, and be provided with safeworking forms for issuance to the driver.

Should the TCS corrupt location or form information the validation functions performed by the interlocking cannot be assured to prevent conflicting authorities. As a result the safety-related data is confirmed through the use of the forms display SWSIL 2 component.

As there are many pre-existing TCS products in use in active railways, it is considered favourable from a user interaction point of view to unify the interface between control of train order and signalled area. As such providing the ability to a “safety kernel” to run on the TCS managing the safety-related issuance of authorities to trains allows network operators to leverage existing systems with minimal extra training or hardware requirements. As the validation of requests is managed by the SIL 4 interlocking, analysis has determined that the safety kernel is required to achieve SIL 2 or higher. We refer to this safety kernel as “Safety Display” in Figure 1.

The safety functions provided by the kernel are limited to correct display of validated authority information, and return of network controller confirmation or rejection of the issuance to a train driver via voice. For the purposes of this paper, incorrect or unsafe requests can be considered mitigated by the interlocking design. As

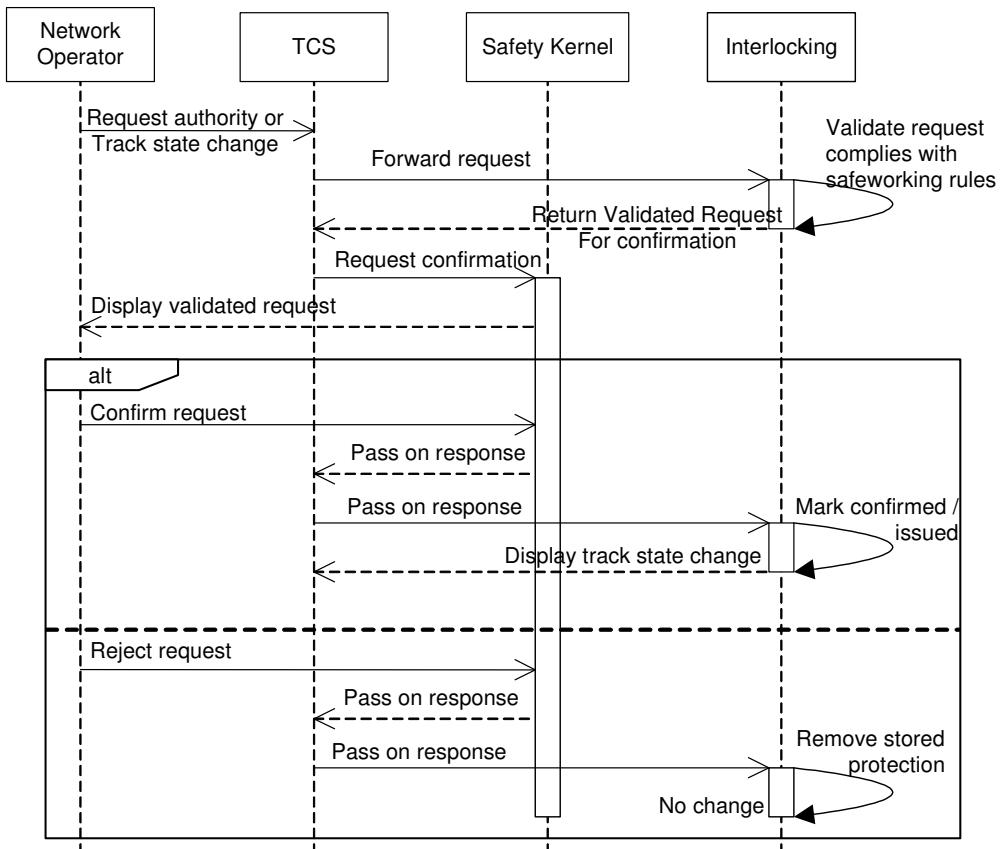


Figure 2: Sequence of Interaction

railway signalling systems are not run in strict real time, it is possible to design these functions such that they can be requested via the existing SIL 0 interface, and confirmed through the safety kernel. The development of a physically separate SIL 2 interface to the interlocking was considered and rejected as unnecessarily obfuscating the Network Controller's (NC) interaction workflows. Note that alarms require special consideration within the analysis, as they do have a timeliness component. A high level design of the sequence of interaction for each the safety function is shown in Figure 2.

The fundamental strategy is that no change to the protection managed within the interlocking is allowed to

progress without user confirmation through the "trusted" SIL 2 kernel, justifying that the remainder of the TCS does not need to achieve any integrity level.

To demonstrate independence of the safety kernel from the SIL 0 TCS and OS the following activities were revisited in light of the modified context: perform hazard analysis on the safety functions provided by TCS, identifying the COTS causes for the hazards; and mitigate each cause in the safety kernel. Following the analysis, the hazards detailed in Table 1 were identified on the interface between SIL 0 and SIL 2 functionality (i.e. between TCS and the Forms Display).

ID	Description	Cause(s)	Safety Requirement(s)
HAZ 1	Confirmation / Rejection is modified by TCS in transit to the interlocking	COTS6	HMI1, HMI8
HAZ 2	Safety Kernel SW fault corrupts message (unsafe)	COTS1, COTS2, COTS3	HMI1, HMI2, HMI3, HMI4
HAZ 3	Unrelated NC HMI interaction leads to inadvertent confirmation	COTS4	HMI5, HMI7
HAZ 4	TCS responds to confirmation request spuriously	COTS5, COTS7, COTS10	HMI8, HMI9
HAZ 5	Multiple HMI failures (SIL 0 code) confirm dialog	COTS4,	HMI6

Table 1: Identified hazards

Threat	Interpretation	Relevant failures from Table 3
Repetition	Previously correct message is resent out of context	COTS5
Deletion	Message to or from Safety Kernel is deleted by SIL 0 components	COTS8 and COTS9
Insertion	Message to Interlocking is generated by SIL 0 components	COTS7 and COTS10
Re-sequence	Messages from Safety Kernel have been changed out of sequence by SIL 0 components	COTS8 and COTS9
Corruption	Messages to or from Safety Kernel are corrupted by SIL 0 components	COTS1, and COTS6
Delay	Message is delayed, considered to be the same as deletion.	COTS8 and COTS9
Masquerade	SIL 0 components attempt to perform functions which the Interlocking is expecting the Safety Kernel to perform.	COTS4, COTS5, COTS7, and COTS10

Table 2: Treatment of EN 50159-2 data transmission integrity threats

Causes of these hazards were identified within the COTS products and addressed as shown in Table 3, the selection of probable failure modes was based upon the “operating system failure modes” detailed in [Pierce02]. When determining the possible COTS failures,

consideration was given to the identified basic message errors, or threats, defined in Clause 5 of EN 50159-2, which deals with transmission systems (shown in Table 2).

ID	SIL 0 / COTS Software Failure	Possible Effect on SIL 2 element	Safety Requirement(s)
COTS1	Corruption of incoming message from interlocking (HAZ2)	Displayed information may not precisely match information stored in interlocking. May lead to confirmation of unsafe state change.	HMI1: Data correctness and integrity shall be confirmed through a sufficiently strong HASH / CRC of all data stored in the message. This shall be repeated during Safety Kernel processing, to detect intermediate memory interference.
COTS2	Corruption of message during processing within Safety Kernel as a result of inappropriate memory access by SIL 0 elements. Could occur at any time, and may occur on volatile or non-volatile memory. (HAZ2)	Displayed information may not precisely match information stored in interlocking. May lead to confirmation of unsafe state change.	HMI1 HMI2: The Safety Kernel is run as a separate process to the TCS, utilising Operating System Level memory and execution protection.
COTS3	SIL 0 Elements may interfere with rendering of data in Operating System level dialog display, causing function to fail in a manner which may modify displayed data. NB: This could occur as an OS level failure regardless of whether there was other SIL 0 code running or not (HAZ2)	Displayed information may not precisely match information stored in interlocking. May lead to issue of incorrect information, or inability to detect operator error.	HMI3: Prior to delivery to the OS dialog renderer the safety-related data shall be rasterised to images (e.g. bitmaps) from a verified library of individual character images. Any image level corruption will be visually detectable, or insufficient to modify the data meaning. HMI4: Design of rendered information shall be sufficient to mitigate undetectable modification of bitmap location i.e. data transposition / removal.

ID	SIL 0 / COTS Software Failure	Possible Effect on SIL 2 element	Safety Requirement(s)
COTS4	<p>Required confirmation response to Safety Kernel may be triggered by SIL 0 elements, or by NC during unrelated interaction with HMI.</p> <p>NB: This could occur as an OS level failure regardless of whether there was other SIL 0 code running or not (HAZ3, HAZ5)</p>	<p>Network Controller may not have sufficient time to interpret, or see all data. If state change is one which modifies track protection (e.g. logging train off, removal of track block) Network Controllers may make unsafe decisions.</p>	<p>HMI5: The safety display is designed such that keyboard entry is disabled, meaning that should the window take focus during unrelated data entry, the NC can't accidentally cancel or confirm the state change.</p> <p>HMI6: The Safety Kernel interactions shall be such that at least three Windows events (related to the NC confirmation action) are received, in the correct sequence, prior to confirmation of state change.</p> <p>HMI7: Confirmation interactions for state changes shall be at least two discrete user interactions with the Safety Kernel dialog, geographically separated on the screen.</p> <p><i>NB: HMI6 and HMI7 are based on FTA not presented in this paper</i></p>
COTS5	<p>Message from Safety Kernel is cached by SIL 0 elements, and subsequently resent to the INTERLOCKING. Alternatively the SIL 0 elements may cause messages to be sent out of sequence.</p> <p>(HAZ4)</p>	<p>Message may match outstanding response, and incorrectly confirm / cancel state change</p>	<p>HMI8: NONCE is included in return message. Should this NONCE not match the expected number then message will be rejected by the interlocking</p>
COTS6	<p>Message from Safety Kernel is corrupted during transmission through TCS subsystem</p> <p>(HAZ1)</p>	<p>Confirmation may be changed to rejection and vice versa</p>	<p>HMI1 HMI8</p>
COTS7	<p>Message generated to INTERLOCKING by SIL 0 elements through some internal failure</p> <p>(HAZ4)</p>	<p>Message may match outstanding response, and incorrectly confirm / cancel state change</p>	<p>HMI9: Messages shall undergo endpoint authentication between the interlocking and the Safety Kernel subsystems.</p> <p>Message Authentication prevents messages from SIL 0 elements being treated as valid by either the interlocking or Safety Kernel.</p>
COTS8	<p>Failure of SIL 0 elements interacts with Safety Kernel</p> <p>(Fail safe, no hazard)</p>	<p>Possible failure to send or receive Safety Kernel messages. Alternatively messages may be sent out of sequence.</p>	<p>HMI8 N/A: TCS Backend Failure: no messages will be sent to or from the Safety Kernel – Fail safe state.</p>
COTS9	<p>SIL 0 elements consume all TCS hardware resources (Safety Kernel process starvation)</p> <p>(Fail safe, no hazard)</p>	<p>Safety Kernel may not receive or respond to messages. Alternatively messages may be sent out of sequence.</p>	<p>N/A: Fail Safe state for the TCS hardware, as the interlocking does not modify protection should confirmation not be received.</p>

ID	SIL 0 / COTS Software Failure	Possible Effect on SIL 2 element	Safety Requirement(s)
COTS10	SIL 0 element generates a message to Safety Kernel through some internal failure (although highly unlikely this is considered to be a credible failure mode) (HAZ4)	<p>Safety Kernel may respond to message, which is passed onto interlocking, and interpreted as valid.</p> <p>If state change is one which modifies track protection (e.g. reset of track detection, removal of track block) Network Controllers may make unsafe decisions.</p>	HMI9

Table 3: Safety Kernel data integrity protection from SIL 0 failure

Based on the above analysis, the nine identified HMI safety requirements must be implemented in order to achieve SIL 2 for the safety kernel. With these safety requirements in place, and confirmation via a Fault Tree Analysis, an argument can be presented that the integrity of the identified safety kernel is commensurate with EN 50128 SIL 2.

7 Issues with Alarms

Whilst the fundamental concept is that the system must be able to fail safe, in the example train order system discussed above, it was identified that the concept presented some issues with alarm management. In all cases where a network controller has requested a change of state, should the system fail to present confirmation; the railway will remain in a safe state. Where the interlocking needs to alert the network controller of a failed railway state however, this approach is not wholly appropriate.

If a train is travelling on an existing authority, the points have been confirmed by the interlocking to be in an appropriate lie for that authority. Should the interlocking then either lose detection of those points, or detect them in the incorrect lie, the train cannot be protected through any means other than the network controller advising them of the situation. As such COTS1, COTS6, COTS8 and COTS9 need to be re-examined. To partially mitigate this risk a further Safety Requirement was identified.

HMI10: The Safety Kernel shall display any safety related alarms with priority over all other messages from the interlocking

Should the TCS be unavailable, this alarm fail to be delivered, or the alarm is corrupted such that it is rejected by the system, HMI10 is insufficient to ensure the train driver can be notified within sufficient time. To ensure appropriate protection, an external mitigation was identified:

Application Condition: All time-sensitive safety alarms shall require acknowledgement within a specified time, should network controller confirmation not be provided, a control centre alarm shall be raised external to TCS to ensure rail traffic can be protected.

This analysis highlights the importance of consideration of the whole of system safety argument when assessing COTS failures within a single subsystem.

8 Further Limitations and Issues

Further to the identified issues with presentation of time-sensitive data, there are several further limitations to the applicability of the strategy. Specifically:

1. Systems of this nature need to have a fail safe state, or have sufficient external mitigations (as for alarms above);
2. Great reliance is placed on the human-in-the-loop, both to detect system failure and to perform actions correctly;
3. At higher integrity levels (SIL3 or 4),
 - a. the required integrity from the operating system is not considered justifiable due to the partial reliance on process execution integrity and separation;
 - b. higher integrity is required from the hardware, e.g. 2-out-of-2 processor architecture.
4. Some integrity is assumed of the operating system. In particular that the SIL 2 binary code shall execute unperturbed by untrusted code, and that memory will remain unchanged during active execution. The assumption on process protection is based on the lifetime of the Windows NT Kernel, and maturity of Windows XP; and
5. Should rich data entry be required, further analysis of the COTS failure modes would need to be conducted.
6. Any changes to the OS configuration (upgrades and patches) will need to be assessed to confirm that they do not impact on the safety kernel argument, as such may change the low level process execution behaviour of the OS.
7. Anti-Virus protection systems present difficulties with the approach detailed in this paper, as by design they have low level access to programs under execution, and can impact on the operating system's scheduling and interrupt executing processes. The current approach has been to forbid anti-virus protection systems as the product under development exists in a completely isolated and controlled network. It is not expected that this approach will be suitable for all applications, and as such analysis of the

possible interactions with anti-virus products will need to be conducted for more general roll-out.

9 Conclusions

This paper has presented further evidence of a practical approach to arguing for a COTS OS used to enable safety-related applications up to SIL 2. Two such systems are currently under development, and the approach determined sound by separate third party independent safety assessors. Therefore it is believed that when used within the stated limitations it is expected that the COTS OS approach described will result in a suitably safe system, whilst providing significant cost benefit to projects, and to customers in various industries.

Further work is required to apply this approach to real-time applications or ones requiring integrity greater than SIL 2.

10 References

- R.H. Pierce, Preliminary assessment of Linux for safety related systems, 2002, UK HSE Research Report 011.
- C. Jones, R. Bloomfield, P. Froome & P. Bishop, Methods for assessing the safety integrity of safety-related software of uncertain pedigree (SOUP), 2001, Contract Research Report 337.
- C. O'Halloran. Assessing Safety Critical COTS Systems. Journal of the System Safety Society, 35(2), 1999.
- Certification Authorities Software Team, Use of a Level D Commercial Off-the-Shelf Operating System in Systems with Other Software of Levels C and/or D, CAST-14, June 2002.
- United States of America Department of Defense. MIL-STD-882: System Safety Program Requirements.
- United Kingdom Ministry of Defence. DEF STAN 00-56: Safety Management of Defence Systems. 2007.
- Defence Science Technology Organisation. Def (Aust) 5679: The Procurement Of Computer-based Safety Critical Systems. DSTO, 1998.
- Radio Technical Commission for Aeronautics. DO178B: Software Considerations in Airborne Systems and Equipment Certification. 1992.
- International Electro-technical Commission. IEC61508: Functional Safety: Safety Related Systems. IEC, 1995.
- CENELEC. EN 50128: Railway applications - Communications, signalling and processing systems - Software for railway control and protection systems. 2001.
- CENELEC EN 50159-2: Railway applications - Communication, signalling and processing systems - Part 2: Safety-related communication in open transmission systems. 2002.
- S. Connelly, H. Becht, Arguing for the use of COTS operating systems with safety-related software, ISSC 28 2010.

Urgent Operational Requirements: Impact on the Safety Case

Tony Cant and Brendan Mahony

Command, Control, Communications and Intelligence Division
 Defence Science and Technology Organisation
 PO Box 1500, Edinburgh, South Australia 5111
 Email: Tony.Cant@dsto.defence.gov.au, Brendan.Mahony@dsto.defence.gov.au

Abstract

Modern Defence systems are complex and software-intensive. In response to the technical challenges posed by such systems Defence has developed a capability lifecycle with suitably rigorous quality control measures. Unfortunately, in today's rapidly evolving Defence environment, unforeseen threats can lead to capability gaps that require rapidly fielded solutions. Such *Urgent Operational Requirements* (UOR) can accelerate (and perhaps curtail) the normal capability lifecycle.

Defence systems are often safety-critical: they have the potential to cause death or injury as a result of accidents arising from unintended system behaviour. For such systems an effective safety engineering process (along with choice of the appropriate safety standards) must be established at an early stage of the capability lifecycle, and reflected in contract documents. This process culminates in a safety case, which is a structured argument, supported by a body of evidence, that provides a compelling, comprehensible valid case that a system is safe for a given application in a given environment.

In this paper we discuss the impact of Urgent Operational Requirements and the above lifecycle issues on the Safety Case. We use the processes and terminology of the recently published standard DEF(AUST)5679 Issue 2. In discussing the impact of UORs on the safety case, we find it useful to distinguish three cases: *Greenfield Acquisition*, *In-Service Modification* and *Modified Operational Context*.

Keywords: Safety case, safety assurance, rapid acquisition, urgent operational requirements.

1 Introduction

Modern Defence systems (such as combat systems, avionics systems, command support systems, precision weapons systems etc) are complex and software-intensive systems. In response to the technical challenges posed by such systems Defence has developed a capability lifecycle with suitably rigorous quality control measures.

1.1 The Capability Lifecycle

In the Australian context the *capability lifecycle* for such systems is divided into the following phases:

Copyright © 2011, Commonwealth of Australia. This paper appeared at the Australian System Safety Conference (ASSC 2011), Melbourne, Australia. Conferences in Research and Practice in Information Technology (CRPIT), Vol. 133. Tony Cant, Ed. Reproduction for academic, not-for profit purposes permitted provided this text is included.

1. *The Needs Phase* — involves the articulation of capability goals in the context of the current and planned force.
2. *The Requirements Phase* — involves the detailed planning required for converting capability needs into an integrated set of changes in each of the Fundamental Inputs to Capability (FIC). The Requirements Phase also incorporates a *Decision Making Process*, consisting of:
 - *First Pass Approval* — at which Government considers alternatives and approves capability development options; and
 - *Second Pass Approval* — at which Government agrees to fund the acquisition and through-life costs of a specific capability.
3. *The Acquisition Phase* — the process by which Defence acquires a specific capability via requests for tender, risk reduction activities and management of system procurement via an Australian Defence contract. At the end of the Acquisition Phase, an *Acceptance into Service* decision is made in light of an assessment made by the relevant service's technical regulator.
4. *The In-Service Phase* — the normal operating life of the system in service.
5. *The Disposal Phase* — controlled and managed decommissioning of the system.

1.2 Urgent Operational Requirements

The Capability Lifecycle is a measured and well-instrumented process, designed to make well-informed decisions about the acquisition and deployment of capabilities with in-service lifetimes of up to a quarter of a century.

Unfortunately, in today's rapidly evolving Defence environment, unforeseen threats can lead to capability gaps that require rapidly fielded solutions. This is often referred to as an *Urgent Operational Requirement* (UOR). In some countries, the term Rapid Acquisition is used instead of UOR. We essentially use them as synonyms in this paper.

The general tendency of UORs is to accelerate (and perhaps abbreviate) the normal capability lifecycle. The Capability Lifecycle is intended to mitigate the risks inherent to capability development. Capability development risk include the following classes: engineering risks (project failure, capability gap); economic risks (budget overrun); security risks (release of classified information) and safety risk (death or injury of personnel or the public). Accelerating the process necessarily reduces the level of risk mitigation, but this is balanced against the mission risk that gives rise to the UOR. The evaluation of these competing

risks is fundamentally different for the various risk classes.

The primary motivation for mitigation of engineering risk lies in the potential for leaving a pressing capability gap, but this is precisely the problem which leads to the UOR, so that it is highly likely that a timely effort is better than a low engineering risk effort.

A high level of mitigation of economic risk is inherent to the time pressures posed by the UOR. By definition, a rapid acquisition will be of relatively fixed duration and cost. It is at low risk of overruns, provided that the rate of spend is well contained. Thus, it is fairly straightforward to make a rational decision in balancing economic and mission risks.

The primary motivation for mitigation of security risk lies in the potential for breaches to lead to mission failure. Again, the very existence of the UOR means that there is already a high risk of mission failure, so that it is highly likely the a timely effort is better than a low security risk effort.

The primary motivation for the mitigation of safety risk is to protect defence personnel and the general public from death or injury. In order to rationally balance safety risk against the mission risk motivating the UOR, it is necessary to properly identify the level of safety risk posed by the system. In contrast to the other risk classes discussed, there is no natural tendency for the UOR to limit the level of safety risk. Making rational decisions about the safety risk associated with a system requires the existence of an appropriately rigorous safety case.

1.3 Safety Cases

In Australia, the Occupational Health and Safety (OH&S) Act¹ requires that parties involved in the acquisition and sustainment of systems for Defence have a duty of care arising from their legal obligation to take “reasonably practicable steps to avert harm to members of the public, as well as their own employees.” A breach of this duty could make them liable in the case of an accident.

Defence systems often have the potential to cause death or injury as a result of accidents arising from unintended system behaviour. For such systems an effective safety engineering process (along with choice of the appropriate safety standards) must be established at an early stage of the acquisition lifecycle, and reflected in contract documents. This process culminates in a *safety case* that is presented to safety evaluators and certifiers for assessment. A safety case has been defined to be (Ministry of Defence 2007):

...a structured argument, supported by a body of evidence, that provides a compelling, comprehensible valid case that a system is safe for a given application in a given environment.

The safety case is the natural vehicle for the assessment and communication of the safety risk that is potentially introduced by use of the system — not least in the case of UORs. In fact, the accelerated nature of Rapid Acquisition requires a corresponding increase in the rate of safety effort to ensure a timely assessment of safety risk. In practice, there are known to be cases in which a UOR system has not been accepted into service due to high levels of safety risk.

In discussing the impact of UORs on the provision of safety cases, we find it useful to distinguish three system acquisition classes.

- *Greenfield Acquisition*: a new capability is acquired from scratch.
- *In-Service Modification*: a system is modified during its operational life.
- *Modified Operational Context*: a system is used in situations for which it was not originally intended.

Each of these classes occur quite naturally in capability development, but each provides different insights into the challenges posed by UORs.

1.4 Outline

In this paper we are interested in the implications that UORs can have on the safety case. First of all, in Section 2 we provide general background on the issue of Urgent Operational Requirements. In Section 3 we discuss the Nimrod Review. In Section 4 we discuss the structure of the safety case using the terminology of the recently released standard DEF(AUST)5679 Issue 2 (Department of Defence 2008c). Section 5 summarises the issues involved in the procurement of Non-Development Systems. In Section 6, we discuss the impact of UOR on the safety case; while in Section 7, we consider the three class of System Acquisition so as to identify situations that are highly favourable to Rapid Acquisition. Finally, Section 8 presents some conclusions.

2 Urgent Operational Requirements

A key driver for Defence organisations in, for example, the US, UK and Australia is the need to support peacekeeping or military operations across a range of environments. Such operations (for example the USA’s Operation Iraqi Freedom or the UK’s Operation HERRICK in Afghanistan) present huge challenges owing to the nature of the terrain, the political landscape and the threat posed by asymmetric warfare. Current Australian Defence operations are: CATALYST (Iraq); SLIPPER (focused on Afghanistan); ASTUTE (East Timor) and ANODE (Solomon Islands). These are smaller in scale than the corresponding UK or US operations, but present a similar range of challenges.

UOR is a complex area: in the following we highlight some aspects of UOR in the UK, USA and Australia that are especially relevant for our later discussions on safety.

2.1 UK

In the UK special Treasury funding is used to support UORs, for example the Ridgback and Mastiff Protected Patrol Vehicles used in Iraq and Afghanistan. The definition of UOR used in the UK is as follows (Ministry of Defence 2011):

UORs arise from the identification of previously un-provisioned and emerging capability gaps as a result of current or imminent operations or where deliveries under existing contracts for equipment or services require accelerating due to an increased urgency to bring the capability they provided into service. These capability shortfalls are addressed by the urgent procurement of either new or additional equipment, enhancing existing capability, within a timescale that cannot be met by the normal acquisition cycle.

¹Occupational Health and Safety (Commonwealth Employment) Act, 1991

In a recent speech entitled “Performance under pressure: the reality of acquisition in the world’s most complex environment”, Andrew Tyler (Tyler 2009) (the UK MOD’s Defence Equipment and Support (DE&S) Chief Operating Officer) points out that the MOD is “an organisation that is on a war footing.” DE&S has around 850 staff involved in Urgent Operational Requirements (UOR); over recent times they have responded to about 1600 urgent requirements, resulting in 700 items of new equipment being delivered into theatres of operations (often in less than six months). Tyler also comments that: “as much leading-edge technology is being brought to bear on the incredibly complex problem of counter Improvised Explosive Devices (IEDs) as is going into low observability on the Joint Strike Fighter”.

Tyler draws the distinction between UOR processes and the “normal” acquisition process:

Applying UOR processes to the purchase of a nuclear submarine is an absolute non-starter. UORs are about meeting an immediate military need, using rapidly modified off-the-shelf equipment where possible, which may be discarded quickly when the immediate requirement is removed. No enduring support solution is required and integration with wider systems is often minimised for expediency. Furthermore the degree of scrutiny of public spending is balanced against the rapid delivery times required to support crucial operations. None of this applies to nuclear submarines and fighter aircraft which take many years to design and build, usually succeed complex equipment already in service and are designed to meet the long-term military capabilities required in future decades.

Highly skilled, versatile and diverse teams tend to be involved in the problem-solving that is necessary to meet UORs.

2.2 USA

In the USA, the Report of the Defense Science Board Task Force on the Fulfilment of Urgent Operational Needs recommends that Rapid Acquisitions be acknowledged as processes that are formally different from (and incompatible with) deliberate (i.e. normal) acquisition processes. It also recommends that a separate funding stream and organisation be established to handle Rapid Acquisitions.

The Office of the Director, Defense Research and Engineering (DDR&E) has commissioned a study of tools suitable for Rapid Acquisition. This study has highlighted the need to focus on the “front-end” of the capability lifecycle by creating a strategic effort in “accelerated concept engineering” (from anticipated or emerging need to initial design). There is heavy emphasis on exploiting gaming technologies for need and concept exploration; explicit accounting of potential threat evolution and vulnerabilities (“red teaming”); modelling and simulation tools to support concept engineering; and agile and adaptive systems engineering.

During the study, it was observed that most Rapid Acquisitions are not new: they start with some existing capability, and their objective is to build on, adapt, or integrate.

2.3 Australia

In Australia, the recently published Defence Instruction (General) DI(G) LOG 4-1-008 (Department of

Defence 2008b) recognises the challenges posed by asymmetric warfare and provides an overall policy framework for Rapid Acquisition of Capability. It makes the Prime Minister the approving authority for Rapid Acquisitions, and includes the following policy statements relating to safety aspects of rapidly procured equipment:

- Procurement via Rapid Acquisition must not be used to circumvent or over-ride extant Government or Departmental policy.
- Capabilities acquired through Rapid Acquisition shall be certified as fit for service, safe and, where appropriate, comply with regulations for the protection of the environment.

However, the document also allows for the acceptance of risks at higher levels of authority:

10. Capability Managers must identify any risks associated with equipment procured under the Rapid Acquisition process. Risks identified under this process must only be waived at the correct level. Only the Government, CDF and Service Chiefs have the authority to accept the risks associated with the use of items acquired under Rapid Acquisition where, due to time critical requirements, normal due process cannot be followed.

It also allows (in Annex E) for some dilution in the degree of technical regulation:

2. Regulation. In developing the Rapid Acquisition proposal, Capability Managers are to refer to Defence Instruction (General) LOG 0815 — Regulation of Technical Integrity of Australian Defence Force Materiel. TRAs are to be mindful of the timeframes by which Rapid Acquisition capabilities may need to be deployed, which will necessitate risk assessments and judgements to be made concerning the degree to which regulation of the materiel is to be applied. Risks in the areas of safety, performance and environmental compliance are to be documented, reported and managed as part of the Rapid Acquisition process.

The recently published Strategic Reform Program (Delivering Force 2030) (Department of Defence 2009) outlines a program of savings within Defence that will deliver gross savings of \$20 billion. This money is to be reinvested in key areas of Defence to deliver stronger military capabilities; to remediate poorly funded areas; and to modernise the Defence enterprise backbone. The following reference is made to safety:

This program is not about compromising capability to save costs; it is about delivering improved levels of capability at less cost by improving productivity and eliminating waste. While efficiencies can be found in support areas, quality and safety will not be compromised.

In the Technical Regulatory framework for the Australian Army, policy has been developed to address issues arising from Rapid Acquisition (RA). The RA process considers: (1) risks to fitness for service (i.e. mission risk); (2) safety risks to the personnel or public; and (3) environmental risks.

In a normal acquisition, these three aspects will be articulated in a User Requirement and subsequent

Functional & Performance Specification (FPS). Then tendered options are assessed against the Statement of Work (a document that includes the FPS), and a preferred solution is selected. The aim is for the matériel to go into service with a residual risk baseline that is LOW, or at a level of risk that is assessed to be As Low As Reasonably Practicable (ALARP).

The policy recognises that, in a Rapid Acquisition, there is sometimes insufficient time to do this work, and that equipment has the potential to enter service with a significant level of residual risk. The aim of a Rapid Acquisition is to minimise risk as low as reasonably practicable in the time frame available, that is, as much of the above process should be followed as time permits.

The Defence Materiel Organisation (DMO), in carrying out a Rapid Acquisition, often cannot assure risk free operation to the user. Rather, the emphasis is on the DMO to understand the technical risks and inform the user accordingly so that they are able to make informed decisions regarding the equipment's use and operational impact.

3 The Nimrod Review

The recently released Nimrod Review (Haddon-Cave 2009) is an independent review by Charles Haddon-Cave QC into the broader issues surrounding the loss of the RAF Nimrod MR2 Aircraft XV230 in Afghanistan in 2006. It is an example of the issues that can arise with UORs.

3.1 Background

The Falklands War in April 1982 gave rise to an Urgent Operational Requirement (UOR) to equip the Nimrod MR2 with an Air-to-Air Refuelling (AAR) capability, thereby extending the Nimrod's endurance to 20 hours in the air so that they could better support British operations during the war (a new operational context). An in-service modification was made to the aircraft to provide the required AAR capability.

The initial UOR design was modified in 1989 to meet the requirements of Def-Stan 00-970 (Ministry of Defence 1983). The AAR modification changed the function of refuel pipes within No. 7 Tank Dry Bay (previously they had not been used in flight). The review states that: "In making these pipes 'live', the AAR modification introduced a significant new element to the risk of fire because of their close proximity to the hot Cross-Feed/SCP duct".

The review concludes that the accident most likely resulted from ignition (via the Cross-Feed/SCP duct) of fuel in the No 7 Tank Dry Bay that had accumulated as a result of AAR. The review further states that design flaws introduced over the life of the aircraft played a crucial part in the loss of the aircraft.

The review also claims that organisational factors also played a major role in the loss of XV230, and is critical of the Military Airworthiness System. Following the 1998 Strategic Defence Review, financial pressures and the shift in culture towards business and financial targets led to a "dilution of the airworthiness regime and culture within the MOD, and distraction from safety and airworthiness issues as the top priority".

The loss of the XV230 aircraft is illustrative of the consequences of extending the lifecycle beyond its intended end-point. The Nimrod Review points to "an inadequate appreciation of the needs of Aged Aircraft" and goes on to state: "But for the delays in the Nimrod MRA4 replacement programme, XV230 would probably have no longer have been flying in

September 2006, because it would have reached its Out-of-Service Date and already been scrapped or stripped for conversion."

3.2 Safety Case Criticisms

The Nimrod Review is especially critical of the inadequacy of the Nimrod Safety Case. For example, the safety case had a number of open or not properly assessed hazards, including the catastrophic fire hazard relating to the Cross-Feed/SCP duct that was the ignition source in the accident.

The Nimrod Review is likely to have a significant impact on the UK MOD procurement policy for safety-critical systems. We will not reflect on all of these in this paper, but concentrate on the comments and recommendations relevant for safety cases that are made in the report. The Nimrod Review (in Chapter 22) says: "The safety case regime has lost its way. It has led to a culture of 'paper safety' at the expense of *real safety*." Safety cases are too lengthy and complex; use obscure language; lack operator input; tend to be compliance-only exercises; involve audits of process only; and make prior assumptions of safety of 'shelf-ware' (another term for non-development items).

The Review makes the point that the definition of safety case given earlier tends to encourage a "laborious, discursive, document-heavy argument ('a structured argument', 'a body of evidence') aimed at justifying a self-fulfilling prophecy ('system is safe')."

It is recommended by the Nimrod Review that safety cases be re-named *risk cases*, ("to focus attention on the fact that they are about managing risk, not assuming safety"). The risk case is intended to provide "reasonable confirmation that risks are managed to ALARP." As used in the Review, the term 'risk case' implies the need to focus attention on the most significant hazards and the ways that they can lead to dangerous situations.² It must conform to six principles (abbreviated as *SHAPED*): Succinct; Home-grown; Accessible; Proportionate; Easy to understand; and Document-lite.

The Review also comments that "care should be taken when utilising techniques such as Goal Structured Notation or Claims-Arguments-Evidence to avoid falling into the trap of assuming the conclusion (the platform is safe), or looking for supporting evidence for the conclusion instead of carrying out a proper analysis of risk."

The Review states that "care should be taken when using quantitative probabilities ... Such figures and their associated nomenclature give the illusion and comfort of accuracy and a well-honed scientific approach. Outside the world of structures, numbers are far from exact. Quantitative Risk Assessment is an art not a science. There is no substitute for engineering judgement."

4 Structure of the Safety Case

The exact structure of a safety case depends on the application domain and the relevant standard(s); however, all safety cases have a number of features in common. The safety case structure described in

²While the term 'safety case' is shorthand for 'the (evidence-based) case for system safety', the term 'risk case' means something like: 'the (evidence-based and streamlined) case for system safety in which the system hazards and safety risks are clearly stated, understood and accepted'. The term 'risk case' does not imply, as some might think, a focus on consideration of system risks other than safety. The authors do not believe that 'risk case' is a helpful concept and it will not be used in the rest of this paper. Having said that, we recognize that technically unsound terms can nevertheless be effective in a management or political context.

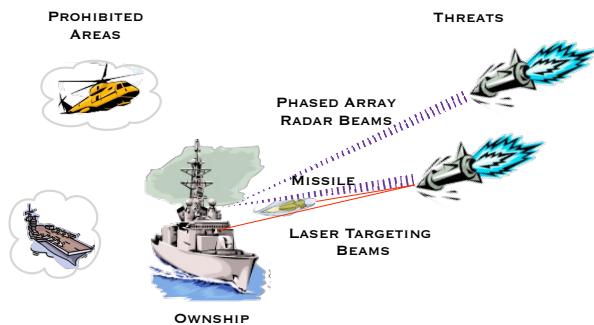


Figure 1: PARTI System Overview

this paper is taken from DEF(AUST)5679. There are three key reports in the DEF(AUST)5679 safety case:

- *Hazard Analysis* – identify the potential hazards posed by the system;
- *Safety Architecture* – demonstrate that the system is architected to be safe; and
- *Design Assurance* – demonstrate that the components are designed to be safe.

We illustrate the structure of the safety case using a case study from DEF(AUST)10679 (Department of Defence 2008a, Mahony & Cant 2008). The PARTI (Phased Array Radar and Target Illumination) System is a ship-borne Surface to Air Missile (SAM) targeting support system. It uses a Phased Array Radar (PAR) to direct laser illumination of hostile missiles and aircraft. The laser illumination provides targeting information to an existing ownship SAM capability. The main items of interest in the PARTI and environment are depicted in Figure 1.

4.1 Hazard Analysis

The first report of the safety case is called the *hazard analysis*. It provides an assessment of the danger (or threat to safety) that is potentially presented by the system. The hazard analysis must describe the system, its operational context and how the two interface from a safety viewpoint. Potential hazards posed by the system are then identified through a series of thought experiments about possible ways in which the system and its environment may interact to cause harm.

An *accident* is an external event that could directly lead to death or injury. An *accident scenario* describes a causally related mixture of system behaviours (*hazards*) and environment behaviours (*co-effectors*) that may culminate in an accident. The *severity* of an accident is a measure of the degree of its seriousness in terms of the extent of injury or death that may result. The *external mitigation level* associated with a hazard is a (qualitative) measure of the likelihood that an accident will result, given that the hazard is raised. The combination of severity and external mitigation level determine the *danger level* posed by each of the hazards individually and thereby the system in aggregate.

The need for a comprehensive identification of the relevant system hazards is probably self evident, but of particular interest to the discussion of UORs is the need for a complete description of the operational context. In the case of the PARTI system this involves such factors as: the ownship CMS (Combat Management System); the SAM capability for which the PARTI is providing a targeting service; ship support systems that provided power and physical security;

ship sensors that provided situation awareness; ship helicopters and other ordnance; personnel placements and procedures; friendly ships and aircraft; weather and sea conditions etc.

While such factors have immediate and obvious implications for the level of danger posed by a system, there are also more subtle implications for the suitability and effectiveness of a system's safety architecture or even on the nature of the system level hazards. The two primary hazards of the PARTI system are derived from the emission of radar and laser beams. These beams are both inherently hazardous (when directed at friendly assets) and necessary to the function of the system (when directed at missile threats), so the system can only be safely operated in a context that is aware of the hazard and is regulated to mitigate the hazard. In this case a protocol of prohibited areas is introduced into which the PARTI does not radiate and that vulnerable assets in the environment do not leave. The system hazards associated with the beams then become *radiating into a prohibited area*, rather than the unavoidable *emitting hazardous radiation*.

4.2 Safety Architecture

The aim of the *safety architecture* report is to describe and analyse the broad structure of the system from a safety viewpoint.

The first step is the development of a collection of *system safety requirements*, which collectively assert that the system hazards do not occur. The next step is to decompose the system into *components* and to describe how they combine to carry out the safety functions of the system. The interaction between components is described in terms of component *interfaces*, both between components and with the environment. Finally, the effectiveness of the safety architecture is shown by proposing *component safety requirements* and providing a *correctness* argument that shows how these component safety requirements ensure satisfaction of the system safety requirements (this is called *architecture verification*).

The architecture verification shows that the system will operate safely in its intended or *nominal* mode of operation. The safety architecture will, in general, also include *internal mitigations* that serve to make the system robust to unintended or *failure* modes of operation. Internal mitigations generally serve either to contain hazards (*partitioning*) or to distribute risk (*redundancy*). An argument must be made that the robustness of the internal mitigations and of the individual components is adequate to the dangers posed by the system.

The safety architecture of the PARTI system is depicted in Figure 2. The system's heavy reliance on situational awareness provided by the ownship's CMS is explicit in the diagram, but the safety argument may also make use of assumed properties of the operational context such as the deck placement of components, personnel placement at combat stations, sea state limitations etc. In fact, the system's two most prominent safety features, namely the components *PAR Filter* and *Interlock* provide redundancy in the safety functions of not radiating into protected zones. As described above, the effectiveness of these safety functions derive directly from the presence of mitigating factors in the operational context.

4.3 Design Assurance

The aim of the *design assurance* report is to provide evidence that components are designed and implemented so as to satisfy their component safety requirements.

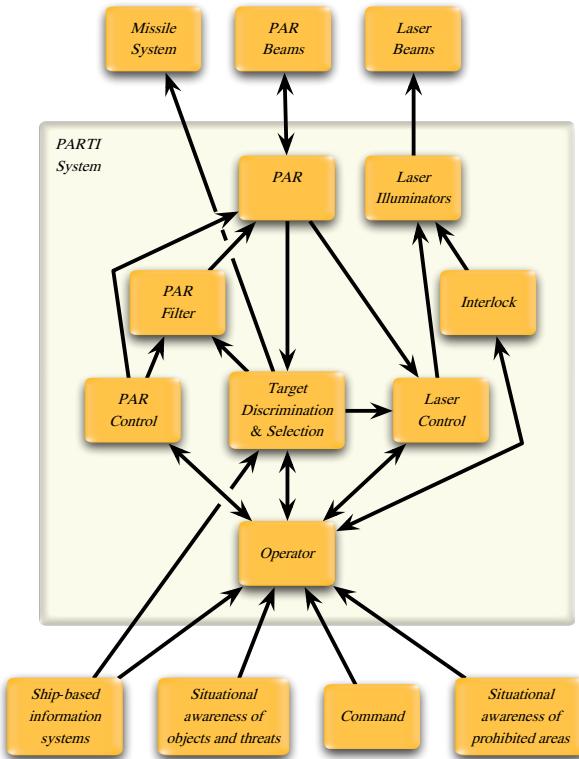


Figure 2: PARTI Architecture

The first step is to describe and justify the *implementation technology* for each component. This encompasses the design techniques and equipment used in the component. The choice of implementation technology must be justified as appropriate. In particular, the equipment must be shown to be suitably robust in consideration of the dangers posed and the assumed operating conditions.

Implementation technologies fall roughly into one of four classes, depending on whether the functions of the component are carried out using *analogue hardware*; *digital hardware*; *software* or *operator*. Specific assurance activities are prescribed, depending on the given implementation technology class.

Design analysis proceeds by re-expressing the component safety requirements in a form appropriate to the chosen implementation technology: this is called the *component safety specification*. A *design model* is developed for each component and a correctness argument developed that this model meets the component safety specification. *Design testing* is also carried out. Both design verification and testing may depend on assumptions about the behaviour of other components and even of the environment.

4.4 Safety Case Summary

The *safety case summary* is an overall narrative (or high-level argument) that is convincing to a third-party and pulls the results of the above phases together.

4.5 Observations on the Nimrod Review

Having discussed the structure of the safety case as provided in DEF(AUST)5679, and its application to the PARTI system, we now return to the conclusions of the Nimrod review and ask how DEF(AUST)5679 measures up against its recommendations.

First of all, consider the Nimrod review's recommendation to rename "safety cases" to "risk cases".

Although we do not agree with the change of name, we do agree with the intent: safety risks must be clearly identified. For this reason, we strongly believe that the hazard analysis phase remains central to *any* safety case. The main reason for this is that hazard analysis *identifies* the potential risks to human safety: without it, no sensible decisions can be made about whether sufficient effort has been made to eliminate or reduce these risks. It is the starting point for effective safety assessment of *any* system. This view is in direct accordance with the conclusions of the Nimrod Review. Roughly speaking it presents "evidence for unsafety" rather than "evidence for safety".

We strongly concur with the Nimrod Review's criticism of safety cases that are too process-focused. A primary aim of DEF(AUST)5679 is to focus attention on the safety of the actual system (product assurance). To this end a small, targeted collection of documents are mandated, each addressing a crucial aspect of system safety. There is a minimal number of purely process requirements. We call this approach *document-focused*. Adopting a document focus helps direct attention back to the system itself and its safety properties. It promotes a more general ownership of the safety case by de-emphasising the agents and processes involved in safety management and engineering. Similarly, it promotes reuse of safety case artefacts in subsequent maintenance and re-development phases.

We also concur with the need to properly involve operators (called *End Users* in the Standard) in all aspects of safety engineering. A number of the process-focused requirements of DEF(AUST)5679 are specifically designed to ensure appropriate levels of End User input to the safety case.

The danger of safety engineering devolving to a "compliance only exercise" is necessarily a concern regardless of the safety standard adopted. DEF(AUST)5679 addresses this through the *Evaluator*, an independent agent whose primary responsibility is to assess the technical safety of the system itself, focussing his/her attention on the quality and completeness of the arguments presented in the safety case. By bringing an independent set of experiences and biases to the Safety Case, the Evaluator serves as a second line of defence against "safety as a self-fulfilling prophecy".

Example: for a software-controlled explosive round, it is claimed in the safety case that there are no safety issues once the weapon has successfully been fired, and consequently there is no analysis of hazards relating to impact of the weapon in areas other than the intended target area. The hazard analysis (and subsequent safety case phases) may appear to address the requirements of the standard and the evaluator may acknowledge that the safety case is process compliant with the standard. However, in the absence of proper treatment of post-firing safety the evaluator will find the hazard analysis to be incomplete and thus unacceptable.

The abuse of quantitative risk assessment techniques has long been a concern of the authors. Numbers are often used to hide qualitative assessments on the basis that it helps them to fit into the overall risk management process. However, hiding qualitative assessments behind hard numbers can give them an unjustified level of technical authority — "You can't argue with the numbers." Often the underlying safety argument has little technical merit, safety becoming essentially a "self-fulfilling prophecy."

DEF(AUST)5679 strongly downplays the role of quantitative risks in safety management. There is no explicit requirement for quantifying risks; qualitative safety arguments are allowed (and usually preferred) at every level. This position derives from the soft-

ware focus of the standard and technical inadequacy of quantitative risk assessment for software based systems.

We note in passing that the most widely used military safety standard, being MIL-STD 882C (Department of Defence 1993), is both strongly process-focused and driven by quantitative risk assessment methodologies.

Our comments against the Nimrod Review's six principles for risk cases (*SHAPED*) are as follows:

- *Succinct* — this principle is reflected in the process described in DEF(AUST)5679. It focuses on system safety requirements and their decomposition into component safety requirements. It does not require elaborate flowing down of hazard analysis into subsystems. It uses diagrams to provide a clear picture of accident sequences.
- *Home-grown* — DEF(AUST)5679 stresses the need for End User participation in Safety Case activities and in particular the vital Hazard Analysis. This serves to promote End User awareness of system hazards and ownership of the Safety Case.
- *Accessible* — the safety case summary provides an overview of the safety case, and the safety case documentation should allow for easy searching and viewing of information.
- *Proportionate* — we believe that the process described by DEF(AUST)5679 represents an approach to safety case development that is proportionate to the level of danger presented by the system.
- *Easy to understand* — we agree with this in principle, although we consider the more basic principle to be that simple architectures promote safe systems. That said, the safety case should reflect the actual system. A simple easy to understand safety case for a complex hard to understand system will almost certainly be a wrong safety case. Furthermore, for any system, some of the assurance artefacts will, by their nature, only be understood by experts. The Evaluator's role is to provide independent judgement of the validity and strength of these artefacts in such a way as to be understood by the general reader. The safety case summary should also as a rule be simple and easy to understand.
- *Document-lite* — this is reasonable if the system is not too complex. In accordance with our remarks above, we would prefer to say 'document-focused'.

5 Non-Development Items

Defence procurements often involve what are called *non-development items* (NDIs). These are essentially items over which the supplier of the system has no design control. The use of NDIs, and their role in safety-critical systems, presents a number of complex issues that also arise for Urgent Operational Requirements.

Issue 2 of DEF(AUST)5679 views the use of NDIs as a necessary part of System Development. However, it makes no provision for tailoring or modification of Safety Case requirements for NDIs. The Safety Case is intended to discharge the responsibility under the OH&S Act to "take reasonably practicable steps to avert harm" which is not diminished by a decision to make use of a third party's development effort. It is not acceptable to make prior assumptions of the

safety of NDIs. Similarly, this OH&S responsibility is not diminished by Urgent Operational Requirements (see Section 6).

DEF(AUST)10679 – which provides Guidance Material for DEF(AUST)5679 – includes an Issues Paper on the use of NDIs (Department of Defence 2008a, IGP-004). This Issues Paper discusses the implications of NDIs for safety, with special reference to DEF(AUST)5679. The Issues Paper highlights three cases where Non-Development Items may appear.

- In general, *all* systems will normally make use of *non-development equipment* as part of the implementation technology of a specific component. Non-trivial examples include software components built in the framework of a commercial operating system; disk drives used for logging data etc.
- A specific *non-development component* may be used as part of the overall system design. For example, the PARTI system includes an already developed laser illuminator component.
- The system itself may be a *non-development system*. Examples include:
 - a commercial or military "off-the-shelf" response to a capability gap;
 - a system that was developed for another military context and is to be customised for use in a new operational environment; or
 - an upgrade of an existing or 'legacy' system (this could involve replacement of obsolescent hardware or a modification to software).

The Issues Paper stresses the importance of the hazard analysis phase — no matter what kind of NDIs are used in the system. This theme will be taken up again in the next section. Even for (perhaps especially for) a non-development system, a full hazard analysis must be carried out, identifying and analysing the proposed operational context. The paper then goes on to discuss in detail NDI issues for the safety architecture and design assurance phases.

Notable examples illustrating the key roles played by NDIs in the Australian context are the Collins Class Submarine and the Air Warfare Destroyer (AWD).

The six Collins class submarines are the largest conventionally powered submarines in the world. They are based on the Västergötland class design built by Kockums Marine AG of Sweden. Long-standing issues with the originally envisaged combat system are being addressed by a replacement program using an "off-the-shelf" system (AN/BYG-1) from the US.

The Air Warfare Destroyer exemplifies a modern sophisticated defence platform that incorporates a number of capabilities. It will provide air defence for accompanying ships (as well as land forces and nearby coastal infrastructure), and offers self-protection against attacking missiles and aircraft. The AWD will make use of a special-purpose Aegis Weapon System incorporating long range anti-ship missiles. The AWDs can conduct undersea warfare via modern sonar systems, decoys and surface-launched torpedoes. The existing Spanish Navantia designed F100 class destroyer has been selected as the basis for the Hobart Class AWDs.

Each of these examples illustrates the use of existing designs, significant off-the-shelf subsystems and major system modifications in a complex defence platform.

Systems that involve NDIs present special challenges for the safety case. In particular:

- the safety case for the system may be non-existent, inadequate or developed in accordance with a different safety standard;
- the system may not have been designed and built with a rigorous safety engineering process; or
- there may be limited access to system development artefacts (including assurance evidence).

Nevertheless a safety case must be developed that properly addresses the hazards that arise from introducing the system to its intended operational context.

6 Impact on Safety Case Phases

Having described – at least in the terminology of DEF(AUST)5679 – the structure of the safety case, we consider how Urgent Operational Requirements can or should have an impact on it.

When there is an Urgent Operational Requirement, there might be political or schedule pressure to streamline — or even circumvent — normal safety case activities. However, there is no reduction in the duty of care required by the OH&S Act, so there is an equal need to be able to argue that the system is suitably safe when accepted into service. A safety case must be produced and it must be adequate to make a rational determination of system safety risk.

Beginning from this premise, we consider the vital question: *what reductions in safety case scope might be acceptable in the context of a UOR?* Such considerations are, of course, meta-level ones that are outside the scope of the standard itself. They need to be addressed by the technical regulatory and policy framework of which the standard is part and are finally a matter for the individual acceptance authority. In any case, it is clear that such non-compliance would have to be explicitly highlighted, acknowledged and accepted by the various parties in the safety case.

In the following, we address the implications of various conceivable reductions in the scope of the safety case.

6.1 Hazard Analysis

As discussed above, the hazard analysis phase of the safety case identifies potentially dangerous system behaviour. Of critical interest to those assessing the safety case is the list of accidents (and their severities). The accident list lays out in stark detail how dangerous the system might be. The accident sequences show how these accidents could actually arise from certain system states (hazards).

For the safety case to be adequate to the task of assessing system safety risk, it is absolutely necessary to determine (correctly) the potential hazards that can arise and the severity of the accidents they may cause. No UOR can be sufficiently pressing to justify the acceptance of a safety risk that is unknown.

Hazard analysis is not as onerous as might be thought. It involves a thought experiment by a diverse group of people with sufficient knowledge of potentially dangerous flows from the system, across its boundary and out into the environment. It does need to be done in a systematic way to ensure complete coverage of hazardous interfaces.

Once the severities of the system hazards have been determined, they provide an initial upper bound on the system safety risk. It might be tempting to use this to determine acceptability of the system; however, a determination of safety risk should not be

based entirely on accident severity. An assessment of accident likelihood is required to properly assess system safety risk. A minor accident that occurs with high frequency may be of more concern than a catastrophic accident that occurs with negligible frequency.

In order to address this aspect the operational context must be properly described, allowing the analysis to be further refined by consideration of external mitigations. Once danger levels have been properly assigned to hazards and to the overall system, they reasonably be thought to serve as an upper bound to the system safety risk in its intended operating context. However, this upper bound is likely to significantly over-estimate the system safety risk as the quality and robustness of the system itself have not been assessed and must therefore be assumed to be at the lowest of levels.

Even if this over-estimated system safety risk is assessed as being sufficiently low when balanced against the mission risk posed by the UOR, it may be difficult to argue that this level of safety analysis is sufficient to constitute taking “reasonably practicable steps to avert harm ...”. Generally the acceptance authority will prefer to see argument that steps had been taken to ensure that the system possesses safety qualities and functional robustness commensurate with the identified system danger level.

6.2 Safety Architecture

The safety architecture phase has essentially two components: a safety correctness argument and a safety robustness argument. In response to a UOR, the developer may consider providing a safety case that omits one or the other of these arguments.

First suppose that only the robustness argument is made. This would allow the safety case to identify the internal mitigations present in the system, thus demonstrating that reasonable steps had been taken to make the system safe. This would also allow the strength of these internal mitigations to be used to provide a tighter bound on the system safety risk. However, in the absence of the safety correctness argument it will be hard to defend the technical validity of the robustness argument. Recall that the safety correctness argument demonstrates that the system is architected to be safe to operate when free of internal equipment failure. If the system is not safe in the *absence* of equipment failure, the robustness of system function in the *presence* of failure is cold comfort.

Conversely, suppose that only the correctness argument is made. This provides assurance that the system is architected to be safe to operate in the absence of equipment failure, but it will not be possible to confidently argue a reduced bound on the system safety risk. An understanding of system failure modes and their potential to realise system hazards is critical to assessing the system safety risk.

In summary, the robustness argument is essential to a proper assessment of system safety risk, but it cannot be trusted in the absence of a safety correctness argument. They are complementary activities, mutually informing each other, and both are required to provide a credible assessment of the risk posed by the system safety architecture.

As above, by making worst case assumptions about the quality and robustness of system components, the architecture assessment can be used to determine an upper bound on the system safety risk. Again, at best, this bound remains a significant over-estimate of system safety risk and it is questionable whether the developer can be said to have taken “reasonable steps etc” if appropriate analysis of component design is not undertaken.

6.3 Design Assurance

Having established that the system is architected for safety with an appropriate level of robustness to equipment failure, the design phase of the safety case turns attention to the fitness of individual components for the purpose assigned them by the architecture.

Again, design assurance consists of highly complementary correctness and robustness arguments; so as argued above it is hard to make use of one in the absence of the other.

6.4 Conclusion

We claim (perhaps unsurprisingly) that the body of evidence required in the DEF(AUST)5679 safety case is the minimum needed to provide a credible argument that safety risk has been properly assessed and that the developer has taken “reasonably practicable steps to avert harm to members of the public, as well as their own employees.”

The levels of rigour dictated by DEF(AUST)5679 are perhaps more open to debate and we do not consider them here in any detail. They simply represent a reasonable attempt to provide a mapping between the current range of commercially feasible levels of rigour and system danger levels.

Even if it is considered that the UOR makes lower levels of rigour acceptable, timely safety case development will require the application of a significantly higher safety analysis effort as a proportion of overall development effort. The safety case needs to provide essentially the same body of evidence as for standard acquisition, but over a compressed time period.

7 The Impact of Acquisition Class

Recall the three acquisition classes described earlier: Greenfield Acquisition, In-Service Modifications and Modified Operational Context. Each of these provide different advantages and disadvantages for any attempt to shorten the duration of safety case development. Generally speaking, timely response to UORs is most favoured in circumstances in which significant reuse of existing safety analyses is possible. We consider each class briefly for potential reuse, illustrating our discussion with simple example systems.

7.1 Greenfield Acquisition

For this acquisition class, there is no existing system for addressing the desired capability and hence no existing safety case. Both the system and its accompanying safety case must be developed to meet a pressing deadline.

Clearly, in most cases it will be very challenging to develop a completely new solution to meet a UOR in a timely fashion. For this to be contemplated with significant chance of success, it is likely either that a very simple solution system is envisaged or else that some existing third-party system is known to address the UOR.

In the former case, the simplicity of the system is likely to favour timely safety case development as much as it does general system development. It has often been observed that simplicity is a great friend of safety.

Example: A UOR results in a proposal to develop a new flak jacket based on a novel material. A hazard analysis of the new jackets is likely to focus primarily on the chemical properties of the new material (toxicity, heat resistance, etc) and the ergonomic hazards of the jacket design and it is likely that the safety

case will be relatively small in scope. Such systems as these also benefit from being developed in a highly mature discipline. The science of combat clothing is well studied, with well documented history of use on military operations.³

The latter case essentially devolves to the use of a NDI system. As observed in Section 5, this presents a considerable challenge in the absence of an existing safety case. Producing a safety case for an NDI can require more time and effort than for a bespoke system, even if commercial secrecy does not render it infeasible. The most favourable situation would follow from the NDI being a common consumer level device with few safety hazards or at else produced by an industry with a mature safety culture.

Example: A UOR results in a proposal to make use of commercial tablet devices to gather and communicate military intelligence. Hazard analysis may show that the equipment itself presents few safety hazards. However, depending on the nature of the intelligence and the purpose it is used for, there may be significant safety concerns requiring extensive safety engineering effort to address.

If the NDI system is provided with an extant safety case, the main concerns will revolve around the degree to which the Operational Context of the UOR matches that used in the safety case. The situation is essentially the same as for a modified operational context acquisition as discussed in Section 7.3.

Example: A UOR results in a proposal to procure a commercial bus. Hazard analysis will concentrate on the ways in which the envisaged military operational context may differ from the typical civilian operational context for the bus. If the operational context is essentially unchanged, the safety analysis will be able to depend largely on the civilian safety certification of the bus and may be concluded quickly.

7.2 In-Service Modifications

In this situation we have an existing system, with a safety case that has been accepted, and we intend to modify the system. The safety case must be updated to reflect the modification.

Firstly, we observe that this is a most favourable situation for rapid safety case development. For the contemplated modification to be feasible in a short time frame, it is likely that the scope of the proposed modification is small and much of the existing architecture and design is to be re-used. Often this will also be true of the safety architecture and design, so that much of the safety case is also re-usable.

It is also of considerable advantage if the existing operational context is maintained (we deal with the situation where this is not so in Section 7.3). In this case, it is likely that much, if not all, of the existing hazard analysis remains valid. Even so, it is necessary to reconsider the hazard analysis in a careful manner.

The simplest kind of modification that might be proposed would be the substitution of one piece of equipment with another as it may be possible to reuse the existing safety case almost totally. If hazard analysis does not reveal hazardous properties of the new equipment itself and the modified functionality is not related to component safety functions, then the safety architecture remains unchanged and the component design is changed only in as much as the equipment list changes. For once, the distinction between mission and safety functions may work in favour of speedy safety case development.

³The sinking of the HMS Sheffield by an Exocet Missile during the Falklands War resulted in changes to protective clothing; the synthetic materials worn by sailors were found to melt onto skin, increasing the severity of burn injuries in the victims.

Example: a UOR results in a proposal to swap an armoured vehicle's existing illuminator for night operations with one that provides better performance in harsh conditions. The intent is to improve on performance and reliability. Quite possibly the original illuminator had no direct bearing on the safety case (since it was always regarded as non-development equipment anyway). Thus updating the safety case simply involves re-visiting the hazard analysis (to ensure the higher performance illuminator is not itself dangerous) and noting the change of equipment in the design assurance.

The next step up the design hierarchy is a modification that replaces an existing component in total. Again, although the replacement component may be expected to provide different mission functionality, it very well may retain the same safety functionality. If hazard analysis reveals no new hazards associated with the replacement component, it may be possible that the update to the existing safety case can concentrate largely on design assurance for the new component.

Example: A UOR results in a proposal to improve the ability of the PARTI system (see Figure 2) to illuminate multiple incoming missiles. The existing design makes use of a two laser illuminator component, which has been superseded by a three laser unit. Provided that the individual lasers of the new unit are functionally equivalent, producing the modified safety case may require little more than a re-evaluation of the hazard analysis. A complicating feature of this modification is the fact that laser illuminator is an NDI. The original safety case made use of a DefStan 00-56 (Ministry of Defence 2007) based component safety audit. If the new unit is not provided with similar design assurance data, the re-development of the safety case may be considerably more difficult.

Finally, we note that modifications that involve significant changes to the system safety interface or the safety architecture of a system may require the re-development of significant parts of the original safety case.

7.3 Modified Operational Context

In this situation, we have an existing system, with corresponding safety case, and we intend to make use of it in a different operational context. This can easily be an unfavourable situation, as any change in the operational context has the potential to cause major revision to the safety case and even modification of the safety architecture and component designs. All phases of the safety case could make use of assumptions about the operational context.

Clearly, the operational context is a critical factor in hazard analysis and the accident scenarios have a direct dependence on this context. It follows that if the operational context is modified, then the hazard analysis must be thoroughly reviewed and may need extensive redevelopment. Not only is it possible that new accident scenarios may arise, but existing ones may involve co-effectors that no longer exist or external mitigations that have been weakened or strengthened.

The operational context is also a critical factor in safety architecture analysis. If the architecture correctness argument makes use of properties of the original context that are not present in the new context, it may be necessary to completely re-develop the safety architecture. Of course, the safety architecture may use context properties that are not required by any mission function, so that the need to re-architect for safety may not be immediately apparent when the change in context is first considered. This is especially so where there is little or no safety analysis in early planning.

The operational context may even be a factor in component design assurance. All in all there is considerable potential for a change in operational context to result in significant re-engineering of the system safety case.

Example: A UOR results in a proposal to deploy a PARTI system in a land-based operational context. This project will face considerable challenges due the tight integration of the PARTI system with the ship's CMS, but even if this can be overcome, the heavy reliance of the existing safety case on the ability of the context to define and enforce protected zones for friendly assets will cause significant headaches in producing a modified safety case. Land combat environments are considerably more crowded and less structured than sea environments.

8 Final Remarks

In this paper, we have discussed Urgent Operational Requirements and the impact that they may have on the safety case. We believe that the threats to the safety of personnel and civilians arising from the installation and use of Defence equipment remain of paramount consideration. No UOR can be so urgent as to warrant ignoring safety issues or failing to properly assess them. Those accepting systems into service need to properly understand the safety risks associated with a system if they are to properly weigh them against the UOR and mission goals. We do recognise that, in a combat situation, commanders frequently put their personnel at risk in order to achieve mission goals; in particular, a commander in the field can make a command decision to override or ignore a safety-related issue or procedure.

While a system need not be safe in some absolute sense when adopted into service, a clear and accurate assessment of safety risk is a critical input to good decision making. Many of the properties of the system that have bearing on safety (reliability, robustness, operation context) are also critical to the operational success of the system. Thus, the effort put into understanding the system from a safety viewpoint may even serve to improve the operational outcome more generally.

Armed forces are likely to become more and more required to deploy rapidly in different regions of the world and to adapt quickly to the conditions that they face. Thus, there will be increased pressure for rapid acquisition of capabilities. Those responsible for developing safety cases need to have robust and efficient processes for carrying out hazard analysis and other safety case phases. They must be steadfast in analysing and highlighting safety risk to decision makers — especially when “reasonably practicable steps” have not been taken to avert harm.

References

- Department of Defence (1993), System Safety Program Requirements, Military Standard MIL-STD-882C, United States of America.
- Department of Defence (2008a), Guidance Material for DEF(AUST)5679/Issue 2, Australian Defence Handbook DEF(AUST)10679/Issue 1, Australian Government.
- Department of Defence (2008b), Rapid Acquisition of Capability, DI(G) LOG 4-1-008, Australian Government.
- Department of Defence (2008c), Safety Engineering for Defence Systems, Australian Defence Standard DEF(AUST)5679/Issue 2, Australian Government.
- Department of Defence (2009), *The Strategic Reform Program 2009, Delivering Force 2030*, Australian Government.
- Haddon-Cave, C. (2009), *The Nimrod Review*, The Stationery Office Limited UK.
- Mahony, B. & Cant, A. (2008), The PARTI architecture assurance, in 'Proceedings of the 13th Australian Conference on Safety-Related Programmable Systems', ACS.
- Ministry of Defence (1983), Design and Airworthiness Requirements for Service Aircraft, Volume 1 - Aeroplanes, Defence Standard 00-970, United Kingdom.
- Ministry of Defence (2007), Safety Management Requirements for Defence Systems, Part 1 Requirements, Defence Standard 00-56, United Kingdom.
- Ministry of Defence (2011), *Commercial Guidance for the UK MOD Defence Acquisition Community: Urgent Operational Requirements*, United Kingdom. <http://www.aof.mod.uk/aofcontent/tactical/toolkit/>.
- Tyler, A. (2009), 'Performance under pressure', *The RUSI Journal* 154(5), 30-33.

Establishing Safety Case Strategies for Mission Planning or Situational Awareness Systems

BJ Martin

Nova Systems
System Safety and Technical Airworthiness
Competency Lead
Canberra,
Australia

bj.martin@novasystems.com.au

Squadron Leader Derek W. Reinhardt

Royal Australian Air Force
Deputy Senior Design Engineer – Avionics C-130H/J
Air Lift System Program Office (ALSPO)
RAAF Richmond, New South Wales
Australia

derek.reinhardt@defence.gov.au

Abstract

Mission Planning and establishing Situational Awareness are important risk management strategies in complex and hazardous military aircraft operations.

Software based Mission Planning Systems (MPS) and Situational Awareness (SA) tools supporting operational decision making in circumstances that impact safety are now common place, and are becoming increasingly functional.

Operational approvals for such systems are typically based on satisfactory technical specification compliance and user trials with criteria of: effectiveness, workload reduction over manual methods, sufficiently intuitive interface, verified outputs for selected operational test cases; and qualified user workforce.

However, a conundrum remains for the structure of the system safety case argument, which would, in safety-related software theory, rely heavily on technical design assurances. The origin of many of the software tools forming part of a MPS is sometimes outside the environment where high integrity design assurance practices are common place. Often referred to in system safety literature as Software of Unknown Pedigree (SOUP). In this situation, the determination of a safety criticality / integrity level or hazard analysis activities do not typically drive system design requirements or design assurance activities. Therefore there are often substantial limitations in design development artefacts or other evidence that the software's integrity is likely to support the determination of safety criticality.

Instead, from consideration of instituted MPS and SA tool approvals processes, it may be construed that system Human Machine Interface (HMI) look-and-feel evaluation and user operational procedures are largely responsible for achieving adequate operational safety. Yet, rarely are effective human error or critical task analysis activities employed for these tools and functions, nor are workload assessments used to validate in-mission operators abilities to detect and correct errors before mishaps occur.

Examination of the limited literature or case studies identified of notable mission planning or situational awareness system related accidents, appears to weigh strongly towards user input or data related failures, and errors in correct system use due to incorrect initialisation or inadvertent reversion to default data values. These factors may be attributable to both technical and operational procedure design issues, although in some circumstances the causal factors have heavily favoured one over the other.

Where then, should the strength of argument and emphasis of safety case resources be invested for maximum safety return? What is an effective safety case assessment methodology for MPS or SA systems approvals?

This paper examines the current use of Mission Planning Systems, related accident history and causal factors, current regulatory requirements, and proposes a basis and methodology for architecting the safety case for MPS and SA systems.

Keywords: Mission Planning Systems, Electronic Flight Bags, Situational Awareness, Human Factors, HMI, Safety Case Argument.

1 Introduction

Safety certification of highly integrated technologies intended to perform a pro-safety service, and a bridging function between planning operations and actually conducting safe operations, can be, in the author's experience, a vexed subject among safety engineers and operators alike. The technology in question does not directly control any hazardous energies, or directly cause mishap consequences when it fails; the usually drivers for safety integrity. However Endsley [EBJ03], Sandom [SaFo06], [San07] and Storey [StFa03] (among others) have published extensively on the relationships between SA, Information Systems, Data and safety. They argue that breakdowns in the functions these Mission Planning

Systems and Situational Awareness (from here on 'MPS/SA') tools facilitate, can provide operators with misleading decision support data, which is frequently attributed as a major contributor to accidents.

The introduction of powerful computing and display tools for aviation MPS/SA functions has been gradually underway for nearly 20 years via the evolution to 'data-hungry' flight management systems. The first significant regulatory guidance on tools that support these functions was FAA AC120-76A [FAA03] for Electronic Flight Bags. [FAA03] proposes a traditional aerospace functional safety assessment approach, but only if the system was physically integrated into the cockpit. As with much emerging technology, the market was leading the regulators by several years, and there is much anecdotal evidence to support the notion that regulation is still playing catch-up. Thus, such technologies are yet to be heavily influenced by regulatory requirements, and operational approval authorities are left without a consistent understanding of the technical and operational risks presented by the products on the market (ie. there is no emergent and widely agreed technical certification basis for these MPS/SA tools).

Using primarily an aviation experience basis, this paper will examine the connection between MPS/SA tools and safety outcomes by reviewing noted accidents, incidents and analysis, exploring the contributors and comparing this to the relative emphasis and care given to these factors in development, introduction to service and regulation.

The authors have had involvement in the introduction to service and attempts to create safety case arguments for several MPS/SA and related critical data handling tools. The author's observation is that while technical regulation is essential, much more work needs to be done for the integration of the technical evidence and arguments with the operational approval activities, in order to safely assess and operate these capable tools. This is vital to an overall safety argument architecture and safety case, and ultimately to ensure safety and benefits are realised by the employment of the MPS/SA tool.

The intent of the paper is to identify where the greatest returns for investment of effort are, in developing strong safety arguments for mission planning and situational awareness tools. It will discuss proposed methods of establishing arguments for safe operational release. The case is made for aviation applications, but read across for similar technologies by other safety critical industries will be possible. This paper has not, however, developed a honed methodology or cook-book for a complete safety assessment of MPS and SA tools. Peer validation of the strategies proposed here, further research and experience by application will be required.

The goal is to propose alternative pragmatic strategies for practice in this area, with well reasoned dependence on both technical and operational approval activities and evidence.

2 Background

To provide background to the human and technical factors examined by this paper, and their explicitness in the proposed safety argument, this section examines the MPS/SA tools, how they are intended to be used, how they have failed, and what this might imply for the safety case.

2.1 Mission Planning, Situational Awareness and Safety

The linkages and importance of the acts of planning and provision throughout of situational awareness, to flight safety, is largely accepted wisdom. Locally [CASA11] promotes it as follows - 'planning is important because it constructs a four-dimensional picture of the flight in your mind'. That is, the act of planning builds a foundation for situational awareness. However, the degree of automation and complexity of the modern aircraft means that much of that situational awareness now resides in a machine. Human operators are instead presented with an emerging (but not necessarily intended) paradigm of merely following directions or monitoring automation.

Examples of this paradigm shift in the Australian Defence Force (ADF) context is reflected in the F-111 controlled flight into terrain accident in Malaysia in 1999 and the B707 crash near East Sale in 1991. Flawed mission planning and risk management were major contributors to these accidents, and the resulting investigations made recommendations for significant changes in regulating, and measurably improving, ADF operational airworthiness (safety).

While the paradigm of merely following the machines directions does offer some benefits in many circumstances, the operators 'ignorance will become a problem if the machine stops'. Recent operational safety studies in Europe [Lea10] assessed helicopter safety incidents to identify major opportunities for improving safety. The studies identified four of the most insidious technical/operational hazards including: unexpected encounters with a degraded visual environment; getting into a "vortex ring state"; loss of tail rotor effectiveness; and static and dynamic rollover. A major contributing factor (or rather - weak defence mechanism) was inadequate mission planning, and the resulting shortfalls in situational awareness. These factors have become a focus of training leaflets being distributed by the European Helicopter Safety Implementation Team.

To further illustrate the criticality of effective mission planning, and the important role the MPS/SA tool has in this process, Section 2.3 of this paper will examine specific aviation accidents and incidents. It will identify major contributing factors and will also illustrate challenges of new technology introduction, that brings both opportunities for improvement and new sources of hazards.

2.2 Use of MPS and SA tools

In aviation, trusted rules-of-thumb have always evolved built on lessons learnt, (eg. fuel reserves/flow rates,

Visual Flight /Instrument Flight Rules or Extended Twin Operations diversion limits, Lowest Safe Altitude and buffers for predicted performance/weather inaccuracies, etc.). Calculation tools have been used to support the planning function (eg. performance reference charts, whiz wheels, TOLD cards, and numerous home grown spreadsheets). This has been a natural response in order to reduce time and simple error prone, tedious and repetitive arithmetic. Sources and types of errors that might come from these methods and tools are generally well understood, the subject of explicit training, and are commonly the subject of flying supervision and pre-flight authorisation checks. However, like many aspects of modern life, the proliferation of automated tools is tending to drive complexity beyond the comprehension of most operational users. Undeniably though, the intent of adopting tools has been to improve safety and reduce incidence of hazardous simple errors; but it also creates the predicament of unknown, or at least under-appreciated, potential for error.

In aviation, modern MPS consist of software applications that allow maps, charts, weather, intelligence and aircraft performance data to be used in developing navigation solutions (e.g. routes, approaches, terminal procedures), communication settings, flight/mission calculations (fuel, leg times, etc), and other pertinent aircraft operational data. MPS may include visual software tools, optimised for specific aircraft roles, and automate the computations associated with aircraft specific flight/mission planning. Once the mission information has been generated, it is printed (e.g. kneeboards, strip charts), or alternatively written onto a data storage device (e.g. PCMCIA flash disk, or proprietary data transfer module) for transfer to aircraft systems (e.g. Flight Management System, navigation system, Electronic Flight Bag(EBF)). For the purposes of this paper, when MPS output data is combined with displays, and are used as a live updated decision support reference during operations - it is performing a Situational Awareness function. Some more recently developed MPS also include functions to transmit and receive flight/mission information via datalink, either at the commencement of a flight/mission, or as real time updates throughout a flight/mission. MPS may also be used for post-flight/mission debriefing and analysis. Note that while the FAA do not use the term MPS, preferring Electronic Flight Bags (EFBs) to describe these applications, the ADF use of the MPS term encompasses both flight planning and aeronautical database processing.

Aeronautical Data, which underpins many of the functions of the MPS tool, is provided to the ADF by the Royal Australian Air Force Aeronautical Information Service (RAAF AIS), and to civil operators by Air Services Australia. These agencies are charged with regional responsibilities for military and civil users, providing aeronautical data in electronic and paper form for planning, en-route reference and critical terminal procedures. How this data is integrated into an aircraft automatic flight management functions and used in operational procedures, governs the criticality of the mission planning system.

Similar parallels also exist in the maritime and land domains, such as the Electronic Charting Display Information Systems (ECDIS), integrated ship Navigation systems (eg. ECPINS), and Battlefield Command Support System (BCSS) to name a few.

In commercial maritime operations, similar applications in Electronic Charting Display Information Systems (ECDIS) and integrated ship Navigation systems (eg. ECPINS) have become more common place over the last 10-15 years for safety and economic reasons. Meanwhile the military has been adopting the same technology for arguably higher risk and more dynamic military operations planning for surface and now sub-surface applications. The drive for integrated automation and error reduction has also extended to join the digital dots between the source of the Navigation and operations planning data and the users. In Australia the Hydrographic Office, is responsible for charting and distribution of all Australian Territorial waters and additional areas of military interest. This service has transitioned in recent years from a historically evolved cartographic drafting service, reliant on evolved knowledge and craftsmanship, to a Digital Hydrographic Database importing survey data from a combination of survey tools (Laser Airborne Depth Sounding, Survey ship digital soundings, and more manually collected then hand recorded into digital devices). This survey data is then collated and classified by the hydro data and charting specialists.

Irrespective of the domain the data is received, manipulated, created and transferred via various information systems and geospatial software tools linked together by an automated workflow, and tailored to meet international presentation standards as well as some client specific format requirements to suit particular MPS and SA tools used in operations. Tool developers of multiple application platforms will typically not be aware of the specific user operational context and criticality.

In the land operational environment, planning and control systems such as Battlefield Command Support System (BCSS) and other generic Battlefield Management Systems also rely on combinations of digital terrain data, scanned traditional topographic maps, live 'blue-force' tracking, mission planning overlays, dynamic intelligence data etc.

These non-aviation examples are described here merely to illustrate that the technical and conceptual issues to be discussed in this paper from an aviation perspective, will have applicability in other domains where safety and mission critical decisions will be made based on data presented and manipulated in integrated planning tools.

2.3 Examples of Accidents involving MPS and SA tools

A recent ATSB Report [ATSB11] catalogues and analyses 11 Australian and 20 International civil high capacity transport aircraft accidents and incidents over a 20 year period (Jan89-Jun09), where mission planning and data errors were involved. Three Examples will be reviewed briefly here to illustrate the scenarios:

2.3.1 Emirates A340 Melbourne - Mar2009

The following summary is based on the preliminary results of the ATSB's ongoing investigation, released on 18 December 2009.

On 20 March 2009, the crew of an Airbus A340-541 aircraft arrived at the aircraft about 1 hour before the scheduled departure time. About 30 minutes later, they received the final loadsheet, with a Take Off Weight (TOW) of 362.9 tonnes. Shortly after, the first officer entered a TOW of 262.9 tonnes into the Airbus Less Paper Cockpit (LPC) electronic flight bag system. The first officer recorded the resultant figures on the flight plan and handed the LPC computer to the captain for cross-checking. The captain checked the take-off performance figures and entered the figures into the flight management and guidance system (FMGS). The captain's figures were then cross-checked with the figures recorded by the first officer.

During the takeoff, the captain and first officer attempted to rotate the aircraft, but it did not respond. They tried again applying a greater nose-up command. The nose of the aircraft raised and the tail made contact with the runway. The aircraft did not begin to climb. The captain selected TO/GA thrust and the aircraft commenced a climb.

After establishing a positive climb gradient, the crew received a message from the on-board error system indicating a tailstrike. The crew notified Air Traffic Control (ATC) and advised that they would be returning to the departure airport. While reviewing the aircraft's performance documentation in preparation for landing, the crew noticed that a TOW 100 tonnes less than the actual TOW had been inadvertently entered into the LPC, resulting in low V speeds. At no times during the process did the LPC or on-board systems challenge that the TOW might be incorrect.

2.3.2 MK Airlines B747

On 13 October 2004, a Boeing 747-244SF aircraft, registered 9G-MKJ, was planned to operate a multi-stage non-scheduled international cargo flight departing from Luxembourg, through Bradley and Halifax, Nova Scotia.

The aircraft was taxied to the runway and during the takeoff the aft fuselage momentarily contacted the runway. Several seconds later, the fuselage contacted the runway again with greater force. Contact with the runway continued to about 825 ft beyond the end of the runway, where the aircraft became airborne. The lower aft fuselage then struck an earth bank supporting the instrument landing system antenna and the tail separated from the aircraft. The rest of the aircraft continued forward until it struck terrain. The aircraft was destroyed by the impact forces and subsequent fire. All seven of the crew members received fatal injuries.

The following factors were identified throughout the subsequent investigation:

Flight data recorder comparison

The flight data recorder information for the take-off at Halifax was compared with the takeoff at Bradley to identify any similarities. This comparison identified that the rotation speed and flap setting for both flights were about the same, however, at Bradley the aircraft reached rotate speed 13 seconds before that recorded for the Halifax takeoff, indicating a higher rate of acceleration. Furthermore, the initial pitch rate for the Bradley takeoff was 1.2 degrees per second and the aircraft climbed away about 4 seconds later, with the pitch angle increasing to 6 degrees. For the Halifax takeoff, the initial pitch rate was 2.2 degrees per second, with the aircraft lifting off near 10 degrees. This eventually increased to 14.5 degrees.

The take-off data for Halifax was identified as being nearly identical to that for the takeoff at Bradley, indicating that the Bradley TOW (239,783) kg was used to generate the performance data for Halifax. The calculated TOW for Halifax should have been 353,800 kg.

Boeing laptop tool (BLT)

In order to calculate the take-off performance data, landing performance data, and weight and balance information for a flight, the crew were required to use the Boeing Laptop Tool (BLT), which was located on the upper deck of the aircraft.

It was likely that the use of the wrong TOW came from the misuse or misunderstanding of how the BLT software functioned. When the BLT program was launched, the data for the previous flight would populate all of the fields, in this case, the data for Bradley. These fields would then need to be updated with the data for Halifax. If the user opened up the weight and balance page, and then returned to the take-off performance page, the TOW already in the system would automatically populate the planned weight on the take-off and performance page, which was 240,000 kg for Bradley. If the user was unaware of the software's reversion feature or did not notice the change, and they selected the 'calculate' button, the resulting V speeds and thrust settings for the takeoff at Halifax would have been based on the data for Bradley. If these figures were written on the take-off data card with the correct TOW of 353,300 kg, it is likely that the error would have gone unnoticed.

Other factors identified

It was likely that an independent check of the take-off data card was not performed by the crew as required by the standard operating procedures (SOPs). The crew did not conduct a gross error check in accordance with the SOPs. The crew were at their lowest level of performance due to fatigue, which may have increased the probability of error when calculating the take-off performance parameters, and degraded their ability to detect the error. Crew fatigue and the dark take-off environment contributed to a loss of situational awareness. The airline did not provide formal training on the use of the BLT, nor did they have a proficiency program.

2.3.3 Southwest Flt 1248

After deciding it was safe to land in a snowstorm, the pilots of Southwest Airlines Flight 1248 overran the zone where the plane needed to touch down, resulting in a runway overrun. The result is that it skidded outside the airport and killed a 6-year-old boy who was a passenger in a proximate motor vehicle. The pilots needed at least 800 more feet of runway to avoid a collision, according to the National Transportation Safety Board (NTSB).

As they approached the airport the pilots and a Southwest dispatcher were confident a landing could be accomplished, despite contending with low visibility, a tailwind and reports of poor braking power on snowy Runway 31 Center. The pilots based their decision to land on the dispatcher's positive assessment, their piloting experience and flight data they entered into a cockpit computer. The onboard computer confirmed the difficult landing would be within the capability of the Boeing 737-700 and would conform to Southwest's procedures.

Flight crew used on-board laptop performance computer (OPC) to calculate expected landing performance. The OPC was programmed to assume that engine thrust reversers will be deployed on touchdown in its calculation of the stopping margin. The calculated stopping margin was acceptable to the aircrew. If the OPC did not use reverse thrust credit, it would have indicated that a safe landing on 31C was not possible. It's unclear whether the crew were aware of the assumptions in the OPC calculations. The NTSB now prohibits operators from using reverse thrust credit in landing performance calculations.

2.3.4 Other related accidents

The aviation accident record is focussed on heavy transport aircraft, where reliance on data is tightly coupled to hazardous and automated phases of flight. Other incidents related to similar causal factors and mishandling of automation include: Ryanair in 2006 [Lea06] a Controlled Flight Into Terrain (CFIT) accident was narrowly avoided after crew became fixated on reprogramming the automation via the Flight Management System (FMS) after the discovery of incorrect/outdated data for the airport they were approaching at Knock, Ireland. Another example of a more purely navigational data based accidents are the 1995 Cali B757, Flt 965 [Lad05] crash into mountainous terrain due to erroneous waypoint assignments by the crew when they lost situational awareness, were under time pressures, independent data validation sources had failed and they were subjected to arguably poor HMI design in the FMS.

2.4 Accident Reporting and Analysis

[ATSB11] states that it's major findings corroborated previous findings from US NTSB and studies by Boeing and Airbus and were essentially that all identified incidents had causal factors associated with input errors, poor or non-existent gross error validation practices, time pressures, workload and/or poor coordination and communication within the crew or with external parties

(ATC). These were seen by accident investigators as failures to create adequate procedural defences to error, and failure to recognise and react to abnormal performance aircraft indicators.

ADF Aviation Safety Spotlight Magazine 04/2010 [War10] picks-up this ATSB report and themes in "Deadly Data" and correlates to ADF experience with similar "safety factors" at play, in similar heavy transport operations. With the added complication of military transport involving more dynamic tasking but also subject to time pressures driven by operational imperatives and other safety factors, such as dangers to passengers in hostile zones if not airlifted etc.

The FAA have also reviewed specific incidents involving Electronic Flight Bags at [Chandra10] to identify some common threads and similar causal factors (ie training, familiarity, over trust without validating) but further drew out certain fundamental design features that were co-contributors to incidents and hazard scenarios. Particularly where the equipment was operated during the flight and involved crew workload/distractions from fundamentals of aviation because of legibility and manual manipulations, workload required to pan and zoom and allowing important context data to be missed. Interestingly, this very issue was the primary contributor with a submarine grounding incident ([Per05], [Ham05]) where recently charted hazards had been updated on some resolution electronic charts but because the operator was using a lower resolution, the boat navigated into shoaling waters at high speed, despite active soundings cautioning the crew otherwise. Closer to home we read Newspaper stories of similar things happening regularly with over-reliance of drivers on SatNav directions in road vehicles.

[Sei11] and [FSF05] sites a current study collating NASA Aviation Safety Reporting System (ASRS) data where a general concern has been raised over reductions in pilots' manual flying skills, possibly from an over reliance on automated systems, as well as an incomplete understanding of such computerized controls, planning tools and aircraft operating modes.

Avionics Magazine [Evans06], reviewed the celebrated example of the October 2004 MK Airlines incident in Canada. The planning errors were on an EFB tool, but are no different from those carried out on ground based Mission Planning Tools. The article criticises the design of the Boeing Laptop Tool (BLT) (and essentially it's cousin products from other manufacturers that are no 'smarter') for not including design features that made modes and data manipulation actions more clearly understood by the crew, and checking for gross or non-sensical errors. It also has promoted HMI concept schemas that would help in this role. Finally the author advocates mandatory subjection of the laptop tools to the *"same robust validation required of flight control software. At the end of the day, both are equally capable of killing"*. This last statement in his editorial draws a much closer link between mission planning and catastrophe than is currently supported in regulations. The flaws were not system functional failures or erroneous behaviours. The failures identified were essentially in the requirements set, not having identified

sufficient operational hazards and HMI challenges. So in the absence of systematic hazard analysis requirements in the regulations, success would have to depend on how operationally savvy the developers and testers were and whether they were testing for operator error potential and the validity of their requirements, both in normal and failure modes. This is not a common strength of the software development industry.

Several articles in aviation safety journals have focussed on the importance of recognising the training liability/burden of introducing MPS and SA tools in order to minimise human error and capitalise on the safety (and economic) benefits of such systems. This is a well supported focus from the analysis of causal factors above. But aren't training and procedures the last refuge of the system safety scoundrel? In the MIL-STD-882C design mitigation order of precedence, we are supposed to deal with eliminating the hazard and providing safety features first.

2.5 Summary and Assessment of Contributing Factors to Accidents

All of these analyses support an obvious conclusion that better training and understanding is required in the automated systems and the vulnerabilities of the human and procedural interface. But is this sufficient or even practically maintainable given innate complexity and the pace of change in systems software upgrades? Is it a case of more sympathetic design to human responses and mental states?

Notably, none of the incidents and enquiries examined in the referenced reports, identified faults in the calculations performed by either planning devices or flight management computers. Although it has been suggested that these system human machine interface could have been designed more sympathetically and robustly, to identify and flag non-sensical or inconsistent planning data inputs and outputs. (ie. they could have been designed to add to the defences to assist error detection in critical tasks, if this operational criticality had been understood by the designers). It cannot be concluded that there were no software faults in those equipments used, but it is clear from the above analysis that tool faults (in requirements satisfaction) did not play a profound role in the accidents at hand.

In the absence of consistent application of airborne software standards and certification requirements for MPS design, what value do "robust validation" and operational testing provide? How reliant has current aviation EFBs been on a level of product quality that comes by default from existing trusted avionics suppliers, rather than on any demonstrated achievement of safety goals? As new software development sources and competition from cheaper providers come onto the market, how will regulators evaluate when the integrity of the design is insufficient?

The following section will now review what the current regulatory requirements are, and discuss how they are currently applied.

2.6 Current Civil and Military Regulatory Requirements and Application

The FAA's AC120-76A was released in 2003 providing guidance for certification, airworthiness and operational approval processes of EFBs. It combined facets of existing regulation of airborne avionics device compliance and safety, with the results of sponsored Human Factors research by the Volpe National Transportation Systems Center [Cha03]. In brief, the requirements are graduated by classification of hardware (Classes 1, 2 and 3) and software (Types A, B and C) by features and functions with increasing approvals requirements and rigour as the system becomes more integrated into live operations as a reference and decision support tool. Physical cockpit integration is a primary indicator of safety criticality in this guidance material. As such the AC did not require functional hazard analysis (ala FARx.1309 system design and analysis requirements) of EFB systems unless they are a Class 3, permanently installed device. However, FAA inspector operational approvals include minimum requirements for training, currency and checking, operational and data update procedures, regardless of hardware or software classification, with increasing objective evidence requirements as the criticality increased. The minimum standards required for these operational procedures supporting EFBs installation, is very conceptual and subjective. The accident record collated by the ATSB and other cited examples, seem to indicate that consistent application of the principles and intent in AC120-76A are not yet common place or fully effective.

The ADF technical airworthiness regulator, DGTA, has considered its approach to EFBs and MPS for a number of years while dealing with the more gradual integration into modern military aircraft. The military regulator has also had to consider the broader context of integration of MPS and data input/output into military theatre operations planning systems. Preliminary guidance had been available for design requirements and Technical Regulatory compliance since 2004 and in 2008 DGTA (including one of the authors of this paper) published a Notice of Proposed Rule Making (NPRM) for EFBs [DGTA08] and later for the broader scope of MPS in Dec'09 [DGTA09]. The EFB certification guidance considered AC120-76A applicable in most cases of common mission planning system functions, however the discriminator of criticality of function of mission planning systems was to be dictated by the level of automation and risk associated with the operational roles of user aircraft. This, then identified a need for further guidance of how to establish a basis for judgement of criticality. The MPS NPRM directed that aeronautical data and its intended use was this discriminator. Depending on whether the aircraft and crew would rely solely on the data correctness for safety critical functions or decisions, would dictate the criticality of the MPS functions of generating, manipulating and transferring this data. The systems carrying out these functions – software, hardware and human – inherited this criticality and responsibility for integrity. (This concept of data

criticality will be explained further in Section 3.1 of this paper.)

On the operational regulatory side, however, there are limited ADF requirements or guidance published. The technical regulations outline some operational safety management considerations, but operational approvals are not equivalently or explicitly regulated as they would be for the FAA requirements on EFBs. Military Aviation Regulation 6 [ADF09] is currently interpreted to consider EFBs and MPS as classes of Aviation Support Systems, and thereby requires classification, requiring a form of technical approval and an Operating Permit. The requirements of a basis for an Operating Permit are less explicit in terms of assessment, procedures, currency and approval requirements and not specific to mission planning systems.

CASA's published guidance appears in a very recent publication of an Airworthiness Bulletin [CASA10] which would essentially indicate that the FAA view, requirements and comprehensive approach should prevail.

Alas (and anecdotally) - a recent unattributable presentation on an approach being taken by a low cost airline to 'paperless cockpits', was witnessed at an Aviation symposium in Australia. This presentation underscored the naïve but perhaps understandable approach possible in this immature area of the power of emerging technologies. Automation may be sought as a business solution in order to simply reduce operating costs, without foreseeing a safety implication of the negative sides. In the subject presentation, an in-house developed set of calculation tools, hosted on a commercial mobile computing device, were developed to reduce perceived overhead, improve planning speed and on-time departures via flexibility for re-planning in the cockpit, and reduce take-off settings to minimum margins. According to the presenter, acceptance of the new system was to be based on a judgement of minimum negative feedback from operating crews in trials, and having achieved local CAR35 signatory approval for carriage of the device based on no physical interaction with the aircraft systems. The fact that this approach had reached implementation trial stage in a public transport carrier, indicates a worrying absence of understood technical and operational approval requirements based on functional criticality, or at least a basic lack of awareness of those regulations that should apply.

In summary, it seems that regulations and guidance in the aviation sector with regards MPS approvals is currently available but fluid and dispersed. It is also clear that the technical approvals basis is more rigid, and primarily driven by considerations of physical interface to the operating platforms rather than consideration of the functional interface. The exception to this is where data criticality is being proposed as a discriminator, such as the ADF draft design requirements for MPS. Even in this case, the data criticality discrimination, is being used as a mechanism to drive software assurance requirements (not yet a substantial contributor to the accident record) and not human factors assessments or regulated operational approvals.

2.7 State of practice for certification of MPS/SA tools

Despite the recent emergence of the certification requirements discussed above, many MPS/SA tools in use haven't typically been developed to these frameworks, or have ignored normal airbourne system certification requirements. Further the incidents and accidents record suggest that at this time the frameworks are yet to be totally effective.

MPS tools are not, in totality, considered as aircraft software, and thus they may not be subject to normal aircraft software certification requirements. For example, as described in the section 2.5, the FAA approach to Electronic Flight Bags only prescribes full software assurance certification requirements for specific types and functions of tools (e.g. Type C tools), while limiting software certification requirements for other types of tools (e.g. Type A and B tools). This approach is pragmatic, but it also means a wide range of tools are outside traditional aircraft software certification requirements. Hence, it is relatively common practice of not subjecting such tools to safety and design assurance practices commensurate with airborne software.

So why does a certification authority take this approach? One reason is that many of these tools offer significant improvements to pilot planning efficiency and situational awareness (hence safety dividends). The other is that for many of these tools, the worst credible hazard may only be as severe as Minor when appropriately incorporated into normal cockpit procedural practices. However, when it comes to the application of such policy, developers of these tools may not always understand these underlying assumptions behind the policy. Hence it is common place to see developers promoting products for which very limited certification evidence exists. This practice exists because often some developers are ignorant as to the certification requirements, and are focussed on the perceived benefits of the tool. For example the laptop, netbook and iPad evolution the proliferation of software applications on these platforms has lead these developers to explore applications across a wide range of domains outside the conventional IT domain. Further, the developers are likely not to have undertaken to fully assess the hazards associated with the use of the tool, or have blindly made underlying assumptions that there will usually be a human operator in the loop, and this will provide sufficient mitigation to all classes of errors, faults and failures under all circumstances. Yet, rarely is there any evidence of such an assessment. Very recently news items were posted at [Jep11] that an iPad application has received EFB certification by the FAA at Class1 level, with aspirations for Class 2.

In the military environment, explicit regulation of certification requirements for such tools has lagged somewhat. The ADF's NPRM on Flight and Mission Planning Systems [DGTA09] is still one of the first military authorities to publicly issue certification guidance for MPS and EFB tools. While this guidance is derived from the FAA approach mentioned earlier, it is adapted for military operational circumstances, and

heavily technically focussed, as technical and operational regulation is a separate activity in the ADF.

The most common MPS in use with the ADF are the Portable Flight Planning System (PFPS) and the Joint Mission Planning System (JPMS). The dependability of these products relies substantially on the retrospective Verification and Validation undertaken by the USAF and USN respectively, rather than the prescription of development assurance practices. One of the author's on this paper has direct experience of studying the qualification and release processes, where (for example) the USAF assigned test verification agency is manned with *several hundred* personnel dedicated to testing, verification, validation and support of the system PFPS for every operational platform in USAF service (including FMS export aircraft types). This agency undertakes a substantial V&V, including regression program for each PFPS build delivered. While this approach does not mirror the application of software assurance principles of recognised assurance standards such as the coupling of ARP4754/61 with RTCA/DO-178B, Def(Aust) 5679, and Defence Standard 00-55 (now obsolete), it does contribute evidence to the safety case with respect to confidence in PFPS's behaviours.

Several legacy ADF MPS acquisitions have also relied heavily on one-time-only ADF conducted Verification and Validation to provide some assurance against hazardous behaviours (e.g. the Mission Data Preparation Equipment software for the now retired F-111 aircraft). For example, when MDPE was being accepted by the RAAF, one of the authors of this paper was personally involved in the V&V of the MDPE software consisting of many thousands of V&V cases of the software, including functional, robustness, and crew procedural requirements. This V&V was undertaken over the period of 4-6 months. The fault density in earlier versions of MDPE was relatively high, however, the V&V effort certainly contributed to the subsequent resolution of many of these issues that might have provided an opportunity for a hazard to the crew. Again, this didn't constitute standard software assurance practices that would see this evidence generated during the development of the product, nor is this approach being currently advocated by ADF regulation. ADF's resources to achieve this are not the same as in the mid-90's when much of this work was undertaken. Nonetheless it did contribute to the case for acceptance/employment of MDPE. Still, for other ADF aircraft, assurance of mission planning systems have been almost completely overlooked in the development stages. In either case, the V&V effort possible today in Australia pales by several orders of magnitude to that nominally provided to PFPS and JPMS as a matter of course.

To summarise - assurance practices have had very limited application to MPS/SA tools developed to date. In some cases V&V has been used a means of shoring up the design evidence shortfall, but this is becoming less practical in the current Defence funding climate, and not practical in a competitive airline environment. Therefore, achieving assurance of MPS/SA tools requires a more holistic approach.

3 Case for Data and Design Integrity

Section 2.3 has summarised several hazardous aircraft circumstances related to MPS tools. Of note, most of these were largely the result of operator errors and misunderstandings of the tool's results, however the integrity of the underlying data, and of the tool itself is an essential input to safety. For example, there is evidence of tools providing invalid results that were interpreted to be valid by human operations. This is potentially a shortfall by the human in interpreting this information, but also a requirements validity issue with the tool if invalid data can be interpreted as valid data. There is also evidence of tools invalidly using stale data in calculations. Further still, there is evidence that the workload impost on human operators working with these tools (particularly those used in flight) resulting from unexpected or unintelligible behaviours of these tools is also a factor. All of these circumstances are evidence of potential shortfalls in the integrity of the respective tools, albeit they are concerned more with requirements validity than with latent faults in the specified implementation. For the purposes of this paper, integrity is a qualitative term used to infer the degree of confidence that the software's behaviours are valid in both normal and failure modes of the software and that the behaviours that may impact safety satisfy an explicit and valid requirement for that behaviour. Therefore, software integrity is not just the isolated application of software assurance practices, but the application of software assurance in the context of allocation/derivation of requirements for the software from safety analysis of the system, the software and the operational context.

Further to these more obvious factors, there is another factor to consider which doesn't get tied directly back to the individual circumstances surrounding just the way these tools have failed in practice. There is a broader question of when the tool may be found to be a contributor to an accident, what will the investigation recommendations (hence opportunities for litigation) be targeted at. The authors' view is that accident investigations will often make recommendations for explicit behaviours of the tool. For example the findings made against the Boeing Laptop Tool mentioned in section 2.3 relate to clear annunciation of stale or default data to the operator to avoid misinterpretation for valid date. There are also recommendation relating to interface regarding display of units, etc; all of which are functional requirements for the MPS tools.

The other forms of recommendation may be less explicit but become more apparent if litigation is pursued. For example, if an unassured tool calculated an invalid result which was misinterpreted by the human operator; and it could be demonstrable that the application of industry benchmarks and recognised aviation software practices for this tool would have prevented the issue at reasonable cost then would the developer of the tool be held liable? There are few cases to date where this argument has been explicitly tested. The authors' reason that the comparison of what's been done regarding tool assurance, versus what could have been done at 'reasonable' cost, would

certainly fall against developers who had adopted the former approach.

To a great extent, the regulatory requirements outlined in Section 2.5 provide some benchmarking of what forms a suitable basis of comparison, but in the ADF context, these are yet to be widely adopted.

So in light of these circumstances, and the assertions this paper has made about the behaviours of certification authorities and developers alike, and how these may be traced to the incidents of the previous section of this paper; what is the case for prescribing data and design integrity requirements onto MPS tools? This paper proposes that the case should be centred around two key issues.

The first is that the errors, faults or failures of the tool should not present an unreasonable burden on the human operators in having to dedicate crew resources to detecting any errors, faults or failures. This is because the very purpose of the tool is to reduce workload in planning and situational awareness, and not to add workload burden to these activities. Ideally there should be no errors, faults or failures (but for pragmatic reasons, absolute assurance is not achievable). However, since errors, faults and failures are almost inevitably present, the focus should be to use assurance practices to limit the presence of these errors, faults and failures to within a tolerable operational burden (i.e. the satisfaction of requirements of the tool should be assured to a known confidence). Safety and software assurance practices currently offer the only recognised approach to presenting an argument that qualifies the degree of confidence in the absence of errors, faults and failures. In its frequent absence in the MPS/SA context, however, the 'assurance deficit' must be assessed for acceptability of risk by some means. This is where the linkage into the human factors evaluation elements become explicit (refer to Section 4).

The second is that the suitability of the tool's behaviours should be explicitly treated, through the introduction/confirmation of assured product behavioural requirements for those circumstances which would increase the likelihood of the crew invalidly interpreting a result from the tool (i.e. the tool should be designed to actively minimise crew error and prevent hazardous circumstances). An example of these sorts of circumstances might be a small error in flight performance information that leads to a runway length error in marginal operating circumstances without crew knowledge. Another may be a small positional error that leads to invalid situation awareness regarding navigation, particularly when engaged in tactical low-level flying. Both of these circumstances could be mitigated through the introduction of safety requirements (reasonability checks, cross checks, warnings, etc) to assist in their mitigation. To make the safety case successful, developers should have to argue that the tool has been designed not to reinforce or contribute to a decision process which would lead to hazardous circumstances. For example, if a tool can allow default or previous data to be automatically imported into the tool's fields to expedite repetitive entry tasks, then the possibility of this data being invalid should be explicitly treated. This may

only be achieved via the introduction of specific design safety requirements on MPS tools. If considered for developmental tools, where the opportunity to inject design requirements still exists, this will be relatively straightforward. However for legacy tools, it is rarely possible to retrospectively inject design requirements to introduce such behaviours to the tool. Instead then, such legacy tool circumstances will drive a need for human operational evaluation to assess the real affects of the absence of these features, and determine their tolerability. Until design regulation leads the technology, this latter situation will carry the greater burden of the safety argument.

In light of these two key arguments, the following subsections outline the case for data and design integrity for MPS tools. The paper has been constructed around both data and design integrity because the results of these tools inevitably depend on both the behaviours of the tool, as well as the data that is the input to these tools.

3.1 Aeronautical Data Integrity

Aeronautical data integrity is the degree to which confidence can be placed in the precision and accuracy of the supplied data. In circumstances where aeronautical data, or the calculations being made from it, are used to support phases of flight where errors in that data may be hazardous, then clearly the integrity of the data is paramount.

While it is possible to argue, that aeronautical data integrity uncertainty, and the hazards arising from its use, may be overcome by detection and workarounds by human operators, this approach is problematic. Humans are relatively good at detecting significant gross errors, provided the relevant cues are provided, and the basis of comparison from which the error is detected via comparison of similar information (e.g. similar types of displays and units). Unfortunately, humans are less adept at detecting subtle errors between information. The challenges with aeronautical data are that there is so much of it (i.e. the aeronautical data bases are usually big and complex), and this prevent it being obvious to operators as the how good all that data is, unless appropriate controls have been placed on how the data has been generated, manipulated and managed.

Under what circumstances then would it be reasonable for a human to detect an error within aeronautical data? This will be dependent on the extent to which the MPS tool manipulates the data or derives other values from it. Further, other data types may be used directly in the tool output, but the detectability of their validity (and any margins of tolerability of accuracy and consistent) will depend on how the data is used and correlated by the pilot to other situation awareness cues.) Section 4 of this paper deals with these aspects of the operational assessment. Any data which fits beyond the reasonable detectability and handling by operators should therefore be subject to some form of dependability assurance. This seems a pragmatic approach, but when subject to operational hazard assessment, much data used for the more challenging operations, will always end up requiring a

level of assurance, as it just isn't possible for a human operator to provide appropriate workarounds to potential shortfalls. One approach to aeronautical data assurance might be along the lines of the FAA approach for data integrity:

- RTCA/DO-200 – Standards for Processing Aeronautical Data describes the requirements for the processing of aeronautical data including tool qualification requirements.
- RTCA/DO-201A – Industry Requirements for Aeronautical Information specifies the aeronautical data elements required by the aviation industry and a standard for the accuracy, resolution, and integrity of the associated values.

The FAA approach is consistent with ICAO practices and is one of the more mature approaches available. Of course, given that in many cases the regulation lags innovation, this is not necessarily testament that the FAA approach is the best approach. Note also that the FAA approach to Aeronautical Data Integrity is not a one size fits all approach, and it scales the degree to which confidence is required along similar lines to the Design Assurance Level approach of software standards such as RTCA/DO-178B.

Alternatively the ADF has developed an approach that adapts the FAA approach to the military specific context (refer to the NPRM for AAP7001.054 Section 2 Chapter 24). The ADF approach encourages an even more product focussed assessment of the data in the context of the tool and end application to establish the impacts of invalid data, and this drives data integrity requirements.

In many cases operational evaluations will lead to the conclusion that the data needs to be dependable, and thus the data integrity requirements will be required anyway for the bulk of data being used to support MPS tool functions and the flight operations dependant on them.

3.2 Software Safety and Assurance

Software safety and assurance are the means by which software is developed that meets safety objectives. It usually involves a complementary suite of analysis and verification evidence which seeks to show two key outcomes: requirements validity, and requirements satisfaction. The following sub-paragraphs examine these two concepts in further detail, and explain their relevance to MPS tools.

3.2.1 Requirements Validity

Requirements validity addresses the question: does the software have the right behaviours? It is about ensuring that both the normal functional behaviours of the software, and also the failure behaviours are compatible with the intended safety objectives. While this is a fairly abstract concept, it provides some important pointers for the types of behaviours that have to be considered when developing software requirements, which will ultimately determine the acceptable behaviours for the software.

So what are the acceptable behaviours for an MPS tool? There are several ways to achieve this. One practice is to start with a system with known behaviours and empirically evaluate the suitability of each of these behaviours in the operational context. While in many respects this provides an very effective evaluation of these behaviours, it is costly and time consuming. The alternative approach is to establish a set of behaviours early in the development and subject them analytically (or by targetted operational evaluations) to establish their suitability and completeness. If this second approach is adopted, then the result should be requirements that unambiguously define the functions associated with each piece of information presented to the user, and why this behaviour is appropriate under all scenarios that this information may be used. This will provide the requirements for the normal functional behaviours of the system.

Further to the normal functional behaviours of the software, additional behaviours of the software should be defined. These are the behaviours of the software to deal with circumstances involving errors, faults and failures that might invalidate the normal functional behaviours of the MPS software detailed above. Behaviours dealing with errors, faults and failure normally require two properties, one of which is the means by which the error, fault or failure will be detected, and the other is the means by which it will be handled. In some cases the handling of the error, fault or failure, may be by defining a requirement for a behaviour at the human machine interface that alerts the human operator to the error, fault or failure.

Therefore a key aspect of identifying and analysing the behaviours of an MPS is to ensure that the evidence provides good coverage of both these normal and failure circumstances. One way of achieving this would be to undertake analysis that considers:

- each resultant piece of information presented to the human operator, or prepared for automated transfer to an aircraft (such as to a flight management system),
- each phase of flight, or operational scenario in which this data might be used (e.g. power settings supporting take-off, a GNSS based landing approach, etc.)
- the credible effects (including worst) of the information being invalid in that flight scenario,
- whether the invalid information would be human detectable, and what the impact on pilot work-flow would be as a result of the error (see section 4)
- how the MPS tool could either prevent, or detect and handle the applicable error, fault or failure

In the ADF, this approach as described by AAP7001.054 Section 2 Chapter 24, has had some limited application to a couple of applications, eg PFPS/JMPS, Super Hornet, MRH90 GMMS. In each case, the understanding developed as to what the data each system provided and how it was used, permitted an extremely pragmatic approach to fielding these systems (albeit retrospectively) to be pursued.

However, despite the intentions of the aforementioned analysis, when evaluated, many MPS will have behaviours that aren't appropriate under certain circumstances. If this is the case, then the effects, human detectability and impacts of human work-flow are vitally important. An example of this is some of the limitations promulgated by the ADF on the use of PFPS and J MPS in flight to support more challenging navigation functions, in the absence of full operational evaluation of potential workarounds to the shortfalls. Section 4 describes how these should be evaluated and how it might be established that these collective impacts are tolerable.

3.2.2 Requirements Satisfaction

Having developed a set of requirements that are asserted to provide a set of behaviours compatible with safety objectives, requirements satisfaction deals with the implementation of these requirements such that the required behaviours are implemented in the software product. The key goal is to ensure that in implementing these required behaviours, that unacceptable errors are not introduced that would lead to a hazard to safety, or violate an assumption about treatments to identified hazards.

For aircraft software, requirements satisfaction is normally achieved via the application of the software levels within software assurance standards such as RTCA/DO-178B. While there are arguments in the literature about the effectiveness of such standards, the purpose of this paper shall not be to re-examine these arguments. Instead, this paper will assume that whether it be the framework of a software assurance standard, or the alternative framework provided by an argument and evidence based approach, both are means of providing evidence of requirements satisfaction. For the purpose of simplicity here though, these concepts will be referred to as the application of a software assurance standard.

For new ADF specific developments, ADF contracts (or equivalent instruments) should include requirements for software assurance to the degree required when the MPS tool employment is holistically considered.

However, the earlier paragraphs in this section discussed the limitations to the applications of software assurance standards to all classes of MPS and EFB tools per both the FAA and ADF policy. In these the application of software assurance practices could be retrospective at best; and most likely perceived as not cost effective. Therefore, in the absence of prescription of a software assurance standard, how does the FAA and ADF policies assure requirements satisfaction for these tools. In short, this is where the interactions between the degree of design and data integrity and the human factors elements come into play. There is no blanket answer on whether all the circumstances where safety and assurance practices haven't been employed would result in acceptable tool solutions. Section 5 presents a safety case argument strategy that deals with this conundrum.

4 Case for Human Interface Safety

Mission Planning Systems and SA tools, by their very definition, are an extension of the Human operators thinking and computing space. The operator exports data into the tools in order to simplify the computation of derived parameters and assist in creating and representing mental models back to the operator in preparation for use. This data may then also become "off-line" during operations, but be used for real-time decision making. It follows that the interface and presentations through which this information transfers is critical to crew manipulating data correctly and gaining a correct understanding of the results.

Dr. Carl Sandom, an independent consultant in the arena of safety-critical software based systems and human factors, maintains a thesis of the interdependence of human safety functions and system safety functions in information systems. "Human as Hazard or Human as Hero" is his catch phrase. In [San06] and others, he portrays the interaction of real-time and near real-time information systems, operations and safety outcomes in the following model:

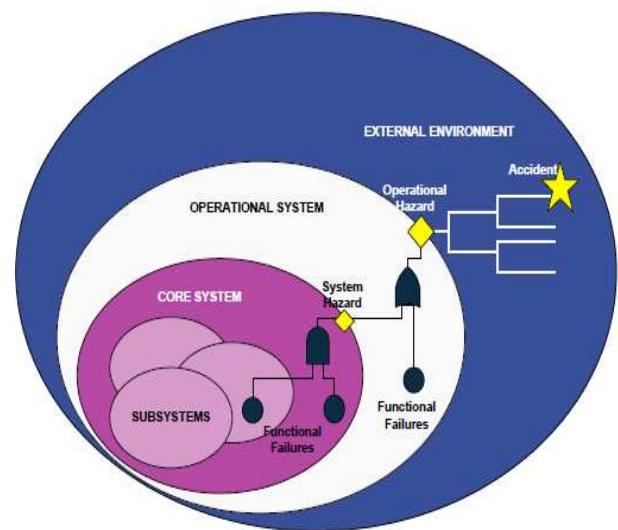


Figure 1: Information System Model [San06]

Fig 1 attempts to convey the global picture of influence on accident sequences where, for safety related information systems, the operational sphere has almost as much influence on the outcome as the core system. The consistent concern is that design and safety assessment analyses focus too heavily on system functions and integrity, without the commensurate analysis to support assertions of human function reliability and the factors that would affect it.

It is reasonably clear from the above referenced accident research, that erroneous human interactions with planning and automated flight management systems, is a vulnerable link in the aviation safety chain. Appropriately, the FAA's AC120-76A approvals requirements guidance, devotes significant proportion attention to Human Factors in design and implementation

requirements. This content was largely taken from the FAA sponsored research published by the Volpe Center in 2000, which has since been significantly superceded and embellished at [Cha03], and recognises that although EFBs may increase efficiency and safety of operations, they “could have negative side effects if not implemented correctly. For example, increasing workload and head-down time, and distract crews from higher priority tasks”.

In 2010 Volpe commissioned Chandra et al again [ChaK10] to review safety incidents involving EFBs and this report validated that workload, data entry errors and crew attention fixation issues as major contributors, all revolving around the way the EFBs were implemented and trained.

The authors believe that several conclusions are reasonable from the accident analysis, and when correlated with extant Human Factors accepted studies collated in [Kel85], [EBJ03], [San06] and others from various industries:

- Humans are traditionally poor at the role of passively and continuously monitoring automation for long periods of time;
- There is a natural bias towards trusting automation even when external cues and training would indicate otherwise; and
- Increases in automation are reducing aircrews fundamental skills to deal with failure scenarios.

The last point has also been discussed recently in aviation safety journals and commentary such as [FSF05] and [Sei11]. Further the accident record seems clear that operational procedural norms and training have not yet been adequately ‘tuned’ to new vulnerabilities introduced by planning information systems and flight management automation.

In other words – managing failure at the MPS/SA human machine interface is an undeniable and critical link to achieving operational safety.

4.1 Assessing Safety Hazards from Human Factors

What does this tell us about hazard elimination or acceptable risk assessment? Safety management will reside in a combination of design features, highly integrated planning and operational procedures, and effective training. Where the design stage is evolutionary or all together independent of implementation, a greater weight of responsibility for safety falls on the design of operational integration.

In either case, a more comprehensive hazard analysis activity is required. An analyst first will have to identify and assess the criticality of where the most hazardous elements (considering both human tasks and system functions) exist, then have to identify what features or procedures would facilitate error detection and correction, in various phases of operation, considering workload and other factors affecting probability of error reduction in order to make risks acceptable.

[SaF06] describes one such approach to address these issues by defining a framework for identifying human safety and system safety requirements through a design phase using established Critical Task Analysis and Human Error Analysis methodologies. [SaF06] describes how to assess an existing COTS/MOTS product’s safety for a new application, in the cases where safety requirements have not been explicitly articulated and tested, and safety or user error reduction features not documented in any analysable form. Data criticality and software integrity requirements can be decided by the methodologies discussed earlier, but it is reasonable to assume (and is the well-worn experience of the authors) that if these considerations were not part of the original design, then objective evidence of design assurance is difficult to achieve retrospectively. However, Critical Task Analysis (CTA) and Human Engineering Assessment (HEA) activities are still valid and certainly able to be applied retrospectively. Human error detection and correction (or handling) is the last line of defence against hazardous system faults and errors created by human functions.

In the realm of aviation electronic flight bag functions, the Volpe design assessment guides are clearly a good starting point for hazard identification sources. Deficiencies against these proposed design features are immediate candidates for hazardous interactions. A data criticality analysis will then focus identification of the more critical functions as related to the target application and user scenarios. In order to complete the data criticality analysis, a complete understanding of operational intent and user operational workload profiles will be required through engagement with platform qualified operational representative/s and reference to an authorised Statement of Operating Intent, which would bound the problem space. Moreover, in the case of pre-existing or non-safety assured systems, it is possible and may be necessary or even essential (in the absence of sufficient human factors analytical resources) for extensive Operational Evaluation to pre-date formal ‘service release’ in order to fully appreciate the critical tasks and human error zones, training needs and procedural defence targets.

4.2 Mitigating Human Interface Safety Deficiencies

Fundamentally and inevitably the human functions will be both “hero and hazard”. No two humans are alike, so it will not always be possible to transfer assumptions from one human operator to another. Where analytical safety assessment or operational evaluation identify vulnerabilities and deficiencies in design, the choices are limited. From [DGTA09], the following approaches are outlined to overcome an identified technical shortfall:

- a. Design Assurance.*
- b. Detection and Handling Mechanisms.*
- c. Operational Limitations.*
- d. Independent Verification.*
- e. Risk Retention.*

Experience in the ADF has shown that options 'a' and 'b' are typically a last resort of project managers, because they have a substantial cost and schedule burden, and usually only pursued after clear demonstration that a combination of options 'c', 'd', and 'e' are unworkable, and the risk is not tolerable.

The most difficult challenge in this step, is typically the lack of sufficient substantive human factors analysis and/or operational evaluation results to make convincing unacceptable safety arguments. In the absence of thorough analysis and data, only opinion and credibility are available to ensure sufficient safety is provided. All of which may be flawed or skewed.

Insufficient benchmarks or other objective measures currently exist. Unfortunately, the apparent convenience and lack solid counter evidence of the efficacy of these mitigations, means they are accepted. Thus, with these emerging technologies and accompanying hazards, the task of demonstrating adequate safety of operational risk management measures, will often be harder or easier depending on specifically relevant experience levels of the operational regulator staff.

4.3 Procedures, Training and Workaround Options

Standardised procedures, supervision and checking are essential and common defences for safety related human functions. These are normally designed around intuitively critical steps, or aspects that were complex and prone to human error, or applied in response to experience of failure. With emerging technologies being charged with automating previously human functions and adding more complex simultaneous activities, intuition is no longer enough and experience is not available.

For complex systems, as identified above, a combination of specific analysis methodologies and structured operational test and evaluation periods are likely to be required to in order to develop appropriately targeted procedural defences.

In particular, human procedures that are intended to compensate for design integrity shortfalls must be based on study of the practicalities of detection and correction of each critical potential error in function or data handling, with due consideration given to a reasonable workload in envisaged phases of flight including degraded visibility or weather conditions or predicted system failure scenarios. In order to detect error, crew must have ready reference "truth data" that is regularly being checked. For example, monitoring aircraft performance against plans is typically well supported by constantly scanned instrumentation and back-ups. However, navigation cross referencing requires visual meteorological conditions and independent sources of position data from the integrated flight displays. It is generally understood as difficult to for operators to detect subtle errors in digital aeronautical data. This should best be achieved via automated and qualified data integrity control. If this isn't in place, then operational procedures to limit the effects of invalid data to minor failure conditions only - some analysis would be required to

work out how each data element is used. Eg. think about how to detect if a runway threshold is in the wrong position for a GPS based landing in Instrument Meteorological Conditions - crew would have to be correlating differences between the GPS solution and the older navigation beacons. A high workload impost and a busy stage of flight.

5 Safety Case Argument Strategy for MPS & SA tools

Based on the two cases presented above (for software and human factors), this paper proposes a top level argument strategy for MPS tools that might form part of the overall safety case argument for such tools. Note that additional factors such as tradeoffs between operational risk, capability and safety during non-peacetime operations have not been covered within the argument strategy presented in this section.

As the case for human operator responses relies on the impact of the behaviours of the MPS tool, and the case for suitability of MPS behaviours relies on the existence of features of the MPS tool to avoid potential sources of human error. The top level argument of the proposed strategy focuses on the duality of the interactions between the human operation and the MPS tool. Figure 2 presents a Goal Structuring Notation (GSN) representation of the overall strategy proposed by this paper.

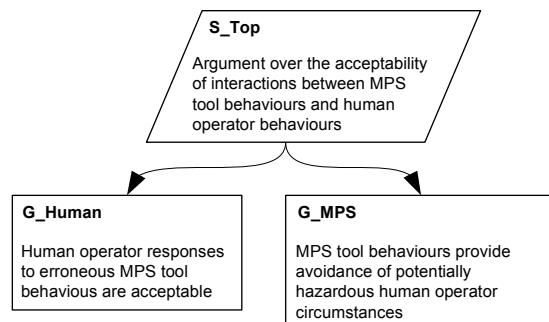


Figure 2: Top Level Argument Strategy

The argument focuses on a systematic evaluation of the interactions between the human and the MPS tool, to ensure that each of these interactions are acceptable. Two main arguments make up examining these interactions, the human factors element (G_Human) and the MPS tool (G_MPS) element. This deliberate breakdown ensures that neither human factors evaluations, nor design and data integrity form a biased role in the argument, and that both have equal intended precedence, and cannot achieve the requisite outcomes in isolation of each other. The following subsections now examine the two key sides of the argument strategy.

5.1 Human Factors Elements

The main thrust of the human factors elements of the argument strategy is to ensure that the human operator responses to the MPS tool's behaviours, with an emphasis on those behaviours which might be considered erroneous or invalid, are appropriate. The strategy for this argument is to systematically examine, both in isolation and

collectively, each identified MPS tool behaviour and to determine if the associated human responses are appropriate.

The multiple instantiation dot on the link to the goal G_Human_X indicates that this goal will require instantiation for each MPS tool behaviour. So how should each MPS tool behaviour be established (per C_Human)? This is one element of the argument where there is an implicit dependency between the human and MPS tool sides of the argument. Where design and data assurance practices have been employed on the MPS tool side the argument, then the behaviours of the MPS tool documented in requirements, verification and validation evidence naturally provide a basis from which to infer MPS tool behaviours. Where the MPS tool argument is substantially weaker, then this puts an imperative on the human evaluation program to focus more analytically on potential behaviours of the MPS tool (such as via some structured software safety analysis, functional analysis, etc), to draw meaningful conclusions about having been systematic about MPS tool behaviours in the human evaluation.

To determine if each MPS tool behaviour is acceptable, both the detect-ability of the MPS tool behaviour is considered, along with the human handling response to resolve. The argument also makes explicit the phase of flight, as the impact of MPS tool behaviours will almost certainly always vary depending on the phase of flight in which the information is required or used. Figure 3 presents a GSN representation of this part of the argument.

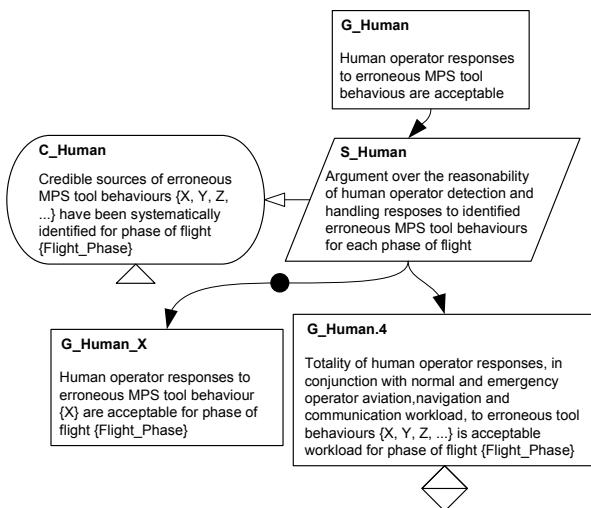


Figure 3: Human Factors Elements

At the lower level of the argument structure for each individual tool behaviour, MPS tool behaviour detectability, handle-ability, and workload are made explicit. While the evidence for each of these goals would normally be derived from the one holistic human engineering program, these sub-goals provide focus that the human engineering program must explicitly evaluate each of these factors in turn. For example, there may be scenarios where the human handling response to particular classes of MPS tool behaviours may be quite

straightforward, but due to limitations in how detectable the MPS tool behaviour is, there may be insufficient time to conduct the handling response prior to the hazardous circumstance manifesting itself.

It is also important to consider the workload impost of these detect-ability and handle-ability responses in the context of the normal crew workload (G_Human.3). There may be many workarounds that when considered in isolation are entirely valid, but which cannot be suitably integrated into existing crew workload. A substantial focus of the human engineering program should be to establish these. It also provides a good vehicle for considering the totality of human workload from all workarounds, and not just issues in isolation.

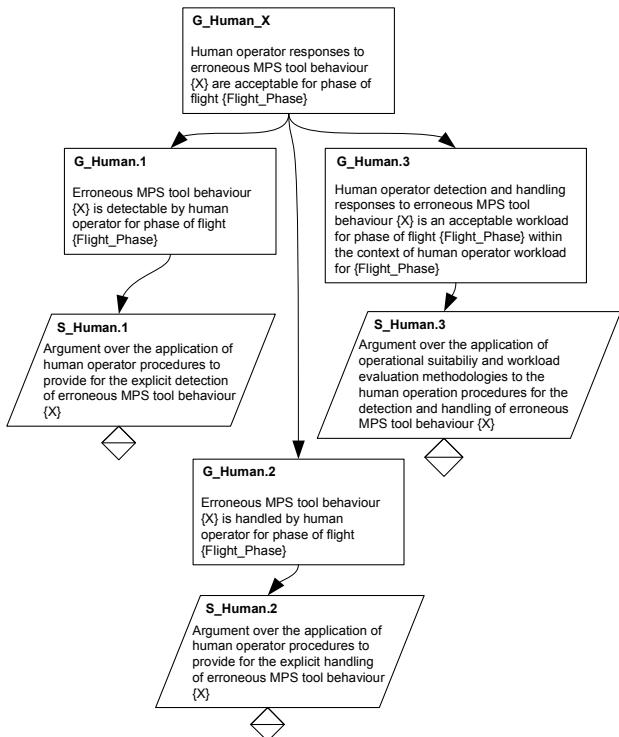


Figure 4: Detect-ability, Handle-ability and Workload

For the purposes of this paper, the remaining lower level parts of the argument are left undeveloped and uninstantiated. Below this level will always be evidence and solution dependent, and it is not the purpose of this paper to provide anything more than a generic argument strategy template.

5.2 MPS Tool Elements

The main thrust of the MPS tool elements of the argument strategy is to ensure that a set of suitable MPS tool behaviours is provided to avoid those circumstances which would result in potentially hazardous human operator circumstances. This argument establishes that within the workload and crewing procedures established for the crew to safety operator the aircraft, no behaviours of the MPS tool should violate these in such a way that leads to hazards. The strategy for this element of the argument is to systematically² evaluate what might constitute potentially hazardous human operator circumstances resulting from MPS tool outputs, and then examine ways the MPS tool might offer additional behaviours to prevent or avoid these circumstances. Figure 5 presents a GSN representation of this part of the argument.

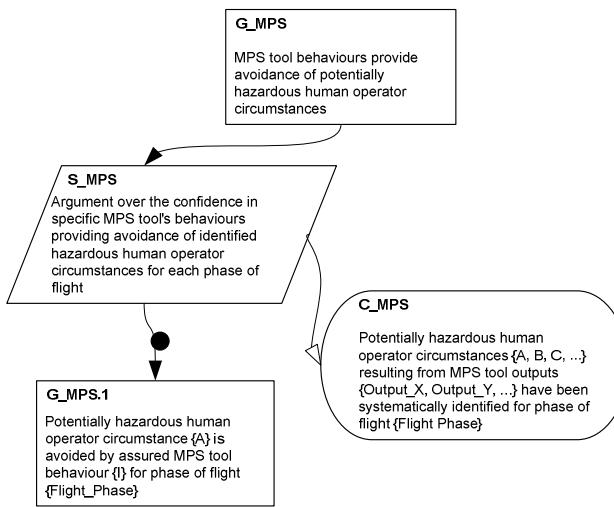


Figure 5: MPS Tool Elements

The key point here is that this element of the argument does not focus on MPS tool errors or failures, instead it should be examining how the MPS tools outputs (even when correct) affect the human workload and procedures. As we discovered with the human factors side of the argument, this here introduces the mirrored implicit relationship between the sides of the argument. To completely understand the human operators' workload and procedural implications for the outputs of the MPS tool, then strong linkages into the human engineering program results will be required. This is reflected by the context C_MPS, and the relationship it infers. Figure 6 shows a GSN representation of a strategy of how C_MPS might be presented, such that the linkages into both the human engineering program and the safety program are made explicit.

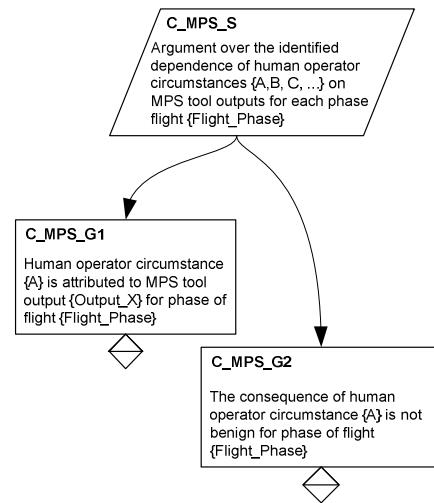


Figure 6: Evaluating Human Operator Circumstances

The multiple instantiated dot on the link to G_MPS.1 in Figure 5 requires that each potentially hazardous human operator circumstance gets considered, so that the MPS tool outputs that affect each of these circumstances get systematically reasoned about. There is no need to introduce any non-interference criteria between the MPS tool outputs at this level of the argument, as this will be made more explicit at lower levels of the argument associated with MPS tool behaviour implementation, or requirements satisfaction in software assurance parlance.

For each MPS tool behaviour that might provide avoidance of the hazardous human operator circumstances identified above, there are two key elements that provide assurance of that behaviour:

- Safety and design assurance of the MPS tool behaviour itself; and
- The integrity (accuracy, precision, etc) of the underlying data used by the MPS tool behaviour.

Figure 7 shows the elements of the argument that introduces the role of software safety, software assurance and data integrity.

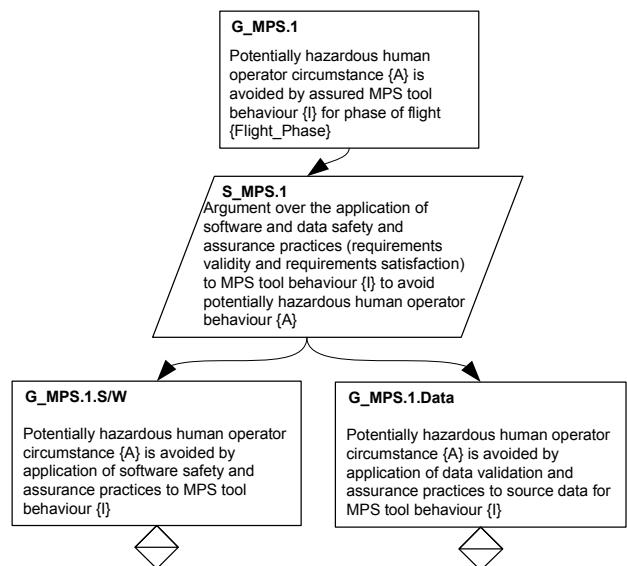


Figure 7: Software Assurance and Data Integrity

² The intention of requiring 'systematic' evaluation is to ensure that the operational context is defined, and the means of undertaking the evaluation provides a measure of coverage of the extent (and confidence) in the circumstances identified by the evaluation.

For the purposes of this paper, the remaining lower level parts of the argument are left undeveloped and uninstantiated. Below this level will always be evidence and solution dependent, and it is not the purpose of this paper to provide anything more than a generic argument strategy template. However, the application of recognised safety standards (ARP4754, DefStan 00-56, etc), software assurance standards (RTCA/DO-178B, Def(Aust)5679, DefStan 00-55 (superseded)), and data integrity standards (RTCA/DO-200A and RTCA/DO-201) would provide an appropriate instantiation of these lower level goals. While it is recognised that many MPS tools are developed outside such frameworks, and the retrospective application of such standards may be problematic, these goals provide the linkages to those circumstances when an absence of assurance (in the context of the overall human operator interaction with the MPS tools), would likely be intolerable.

6 Summary

This paper has considered the challenge of creating a successful safety argument for implementing the emerging technology of sophisticated and integrated mission planning systems and situational awareness tools into safety critical operations, such as aviation.

Usage has evolved over the last 10-15 years in commercial and military aviation and the accident record now provides sufficiently valid data to identify the consistently contributing factors to catastrophic outcomes.

The authors have assessed how well matched current regulatory guidance is to these contributing factors and how it is currently being applied to product development and implementation. The paper then considers the relative contribution of software design and data integrity on balance with human interface design and operational assessment and training, for mission planning systems operational safety. There are challenges for assessing and mitigating the hazards posed by each.

Finally the paper proposes the elements of a safety case argument structure that may be used to achieve approval for use of mission planning and situational awareness systems in safety critical applications.

7 Conclusions

Current certification and operational approvals requirements for aviation mission planning systems (including EFBs) and situational awareness tools are arguably not sufficient. Potential accidents due to MPS or SA tool causes are not mitigated to an equivalent risk level as for other hazardous and catastrophic potential aircraft systems. System design standards do not exist and valuable lessons learnt are therefore not able to feed into an improving and safer design basis to be consistently applied. Operator interface with these systems is the most vulnerable link in the accident causal chain, which needs to be supported by more robust designs and critical task analysis leading most specifically to tailored operational procedures. Analysis should also be supported by conducting thorough operational evaluation in order to develop targeted

training, currency requirements and aeronautical data management processes.

In the interim, individual applications for certification and implementation of mission planning systems should be required to present a more sound and substantiated safety argument, fulfilling goals of a balanced treatment of design integrity and human factors elements, where each arm supports the assurance deficits of the other - as proposed in this paper.

It is hoped that continued interest in the subject for aviation safety and its potential extrapolation to other safety critical industries, and decision support information systems, will result in further research and validation. Most fruitfully, a cut-set of certification design requirements (or System Safety Requirements) could be identified. These would include HMI and safety features, as well as design pedigree and software assurance that will influence the state of the art offered to the market.

8 References

- [CASA10] Visual Flight Rules Guide, Version 4 May 2010, CASA
- [Lea10] D. Learmont, "European helicopter safety team unlocks secrets of rotary wing achilles heel", Flight International, 15/10/10, <http://www.flightglobal.com/articles/2010/10/15/348468/european-helicopter-safety-team-unlocks-secrets-of-rotary-wings-achilles.html>
- [ATSB11] TRANSPORT SAFETY INVESTIGATION REPORT Aviation Research and Analysis Report, AR-2009-052 Final, "Take-off performance calculation and entry errors: A global perspective" January 2011
- [War10] SQNLDR G. Warwick, "Deadly Data", ADF Aviation Safety Spotlight Magazine 04/2010, DDAFS
- [Lea06] D. Learmount, "Ryanair 737 'nearly crashed'", 12/12/06, Flight International, <http://www.flightglobal.com/articles/2006/12/12/211032/ryanair-737-nearly-crashed.html>
- [Lad05] P Ladkin, "The American Airlines B757 Accident in Cali", accident investigation summary and commentary, http://www.rvs.unibielefeld.de/publications/compendium/incidents_and_accidents/cali_american_airlines_b757.html
- [FAA03] AC120-76A, "Guidelines for the Certification, Airworthiness and Operational Approval of Electronic Flight Bag Computing Devices"
- [RTCA] DO-200, "Standards for Processing Aeronautical Data"
- [RTCA] DO-201A, "Standards for Aeronautical Information"
- [ADF09] AAP 7001.048(AM1), ADF Airworthiness Manual, <http://www.defence.gov.au/dgta/Documents/Publications/7001048/7001.048%20AL1%204%20Mar%09.pdf>

- [DGTA08] NPRM01/08, "Electronic Flight Bags", Preliminary DRAFT AAP7001.054, Sect2,Ch22
- [DGTA09] NPRM 03/09, "Flight and Mission Planning Systems", Preliminary DRAFT AAP7001.054, Sect2,Ch24
- [CASA10] "Electronic Flight Bags" Airworthiness Bulletin AWB00-0017, Issue 2, 18May10)
- JAA Leaflet No. 36 Use of EFBs
- [EJB03] M.R. Endsley, B. Bolte, D. Jones, "Design for Situational Awareness – An Approach to User centered Design", Taylor and Francis Group, 2003
- [StFa03] N. Storey, A. Faulkner, "Data-The forgotten System Component?", The Journal of System Safety, Q4 2003
- [Kle85] T.A. Kletz, "An Engineer's View of Human Error", The Institution of Chemical Engineers, 1985
- [FSF05] "Increased Reliance on Automation May Weaken Pilots' Skill for Managing System Failures", *Human Factors and Aviation Medicine*, Vol52, No.2 Mar-Apr2005.
- [Sei11] J Seigfreid, "Automation Issues Raise Safety Concerns", <http://www.examiner.com/airlines-airport-in-national/aircraft-automation-issues-raise-safety-concerns>, 18Feb11
- [Ham05] R.A. Hamilton, "Navy Faults Navigational Procedures in Crash of Sub"" http://www.ssbn611.org/uss_san_francisco.htm,
- [Per05] R. Perry, "Why We Almost Lost the Submarine (USS SAN FRANCISCO SSN711)" 13 April 2005, http://subveteran.org/SSN%20711/711_summary.htm
- [Cha03] Human Factors Considerations in the Design and Evaluation of EFBs, v2, DOT-VNTSC-FAA-03-07, September 2003, Volpe centre
- [ChK10], Chandra/Kendra, "Review of Safety Reports Involving Electronic Flight Bags", DOT/FAA/AR-10/5, DOT-VNTSC-FAA-10-08, April 2010.
- [Spa11] N. Sparano, "Pilots go paperless with Denver developed FAA iPad app", 17Feb11 Fox31 KDVR news item <http://www.kdvr.com/news/kdvr-faaipadapp-txt0,3172069.story>
- [Jep11] Jeppesen, "Jeppesen and Executive Jet Management Collaborate to Gain FAA Authorization for Use of Jeppesen Charts on iPad", 11Feb2011, http://www.jeppesen.com/company/newsroom/articles.jsp?newsURL=news/newsroom/2011/iPad_EFB_authorization_NR.jsp
- [SaF06] C. Sandom and D. Fowler (2006), "People and Systems: Striking a Safe Balance Between Human and Machine", in Redmill F and Anderson T, Developments in Risk-based Approaches to Safety, Proceedings of the 14th Safety-critical Systems Symposium, Bristol, UK, 7 – 9 February 2006, Springer-Verlag
- [San07] C Sandom, "Success and Failure: Human as Hero – Human as Hazard" Keynote Presentation, proceedings of 12th Australian Conference on Safety Related Programmable Systems, 30-31Aug 2007, Adelaide. Conferences in Research and Practice in Information Technology (CRPIT), Vol. 57.
- [TSBC06] Transportation Safety Board of Canada. (2006). Reduced power at take-off and collision with terrain, MK Airlines Limited, Boeing 747-244SF 9G-MKJ, Halifax International Airport, Nova Scotia, 14 October 2004 (A04H0004). Quebec, Canada: Transportation Safety Board of Canada. [Transport Canada www.tsb.gc.ca/en/reports/air/2004/A04H0004/a04H0004.pdf]

Managing Systems and Software Safety Risks in Emerging Technologies – A Surface Transport Perspective

Len Neist

NSW Independent Transport Safety Regulator
Level 22, 201 Elizabeth Street, Sydney, Australia

Leonard.Neist@kensarypark.com

Abstract

This paper was a keynote address at the Australian System Safety Conference (ASSC 2011). It provides a surface transport perspective on safety risk management and the need for better human systems integration to better build in error tolerance. The paper also discusses the use of scenario based planning to better understand an operational risk context.

Keywords: human systems integration, risk mechanisms, safety leadership.

1 Introduction

The theme of *Managing Systems and Software Safety Risks in Emerging Technologies* is almost a timeless subject. As we try to do more with technology to speed things up, provide greater endurance, increase lethality, increase logistic performance and in general get more with less, the risks to safety increase.

This is as true in surface transport as it is in aerospace, health, defence, energy or any other industry. As rail operators strive to get more out of the infrastructure, trains are getting longer, heavier, faster and they need to move closer together across more complex networks. Buses need to operate in overstressed traffic networks yet still try to maintain timetable in order to coordinate with rail as part of a seamless passenger transport network. Designers are reaching for new technologies to make this possible.

As the NSW Independent Transport Safety Regulator, I have a role in making sure the safety risks are appropriately considered and are proactively managed, and hazards or threats to safety are either removed or controlled so that the risk to safety is managed to be tolerable so far as is reasonably practicable.

Despite the investment of times past and the diligence of many great engineers, catastrophic incidents still occur. Human error, intentional violation, system failure and underestimated complexity are some of the

mechanisms that still require system safety specialists to design in appropriate controls and defences.

In the rail environment a key near term challenge is to make sure that the adoption of technology to improve location awareness, provide high reliability communications and high dependency authority control systems, is done safely and effectively in the context of the rail operating environment.

As we move trains faster, closer together and with more complex technology, the need to ensure that the human systems are fully considered along with the software and hardware during design, integration and test becomes increasingly important. Increasing the maturity and effectiveness of the technical and management systems to understand and improve safety risk management will require three things from the rail industry (and I include the regulators in the definition of rail industry):

1. growth in understanding and practice in regards to methods and processes associated with human system integration, particularly building in error tolerance in technical systems;
2. increased maturity in risk management practice such that the mechanisms that translate hazards into harm are identified, understood and controlled; and
3. improved safety leadership from the chief executive down so that everyone is behind the push to keep risk to safety as low as is reasonably practicable.

I would like to discuss these three concepts at a high level.

Following a study into major industrial incidents in the 90's, Charles Perrow said:

"Despite improvements in technology, the number of catastrophic incidents is expected to rise, if for no other reason than opportunities for both human and machine failures increase with complexity."

In order to develop and establish a truly effective safety management system capable of managing the risk to safety involved in increasing complexity, safety leaders and managers need to establish a firm understanding of how humans perform and are integrated with organisations and technology.

People need to be competent to operate in such complex environments. By competent, I mean they must possess the required training, experience and qualifications to allow them to exercise safety leadership accountability and be responsible for managing and

Copyright © 2011, Australian Computer Society. This paper appeared at the Australian System Safety Conference (ASSC 2011), Melbourne, Australia. Conferences in Research and Practice in Information Technology (CRPIT), Vol. 133. Tony Cant, Ed. Reproduction for academic, not-for profit purposes permitted provided this text is included.

accepting risk. Complexity is increasing as more and more safety functions are implemented through software which must then be integrated with hardware and the operator interface. The processes, procedures and policies of an organisation that works in complex hazardous environments must be developed to manage the new risks associated with software driven solutions to ensure that the safety integrity of the final product is maintained at an appropriate risk level commensurate with the experience and maturity of the system operators.

The safety management system used to manage the risk to safety must be sufficiently broad to facilitate effective and safe integration of these three elements: people, organisations and technology. When I was involved in airworthiness in Defence, we used to refer to this as the OPPD model, i.e. competent organisation, authorised people, current approved procedures and current, valid technical data.

The concept of integrating humans with organisations is well understood. Similarly human factors and human performance in relation to technology is also well known. However, human systems integration is a new technical and managerial concept that attempts to understand and utilise the complex relationships amongst people, organisations and technology to provide positive outcomes. In other words, managing the integration of humans with technology in an attempt to keep residual risk acceptable, or at least tolerable.

When I refer to people it is not just the operators: the term includes designers, customers, users and repairers. In reference to organisations I mean: regulators, acquisition organisations, design houses, manufacturers and operating groups. Similarly, technology methods and processes include those associated with design, production, systems operation and equipment.

When railway organisations commence the planning to acquire new systems, they need to consider the human aspects in addition to cost, schedule, and technical performance. Integrating human consideration into system acquisition processes involves ergonomics, human performance assessment, knowing physical and psychological limitations and understanding error mechanisms. Modern IT systems provide designers with significant capabilities to quantify measure and simulate human characteristics allowing for better design decisions early in system design. Many catastrophic outcomes are the result of a technical failure, followed by human error in attempting to control or recover from those failures. This is why system test must include degraded mode testing particularly to gauge operator response.

However, one of the significant problems still to be fully overcome is the answer to the question:

Why do humans still fail to see the potential for a chain of events leading to a catastrophe?

The Three Mile Island, Chernobyl, Longford and Waterfall accidents were all foreseeable if the people involved in judgement and decisions recognised that events were heading in a bad direction.

Arie de Geus, one of the first to use scenario based planning, studied this question and examined four theories.

- Theory 1: managers are stupid – a conclusion often reached with hindsight by media, academics and investigators. Arie discounted this theory.
- Theory 2: we can only see once the crisis has opened our eyes - this theory is liked by those who think a crisis allows quick decisions, heroic management and centralised power.
- Theory 3: we can only see or understand what has been already experienced – some truth in this as other people's mistakes are the cheapest, human's rationalise only what has a basis of understanding through experience.
- Theory 4: we do not see what is emotionally difficult to see – those in the lead are reluctant to change until it is too late, the biggest fall the hardest!

None of these theories made complete sense to Arie who was studying some of the world's most successful companies that have been in continuous operation for over 50 years.

He proposed the following:

- Theory 5: we can only see what is relevant to our view of possible futures. - this theory assumes we are constantly creating time paths of hypothetical futures in our minds.

For example, some of you are probably already mapping out morning tea or lunch or the weekend and have made some preliminary decisions or set a few options.

Each of the possible futures has accompanying options for action – our mind records and stores the options, these can be referred to as our memory of the future. Events and things become meaningful if they fit with our memory of anticipated futures and we tend to take action in accordance with the preconceived options.

The more future memories we create, the more open and receptive we become to change, and the better our minds are prepared to recognise tell tale signs or react to precursor events.

Arie concluded that scenario based planning (gaming) helps create an environment for building organisation memories of the future.

Scenario based planning is a powerful tool for understanding risks to safety, testing and verification of risk control effectiveness and help in planning contingency actions to cope with error, vulnerability or threats. If organisations and individuals experience events that lead to harm or unacceptable risk via gaming or exercise they can develop memories of the future to better prepare them to take notice and react when the real events are about to occur or have occurred.

Rail is a complex hazardous industry. In understanding the context we need to recognise that:

1. hazards exist because of what we do, how we do it, where we do it and what we use to do it;

2. operations can either produce positive or negative outcomes;
3. operations must be managed to ensure the outcomes remain acceptable;
4. there is always a chance that sometimes things do not go as expected, things fail unexpectedly or go wrong. These events need to be understood, simulated under controlled circumstances and tested, for example:
 - Have you ever fired a high powered, fully automatic weapon? The first time you do and feel the power, hear the noise and see the results it can be very unsettling. Accordingly, the military need to practice and train with such weapons before they need to use them in real situations otherwise the outcomes become too unpredictable.
 - Have you ever been shot at? This is not something you hope to experience but the probability of this for the special operations groups is very high, hence their need to make their training as realistic as possible, sometimes even to the point of using live fire. This is to build their future memories so that they can focus on the mission, not be concerned with live fire situations.
5. Because hazards exist and there is the probability that things may go wrong or right there is Risk.

Conducting simulations or gaming operations in degraded states or failed states can provide invaluable insight into risks and help test control effectiveness.

To combat increased complexity and high tempo of rail operations, the rail industry needs to increase in its maturity in understanding and managing risk.

Increased maturity in risk management requires a full understanding of the operational context including what could possibly go wrong. It requires a comprehensive knowledge of the mechanisms that lead to unacceptable safety risk and harm.

Operating context comprises the functions, organisation, hazards, processes, technology, risk controls, systems, standards etc involved in railway operations. These things exist because of what you do. Because of this context, there exists a set of possible consequences, the potential for harm and the probability that things may go wrong.

The hazards and the harm normally exist in isolation of each other. It takes a mechanism such as error, an intentional act, a technical failure, a latent defect, poor control effectiveness, or some escalation action for the hazard to produce the harm. To explain what I mean by escalation action, think of a situation where a technical failure places an operation at risk. If the operator is not trained to expect or react to such a situation their subsequent action may make the consequences worse.

One of the signs of an organisation's maturity is if those charged with leadership accountability and judging significance in a risk context set the organisational goals to challenge them to move towards Zero Harm.

It is important to understand that:

- Zero Harm does not mean Zero Hazards
- Zero Harm does not mean Zero Risk

However, it does mean that the environment and the risk are studied, identified and understood such that appropriate actions are planned or have been taken to ensure that the risk of a hazard resulting in harm is managed so that it is acceptable so far as is reasonably practicable. Organisations need to understand that if you identify hazards, determine the level of risk, identify controls but do nothing more, you have deemed the risk to be acceptable whether you know it or not, at least from my perspective as a regulator.

Risk leadership in an organisation with a mature attitude requires proactive acceptance of risk and competency in leadership to comprehend, treat and resolve unacceptable risk. Complex scenario based planning and setting organisational goals that cause an organisation to strive for Zero Harm are just consultant buzz words unless there is strong safety leadership. Strong safety leadership needs to determine what the organisation's safety culture should be, how the organisation views and manages risk, and what is judged to be acceptable or unacceptable risk behaviour. At this point it is important to understand that leadership is not just vested in those with the high pay grades. All levels of management carry leadership accountability.

Changing and maintaining the right safety culture and developing maturity in the organisation's risk environment requires significant transformational change management and leadership. Such a manager needs:

- to accept ambiguity while managing complexity – expect the unexpected, there may be more than one right answer
- be flexible in their thought processes – i.e. open to new ideas, new concepts and be prepared to change direction if necessary
- have great personal integrity to inspire trust and fellowship – the standard of risk that you walk past is the standard of risk you set for the organisation

If a CEO shows that safety is important to them, everyone else will get on board. If a CEO states that 'at risk' behaviour is unacceptable and will not be tolerated, everyone else will hold the same view.

If the CEO wants to hear about 'near hit' and other safety incidents and tracks repeat incidents treating them as indicators of ineffective risk controls and potential mechanisms, that may lead to harm – the organisation will be watchful and report such things.

As previously mentioned, this accountability does not just rest with the CEO; all levels of management must demonstrate the importance of safety to themselves if they expect subordinates to treat it seriously.

The future of system safety is in the hands of leaders whether they are managers of design, system operation, or system maintenance and the manner in which they discharge their safety leadership accountability. They need to evolve in maturity with respect to understanding, managing and accepting risk. That maturity will depend on their understanding of the mechanisms that turn

hazards into harm, especially the mechanisms associated with violation, slips, lapses and mistakes. Controlling or isolating those mechanisms can be helped through simulation and scenario based management to help organisations build up their memories of the future so they might react in time when the need arises.

It is hoped that as we turn more to technology to help us do more with less, that we ensure the systems are designed to be fit for purpose but also are designed to be error tolerant and have effective recovery controls and defences.

I am sufficiently passionate about safety to continuously strive to understand the mechanisms that drive risk and improve the judgement capacity of those charged with making the decisions that count. If you are involved in system design, development, integration, test or operation; I urge you to develop a similar passion, especially in respect to human systems integration with technology solutions.

2 References

- Booher, H.R. (2003): *Handbook of Human Systems Integration*. Wiley Interscience.
- Gues, Arie de. (1997): *The Living Company; growth, learning and longevity in business*. nb publishing.
- Jacques, E., Clement, S.D. (1991): *Executive Leadership, a practical guide to managing complexity*. Blackwell publishing.

Safety Assurance: Fact or Fiction?

Carl Sandom

iSys Integrity Limited
10 Gainsborough Drive
Sherborne, Dorset, DT9 6DR, England

carl@iSys-Integrity.com

"If the facts don't fit the theory, change the facts" Albert Einstein (attributed).

Abstract

Many safety-related systems are also socio-technical systems and providing safety assurance for these systems is extremely challenging. Providing comprehensive safety assurance evidence for the technical elements of anything but the simplest of systems is impossible due to the complexity involved and these difficulties increase dramatically when the human and organizational factors have to be considered. Apart from the inherent complexity associated with the development of safe socio-technical systems, there are other reasons to believe that safety assurance claims can be overly optimistic and based more upon fiction than fact.

This paper will examine where improvements could be made to the safety assurance process. The paper will first consider some of the reasons why safety assurance claims may be based too much upon 'self-fulfilling prophesies' appealing only to confirmatory and highly subjective evidence because of inherent methodological limitations with the safety assurance process and an overreliance on professional judgement. The paper will then examine a significant but common area of neglect for safety assurance claims; specifically, the widespread fixation on technology despite the prevalence of socio-technical issues for many safety-related systems. Finally, suggestions will be made regarding how to improve the validity of safety assurance claims through the use of metaevidence.

Keywords: argument, claim, evidence, induction, metaevidence, professional judgement, safety assurance, socio-technical.

1 Introduction

Systems engineering is hard enough without adding to the complexity; yet the use of socio-technical systems in high-risk environments is prevalent despite the fact that these systems often contain a complex mix of hardware, software and firmware designed, operated and maintained by people and organisations within highly-dynamic

environments often using complicated rules and procedures. The rapid rate of technological change and the use of emerging technologies in safety-related environments have also brought with it added complexity for systems engineers and new or improved processes are required to maintain the status quo.

Safety assurance is often claimed with reference to a safety argument supported by evidence that a system is acceptably safe; this broad framework for making safety assurance claims has been around for some time and is now the generally accepted paradigm within the safety engineering discipline. This paper challenges some of the fundamental assumptions underlying the current safety assurance paradigm and argues that there are some major limitations with this approach regardless of the particular safety standard or guidance adopted.

The aim of this paper is to stimulate debate on the limitations associated with safety assurance claims made for systems which are too often overly reliant upon subjective judgement and incomplete evidence to support tenuous claims regarding mainly the technical aspects of socio-technical systems safety.

Many safety assurance process improvements could be suggested; however, this paper will restrict itself to an examination of three significant and prevalent shortcomings namely: methodological limitations; professional judgement and technology fixation.

2 Methodological Limitations

Without wishing to get too deep into the philosophical discussions regarding questions of reasoning and knowledge (see Hume (1777), Popper (1959) and Kuhn (1962) for detailed discussions); it is useful for systems engineers to consider the common approaches that underpin reasoning and the acquisition of knowledge; we do this to focus on the limitations associated with the approaches used to reason about safety. (Note: there is no definitive view on the validity of knowledge; this paper will restrict itself to the prevalent view which has prevailed since the mid 20th Century. Also, some intentional simplifications are made here for the sake of brevity).

2.1 Problems of Induction

There are two broad approaches to reasoning known as deductive and inductive. Briefly, deductive reasoning

progresses from the general to the specific. Deductive reasoning begins with a theory which is then refined into more specific hypotheses that can be tested. Specific hypotheses are further refined by collecting supporting observations. Finally, hypotheses are tested with specific data and the original theory is either confirmed or rejected. In contrast, inductive reasoning works the other way, moving from specific observations to broader generalizations and theories. Inductive reasoning begins with specific observations which suggest certain patterns or trends. From these patterns, tentative hypotheses (note the word *tentative* for the discussion later) are formulated from which general conclusions or theories are developed (Figure 1).

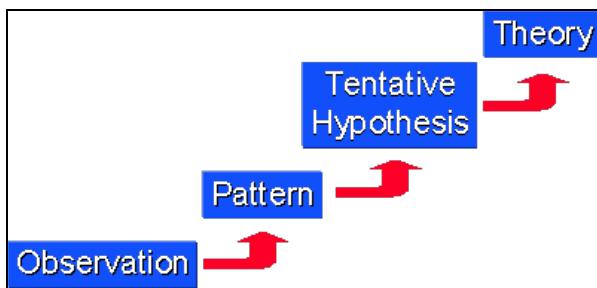


Figure 1 – Inductive Reasoning Stages

Deduction and induction processes are inextricably linked as, at some point one relies upon the other for validation. For example, a deductive safety argument may claim that a system is safe (hypothesis) then construct an argument based upon evidence (observations) to support the original claim; at some point the process will reverse and become inductive to validate the original deductive claim and vice versa.

Both inductive and deductive methods have been used for reasoning about safety even if systems engineers don't recognize those terms or use the same terminology; as discussed, the current practice for reasoning about safety assurance is for a claim (hypothesis) to be made with reference to a safety argument (pattern) supported by evidence (observation). Consequently, it is argued here that any limitations with the basic scientific approaches are also limitations with the safety assurance process.

The first significant work on the *problem of induction* was attributed to Hume (1777) and later refined by Popper (1959); Hume raised the important question of whether inductive reasoning actually does lead to knowledge and the main limitations of induction can be simplified as (Okasha 2001):

1. Hypothesizing about patterns or trends based on some number of observations can be flawed as it only takes one counter observation to nullify the hypothesis (e.g. the inference that "all swans we have seen are white, and therefore all swans are white," before the discovery of black swans). Put another way, making a safety assurance claim based upon an argument supported by some arbitrary quantity of evidence may lead to a false claim as the safety engineer may have overlooked the 'black swan' piece of evidence.

2. Past data tells you nothing about the future; therefore, it is possible that the future will turn out differently from how we believe; therefore knowledge of the future is impossible. All experimental conclusions proceed upon the basis that the future will conform to the past. Or, to put it another way, any safety assurance claim is based upon evidence that *suggests* a certain outcome based upon our past experiences; but the suggestion may be false (again, the black swan).

The problems of induction have been understood and generally accepted since 1777 but, despite that, there have been major scientific advances based upon inductive and deductive reasoning. If we accept that the problems with induction are irrefutable, and most philosophers and scientists do, we could conclude that the limitations are academic and meaningless in the context of engineering methods; however, for safety engineering at least, this is not so as flawed hypotheses may lead to unexpected failures and catastrophic accidents.

2.2 Tentative Hypotheses

For safety assurance purposes we must be proactive in trying to identify the flaws in a safety assurance claim otherwise a claim made at the outset that a system is safe may simply become a self-fulfilling prophecy as supporting evidence is sought to the exclusion of any counter-evidence that may negate the claim. Put simply, safety claims should be considered only as *tentative hypotheses* until strongly challenged by attempts to prove them false.

Kinnersly (2011) puts forward a similar opinion and suggests an alternative view to the accepted safety assurance paradigm; he argues that scientific methods should be adopted in safety engineering whereby a safety claim is examined from the view of hypothesis and challenge rather than the current norm whereby a claim that a system is safe is shown to be true as a logical consequence of appropriate (or compelling) evidence. One of the findings of the Haddon-Cave report (2009) into the loss of a UK Nimrod aircraft in Afghanistan made numerous criticisms of the way safety claims are made and concluded that the safety assurance process is not 'new' suggesting that well established (i.e. old) scientific methods have relevance for the current paradigm.

These points are consistent with the assertion made here that the inherent problems with induction should lead to a change in approach for safety engineers to challenge tentative hypotheses by proactively seeking evidence to counter claims made about the safety of a system.

2.3 Black Swans

The term 'Black Swan' is used in philosophy as a metaphor for something that hasn't been observed and therefore its existence is assumed to be improbable but not impossible. The term originates from the ancient Western conception that all swans that had been observed were white and (by the logic of induction) it was

therefore concluded that black swans could not exist. However, black swans do exist and they were first discovered in Australia in the 17th Century. Taleb (2008) takes the metaphor further and raises the prospect of 'Black Swan Events' which he characterizes as:

1. Having a central and unique attribute and high impact; his claim is that almost all consequential events in history come from the unexpected, yet humans later convince themselves that these events are explainable in hindsight.
2. The probability of rare Black Swan Events cannot be computed using scientific methods owing to the nature of the small probabilities involved.

Taleb (2008) makes the general point regarding the shortcomings of the inductive scientific method and makes a case for a new approach which attempts to answer improbable "what if" questions which he refers to as 'counterfactuals'. Interestingly, Perrow (2011) used a counterfactual approach (although he didn't refer to it in these terms) when he predicted almost exactly the failure mode of the recent Fukushima Daiichi reactors (Ladkin (2011)):

"A hurricane could take out the power, and the storm could easily render the emergency generators inoperative as well" (Perrow 2011, p134);

"No storms or floods have as yet disabled a plant's external power supply and its backup power generators". (Perrow 2011, p173).

The failure modes were evidently not foreseen by the Fukushima safety engineers as a claim was made that the Fukushima plant was acceptably safe; however, a counterfactual safety argument like Perrow's could have challenged that assertion. Clearly there is a degree of hindsight to this now, and safety engineers typically deal only with 'credible' issues but the general point being made here is that safety-related systems developers should question, justify and document what is assumed to be credible and consider potential Black Swan events.

2.4 Summary

The key point made here is that the collection and analysis of safety evidence should be based on proactively and explicitly challenging any claim that the system is safe rather than merely seeking evidence to confirm it. To paraphrase Kinnersly (2011), safety professionals need to adopt a 'challenge the claim' mentality to safety assurance rather than accept self-fulfilling arguments backed up only by confirmatory evidence. In addition, the boundaries of credibility should be challenged and Black Swan events considered; after all it is usually improbable events such as those at Fukushima that are found to be the primary causal factors for most major disasters.

It has been argued here that safety assurance evidence can be deficient due to the inherent problems of induction and improbable events; however, it is also argued that the evidence that is presented can be over-reliant on

professional judgement which is also an inductive process.

3 Professional Judgement

Professional judgement (or expert opinion) can be defined as the ability of a person or group to draw conclusions, give opinions and make interpretations based on a combination of evidence from diverse sources such as experiments, measurements, observations, knowledge and experience (McKenna and Mitchell 2006). Professional judgement is frequently used by systems developers of all disciplines and it relies upon a combination of impartial and biased facts and opinions and, for anything but simple scenarios, subjectivity can be hard to discriminate from objectivity. For example, the problems of perception when applying professional judgement to decisions on risk have been well documented (see Adams 1995).

Professional judgement is often used when an expert doesn't have any accurate or statistically significant data and the order of magnitude required for the solution to be acceptable is estimated by applying judgement gained through a combination of: academic training; experience and professional development. Professional judgement can be considered poor if highly subjective evidence is accepted as fact without consideration of where or how the evidence is derived and without an appreciation of when it is invalid. Safety assurance claims are founded upon professional judgement and it is useful to consider examples of how conclusions, opinions or interpretations may be derived from incomplete or inadequate evidence.

3.1 Statistical Inference

Safety assurance claims often need to be made for systems which are fielded before the existence of sound empirical data and claims are therefore based upon a high degree of professional judgement. In the absence of empirical data, systems developers must make statistical predictions *a priori* when, for example, considering technical or human failure rates and their associate risks. Clearly, professionals do not need to be 100% certain about something before it can be considered *a priori* knowledge; however, the point made here is that making safety claims based upon subjective judgements for which there is little evidence must be avoided; particularly in safety-related systems.

However, that is not always the case, professional judgement may be applied for example for software safety assurance and some level of inferred safety integrity may be claimed based upon evidence of software reuse in an evolving product which has been fielded on multiple platforms over a significant period of time. However, claims based upon software reuse can be based upon flawed assumptions; for example, the software (and perhaps even the hardware platform) may have been subject to considerable changes for maintenance or improvement over the period of time considered effectively invalidating any claims.

Statistical inference can lead to systems safety claims based upon a circular argument whereupon a judgment is

based on a probability when the probability was based on judgement. Vick summarizes this situation neatly with the phrase:

“...subjective probability is judgement’s quantified expression” (Vick, 2002, p393)

This situation occurs throughout the safety assurance process; particularly in those analyses based upon quantitative techniques and methods where subjective opinion is based upon subjective opinion without taking into account their source.

3.2 Assurance Gap

In addition to using judgement for statistical inferences, opinion is also often used to bridge assurance gaps. Complex systems cannot be tested exhaustively to provide definitive evidence that the required standards of safety assurance have been achieved; for example, a system would need to be tested continuously for more than 10 years, under operational conditions, with no dangerous failures and no system modifications to demonstrate that it met the IEC 61508 (2010) SIL1 target of $10E^{-6} < p_{fh} < 10E^{-5}$ (Littlewood & Strigini 1993).

Thomas (2004) points out that the lowest integrity level that current safety standards consider safety-related are associated with a probability of dangerous failure per hour that is in practice too low to be demonstrated and therefore engineering judgement must be applied by various professionals to justify claims made about systems safety. If a system cannot be exhaustively tested, the resulting assurance gap must be bridged with reference to professional judgement which, as history has shown, is not infallible.

3.3 Summary

For these reasons, and many others, safety assurance is ultimately a matter of professional judgement. Safety-related system developers in particular have a responsibility to show that where professional judgement has been applied and, for safety assurance claims, that it must be defensible. The application of professional judgement is a necessity for any systems development; however, it remains problematic; particularly for safety-related systems development.

It has been argued here that safety assurance evidence can be deficient due to methodological limitations with the safety assurance process and also that safety claims may be over-reliant on professional judgement. However, perhaps the most significant limitation for safety assurance claims is the widespread fixation on technology even for obvious socio-technical systems.

4 Technology Fixation

A socio-technical system is a system composed of technical and social sub-systems or elements; for example, Air Traffic Control Centres or Nuclear Power Stations are socio-technical systems with people organized into social structures, such as teams or departments, to do work for which they use technical sub-systems like radars, computers, radios etc. The term

‘*socio-technical system*’ and the socio-technical approach to systems design was first used by Eric Trist (1981) and presented as a radical alternative to the scientific management approach (Taylor 1911).

The socio-technical systems approach is devoted to the effective integration of both the technical and social systems and these two aspects must be considered together for safe systems development because what is optimal for one component may not be optimal for the other and design trade-offs are required. Paradoxically, the prevalent approach to safety-related systems development is often to design the technical ‘system’ and let the operators and maintainers adapt to it. It is useful to consider why safety-related system developers do not always address the socio aspects as well as the technical.

4.1 Scope & Complexity

Many safety-related systems are socio-technical systems; yet, they are often developed predominantly by systems engineers and often have little or no explicit input from human or organizational factors experts. As well as traditional systems engineering expertise, knowledge is also required from other disciplines such as human factors and organizational factors experts to ensure that socio-technical systems are designed to balance the trade-offs necessary for safe systems.

Simplistically, a socio-technical system may be considered a combination of people and technology; however, they are much more complex. Consider the typical elements that comprise a socio-technical system and the full diversity of expertise required to provide safety assurance for each element (Computing Cases 2011):

1. **Hardware and software.** These elements are likely to be an integral part of any socio-technical system. Software often incorporates social rules and organizational procedures as part of its design making them difficult to identify and to change in safety-related systems. Providing safety assurance for system hardware and software elements is relatively easy compared with the non-technical elements.
2. **People.** Individuals, groups, roles (e.g. support, training, management, engineer etc.). People can exert a positive and a negative influence on system safety and humans can alternatively be considered as ‘hazard’ or ‘hero’ depending upon the circumstances (Sandom 2007). Ideally, an interdisciplinary approach should be taken to safety-related systems development through an integrated application of Human Factors and Systems Engineering methods and techniques.
3. **Procedures.** Official and actual procedures, management models, reporting relationships, documentation requirements, rules and norms are all parts of a system and can affect its safety. Procedures describe the way things are done in an organization (or at least the official version of how they should be done) and their analyses are

essential for understanding complex socio-technical systems.

4. **Laws and regulations.** Laws and regulations are like procedures but they carry special societal sanctions if the violators are caught. Regulations are often the basis upon which system requirements are derived and they must be taken into account for the design and maintenance of the other system elements throughout the life of a system.
5. **Environment.** The complexities of the environment within which a system operates must be taken into account for any safety assurance claim. This includes aspects such as weather, and other physical conditions within which the socio-technical operates.

This vast scope, and the resulting complexity, presents a challenge for systems developers who need to consider the safety-related aspects of the entire system and then to focus the limited resources available on the most critical system functions.

The scope of any safety assurance claim must cover all these elements for socio-technical systems. If the risks associated with the non-technical elements are not considered a system will not achieve the required level of safety assurance. If the mitigations provided by the non-technical elements are not considered the technical elements may be over engineered at unnecessary cost to achieve a target level of safety assurance.

4.2 Summary

In the absence of a holistic approach to socio-technical systems safety assessment, it is tempting to concentrate safety assurance 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 and organizational 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.

Perhaps the most significant shortcoming of many safety assurance claims is the widespread fixation on technology. The conclusion to be drawn from this is that in many instances safety claims at best provide only limited safety assurance as the prevalent errors in socio-technical systems are often related mainly to issues associated with human and organizational factors.

5 Improving Safety Assurance

From the previous discussions, it was asserted that there are some significant limitations on the veracity of the evidence supporting safety assurance claims which are caused by methodological limitations, professional

judgement and technology fixation. A safety claim can be backed up with a perfectly logical argument but still fail to provide assurance if the evidence is inadequate (McDermid 2001). The main aim of this paper is to stimulate debate on the limitations associated with safety assurance; however, some suggestions will now be made on how to improve the validity of safety assurance claims through the use of what is described here as metaevidence.

The prefix ‘meta’ is used to describe a concept which is an abstraction from another concept; for example metacognition could be described as ‘thinking about thinking’. Assertions have been made in this paper regarding the perceived shortcomings of safety assurance claims and, specifically, their reliance on incomplete and/or unconvincing evidence. To address the shortcomings described, it is suggested here that metaevidence (i.e. evidence about evidence) should be sought to support a claim that safety assurance evidence is both comprehensive and compelling.

5.1.1 Comprehensive Evidence

Some general improvements can be made to the safety assurance process by ensuring that the scope of the safety evidence is comprehensive by addressing the issues previously discussed. Specifically, metaevidence should be sought to take into consideration the following:

1. **Challenge Claims.** Evidence should be actively sought to challenging systems safety claims rather than simply focusing upon confirmatory evidence which is the norm. A review of three of the major safety standards in common use today revealed that only UK Defence Standard 00-56 (MoD 2007) contains a requirement to consider counter-evidence and this is not developed further in the guidance (Kinnersly 2011). Pragmatically, this will require a sufficient degree of independence in the overall safety assurance process as the person(s) responsible for making a safety claim are not well placed to try breaking a safety claim; the same principle is applied for independent validation and verification in systems engineering.
2. **Consider Black Swan Events.** Safety assessments must necessarily be bounded and it is normal practice to focus only on what is perceived to be credible; however, the bounds of credibility need to be agreed and evidence should be presented to back up all related assumptions made by systems developers. Something that may be considered incredible during system development may be considered probable later in the operational life of the system so assumptions must be revisited periodically in light of emerging technologies and other changes. Analysing the incredible may seem like an unnecessary task; however, a brief examination of many disasters will reveal that the improbable has actually occurred (e.g. Fukushima).

3. **Examine Subjectivity.** All safety assurance activities rely upon professional judgement which is inherently subjective and should therefore be critically examined to ensure that the resulting safety claims are reasonable and remain so over time. Statistical inference is particularly sensitive to error for quantitative analyses (e.g. Fault Tree or Human Reliability Analyses) and the assurance gap created by a lack of testing is another area of focus. Systems developers should seek evidence that any professionals applying professional judgement to safety assurance claims are competent to do so. In addition, it is equally important to ensure that the application of professional judgement to safety-related issues is not simply the opinion of a single person and a consensus from a group of competent professionals should be formed.
4. **Extend Scope of Analyses.** The scope of systems safety assurance activities should be extended from the norm to include all elements of socio-technical systems which requires expertise and contributions from different disciplines (e.g. engineering, sociology, cognitive psychology etc.). Ignoring the non-technical aspects of many safety-related systems has a significant impact on the actual safety assurance provided. Programme managers should ensure that interdisciplinary teams are formed for the analysis of safety in socio-technical systems; despite the lack of regulation or guidance in this area provided by the primary safety standards. Consider the simple reality that in some domains human factors account for more than 90% of accident or incident causal factors (Sandom 2004); yet the human factors are often not been properly addressed making system safety assurance claims fictional.

5.1.2 Compelling Evidence

In addition to questions of comprehensiveness, safety evidence should be assessed to determine if it is convincing. The credibility of safety evidence should be assessed to determine where it comes from and if it is adequately representative of the claims being made. Metaevidence should be sought to take into consideration the following possible evidential criteria:

1. **Misrepresenting data.** Data can be deliberately or unintentionally represented. Data can be misrepresented deliberately by claiming that it suggests something when it does not; this can be the case with safety evidence for example when programmes are under severe pressure to meet budgets, milestones and targets. A further way in which data may be misrepresented is if it is presented selectively and a varied data set is described by focusing only on certain sections of it. Data can be unintentionally misrepresented as conclusions are hurriedly based upon initial evidence found to fit a given proposition.

2. **Insufficient data.** A common problem with evidence sampling is drawing conclusions from insufficient data; this is related to the problem of induction (see 2.1). It is not enough to observe a couple of instances of data that support a safety claim; however, it is not easy to decide how much data is statistically sufficient. Sufficiency of data is a matter of degree; the more evidence the better and the amount of confidence that we can have in an inference grows gradually as more evidence is brought in to support it.
3. **Unrepresentative data.** Simply having a lot of data is not enough to guarantee that a claim is valid; it is generally important that the data has been drawn from a representative sample of sources and obtained under a variety of different conditions. For example, it may not be enough to show that requirements-based testing has been undertaken for software, a valid claim may also require some proof of absence of errors during operation of the system. Special attention should be paid to evidence relating to evolving products where claims are made based on past performance without properly considering the impact of configuration changes or changes in the context of use. For example, software safety evidence taken from use in fixed wing aircraft may not be valid in rotary winged aircraft.

5.1.3 Summary

In summary, it is suggested that metaevidence should be sought to support a claim that safety assurance evidence is both comprehensive and compelling before a system is operational and throughout the operational life of a system.

It is recognised that metaevidence is itself evidence and it can be argued recursively that it should also be comprehensive and compelling and require evidence to demonstrate that it is so. However, at some point the law of diminishing returns must apply and professional judgement (or consensus opinion) must be applied to bring the process to a halt when little value is being added. Nonetheless, it is asserted here that at least one-level of metaevidence should be sought for all but the simplest safety-related systems.

6 Conclusions

There are many safety-related, socio-technical systems in operational use today and many of these have based safety assurance claims on inductive arguments, a great deal of professional judgement and have only considered the technology; yet, thankfully there are few catastrophic accidents or serious incidents associated with these systems. The relatively small number of catastrophic accidents or serious incidents associated with these systems could lead us to conclude that our safety assurance processes are sufficiently robust; however, this is not the case.

A safety claim will usually be made relative to an acceptable level of risk and it is suggested here that a

great deal of uncertainty and sensitivity of these claims can be attributed to the issues raised in this paper. A safety claim is *not an incontrovertible fact* and the nature of the safety assurance process means that it is often difficult to determine the robustness or validity of a claim. It is often impossible to determine how close to being unsafe a system might be.

From the arguments presented in this paper, it may be concluded that it is not possible to provide valid system safety assurance without major professional input from sociologists and cognitive psychologists and without using sound scientific methods. However, safety professionals shouldn't 'throw the baby out with the bathwater' as, despite the issues raised in this paper, there are relatively few accidents given the vast number of complex, safety-related systems in existence.

Although there is room for improvement in current safety assurance best practice it is not suggested here that a paradigm shift is required, merely an evolution of the existing practice to address the major limitations, some of which have been discussed in this paper, and to enable safety professionals to better separate fact from fiction.

7 References

- Adams. J (1995): *Risk*. Routledge, London.
- Computing Cases: <http://computingcases.org>. Accessed 26 April 2011.
- Haddon-Cave, C. (2009): *An independent review into the broader issues surrounding the loss of the RAF Nimrod MR2 aircraft XV230 in Afghanistan in 2006*. Her Majesty's Stationery Office.
- Hume, D. (1777): *Enquiries Concerning Human Understanding and Concerning the Principles of Morals*, Niddich. P. H. (Ed.) 3rd Edition 1975. Oxford University Press.
- Kuhn, T. S. (1962): *The structure of scientific revolutions*. University of Chicago Press.
- IEC 61508 (2010): *Functional safety of electrical/electronic/programmable electronic safety related systems*. International Electrotechnical Committee. Ed. 2, 2010-04. Geneva, Switzerland.
- Kinnersly, S. (2011): Safety Cases – what can we learn from Science? in Dale, C., and Anderson, T. (eds), *Advances in Systems Safety, Proc. Safety-Critical Systems Club Symposium 2011*, Springer-Verlag, London.
- Ladkin, P. (2011): posted on Safety Critical Mailing List, <http://www.cs.york.ac.uk/hise/safety-critical-archive/2011/>. Accessed 26 April 2011.
- Littlewood, B. and Strigini. L. (1993): Validation of Ultra-High Dependability for Software-based Systems in *Communications of the ACM* 36 (11) 69-80.
- McDermid, J. (2001): Software Safety: Where's the evidence? *6th Australian Workshop on Industrial Experience with Safety Critical Systems and Software* (SCS'01), Brisbane. Conferences in Research and Practice in Information Technology, Vol. 3 P Lindsay, Ed.
- McKenna. S. and Mitchell. J. (2006): *Professional Judgment in Vocational Education and Training: A Set of Resources*. 2nd Ed. Commonwealth of Australia, Department of Education, Science and Training.
- MoD (2007): *Defence Standard 00-56 Issue 4. Safety Management Requirements for Defence Systems: Part 1 Requirements; Part 2 Guidance on Establishing a Means of Complying with Part 1*. UK Ministry of Defence.
- Okasha, Samir. (2001): What did Hume really show about induction? *The Philosophical Quarterly*, 51 (204).
- Perrow, C. (2011): *The Next Catastrophe: Reducing Our Vulnerabilities to Natural, Industrial, and Terrorist Disasters*. Princeton University Press.
- Popper, K. R. (1959): *The Logic of Scientific Discovery*, New York: Basic Books.
- 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.
- Sandom, C., and Harvey, R. S. (2004): *Human Factors for Engineers*, The Institution of Electrical Engineers, UK.
- Sandom, C. (2007): Success and Failure: Human as Hero – Human as Hazard. *Conferences in Research and Practice in Information Technology (CRPIT)*, Vol. 57. T. Cant, (Ed.), *12th Australian Conference on Safety Related Programmable Systems*, Adelaide.
- Taleb, N. N. (2008): *The Black Swan: The Impact of the Highly Improbable*. Penguin.
- Taylor, Frederick Winslow (1911), *The Principles of Scientific Management*, New York, NY, USA and London, UK: Harper & Brothers
- Thomas, M. (2004): Engineering Judgement. *Conferences in Research and Practice in Information Technology*, Vol. 38. Cant, T. (Ed), *9th Australian Workshop on Safety Related Programmable Systems*, Brisbane.
- Trist, E. L. (1981). *The evolution of socio-technical systems: A conceptual framework and an action research program*. Ontario Quality of Working Life Center, Occasional Paper No. 2.
- Vick. S. G. (2002): *Degrees of Belief: Subjective Probability and Engineering Judgement*, American Society of Civil Engineers Press.

System safety in hybrid and electric vehicles

Dr David D. Ward

MIRA Limited

Watling Street, Nuneaton, CV10 0TU, UK

david.ward@mira.co.uk

Abstract

Road vehicles have an increasing reliance on electronic systems to control their functionality and to deliver the feature and attribute demands made by manufacturers, legislators and consumers. This trend is particularly evident in the new generation of more energy-efficient vehicles that includes hybrid vehicles and full electric vehicles. The architectures of these vehicles are characterized by a greater degree of integration and interaction between the systems, as well as the introduction of new types of system with unique potential failure modes. As a result, system safety is a central part of the design and implementation process for these vehicles.

In this respect a new standard, ISO 26262 "Road vehicles — Functional safety" is in preparation. It sets out requirements for managing functional safety, hazard analysis and risk assessment, and the development and verification of systems, hardware and software. Nevertheless, in hybrid and electric vehicles functional safety is only one part of the overall process of system safety, which encompasses other domains such as electrical safety and crashworthiness.

This paper will give a brief introduction to the concepts and challenges of system safety when applied to such vehicles, including a discussion of the role of ISO 26262 and some of the key principles of that standard, including the concepts of automotive safety integrity level (ASIL), safety goals and safety concepts. The implications of the standard on emerging vehicle technology will also be examined. Finally, the need for an holistic approach to system safety in such vehicles will be presented.

Keywords: Functional safety, electrical safety, hybrid vehicles, electric vehicles, autonomous vehicles, UGV, ISO 26262.

1 Introduction

Modern road vehicles have an increasing dependence on electronic systems to control their functionality and to deliver the demands made by manufacturers, legislators and consumers for safety, environmental efficiency, comfort and brand differentiation. This trend is seen in particular focus in the new generation of more efficient vehicles, typically called "low carbon" vehicles. Examples of low carbon vehicles include hybrid vehicles and electric vehicles. Low carbon vehicles are

characterized by a greater degree of integration and interaction between the electronic systems, as well as the introduction of new types of electronic system. As a result, system safety is a central part of the design and implementation process for these vehicles, and continues to grow in importance.

2 Automotive system safety and functional safety

System safety uses the concepts of systems engineering and systems management in the processes of ensuring the safety of a product. In outline the process for addressing system safety takes the form of:

- A hazard analysis and risk assessment to identify the potential hazards associated with the system and the associated risk;
- The identification and implementation of measures to control, reduce or remove the risks, such that the residual risk associated with the hazards is at a defined acceptable level;
- A safety assessment to demonstrate that the risk reduction has been correctly identified and implemented. The safety assessment is frequently conducted by a party with a degree of independence from the developers of the system.

It should be emphasized that system safety is a very wide area. In the automotive context, system safety covers many of the traditional safety disciplines as well as the new safety challenges introduced by innovative systems. Safety aspects in a vehicle have traditionally focused around crash safety. Safety measures can for example be categorized as "active safety" (measures which help prevent a vehicle from being involved in an accident or which can reduce the severity of an impact) and "passive safety" (measures which help reduce the risk of injury to the occupant if the vehicle is involved in an accident). More recently, the deployment of advanced electronic systems has lead to the introduction of terms such as "integrated safety" (EASIS 2011) to describe more wide-ranging and integrated approaches to safety.

These overall trends towards the greater use of electronic systems to achieve safety serve to emphasize the importance of the inherent safety of these systems. Frequently these systems involve a higher degree of integration of the systems, and a higher degree of interaction between them. Thus, taking a systems-led approach to the design and development of these features is essential, and this philosophy should also be reflected in the approaches to the safety of the systems.

Furthermore, in low carbon vehicles further levels of integration and interaction are introduced, as well as novel systems that have their own safety aspects.

- In low carbon vehicles, there is a trend away from imperative control of the functions to goal-based control. In a traditional vehicle, for example, the driver directly commands the engine and brakes to speed up or slow down the vehicle. In a typical hybrid vehicle, the driver requests that the vehicle speeds up, and the hybrid system controller decides whether the required torque should come from the internal combustion engine, or the electrical machine, or both.
- Hybrid and electric vehicles introduce higher voltage components, meaning that electrical safety (that is, preventing human contact with potentially fatal levels of voltage and/or current) is a new issue to be considered.
- Linked to this, the size and location of the components (particularly the traction battery) require additional considerations in the crash engineering of the vehicle.

These areas cannot be considered in isolation from each other. In the example of electrical safety, part of the necessary level of safety is achieved through design measures to prevent contact with the hazardous voltage, such as specially-constructed connectors that prevent direct contact with conductors. However, part of the safety is also achieved through electronic systems, such as a fault monitoring system that checks whether there is a leakage of hazardous voltage onto the vehicle chassis and shuts down the higher voltage system if so. Thus, to achieve the necessary level of electrical safety, correct functionality of an electronic system is also required.

The discipline of ensuring that safety is maintained through the correct functionality of electronic systems is known as “functional safety”. However the foregoing discussion serves to emphasize that functional safety is a subset of system safety. Whilst the state-of-the-art practices for functional safety are based on system engineering principles, in the modern vehicle an overall approach to the safety of the vehicle treating the entire vehicle as a system is clearly necessary.

3 An automotive standard for functional safety — ISO 26262

The discipline of functional safety is generally a mature one. A particular milestone is that work started in the early 1990s on what has now become the international standard IEC 61508 (IEC 2010). First published in 1998, the standard has recently been updated to a second edition. Although originating in the industrial process control sector, IEC 61508 has become a generic standard and the baseline standard for any industry to develop its own requirements for functional safety. As early as 1994, an automotive interpretation of the requirements of this standard was published by a UK consortium (MISRA 1994) and IEC 61508 has also been applied directly to automotive systems.

Nevertheless, there are some key challenges in applying IEC 61508 to road vehicle systems. Perhaps the most significant issue is that in IEC 61508, safety functions are considered separately from the control functions. IEC 61508 has the concept of the “equipment under control” with its own control systems, and designated separate safety functions are added where necessary to achieve the required level of safety. In contrast, in traditional automotive systems the safety functionality is rarely distinguishable from the normal functionality. For example, in an electronic engine controller, the required functionality is to produce torque in response to driver demands; however if this torque is produced incorrectly this is potentially a safety issue.

Some further issues with applying IEC 61508 directly are discussed in (Ward 2008) and include:

- The principles for hazard analysis and risk assessment in IEC 61508 always require calibrating to the specific industrial application, and contrary to popular misconceptions IEC 61508 does not give a normative basis for this.
- The use of distributed development responsibilities in the automotive supply chain, including the relationship between vehicle manufacturers, major systems suppliers and the lower supply chain is not reflected in IEC 61508;
- Final safety validation for automotive systems is performed before release of a vehicle to volume production, often in conjunction with a statutory process such as “Type Approval” in Europe.
- Vehicles are not restricted to being operated in a specific location or restricted environment.
- The human is an important part of the control loop for vehicle systems, and so human reactions must be considered in designing systems. In this context it should be observed that compared to other industries, the operators of vehicle systems generally receive little or no training (either initial or ongoing) in the operation of the vehicle’s safety-related systems. Therefore the reactions of an “average” human to perceived failures have to be considered.
- There is only a limited formal maintenance regime for automotive systems.
- There are few if any systems for collecting in-service data about incidents that are potentially attributable to safety-related systems.

From the foregoing discussion it is clear that an automotive-specific version of IEC 61508 should be developed. One example of such a standard is ISO 26262 (ISO 2010). ISO 26262 was developed against the background of the issues listed above and seeks specifically to address these.

Although currently in development and not due to be published as a full international standard until later in 2011, a public draft has been available since July 2009 and the standard has rapidly become established as representing “state-of-the-art” in the development of automotive electronic systems, particularly in Europe, North America and Japan.

4 Key concepts in ISO 26262

In this section, some of the key concepts of ISO 26262 are introduced; in particular:

- The safety lifecycle;
- Automotive safety integrity levels (ASILs);
- The processes for specifying safety requirements.

4.1 The safety lifecycle

In common with IEC 61508, ISO 26262 specifies a safety lifecycle to cover the essential requirements for achieving functional safety. The safety activities are divided into three main areas.

4.1.1 Management of functional safety

This subject is covered in ISO 26262 Part 2 and specifies requirements for overall safety management in an organization, including requirements for a safety culture within the organization and for competence management of personnel who will undertake functional safety activities. This Part further specifies the requirements for management of functional safety during the development of the item, including the need for appointment of a safety manager, the production of a safety plan for the functional safety activities, and the required confirmation measures. “Confirmation measures” are requirements for reviews of certain work products that have to be performed with a degree of independence from the persons responsible for generating the particular work product. These confirmation measures also include a requirement for an independent safety assessment at the highest ASILs.

4.1.2 Concept phase

This subject is covered in ISO 26262 Part 3 and specifies requirements for item definition, hazard and risk analysis, and the specification of the functional safety concept. These requirements are discussed further in the next two sections.

4.1.3 Development phase

This subject is covered in ISO 26262 Parts 4 to 9 and specifies requirements for the design, implementation and verification of the item. Part 4 in particular covers product development at the system level; whilst Parts 5 and 6 cover product development at the hardware and software level respectively. It is important to note that development of any item is led through Part 4, which includes the requirements for safety requirements specification at the top level of the design (see below) as well as the integration and safety validation. Parts 5 and 6 are concerned with the specific processes for designing and implementing hardware and software. Parts 4, 5 and 6 draw heavily upon the concept of the “V model” for developing systems.

4.2 Automotive safety integrity levels

A key requirement of ISO 26262 is the use of automotive safety integrity level (ASIL), which is defined as “one of four levels to specify the item’s or element’s necessary requirements of ISO 26262 and safety measures to apply

for avoiding an unreasonable residual risk with D representing the most stringent and A the least stringent level”. This is analogous to the concept of safety integrity level (SIL) in IEC 61508, with the following important differences:

- The 4 ASILs (A, B, C, D) of ISO 26262 do not map directly to SILs of IEC 61508. ASILs A, B and D are very approximately equivalent to SILs 1, 2 and 3 respectively; although there are some important detailed differences. There is no equivalent to SIL 4 in ISO 26262, and ASIL C represents requirements that correspond roughly to SIL 3 on the left-hand side of a “V” model and to SIL 2 on the right-hand side of a “V” model.
- ASILs do not contain any normative (i.e. “must do”) requirement for probabilities. In contrast, IEC 61508 SILs have a normative probabilistic requirement, although IEC 61508 does acknowledge that in practice this can only be demonstrated in respect of the random failures of hardware. ISO 26262 does however specify optional probabilistic targets for ASIL, which are associated with the failure to achieve the safety goals (see below).

The ASILs are allocated through a process of hazard analysis and risk assessment. Such a process covers:

- Hazard identification — using a well-defined and structured process to identify the potential hazards associated with the item.
- Hazard classification — using three parameters to assess the risk associated with the potential hazards. The parameters are severity of the (eventual outcome of the) hazard, likelihood of exposure to the hazard depending on operational conditions, and the controllability of the situation by the driver. Each parameter is ranked on a subjective basis using qualitative classes. There are typically three or four classes for each parameter.
- Risk assessment — by combining the three parameters the risk associated with the hazard is determined. This is specified using ASIL, which is also the means of specifying the risk reduction requirements if all of the risk reduction is to be achieved through an electrical or electronic system. ISO 26262 does permit the risk reduction to be allocated to safety elements of “other technologies” but ASIL is not to be used for the purposes of this allocation.

4.3 Safety requirements specification

The specification of safety requirements in ISO 26262 is given at four levels:

- Safety goals, which are the top level statements of the safety requirements necessary to prevent or mitigate the hazards. Each hazard is required to have at least one safety goal. Crucially, the ASIL identified for the hazard is allocated to the safety goal, and all the safety requirements subsequently

derived from a safety goal are required to inherit this ASIL.

- Functional safety concept. This is the top level specification of functional safety requirements to fulfil the safety goal. At least one functional safety requirement is required for each safety goal. The functional safety concept can be created without knowledge of the system architecture.
- Technical safety concept. This is created during the initial design of the system, and refines the functional safety requirements into specific technical safety requirements that can be implemented, taking into account the system architecture. This step includes the allocation of technical safety requirements to hardware and software.
- Detailed hardware and software safety requirements. As the detailed hardware and software design progresses, the technical safety requirements are iteratively refined into specific requirements that can be implemented at the hardware and software level.

The safety goals and functional safety concept are specified during the “concept phase”. Since the functional safety concept can be specified independently of any knowledge of the implementation of the system, this is typically viewed as being the responsibility of the developer of the item. In the typical automotive supply chain this is often the vehicle manufacturer. In contrast, the technical safety concept is developed during the “development phase” (Parts 4 onwards) and with knowledge of the system design. It is therefore often viewed as a supplier responsibility.

A key contrast with IEC 61508 can be seen here. In IEC 61508, SILs are related to assuring the reliability of safety functions. In ISO 26262, ASILs are related to assuring that the safety goals are not violated. This distinction reflects the fact that in traditional automotive systems, it is not usually possible to identify a “safety function” that is completely separate from the nominal performance of the system.

An example of the thinking behind the structure of the safety requirements in ISO 26262 can be seen in the “E-gas” concept that has been a standardized approach between some of the European vehicle manufacturers for many years (VDA 2004):

- A hazard of electronic throttle control is incorrect torque generation;
- The safety goal is to prevent incorrect torque;
- Part of the functional safety concept is to monitor the torque generated by the engine, compare it with the torque demanded by the driver through the accelerator pedal (as well as torque up/down requests from other systems e.g. cruise control, stability control), and limit torque if the delivered torque is significantly different from the demand.
- The technical safety concept specifies how this will be achieved, for example through hardware and software plausibility checks and redundant engine shutdown paths for both ignition and fuelling.

5 Implications for emerging technology

The previous sections have introduced ISO 26262 and demonstrated how it fulfils many of the requirements for a functional safety standard for the automotive industry. Nevertheless, the standard was developed against the background of the current generation of automotive electronic systems and may not be fully applicable to some emerging technologies. This is particularly the case in low carbon vehicles and autonomous vehicles.

5.1 Low carbon vehicles

A key difference between low carbon vehicles and conventional vehicles is the much greater level of integration and interaction between systems and functions. This paper has already argued that a systems-led approach to the safety of such vehicles is required, encompassing crash safety and electrical safety as well as functional safety. The overall system safety approach of identifying hazards and their associated risks, identifying the required risk reduction methods and confirming their correct implementation is equally applicable to any safety domain in the vehicle. It is therefore recommended that a unified approach be adopted, whereby the means of hazard classification in particular is not restricted to a particular technology. An example of such an approach can be found in the MISRA Safety Analysis guidelines (MISRA 2007), where an intermediate parameter of “presumed hazard risk” is used. Allocation between different means of risk reduction can be performed based on this parameter. For example, considering the hazard of “electric shock during maintenance” the MISRA risk parameter could be used to determine the allocation of risk reduction between electronic systems (e.g. a high voltage interlock loop) and the regulatory requirements for protected connectors. The principles of this allocation are discussed further in (Ward *et al* 2009) and will be the subject of a future MISRA publication.

Furthermore, for achieving the required functional safety of functions such as high voltage interlock, fault detection and even certain aspects of battery management, it may be that the IEC 61508 model of risk reduction through a separate “safety function” is more appropriate. Again the MISRA Safety Analysis approach (MISRA 2007) includes an alternative means of performing hazard classification that is more appropriate for such functions.

Finally, some of the technologies used in low carbon vehicles (notably electrical machine control, battery management and high voltage fault detection) are not unique to the automotive industry. Other safety-relevant industries where these technologies may be used may have their own interpretations of IEC 61508 (e.g. IEC 61800-5-2 for variable speed electrical drives (IEC 2007)). One of the guiding principles of applying IEC 61508 and producing industry-specific versions of that standard is that a specific safety integrity level (SIL) should mean the same level of risk reduction regardless of the industry sector, even though the definition of risk and the level of acceptable risk may be different. Thus, an electrical machine controller developed to SIL 3 requirements in the automotive sector should in theory be capable of being used in the machinery sector where the

risk reduction requirements allocated to this device are SIL 3 or less. However, since the ASILs of ISO 26262 do not translate directly to and from SILs, this may prove a major challenge in such a cross-sector application.

5.2 Autonomous vehicles

There is considerable interest in both civilian and defence applications of the use of autonomous ground vehicles (including uninhabited ground vehicles or UGVs). There are several concepts under wide investigation ranging from augmenting of driver tasks through remote operation to fully-fledged autonomous operation.

ISO 26262 is primarily intended to apply to series production vehicles and as such does not address modification of standard production vehicles. Furthermore topics such as the exchange of data between a vehicle and other vehicles and/or the transport infrastructure, or any kind of autonomous operation, are not considered to be in scope. In this latter respect frequent reference was made during the development of the standard to the 1968 Vienna Convention on Road Traffic (UN 1968) which states that “every moving vehicle or combination of vehicles must have a driver” and that “every driver shall at all times be able to control his vehicle or to guide his animals”, and thereby that any kind of autonomous operation could not be considered “in scope”.

However it is clear that the capability of technology has already reached the point where remote or autonomous operation of ground vehicles is feasible and several public demonstrations of such concepts have been made. Where future applications rely on donor platforms from production vehicles that have been developed according to ISO 26262 or other processes with a similar mindset, it could well be a challenge to derive and apply appropriate safety processes.

6 Conclusions

This paper has described the discipline of functional safety, and how it is part of the wider discipline of system safety. The importance of system safety and functional safety in vehicles, particularly the emerging “low carbon” vehicles and also autonomous vehicles, has been discussed. A key recommendation made is that safety of vehicles should consider the vehicle as a system, and ensure a co-ordinated and systems-led approach to managing safety.

In the specific domain of functional safety, the new international standard ISO 26262, which is rapidly becoming established as the state-of-the-art, was presented and an overview given of some key features of the standard. The paper also discussed some of the challenges in applying this standard, particularly for emerging technologies such as “low carbon” and autonomous vehicles, and the cross-sector application of components such as electrical machine control and battery management.

7 References

EASIS (2011): EASIS European Project.
http://www.esafetysupport.org/en/esafety_activities/related_projects/research_and_development/easis.htm

Accessed 8 April 2011.

IEC 61508 (2010), *Functional Safety of Electrical/Electronic/Programmable Electronic Safety-related Systems*, in 7 Parts, Second Edition, International Electrotechnical Commission.

MISRA (1994): *Development guidelines for vehicle based software*, (The “MISRA Guidelines”), ISBN 0-9524156-0-7, MIRA, also available as ISO/TR 15497:2000.

Ward, D.D. (2008): The need for safety-related software development standards, in *SAE Convergence 2008*, Detroit, USA, SAE Paper Number 2008-21-0018.

ISO/DIS 26262 (2009): *Road Vehicles – Functional Safety*, in 10 Parts, International Organization for Standardization.

VDA (2004): *Standardized e-Gas monitoring concept for engine management systems of gasoline and diesel engines*, V 2.0.

MISRA (2007): *Guidelines for safety analysis of vehicle based programmable systems*, (“MISRA SA”), ISBN 0-9524156-5-8, MIRA.

Ward, D.D., Jesty, P.H. and Rivett, R.S. (2009): Decomposition scheme in automotive hazard analysis, in *SAE World Congress 2009*, Detroit, USA SAE Paper Number 2009-01-0745.

IEC 61800-5-2 (2007), *Adjustable speed electrical power drive systems — Part 5-2: Safety requirements — Functional*, International Electrotechnical Commission.

United Nations (1968): *Convention on road traffic*, Vienna.

The Language of System Safety Engineering: Loose Language Surrounding ALARP

Tracy A. White

AMOG Consulting,

Sea Technology House, Monash Business Park,
19 Business Park Drive, Notting Hill 3168, Victoria

tracy.white@amogconsulting.com

Abstract

Whilst there may be some debate as to what exactly qualifies a person as a *System Safety Engineer* (or how each professional institution/domain may perceive such a creature), one factor which acts as a significant discriminator in identifying said *Engineer*, is the precision of the language they use to describe the various safety attributes of a design or engineering process. Due to the potential ambiguities of natural language (or more particularly Engineering-English), and the ever-present emotional bias, which pervades discussions of *safety*, it is vitally important to consider meaning, perception and interpretation in the choice of language in everything we say and do. Regardless of the specific engineering domain there will always be a fundamental requirement to make some statement that the equipment under consideration is acceptably safe and that the associated risk is ALARP (As Low As Reasonably Practical). Experience has shown that the precise meaning of acceptability and the underlying concept of ALARP is poorly understood and articulated. In this paper we will consider why language is so important to the discipline of System Safety, particularly when talking about risk *acceptability*, and why we should always be vigilant, to the use of loose or sloppy safety language amongst our fellow engineers, recognising that clarity, as with any other aspect of the engineering design process, is vital to the success of the endeavour. The potential for misunderstanding is ever present and ignoring this danger ultimately has the potential to undermine any claims that safety has been assured.

Keywords: System Safety, safety language, acceptability, acceptable, ALARP, tolerable, broadly acceptable

1. Introduction

The engineering world has been grappling with the issue of what constitutes *System Safety Engineering* and indeed, if it is merely an inherent part of Engineering, or actually a specialisation in its own right. But whether you are discussing aspects of System Safety or the intricacies of quantifying residual safety risk¹, in all cases there is a

¹ With a project, the term *risk* is applied to many aspects which have some influence on the success of a project but in this paper we are only concerned with safety risk i.e. the risk associated with a hazard, so the term risk here will only be related to safety

language that will be adopted, rooted in English, but adapted to enable effective communication between the affected parties.

The problem with system safety activities in particular and, as this paper discusses, the formulation of an ALARP argument, is that the interested parties are many and varied with no obvious singular unifying language. It is clear from this situation that an essential component of System Safety Engineering is the need to establish and use clear, unambiguous language which supports a common understanding about the risks, the extent of those risks, the overall safety acceptability of a particular system design and the relationship of those risks to the design and design processes. Whilst System Safety engineers may think, that within their fraternity, there is a common understanding and language articulated in the various safety standards and guides. What this paper will examine is, even within this narrow group of Engineers, we are often guilty of using lax and imprecise language that leads to misunderstanding, or a less than clear statement of risk. If we fail to communicate effectively with other System Safety engineers how can we expect the wider Engineering community to fair any better. The issue of System Safety language is extensive, more so when considering the various flavours within the different engineering domains, and it is not anticipated that it will be possible to do this subject justice in a single paper, so for the purposes of this paper we will be concentrating on the use of the language we use to describe the *acceptability* of systems as determined by their underlying risks, in particular we will be considering this *acceptability* in relation to any ALARP claims.

2. Importance of Language

2.1. Colloquial and Technical Language Usage

When considering language, or specifically System Safety language, it is important to recognise that it is not simply the words or phrases that we use, but the understanding this invokes in the listener or reader; this is often dependent on the underlying concepts, which drive that understanding, or interpretation.

The use of language as a source of misunderstanding is a potential problem within many pursuits, but none more acute than in the field of aeronautics, specifically in the

controlling of aircraft flight paths and approaches to/from airports. NASA sponsored a study of these language/communications aspects in an attempt to categorise and identify causal factors in aircraft accidents and improve safety in this area. In a publication of these findings, Fatal Word (Cushing 1994), it was detailed that factors such as ambiguity, homophony, punctuation/intonation and speech acts as characteristics of human language and communication, which had the potential to lead to misunderstanding and, in the case of aircraft movements, ultimately fatal outcomes. The research findings focus almost exclusively on verbal communications, or linguistics, and whilst it can be argued that System Safety Engineering is primarily about the written word and the perception/understanding of those words, valuable lessons can still be drawn. Linguistic characteristics such as homophony and intonation are very much aurally based, but aspects such as ambiguity and punctuation apply equally to the written word. One pertinent aspect that Cushing identifies is that our use of language, or specific words or terms, are subject to interpretation based on their categorisation, namely whether they are used in a *technical* or *colloquial* sense; the use of the same word in these two different contexts produce significantly different meanings. In a simple example Cushing shows that the word *hold*, when used in a technical sense in the air traffic control world has the meaning: *stop what you are doing*, but a colloquial application would mean: *continue with what you are doing*, significantly different meanings and responses.

In this paper we will be examining how that *technical* and *colloquial* language interpretation can lead to confusion when talking about the *acceptability* of a system from the perspective of risk and how that acceptability relates to any claims of ALARP; the ALARP concept invokes a technical meaning for *acceptable* which is not always used consistently and correctly within System Safety Engineering.

2.2. Erosion of the appreciation of the risk inherent in a system

It is not being suggested that loose or careless System Safety language can result in the same immediate fatal results considered by Cushing, but those same flawed language characteristics can lead to misunderstanding, or lack of appreciation, of the importance and level of risk which is being exposed; this lack of appreciation can, particularly overtime, result in a fatal outcome. If we consider disasters such as Piper Alpha oil rig disaster (Cullen 1990), the loss of the Space Shuttle Challenger (Presidential Commission 1986) and, more recently, the loss of the RAF Nimrod (Haddon-Cave 2009), et. al. it can be seen that these disasters contained aspects related to a failure to clearly appreciate and express the continuing risk exposure as it was impacted by: changing designs, incident findings, changes in processes and procedures; more importantly it was not appreciated at what point that risk had progressed to an *unacceptable* state. This lack of clarity, as to the actual level of risk being exposed was, in part, down to poor articulation of the risk where language played a significant role. Where

it was claimed, or considered, or thought at the time, that this risk was *acceptable*, the resulting disaster proved that was not the case.

3. Acceptability and ALARP

As mentioned previously, for the purposes of this paper, the focus is on the use of the term *acceptable* (or risk *acceptability*) in relation to the ALARP concept. In order to appreciate that there is a technical applicability for *acceptable*, it is necessary to refer back to UK Defence Standard DEFSTAN 00-56 (UK MoD 2004) safety standard, which provides guidance on the meaning, and application of ALARP; this is shown graphically in Figure 1². It should be noted that although the title (taken directly from the standard) mentions the ALARP relationship, no mention of ALARP is made in the diagram body. The reason that there is no need to show ALARP is that its application is closely coupled to the *tolerable* category; the term *acceptable* features twice: at the upper limit (*unacceptable*) and lower limit (*broadly acceptable*) – with the *tolerable* region being the area in between. Another important feature of the concept is the upside-down triangle which depicts the level of increasing effort and rigour required for dealing with the increasing risk; there is significant effort required managing a risk which has been determined as *unacceptable* considerably more so than if a risk has been assessed as *broadly acceptable*.

² Whilst the diagram is clearly indicated as an example, it can be seen that a safety program would identify levels of risk ranging from high to low (in this example, quantified as: very high, high, medium and low) and that for these levels, would determine which of those risk levels would be considered (mapped) as *unacceptable*, *tolerable* and *broadly acceptable*.

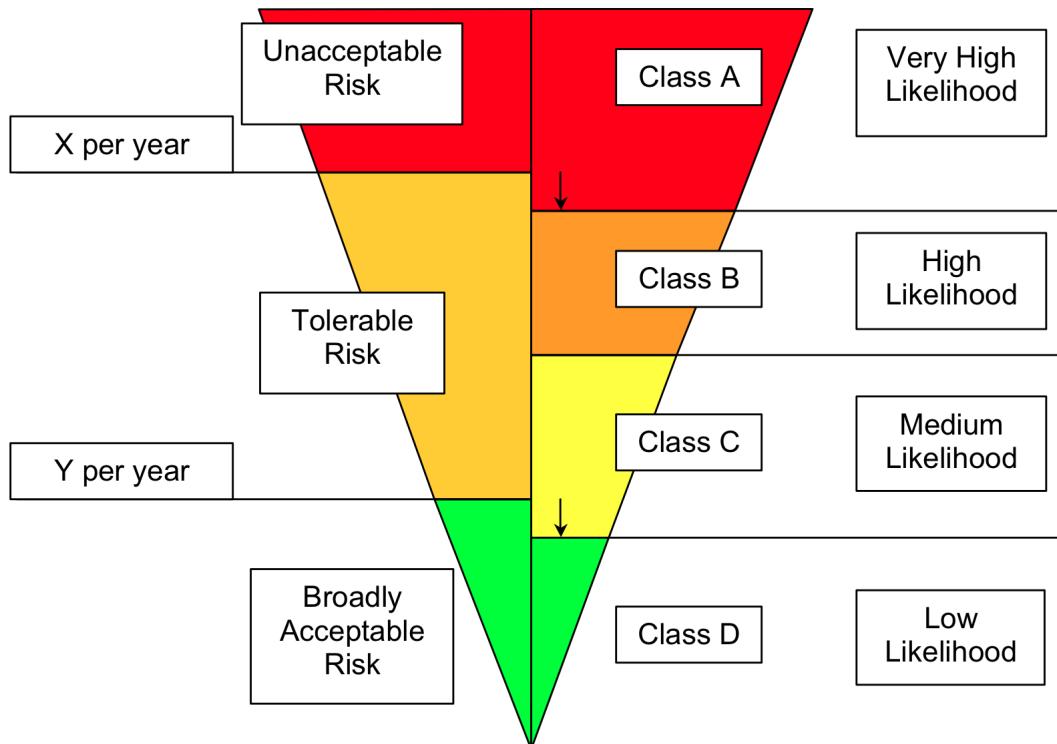


Figure 1 – Example tolerability criteria and ALARP relationship (DEFSTAN 00-56)

Where a risk is considered to fall in the *tolerable* region, additional effort is required to provide further argument/justification that the risk is ALARP which would include considerations of: statutory and regulatory requirements, current best practice, reasonable additional controls and possibly some aspect of cost benefit analysis³; ALARP is therefore synonymous with the *tolerable* region. That is not to say that considerations of reasonableness and practicability are not made for *broadly acceptable* risk, something that is acknowledged in DEFSTAN 00-56. There is a distinction between different levels of reasonableness and practicability that are explicitly recognised and defined in EN 50126-1 Railway applications (CENELEC 1999), the ALARP relationship to the *tolerable* region is more explicitly defined (Figure 2) with the middle region being described as the ‘ALARP or tolerability region’ – this is consistent with the Defence Standard but a more explicit depiction.

Another important aspect to appreciate about a *tolerable* risk, given a robust ALARP argument, is that risk is then, according to DEFSTAN 00-56, *tolerated*. It is sometimes incorrectly expressed that a *tolerable* risk is *acceptable*⁴ in the presence of a robust ALARP argument; this undisciplined and loose language should be avoided. The system can be ‘accepted’ by a regulatory party, or the customer (the *tolerable* risk/s are *tolerated*) but following the ‘acceptance’ of the system, it does not follow that the

risk should be considered *acceptable*, albeit it may be perceived that way.

On the other aspect of relative rigour and effort, EN 50126-1 provides further clarification where, in the *broadly acceptable* region it states that there is ‘no need for detailed working to demonstrate ALARP’. It is not particularly laudable or reasonable to call for more rigour (in the provision of an ALARP argument/justification) for *broadly acceptable* risks. The question needs to be asked whether it is wrong to consider doing more than would be considered *reasonable*? The reality is that system safety, like any other area of engineering, has a finite set of resources and time to provide safety assurance; the more effort spent arguing about a *broadly acceptable* risk, means less time available to address *tolerable* risks – intuitively therefore, mandating disproportionate effort in the provision of ALARP arguments for *broadly acceptable* risk should not be regarded as being anything other than irresponsible⁵.

³ This paper does not argue what should or should not feature in an ALARP argument, but merely recognises that there is a level of effort required in formulating such an argument; but the Health and Safety Executive advice on the subject indicates that costs should feature where the systems exhibit particularly novel or complex features

⁴ In DEFSTAN 00-56 terms a *broadly acceptable* risk is also *tolerated*, but differs from a *tolerable* risk in that it does not require a robust or full ALARP argument

⁵ How much time was spent on Nimrod providing arguments for *broadly acceptable* risks that should have been spent assessing the *unacceptable* fuel leak near the SCP duct (Haddon-Cave, 2009)?

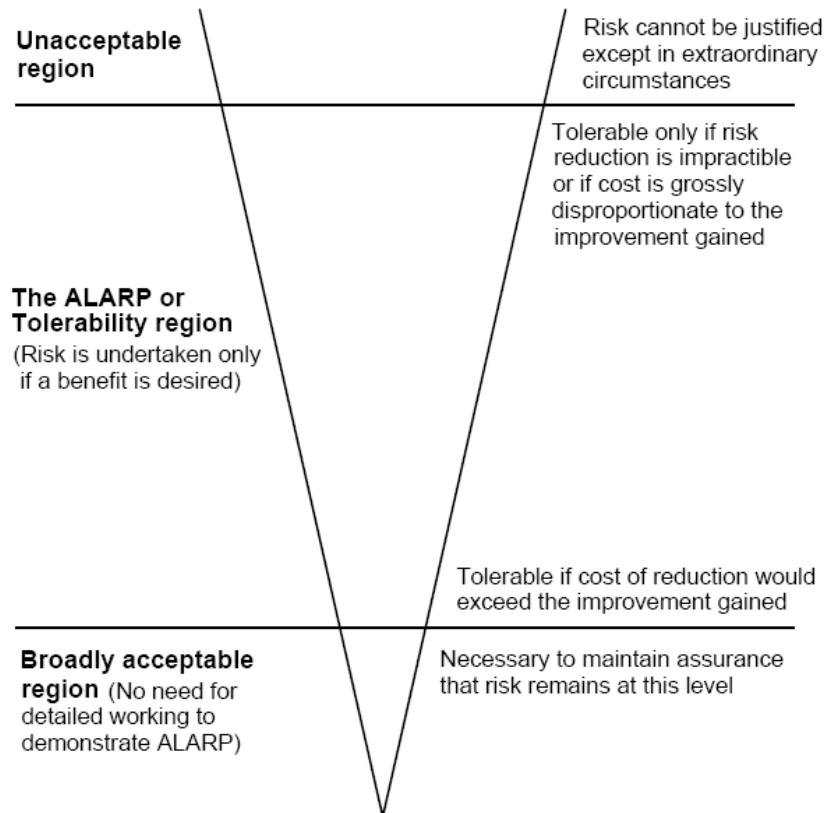


Figure 2 – EN 50126 ALARP triangle

Although exceptional it is possible, from the perspective of system safety, to accept a system (implicitly then in colloquial terms the system is ‘acceptable’) even when it contains *unacceptable* risks, if it can be demonstrated that it there is no practical way of meeting the tolerability criteria for those risks; the *unacceptable* risk would therefore ‘accepted’, possibly with a time (one-off) usage or operational constraint/limitation⁶; the language we use when articulating this circumstance needs to make the distinction very clear. It should not be a possible misunderstanding that an *unacceptable* risk, the system of which has been ‘accepted’ for deployment, remains anything other than *unacceptable*; this is particularly pertinent when that *unacceptable* risk only becomes apparent following system deployment.

It is clear from the shuttle disaster (Presidential Commission 1986) that the o-ring operations in low temperatures presented an *unacceptable* risk. Whilst the Solid Rocket Motors (SRM) had been ‘accepted’ for use, their subsequent performance revealed an *unacceptable* risk with the o-ring blow-by⁷; the now *unacceptable* risk could have been ‘exceptionally justified’ with a limitation whilst the extent of the problem was assessed, quantified and/or mitigated, but it was clear from management discussions that the fact the SRMs had previously been ‘accepted’ for use had coloured their thinking as to the

acceptability of the risk when the risk associated with the o-ring blow-by was finally considered (too) *unacceptable* to give clearance for continued use⁸.

Similarly, the Nimrod (Charles Haddon-Cave QC 2009) had been flying around with what was essentially an *unacceptable* risk (fuel supply line near a heat source), but the prior system ‘acceptance’ presented a disproportionate and unjustified influence on the perceived ‘acceptability’ of the risk. This undue influence ultimately drove the safety case program along the lines of a ‘documentation exercise’ rather than any serious analytical activity, had that not been the case then there would have been a requirement to produce, and have agreed, an ‘exceptional justification’ for the risk associated a potential fire presented by the fuel lines being routed near the a source. Prior ‘acceptance’ of a system should in no way retrospectively alter the ‘acceptability’ of the underlying risks, the two things, whilst related, are essentially independent in terms of the determinations made.

4. Universal Appreciation

Given the preceding discussions about the acceptability and acceptance of risks, it is worth considering the degree of application, appreciation and understanding within the wider safety community. In the following two paragraphs we consider examples from the Naval and Rail domains.

⁶ the system may be constrained from operating at its maximum speed, height or capacity

⁷ the flexing of the structure during launch had imposed loads on the seals that had not been anticipated and catered for in the original design.

⁸ it is accepted that this disaster, as with most other disasters, there were a considerable number of influencing factors, but the lack of clarity of ‘unacceptable risk’ vs. acceptance of the system for previous operations was given disproportionate weighting that was in no way a measure of risk acceptability.

4.1. ABR 6303

The safety policy for the Royal Australian Navy (RAN) is promulgated through a series of publications known as the Australian Book of Reference (ABR), specifically ABR 6303 (Director Navy Safety System 2002), also known as the NAVSAFE chapter. The NAVSAFE publication explicitly acknowledges its major references as being Risk Management (Standards Australia 2004) and Occupational Health and Safety (Standards Australia

1997), rather than a more specific system safety standard such as DEFSTAN 00-56 (UK MoD 2004). Interestingly ABR6303 makes no explicit reference or acknowledgement of DEFSTAN 00-56 but has included the identical ALARP triangle (Figure 3) in Chapter 5 of its publication. It faithfully repeats 3 regions, preferring to refer to them as: *Extreme*, *Tolerable* and *Negligible*, which align effectively with the same in DEFSTAN 00-56 and EN50126 (CENELEC 1999), see Table 1.

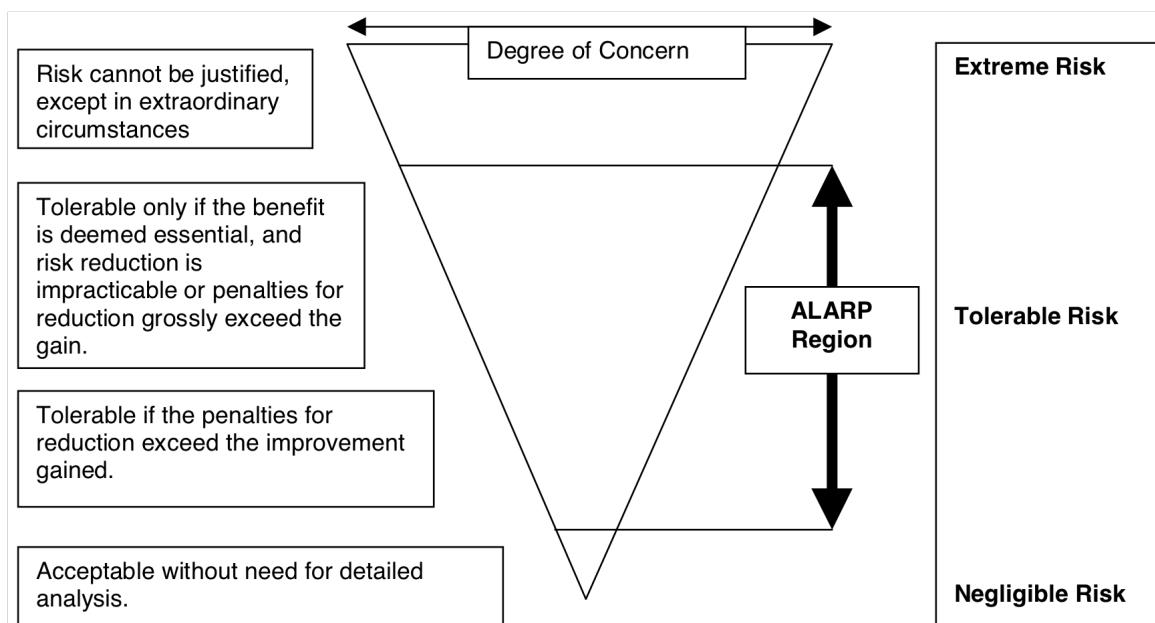


Figure 3 - ABR 6303 - ALARP triangle

A review of the left-hand column of the triangle appears to indicate that the concept (despite the absence of any citing or acknowledgement) has been well adopted. The upper region only allows for ‘acceptance’ (in the colloquial sense) in *extraordinary circumstances*, whilst at the other end of the scale the ‘acceptance’ (again in the colloquial sense) of the risk does not come with the penalty of *detailed analysis*. In a subtle, but significant divergence, the original purpose of the triangle was to indicate some measure of the relative effort required in order to address the relative level of risk; to instead claim that the triangle is some measure of *concern* is an unnecessary detraction. It could be claimed that the more the concern, the more the effort, but given that the reason for forming an argument of reasonableness and

practicability is to provide a legal defence, in the event that some harm has arisen from a system or activity, the defence will be founded on how much was done to address the risk (the level of effort), not on how much you cared (degree of concern).

It is not possible to verify whether the authors of the standard meant the ‘degree of concern’ was an indication of the level of effort as there is no definition of the phrase/term in the accompanying Risk Management Terminology, chapter 5, Annex D; for that matter, there are no definitions for *tolerable*, *tolerated*, or *acceptable* which might help to clarify the potential confusion.

DEFSTAN	EN50126	EN 50126 Description	ABR6303	ABR6303 Description
Unacceptable	Unacceptable	Risk cannot be justified except in extraordinary circumstances	Extreme	Risk cannot be justified except in extraordinary circumstances
Tolerable	Tolerable	Tolerable only if risk reduction is impracticable or if cost is grossly disproportionate to the improvement gained. Tolerable if cost of reduction would exceed improvement gained	Tolerable	Tolerable only if the benefit is deemed essential... Tolerable if the penalties for reduction exceed the improvement gained
Broadly Acceptable	Broadly Acceptable	No need for detailed analysis	Negligible	Acceptable without the need for detailed analysis

Table 1: Comparison of DEFSTAN 00-56, EN50126 and ABR6303 risk acceptability regions

Despite the issue over concern vs. effort and some missing definitions, the concept is relatively well understood and adopted into the NAVSAFE guidance. Unfortunately though when examining NAVSAFE, Annex A, an alternative ALARP triangle is presented (Figure 4). The alternative ALARP triangle now divides the *tolerable* region into *unacceptable* and *acceptable* areas. In addition, the previously *unacceptable* region, from the parent document, is now referred to as *(in)tolerable*. This liberal juxtaposition of the terms *acceptable* and *(in)tolerable* produces a confused and potentially contradictory understanding. There is further confusion where the upper end of the ALARP region of the alternative triangle (Figure 4) describes the risk as being considered in *exceptional circumstances* which is

not significantly different from the description of the *unacceptable* region (*extreme* in Figure 3) where it states that the risk can only be justified in *extraordinary circumstances*, but these risk levels which appear to read the same are now in different regions; there is no technical justification for inconsistent use of terms and concepts from the parent NAVSAFE chapter and the introduction of yet more, unexplained, terminology. The issue has arisen because the author has failed to apply the *technical* language underpinning the concept, in a disciplined and consistent manner, resorting to the colloquial, ambiguous and confusing usage; loose and casual language is not warranted at the best of times and certainly should not feature in a publication, likely to be used by many a corner stone of their risk assessments.

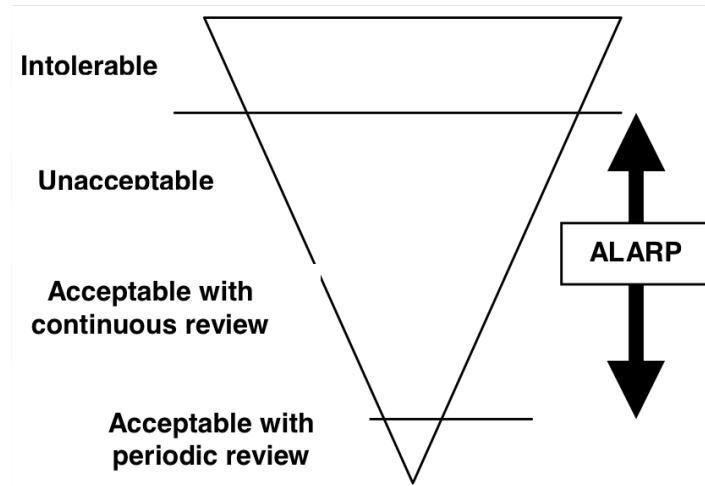


Figure 4: NAVSAFE (Annex A) ALARP triangle

4.2. RailCorp

Rail Corporation New South Wales (RailCorp) is a statutory authority of the New South Wales government. RailCorp owns, operates and maintains the Sydney suburban and interurban rail network, which is marketed under the CityRail brand; in addition to operating rural passenger services under the Country Link brand. It also provides freight operators with access to the rails of the Sydney metropolitan area.

RailCorp has a mature and extensive Safety Management System (SMS), which support all branches of system safety, inclusive of which is ALARP guidance (RailCorp 2010). Given the business is the operation of

railways it is not surprising to find their application of ALARP (Figure 5) to be aligned to the rail standard EN 50126-1 (CENELEC 1999) discussed earlier. The ALARP principles are listed down the left-hand side of the triangle, which in turn are mapped to the RailCorp safety risk application of the principle on the right; the ALARP principle, when compared to the underlying rail standards (and DEFSTAN 00-56 (UK MoD 2004) for that matter), provide evidence of close correlation (Table 2); *unacceptable risk* are justified only in exceptional/extraordinary circumstances, and notably (in contrast to ABR6303, see Section 4.1), the *tolerable* risk is *tolerated* (not ‘accepted’).

DEFSTAN	EN50126	EN 50126 Description	ALARP Principle Categories	ALARP Principle Descriptions
Unacceptable	Unacceptable	Risk cannot be justified except in extraordinary circumstances	Unacceptable	<p>Risk unacceptable regardless of the associated benefit</p> <p>Activity ruled out or risk reduced to a lower category</p> <p>Activity or practice can be retained only in exceptional circumstances</p>
Tolerable	Tolerable	<p>Tolerable only if risk reduction is impracticable or if cost is grossly disproportionate to the improvement gained.</p> <p>Tolerable if cost of reduction would exceed improvement gained</p>	Tolerable	<p>Risk can be tolerated in order to secure the associated benefit</p> <p>Risk must be properly assessed and controlled so that the residual risk is kept as low as reasonably practicable (ALARP)</p> <p>Risk is to be reviewed periodically to ensure it remains ALARP</p>
Broadly Acceptable	Broadly Acceptable	No need for detailed analysis	Broadly Acceptable	<p>The risk is considered insignificant and well controlled</p> <p>Further risk reduction required only if reasonably practicable measures are available</p>

Table 2: Comparison of RailCorp ALARP Principles and EN50126 risk acceptability regions

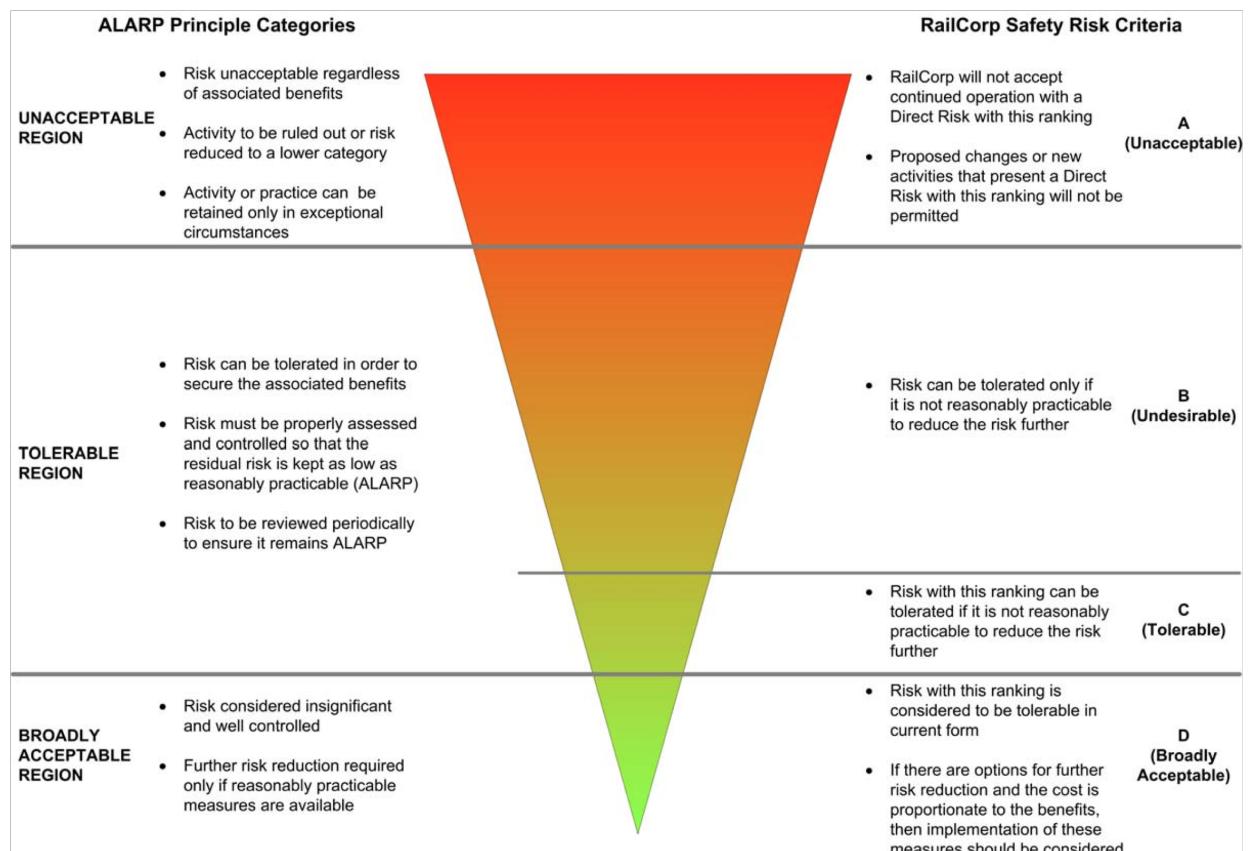


Figure 5: RailCorp ALARP triangle

The splitting of the tolerable region into *undesirable* and *tolerable* has further correlations to the EN50126 ALARP triangle (Figure 2), which shows a similar higher and lower ALARP emphasis where, at the upper end of the region, there is a requirement to demonstrate gross disproportionality of risk reduction vs. improvement, whilst at the lower end there is a lesser emphasis by considering cost vs. benefit. The only difference between *undesirable* and *tolerable* are the phrases: ‘risk can be tolerated only if...’ and ‘risk with this ranking can be tolerated if...’ The ALARP triangle concept would indicate an increased level of justification effort for *undesirable* over *tolerable*, but how much and the degree of effort variation is not evident from the imprecision of the language used; arguably the sole difference is the use of the word ‘only’.

The description for *broadly acceptable* raises further confusion by describing the risk as *tolerable*, which raised the question: if a *broadly acceptable* risk is *tolerable*, does it reside in the *tolerable* region or the *broadly acceptable* region? The author has fallen into the trap of using the colloquial form of the word *tolerable*, rather than the technical form (as used elsewhere in the same figure) and has brought with it the same confusion found by the NASA study (Section 2.1). There is a further question mark, given that *broadly acceptable* indicates that the risk is *insignificant and well controlled*, over the call for the lowest risk to be subject to cost considerations proportionate to the benefit. The implication is that more risk justification effort is been called out for *broadly acceptable* than for *tolerable* or, in this case *desirable*, which tips the triangle concept on its head.

An area where a clearer distinction of relative effort can be made (drawn from the principle, left-hand column) is how additional controls are approached; having proved that a risk falls within the *broadly acceptable*, additional controls should be considered *where they are available*, if that risk is found to be in the tolerable region then it would be expected that additional controls should be considered even where they might not be immediately available. Whatever solution is adopted, there should be a distinction in the relative level of effort required for each region to justify the risk, otherwise the principle/concept of the ALARP triangle is not being effectively applied. But however the principle is applied, the language used to describe it should not introduce confusion as to what level of risk applies and thereby what degree of risk justification is expected; confusion arising from describing a *broadly acceptable* risk as *tolerable*, or a *tolerable* risk as *acceptable* (see Section 4.1) needs to be avoided.

5. Conclusions

The discipline of System Safety involves terminology and language semantics, a necessary requirement to reflect the underlying concepts and truths. It is important to recognise that there is a technical dimension to the language of System Safety that requires discipline in ensuring its consistent and cohesive application if System Safety engineers are to communicate effectively within the discipline and inform other affected activities including design, engineering and project management. It is the language we use that enables us to avoid confusion

or misunderstanding that can lead to a flawed argument, decision or outcome.

It is necessary to avoid colloquial vernacular used by non-safety professionals particularly when articulating the degree of risk inherent in a system, or what residual risk (as represented by the individual risks) is inherent in operating or accepting said equipments/systems. It may be necessary to clarify or correct the language used by involved parties (particularly non-safety specialists) even if, at times, that may be considered particularly pedantic and esoteric.

Engineers with safety responsibilities need to educate themselves in the precise meaning and application of the language of System Safety, which more clearly reflect the underlying concepts that are generally accepted and applied by the wider safety profession.

This paper has only considered the language involved in communicating the relatively small, but significant, aspect of risk acceptability and acceptance within System Safety yet it has highlighted important facets, which can have significant effects on the appreciation of the degree and acceptability of risk; it can be also be appreciated that the potential for confusion, miscommunication and misunderstanding due to inappropriate language certainly exists in other areas of System Safety.

6. References

- Steve Cushing (1994): Fatal Words: Communication Clashes and Aircraft Crashes.
- The Honourable Lord Cullen (1990): The Public Inquiry into the Piper Alpha Disaster.
- Presidential Commission (1986): Report of the Presidential Commission on the Space Shuttle Challenger Accident
- Charles Haddon-Cave QC (2009): The Nimrod Review.
- UK Ministry of Defence (2004): DEF STAN 00-56 issue 3, Safety Management Requirements for Defence Systems, Part 2
- CENELEC - European Committee for Electrotechnical Standardization (1999): EN 50126-1 Railway applications - The specification and demonstration of Reliability, Availability, Maintainability and Safety (RAMS) - Part 1: Basic requirements and generic process
- Director Navy Safety Systems (2002): Australian Book of Reference 6303 – NAVSAFE Manual
- Standards Australia (2004): Australian/New Zealand Standard, AS/NZS 4360:2004 - Risk Management
- Standards Australia (1997): Australian/New Zealand Standard, AS/NZS 4804:1997 - OHS Management Systems—General Guidelines on Principles, Systems and Supporting Techniques
- RailCorp (2010): SMS-06-PR-1382 — ALARP Determination & Demonstration.

Author Index

Bailes, Murray, 5
Becht, Holger, 21, 29

Cant, Tony, iii, 39
Connelly, Simon, 29

Mahony, Brendan Mahony, 39
Martin, Brett J., 51

Neist, Len, 69
Reinhardt, Derek W., 51

Sandom, Carl, 73

Ward, David D., 81
White, Tracy, 87

Recent Volumes in the CRPIT Series

ISSN 1445-1336

Listed below are some of the latest volumes published in the ACS Series *Conferences in Research and Practice in Information Technology*. The full text of most papers (in either PDF or Postscript format) is available at the series website <http://crpit.com>.

Volume 113 - Computer Science 2011

Edited by Mark Reynolds, The University of Western Australia, Australia. January 2011. 978-1-920682-93-4.

Contains the proceedings of the Thirty-Fourth Australasian Computer Science Conference (ACSC 2011), Perth, Australia, 17-20 January 2011.

Volume 114 - Computing Education 2011

Edited by John Hamer, University of Auckland, New Zealand and Michael de Raadt, University of Southern Queensland, Australia. January 2011. 978-1-920682-94-1.

Contains the proceedings of the Thirteenth Australasian Computing Education Conference (ACE 2011), Perth, Australia, 17-20 January 2011.

Volume 115 - Database Technologies 2011

Edited by Heng Tao Shen, The University of Queensland, Australia and Yanchun Zhang, Victoria University, Australia. January 2011. 978-1-920682-95-8.

Contains the proceedings of the Twenty-Second Australasian Database Conference (ADC 2011), Perth, Australia, 17-20 January 2011.

Volume 116 - Information Security 2011

Edited by Colin Boyd, Queensland University of Technology, Australia and Josef Pieprzyk, Macquarie University, Australia. January 2011. 978-1-920682-96-5.

Contains the proceedings of the Ninth Australasian Information Security Conference (AISC 2011), Perth, Australia, 17-20 January 2011.

Volume 117 - User Interfaces 2011

Edited by Christof Lutteroth, University of Auckland, New Zealand and Haifeng Shen, Flinders University, Australia. January 2011. 978-1-920682-97-2.

Contains the proceedings of the Twelfth Australasian User Interface Conference (AUIC2011), Perth, Australia, 17-20 January 2011.

Volume 118 - Parallel and Distributed Computing 2011

Edited by Jinjun Chen, Swinburne University of Technology, Australia and Rajiv Ranjan, University of New South Wales, Australia. January 2011. 978-1-920682-98-9.

Contains the proceedings of the Ninth Australasian Symposium on Parallel and Distributed Computing (AusPDC 2011), Perth, Australia, 17-20 January 2011.

Volume 119 - Theory of Computing 2011

Edited by Alex Potanin, Victoria University of Wellington, New Zealand and Taso Viglas, University of Sydney, Australia. January 2011. 978-1-920682-99-6.

Contains the proceedings of the Seventeenth Computing: The Australasian Theory Symposium (CATS 2011), Perth, Australia, 17-20 January 2011.

Volume 120 - Health Informatics and Knowledge Management 2011

Edited by Kerryn Butler-Henderson, Curtin University, Australia and Tony Sahama, Queensland University of Technology, Australia. January 2011. 978-1-921770-00-5.

Contains the proceedings of the Fifth Australasian Workshop on Health Informatics and Knowledge Management (HIKM 2011), Perth, Australia, 17-20 January 2011.

Volume 121 - Data Mining and Analytics 2011

Edited by Peter Vamplew, University of Ballarat, Australia, Andrew Stranieri, University of Ballarat, Australia, Kok-Leong Ong, Deakin University, Australia, Peter Christen, Australian National University, Australia and Paul J. Kennedy, University of Technology, Sydney, Australia. December 2011. 978-1-921770-02-9.

Contains the proceedings of the Ninth Australasian Data Mining Conference (AusDM'11), Ballarat, Australia, 1-2 December 2011.

Volume 122 - Computer Science 2012

Edited by Mark Reynolds, The University of Western Australia, Australia and Bruce Thomas, University of South Australia. January 2012. 978-1-921770-03-6.

Contains the proceedings of the Thirty-Fifth Australasian Computer Science Conference (ACSC 2012), Melbourne, Australia, 30 January – 3 February 2012.

Volume 123 - Computing Education 2012

Edited by Michael de Raadt, Moodle Pty Ltd and Angela Carbone, Monash University, Australia. January 2012. 978-1-921770-04-3.

Contains the proceedings of the Fourteenth Australasian Computing Education Conference (ACE 2012), Melbourne, Australia, 30 January – 3 February 2012.

Volume 124 - Database Technologies 2012

Edited by Rui Zhang, The University of Melbourne, Australia and Yanchun Zhang, Victoria University, Australia. January 2012. 978-1-920682-95-8.

Contains the proceedings of the Twenty-Third Australasian Database Conference (ADC 2012), Melbourne, Australia, 30 January – 3 February 2012.

Volume 125 - Information Security 2012

Edited by Josef Pieprzyk, Macquarie University, Australia and Clark Thomborson, The University of Auckland, New Zealand. January 2012. 978-1-921770-06-7.

Contains the proceedings of the Tenth Australasian Information Security Conference (AISC 2012), Melbourne, Australia, 30 January – 3 February 2012.

Volume 126 - User Interfaces 2012

Edited by Haifeng Shen, Flinders University, Australia and Ross T. Smith, University of South Australia, Australia. January 2012. 978-1-921770-07-4.

Contains the proceedings of the Thirteenth Australasian User Interface Conference (AUIC2012), Melbourne, Australia, 30 January – 3 February 2012.

Volume 127 - Parallel and Distributed Computing 2012

Edited by Jinjun Chen, University of Technology, Sydney, Australia and Rajiv Ranjan, CSIRO ICT Centre, Australia. January 2012. 978-1-921770-08-1.

Contains the proceedings of the Tenth Australasian Symposium on Parallel and Distributed Computing (AusPDC 2012), Melbourne, Australia, 30 January – 3 February 2012.

Volume 128 - Theory of Computing 2012

Edited by Julián Mestre, University of Sydney, Australia. January 2012. 978-1-921770-09-8.

Contains the proceedings of the Eighteenth Computing: The Australasian Theory Symposium (CATS 2012), Melbourne, Australia, 30 January – 3 February 2012.

Volume 129 - Health Informatics and Knowledge Management 2012

Edited by Kerryn Butler-Henderson, Curtin University, Australia and Kathleen Gray, University of Melbourne, Australia. January 2012. 978-1-921770-10-4.

Contains the proceedings of the Fifth Australasian Workshop on Health Informatics and Knowledge Management (HIKM 2012), Melbourne, Australia, 30 January – 3 February 2012.

Volume 130 - Conceptual Modelling 2012

Edited by Aditya Ghose, University of Wollongong, Australia and Flavio Ferrarotti, Victoria University of Wellington, New Zealand. January 2012. 978-1-921770-11-1.

Contains the proceedings of the Eighth Asia-Pacific Conference on Conceptual Modelling (APCCM 2012), Melbourne, Australia, 31 January – 3 February 2012.

Volume 131 - Advances in Ontologies 2010

Edited by Thomas Meyer, UKZN/CSIR Meraka Centre for Artificial Intelligence Research, South Africa, Mehmet Orgun, Macquarie University, Australia and Kerry Taylor, CSIRO ICT Centre, Australia. December 2010. 978-1-921770-00-5.

Contains the proceedings of the Sixth Australasian Ontology Workshop 2010 (AOW 2010), Adelaide, Australia, 7th December 2010.