Conferences in Research and Practice in
Information Technology

Volume 145

Australian System Safety Conference 2012

AUSTRALIAN
COMPUTER
SOCIETY

Australian System Safety Conference 2012

Proceedings of the
Australian System Safety Conference (ASSC 2012),
Brisbane, Australia, 23rd-25th May 2012

Tony Cant, Ed.

Volume 145 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 2012. Proceedings of the Australian System Safety Conference (ASSC 2012), Brisbane, Australia, 23rd-25th May 2012

Conferences in Research and Practice in Information Technology, Volume 145.

Editors:
Tony Cant
formerly at Defence Science and Technology Organisation
C3I Division, PO Box 1500, Edinburgh SA 5111, Australia
E-mail: canttony@icloud.com

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 145
ISSN 1445-1336
ISBN 978-1-921770-15-9

Printed April 2013 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 2012), Brisbane, Australia, 23rd-25th May 2012

Preface vii

Program Committee ix

Research Papers

Integrating Safety Management through the Bowtie Concept — A move away from the Safety Case focus 3
Anthony P. Acfield and Robert A. Weaver

Contribution to the characterization and identification of human stability with regard to safety: application to guided transport systems 13
Vincent Benard, Philippe Richard, Frédéric Vanderhaegen, and Patrice Caulier

Rapid Risk Assessment of Technical Systems in Railway Automation 21
Jens Braband

Functional Safety based on a System Reference Model 27
Manfred Broy

Safety Protocols: a New Safety Engineering Paradigm 35
Tony Cant and Brendan Mahony

Practical Early-Lifecycle Application of Human Factors Assessment 47
Simon Connelly, Andrew Hussey, and Holger Becht

Risk Based Safety Assurance: towards a defensible and practical methodology 55
C.B.H. Edwards and M. Westcott

Software and System Safety: Promoting a Questioning Attitude 69
Terry L. Hardy

Applying System Safety Methodologies to Consumer Product Safety 77
Zhuojun Liu, Yongguang Zhang, Peng Yu, and Huina Mu

Risk Assessment in the Wild 83
Andrew Rae and Richard Hawkins

Contracting For Assurance of Military Aviation Software Systems 91
D. W. Reinhardt and J. A. McDermid

Do You Get The Picture? Situation Awareness and System Safety 107
Carl Sandom

Author Index 117

Preface

The *Australian System Safety Conference 2012* was held at the Mercure Hotel, Brisbane, on 23-25 May, 2012. The conference, jointly sponsored by the Australian Safety Critical Systems Association (aSCSa) and the Australian Chapter of the System Safety Society, had the theme: “Value Adding and Efficiencies in System Safety” and was attended by more than 100 participants. This year we had, for the first time, five keynote speakers:

- Dr Claire Marrison, Manager, Safety Systems, Risk and Analysis (Airservices, Australia)
- Mr Robert Schmedake, Technical Fellow of System Safety (Boeing St Louis, USA)
- Mr Terry Hardy, Founder and Director Safety and Risk Management (Great Circle Analytics, USA)
- Dr Jens Braband, Business Unit Rail Automation (Siemens AG, Germany)
- Prof Manfred Broy, Professor of Computer Science (Technische Universität München, Germany)

Prior to the conference, Terry Hardy presented a tutorial entitled “Essential Questions in Software Safety”, and Jens Braband presented a tutorial entitled “Rapid Risk Assessment of Technical Systems”. Full program details are available from `assc2012.org`. 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 acknowledge the conference sponsors for their strong support: the Australian Computer Society; Ansaldo STS; RGB Assurance; Nova Systems; Hyder Consulting; BAE Systems; Airservices Australia; and the Defence Materiel Organisation in the Australian Government Department of Defence.

I wish to thank all those involved in organising the conference (listed below). Once again, I am grateful to my colleagues B.J. Martin, Holger Becht and Derek Reinhardt, who worked hard to make sure that the conference was a success.

Finally, our thanks to the Australian Computer Society: in particular we are grateful to Brian Clegg; to Barry Snashall and Colleen Garard of the Queensland Branch; and to the Computer Systems and Software Engineering Board.

Tony Cant, formerly at Defence Science and Technology Organisation
Organisation,
Program Chair, ASSC 2012
May, 2013

Program Committee

Program Chairs

Tony Cant

Program 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)

ASSC 2011 Organising Committee:

- 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)

Research Papers

Integrating Safety Management through the Bowtie Concept
A move away from the Safety Case focus

Mr. Anthony P. Acfield and Dr. Robert A. Weaver
Airservices Australia
GPO Box 367, Canberra 2601, ACT

anthony.acfield@airservicesaustralia.com, rob.weaver@airservicesaustralia.com

Abstract

To ensure that safety processes such as risk management, change management and incident investigation deliver maximum value, it is essential that they are effectively integrated. As well as providing a means to represent risk, the Bowtie concept also provides a strong basis for integrating these safety processes, both internally within an organisation and cross-organisationally.

This paper provides an overview of how these processes can be integrated, why this integration is essential and why a change in focus from traditional Safety Cases to Bowtie Risk Management is needed within the safety engineering industry. As well as this, the paper describes in detail how a Bowtie Risk model can be used at the heart of safety requirements elicitation and a safety change management argument.

The aim of the paper is to effectively demonstrate that a risk-based approach to safety management, using the Bowtie concept, provides an effective means of achieving both this integration and shift in safety argument methodology.

Keywords: Integration, Bowtie, Risk, Safety Case.

1 Introduction

A Safety Management System (SMS) provides a systematic way to control all processes relating to the management of safety for a system or organisation. The International Civil Aviation Organisation (ICAO) Safety Management Manual (ICAO, 2009) identifies the following functions of an SMS:

a) Identify safety hazards;
b) Ensure the implementation of remedial action necessary to maintain agreed safety performance;
c) Provide for continuous monitoring and regular assessment of the safety performance; and
d) Aim at continuous improvement of the overall performance of the safety management system.

To achieve these functions, ICAO identifies the elements shown in Figure 1 as necessary for a successful SMS. While the SMS concept brings together all elements of a safety process into one system, manual or document, it is only when these elements are successfully integrated that the value of the processes can be maximised.

Within an SMS, six key processes need to be integrated to provide the heart of effective safety management:

- Management of Safety Accountabilities;
- Hazard identification;
- Risk assessment and mitigation;
- Safety performance monitoring and measurement;
- The management of change; and
- Incident investigation.

Many approaches to safety management integrate some of these processes. However, it is rare that all of these processes are successfully integrated within the application of a Safety Management System. This lack of integration can lead to safety management being under valued and approached in a "tick box" manner due to the true benefits of the processes not being realised.

Without process integration, we may not concentrate design effort on the correct safety hot spots, which will occur in operation. Similarly, we may not understand which events and occurrences during operation truly represent precursors to or indicators of more severe incidents. Without ownership of risks and controls by operational authorities, safety management can be outsourced to safety departments rather than being actively engaged in by those that have the ability to affect safety performance.

Safety Policy and Objectives

- Management commitment and responsibility;
- Safety accountabilities;
- Appointment of key safety personnel;
- Coordination of emergency response planning; and
- SMS documentation.

Safety Risk Management

- Hazard Identification; and
- Risk assessment and mitigation.

Safety Assurance

- Safety performance monitoring and measurement;
- The management of change; and
- Continuous improvement of the SMS.

Safety Promotion

- Training and education; and
- Safety communication.

Figure 1: Components and Elements of an SMS (ICAO, 2009)

1.1 Current Industry Focus on Safety Cases

Currently the primary technique used for integrating safety processes is the Safety Case, which provides an argument as to why the system is believed to be safe to deploy in its intended operational context (Kelly 1998). Safety Cases were originally developed in the Nuclear Industry with the UK Windscale accident in 1957 providing an impetus. In a similar way, the UK Piper Alpha Disaster and Lord Cullen's subsequent public inquiry (Cullen 1990) led to recommendations for the use of Safety Cases for Offshore Installations. Cullen's 1990 report is commonly seen as an important milestone in the promulgation of Safety Cases.

Since that time, the Safety Case as a concept has grown in stature to the extent that it is now recommended practice in many safety related industries. Standards and guidelines for systems, hardware and software usually require the development of a Safety Case for the certification process before operational use. The "Safety Case" has become terminology used by CEOs (The Australian, 2011) and it is often seen as the panacea for safety process integration. While internationally and cross industry there is alignment on the definition of a Safety Case, there is less clarity on when its use is appropriate. This has led to the over proliferation and application of the concept in ways which do not add value.

While the Safety Case concept is effective during change management for integrating the results of safety processes through the construction of a safety argument and supporting safety evidence, the authors believe that the concept is not as effective during operational service. We contend another concept - Bowtie - is more effective during operations for integrating safety processes and that the current industry focus on Safety Cases should be changed to a focus on Operational Risk Management using concepts such as Bowtie.

In Section 2 of this paper, we provide an overview of some of the issues surrounding the application of the Safety Case concept during operations. In Section 3 we go on to introduce the Bowtie concept as an approach to Risk Management and in Sections 4 and 5 we demonstrate how this technique can be used to effectively integrate safety processes during operations, while still maintaining a strong link with the development of change focussed Safety Cases. Section 6 provides a conclusion regarding the importance of having effectively integrated safety processes and why it is important to centralise our SMS processes around a concept such as Bowtie, as opposed to a concentration on Safety Case development. It also draws a link between the concepts suggested in this paper and some of the conclusions of Charles Haddon-Cave QC in his review of the loss of the RAF Nimrod XV230 in Afghanistan in 2006 (Haddon-Cave 2009).

2 Change in Focus from Safety Cases to Bowtie

Over the past 20 years, there has been an increasing focus in many safety industries on the Safety Case as the central pillar of the safety processes as defined in an SMS. With the growth of the Safety Case concept, the authors believe that insufficient attention has been given to the different roles for which Safety Cases are being used. In this section, we explore some of the issues to do with current application of the Safety Case concept and we suggest that it may not be appropriate to focus on the Safety Case during operation. Instead, it is believed that a focus on Operational Risk Assessments (or more traditionally Risk Registers) during operation is more appropriate. We believe that in some areas the Safety Case concept has grown larger than is of value and a refocussing of the industry back to risk management is needed.

Historically, the one term "Safety Case" has been used for two different purposes – safety change management and operational safety management. It is the authors' experience that these two areas require fundamentally different approaches due to the fact that different information is available at these times and the information is used in different ways to make different types of decisions. We believe that the Safety Case concept (as it is traditionally known – safety argument & evidence) is most applicable in the area of change management, while its value is diminished in operations.

As discussed in the following sections, there is a significant difference between a Safety Case for continued operation and a Safety Case for change management. In this paper, the terms "Operational Safety Case" and "Change Management Safety Case" are used to signify these two different types. Different industries use other terminology, for example the Eurocontrol Safety Case Development Manual (Eurocontrol, 2006) uses the terms "Project Safety Case" and "Unit Safety Case". However, the principle is the same and, as explained below, Safety Cases do not all perform the same role.

2.1 Change Management Safety Case and Placing Arguments within Safety Plans

Change management includes commissioning and decommissioning as well as change in application or operation of a system or service. For change management, safety arguments in Safety Cases (and Safety Plans) have two primary purposes:

- As part of planning, to determine what activities need to be conducted to ensure acceptable levels of safety for the system or service during and after the change; and
- As part of acceptance/certification/endorsement to assist in convincing risk owners and other stakeholders (including regulators) that the change is acceptably safe.

Both are important. However, more importance should be placed on the first item – planning – as it occurs earlier in the development process and thus in general has a greater impact on ensuring or "designing in" safety. When reviewing Safety Case literature, most guidance concentrates on placing arguments in the Safety Case (rather than the Safety Plan). This does not achieve the greatest cost/benefit from the safety argument development process.

In recent times, we have seen a great focus on the development of Safety Cases and, with the growth of the concept, a whole Safety Case development industry has grown up. Detailed techniques for argument creation and presentation have been established. For example, the Goal Structuring Notation (GSN) (Kelly 1998) since its establishment in the 1990s has been extended to include many advanced argumentation concepts – Patterns, Modularity, Assurance Levels to name a few (Origin Consulting 2011). The extent to which these additional concepts add value above and beyond a Safety Case template and a well written clear argument (particularly during certification) is yet to be fully demonstrated. Concepts such as GSN (and the extensions to the notation) provide freedoms which may present Safety Case developers (particularly inexperienced ones) with more problems than added benefits. The number of people applying these concepts and the lack of argument prescription means that we may be seeing as many issues with notation based safety arguments as we see with those written in natural language.

Given the limited budgets available for undertaking safety work during change management, the time invested in the development of a complex safety argument reduces the time spent on other safety activities. It is questionable whether this time is best spent in argument construction or whether greater effort in Risk Management and Safety Requirements definition is more appropriate. Across all industries, the majority of changes would need to make the same risk based argument and the same Safety Requirements Validation argument. Thus it would be more advisable to prescribe these processes and invest the time in conducting them and describing results (product & process) rather than creating bespoke arguments. Depending upon the level of risk, the level of process prescription can be varied. This is in-line with approaches defined in standards for software assurance such as DO-178B (RTCA EUROCAE 1992). Freedoms of Safety Case structure and argument construction may well be wasting valuable resources, which could be spent adding greater safety value.

In summary, it is possible that we have invested too much time focussing on the Safety Case rather than designing safety into a system and preparing for safety management during operations.

In change management, the traditional Safety Case concept (with the use of safety argument and evidence) whether in natural language or the Goal Structuring Notation (Kelly 1998) is at its most useful. However, its use should be controlled carefully. As we describe in the next section, once the transition into operations occurs, the Safety Case concept loses its value.

2.2 Operational Safety Cases

When it comes to "Operational Safety Cases", the focus of purpose shifts from achieving operational acceptance and certification to:

- knowing what the risk baseline is;
- understanding whether it is acceptable; and
- determining what direction the risk-level is going.

Practically, the case for safety at this stage should be heavily focussed on risk management without the need for an explicit safety argument. Here we need to know whether the in-service experiences that are occurring validate the risk assessment or whether they imply that the risk assessment needs updating. The risk assessment is managing changes that occur in the environment and the system, which are not subject to a change management process. At this stage, the accessibility of the risk assessment, including an understanding of controls and their effectiveness is more important than the truth of a historical argument at the point of commissioning. The argument is implicit within the risk assessment.

Change Management Safety Cases tend to be written from the perspective of the point of certification, with events before this written in the past tense, system attributes in the present tense and in-service planned activities in the future tense. This grammatical approach to the argument, while totally appropriate for making the decision to deploy, or change the system or service, exacerbates the static nature of Safety Cases during operation. Safety Cases can become frozen in time as they enter operation.

During operations, the authors believe that maintenance of a large complex Safety Case with a safety argument is not appropriate. The terminology of risk (ISO 31000 (ISO 2009)) inherently contains a standardised safety argument and these principles should be applied at this stage. This is because Operational Authorities who hold safety accountabilities are more likely to find risk concepts more understandable and relevant than argumentation concepts. Terms such as Threats, Controls, Likelihood and Consequence are more applicable to operational services and operational safety processes (e.g. investigations and event reporting) than safety argument terms such as Claims, Goals, Strategies, Justifications and Evidence.

Within operations, the safety argument is usually the same – it is a risk based argument focussing most commonly on the As Low As Reasonably Practicable (ALARP) principle. Other safety arguments, such as legal compliance, are also usually implicit through the application of procedures (e.g. application of a Manual of Air Traffic Services in Air Traffic Control). Thus we do not need to manage or maintain an explicit safety argument. In fact, focussing on maintaining a Safety Case during operations can remove effort from more practical integration of safety processes.

During a development project we need to prepare for a risk focussed operation. Given that the majority of accidents occur during operations, the change management process, while developing a Safety Case for commencement of operations, should also prepare for operational safety management. Often the change process concentrates on the former of these rather than the latter.

Having said this, during operations it is still necessary to maintain a record of what high level goals should be achieved when a change is made to that specific system or service. This would be the upper section of a safety argument, but is something that would most likely be generic to all changes, concise and provide a starting point for any future change management Safety Plans and subsequent Safety Cases.

We believe that the focus for operational safety management should be operational risk management rather than safety arguments (within a Safety Case). This focus means that the term "Operational Risk Assessment" is more appropriate than "Operational Safety Case", "Unit Safety Case" or "Risk Case" (as identified by Haddon-Cave). At the operational stage of the lifecycle they are not a "Case" in the traditional sense of the word and their focus is on risk management, not the argument. Like the Safety Case, the Operational Risk Assessment should be a logical concept, which might be recorded as a document, within a software tool or a combination depending on what is fit for purpose.

The authors propose that the Bowtie concept is not only a useful technique for recording Operational Risk Assessments, but that it is also appropriate for integrating the key safety processes described in Section 1. The Bowtie concept is introduced in the next section.

3 Bowtie Concept

The Bowtie concept (ABS Consulting, 2012) was originally developed by The Royal Dutch / Shell Group and provides a means by which risk information, that would commonly appear in a risk register, can be represented graphically. The resultant diagram (Figure 2) approximates the shape of a Bowtie. Bowtie has been applied in Oil and Gas Exploration and Production, Chemical Processing, Defence and Security, Shipping (including ports and harbours), Packaging and Logistics, Medical, Aviation, Mining, and Emergency Response (ABS Consulting, 2012). The concept is used by Airservices Australia to manage its Operational Risk Baseline.

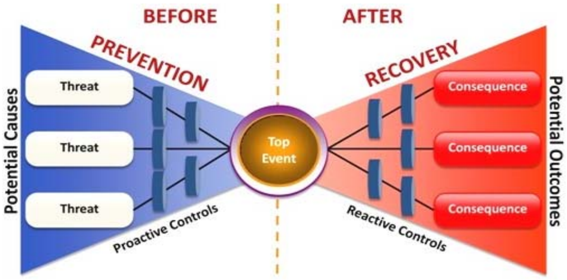


Figure 2: Bowtie Diagram (Sharif 2011)

Each Bowtie presents a single Hazard & Top Event combination and pictorially represents the Threats that can lead to the Top Event (release of the Hazard), and the Consequences that may be produced as a result. Also represented are the Controls in place to prevent each Threat from releasing the Hazard and the Controls in place to mitigate the severity and/or likelihood of each Consequence. It should be noted that the term "Top Event" is used here in a way that the term "Hazard" is used in some industries. In Bowtie terminology the term Hazard is used to describe the activity - e.g. "driving" is a Hazard and "inability to decelerate" is a Top Event. With this approach, there can be multiple Top Event Bowties associated with a hazardous activity. Threats can be both internal and external to the system. Consequences are as per traditional risk terminology "outcome of an event affecting objectives" (ISO, 2009) and occur at the system boundary. As with other notations, it is essential to ensure discipline with respect to terminology and the application of concepts.

The Bowtie concept can be used to represent risk associated with Systems, Services, Processes and Organisations. In order to define an organisation's entire Operational Risk Baseline, Bowtie diagrams must be developed representing all Hazards and Top Events associated with the service provision (Figure 3), addressing both system failure and organisational failure.

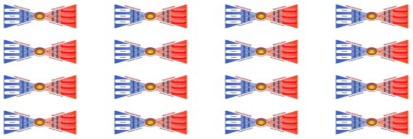


Figure 3: Bowtie Operational Risk Baseline

Once identified and recorded, the level of risk associated with each Top Event can be assessed. The likelihood of the worst credible Consequence can be determined qualitatively or quantitatively, through a combination of the likelihoods of occurrence of each Threat and success of each Control. Once determined, this likelihood can be combined with the Consequence's severity in order to obtain a risk level for the Top Event.

3.1 Benefits of the Bowtie Concept

The main advantage of the Bowtie concept is that it provides a visual representation of risk, including not only each applicable element, but more importantly, the relationships between them. It is this relationship illustration that enables many of the benefits of the concept when compared with textual or tabular risk information (in a similar way to the use of GSN for safety arguments). It allows areas of concern, such as inadequately controlled Threats or Consequences, to be readily identified and subsequently targeted for further treatment. It is the authors' experience that this visualisation of the interactions between risk elements allows the representation to be more easily comprehended and understood by those with accountability for the risk in question, who are generally not experts in safety and risk (and the associated semantics), but rather experts in the applicable subject matter (e.g. Air Traffic Control). This is crucial if risk management is to be an activity undertaken by those who are accountable for safety rather than being outsourced to a safety department.

The linear nature of the Bowtie concept (Threat leads to Top Event leads to Consequence) facilitates the linking of sequential Hazards. For instance, one Hazard's outcomes may be a subsequent Hazard's causes depending upon your area of concern (or your system boundary). This can be performed both internally within an organisation and also involving an organisation's vendors, stakeholders and customers and is described in Section 4.

A further benefit of the Bowtie concept is the ability to include elements from domains traditionally treated separately, on a single representation. Threats due to human error, procedure error, equipment failure and also external, management and organisational factors that can each contribute to a common Top Event can all be represented on a single Bowtie. Additionally, Controls from each of these aspects can be included regardless of the nature of the parent Threat, such as equipment based control of a human error Threat or procedural control of equipment failure. Beneath this top level representation of risk, safety engineering techniques (such as Fault Tree Analyses, Event Tree Analyses, FMEAs, HAZOP Studies, Common Cause Analyses, Software Assurance Techniques and Human Factor Analyses) can be linked to provide greater analysis and, where practical, quantification to a level which is of benefit. This is essential as (with all techniques) Bowtie does not provide all necessary information for safety analysis. For example, the Bowtie Concept is not a suitable basis for conducting common cause analysis.

3.2 Safety by Design

When managing an Operational Risk Baseline defined in the Bowtie format, a clear priority order can be applied to reducing risk through Safety by Design (or Safety Engineering):

- Remove Hazards – Changes at the service level that remove Hazards completely, eliminating the risk from the baseline;
- Remove Threats – Changes at the system level that remove potential causes of a Top Event;
- Reduce Threats – Changes at the system level that render the potential causes of a Top Event less likely to occur;
- Prevent Top Events – Changes at the system or unit level that make the potential causes less likely to lead to a Top Event (i.e. additional or improved preventative control); and
- Reduce Consequences – Changes at the system or unit level that reduce the likelihood or severity of the potential outcomes of the Top Event (i.e. additional or improved recovery control).

This hierarchy is more effective than the traditional hierarchy of control, which tends to focus more towards Work Health and Safety concepts rather than risk management concepts. In reality, using the hierarchy described above, the closer the risk treatment approach is to the top of the list, the harder it is to achieve. In all of the above cases, care must be taken that the change being applied, while intending to remove or reduce one element, does not have the contrary effect, or result in the addition or exacerbation of other elements. One method of achieving this is using any proposed changes to Bowtie elements to generate safety objectives and requirements; this process is described in Section 5.

4 Integration of Safety Processes

The recording of an Operational Risk Baseline using the Bowtie concept affords the opportunity for the integration of safety processes that this paper contends is so essential for effective safety management. This section will demonstrate how this integration can be achieved, both through an organisation's internal safety management processes and across the linkages between organisations.

4.1 Internal Integration

The internal integration of an organisation's safety processes such as operational risk management, safety change management and event reporting & incident investigation is a straightforward process when centred on an Operational Risk Baseline defined using the Bowtie concept (figure 4).

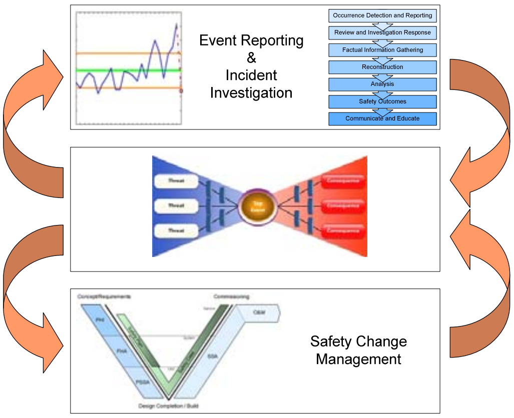


Figure 4: Internal Safety Process Integration

Key to this integration is the use of the concept as the "centre" of the related safety processes and primarily, use of the Bowtie concept to define, record, assess, maintain and accept accountability for the Operational Risk Assessments that make up the risk baseline associated with the organisation's service provision. An effective Operational Risk Baseline must record the contribution to the overall "risk picture" from all aspects of the organisation that play a role in the provision of the end service, not simply that of the operational arms. The Bowtie concept provides a mechanism for not only the inclusion of these upstream considerations but also for demonstration of how they interact. This is made possible by the linear nature of the concept, allowing the outcomes of Hazards from supporting areas of the business, such as system maintenance, to be linked to the causes of Hazards from the operational areas to which they contribute, as shown in Figure 5.

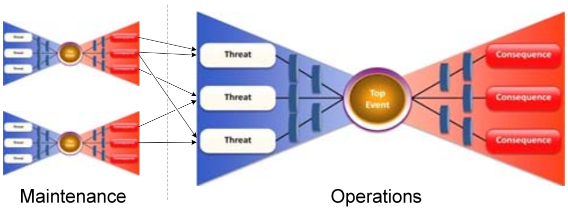


Figure 5: Operational Risk Baseline Linkages

Once defined, the Bowtie based Operational Risk Baseline must then be integrated with other related safety processes, the most important (from this integration point of view) being event reporting, incident investigation and change management.

The linking of an organisation's event reporting and incident investigation processes with its Operational Risk Baseline provides benefits for not only those processes, but also for the baseline itself. This requires the establishment of suitable monitoring and reporting of the occurrence of Threats, Top Events and Consequences as well as the success and failure of Controls, and the investigation of these incidents where appropriate.

Often safety monitoring focuses too much on the events which occur on the right hand side of a Bowtie (for example Breakdowns of Separation (BoS) in Air Traffic Control). These events may not always be accurate predictors of the Bowtie Consequence (was there an increase in BoS rate before the Überlingen collision in 2002?). By also including event reporting on the Threats on the left hand side of the Bowtie and the failure of Controls there is a greater potential for uncovering the low level indicators of future accidents.

Commonly, incident investigations tend to concentrate on what went wrong. By using the Bowtie defined risk baseline as a basis of investigation, we can also capture what went right. Over time, knowledge of the success and failure of Controls through investigations can be built up and overlaid on the Bowtie. This is something that can be difficult to achieve across multiple investigations without integration with a risk assessment.

Once collated and analysed, safety performance monitoring data and investigation knowledge provide those accountable for safety with evidence as to the safety performance of the service in the context of the risk that they have accepted. The benefit to the Operational Risk Baseline comes through the ability for the continual review and validation, or otherwise, of the Operational Risk Assessments through the incorporation of ongoing in-service data. However, we must also integrate this operational safety management with the safety change process.

Integration of an organisation's safety change management procedures with its Operational Risk Baseline, involves a closed-loop process whereby changes' potential effects on the baseline are identified and managed (one method of achieving this is described in Section 5) with the impacts realised through implementation fed back into a subsequent baseline iteration upon change commissioning.

This allows those accountable for the safety of the service to understand each change's impact on their accepted and known baseline, and provide re-acceptance of the revised Operational Risk Baseline upon transition to the change.

4.2 Integration with Other Organisations

An additional advantage of the Bowtie concept over traditional operational risk registers is the improved ability for integration of baseline information across an industry, through the linking of an organisation's Bowties with those of its vendors, service providers, regulatory bodies, peer organisations and customers.

Similar to the internal linking of sequential Hazards described in Section 4.1, an organisation's Operational Risk Baseline should be linked to those of its industry counterparts (Figure 6) in order to provide clarity as to how its:

- Threats may be influenced by the Consequences of vendors, suppliers, service providers and regulatory bodies if applicable;
- Consequences may impact the Threats of those customers relying on the service provided for the safe provision of their own service; and
- Hazards and Top Events interrelate with those of its peer organisations.

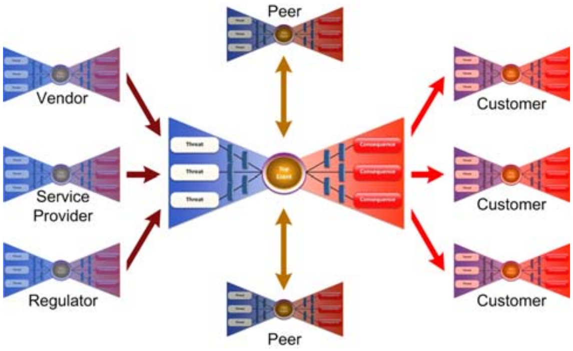


Figure 6: Industry Linking of Operational Risk

Linking Operational Risk Baseline Bowties in this way provides an organisation with greater visibility and understanding of how breakdowns in its service provision may affect the industry as a whole, and in turn how it may be affected by breakdowns of service from other related organisations. This approach provides a structured means for dialogue between organisations. This is essential when one organisation sees an issue as critical while others may not agree. The highly integrated and complex nature of services and systems means that systematic approaches are needed for these dialogues.

5 Deriving Safety Objectives & Requirements from Bowtie Changes

At the heart of integrating a Bowtie based Operational Risk Baseline with an organisation's operational safety change management process is the use of this baseline in the derivation and decomposition of Safety Objectives and Safety Requirements. It is through the satisfaction and substantiation of these derived objectives and requirements that a baseline that is considered acceptably safe is demonstrated to remain so under the change in question.

In order for the process described in the remainder of this section to be applied successfully, the service subject to change must have a defined Operational Risk Baseline that is considered correct and complete and is accepted by the relevant authorities. During the Preliminary Hazard Identification (PHI) phase of the change, this Operational Risk Baseline must be examined and all potential negative impacts due to the change identified. The impacts are described at this stage as "potential", as they may or may not be realised through change implementation. Potential negative impacts on the baseline include:

- Exacerbating an existing Threat;
- Adding a new Threat to an existing Top Event;
- Removing or weakening an existing Control;
- Adding a new Consequence to an existing Top Event;
- Exacerbating an existing Consequence; and
- Adding a new Top Event.

In order to ensure safety objectives and requirements are imposed for both intended and unintended potential negative baseline impacts, all baseline elements that fall within the scope of change, regardless of positive or negative intent, are identified as having potential negative impacts in this analysis. For example, the intent of a change may be to strengthen a particular Control, however the possibility of adverse effects must be managed therefore the potential negative baseline impact is identified.

For new systems or services, establishment of a baseline would occur in a similar way to that described above after initial Preliminary Hazard Identification.

5.1 Service Safety Objectives

During the Functional Hazard Assessment (FHA) phase of the change, the change's potential impacts on the Operational Risk Baseline are used to derive the Service Safety Objectives for the change.

Once all potential negative impacts on the Operational Risk Baseline have been identified, these impacts can be used to impose Service Safety Objectives. A Service Safety Objective is imposed upon a Top Event's Consequence whenever a potential negative impact has been identified for a Bowtie element upstream of that Consequence. The Service Safety Objectives are of the form;

Consequence X due to Top Event Y shall occur no more frequently than ϕ

Where ϕ for existing unimpacted Consequences is that Consequence's existing frequency of occurrence. ϕ for new or exacerbated Consequences is the frequency of occurrence of that Consequence that would result in an acceptable level of risk, as defined in the organisation's hazard risk matrix.

Satisfaction of the imposed Service Safety Objectives is necessary and sufficient to ensure the service under change remains acceptably safe.

For example, two potential negative impacts on elements of Top Event A have been identified (Figure 7) due to the change in question.

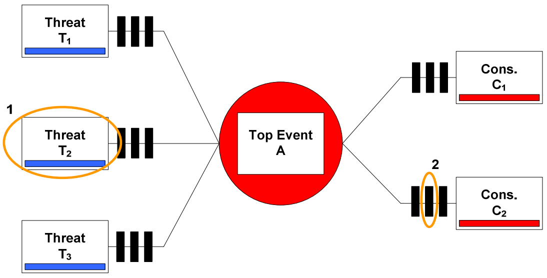


Figure 7: Potential Negative Impacts

Impact 1 – It has been identified that the change has the potential to exacerbate Threat T_2. Threat T_2 is upstream of all Consequences of the Top Event, therefore Service Safety Objectives are imposed upon all Consequences:

- *Consequence C_1 due to Top Event A shall occur less frequently than ϕ_1;*
- *Consequence C_2 due to Top Event A shall occur less frequently than ϕ_2.*

Impact 2 – It has been identified that the change has the potential to weaken one of the Controls of Consequence C_2. The Control is upstream of only Consequence C_2, therefore a Service Safety Objective is only imposed upon Consequence C_2:

- *Consequence C_2 due to Top Event A shall occur less frequently than ϕ_2.*

Note that in this case, as long as both impacts 1 & 2 were identified, the Safety Objective due to impact 2 would be redundant, as it has already been imposed by impact 1.

5.2 Service Safety Requirements

The identified potential negative impacts are also used during FHA to derive the Service Safety Requirements for the change. The format of the Service Safety Requirement derived is dependant upon the element potentially impacted.

For all impacts on Consequences and Recovery Controls, a Service Safety Requirement is imposed upon the sufficiency of the applicable set of Controls in reducing the likelihood that, on the occurrence of the Top Event, it results in the occurrence of the Consequence. For all impacts upon Threats and Prevention Controls, a Service Safety Requirement is imposed upon the sufficiency of the applicable set of Controls in reducing the likelihood that, on the occurrence of the Threat, it results in the occurrence of the Top Event. Additionally, when a Threat is impacted, a Service Safety Requirement is imposed on the acceptability of its rate of occurrence.

Satisfaction of the derived Service Safety Requirements is necessary and sufficient to ensure the achievement of the Service Safety Objectives.

Continuing with the example above; impact 1, being an impact on a Threat, would result in the imposition of the following Service Safety Requirements:

- *The rate of occurrence of Threat T_2 shall be acceptable;*
- *The Controls for Threat T_2 shall be sufficient in reducing the likelihood that the Threat causes Top Event A.*

Impact 2, being an impact on a Recovery Control, would result in the imposition of the following Service Safety Objective:

- *The Controls for Consequence C_2 shall be sufficient in reducing the likelihood that Top Event A causes the Consequence.*

For these requirements during full decomposition it would be necessary to define "acceptable" and "sufficient". These could be based upon in-service experience, similar systems or decomposed from the Service Safety Objectives. At this stage, further analysis techniques will need to be integrated to determine targets and address common causes of Threats and/or failed Controls, as Bowtie provides a technique for presenting information rather than calculating the performance targets.

This approach can be used to provide a structured means to define rates of acceptability of Consequences, Top Events and Threats. This allows balancing of safety prevention and mitigation across the entire Bowtie.

5.3 Functional and Performance Safety Requirements

During the Preliminary System Safety Assessment (PSSA) phase of the change, the functional failures and human errors associated with the change are identified and used to decompose the Service Safety Requirements into Functional and Performance Safety Requirements.

During PSSA a complete set of human errors and functional failures at the equipment, procedure, training and (enabled by the use of the Bowtie approach) organisational level for the service changes must be identified. Also during PSSA, if applicable, a complete set of functional failures at an appropriate level of design decomposition for the equipment changes must be identified. Once identified, in order to allow these failures and errors to be used in the decomposition of the Service Safety Requirements, they must then be linked to their corresponding potential negative Operational Risk Baseline impacts identified during PHI. i.e. Each functional failure or error is linked to the impacted Threat it may cause, or impacted Control it may degrade. This allows the lower level safety techniques to be linked to the Bowtie risk assessment.

The method of decomposition is dependant on the type of Service Safety Requirement and the type of potential impact through which it was derived:

- *Service Safety Requirements imposed on the rate of occurrence of a Threat.* Decomposition is achieved through the imposition of Functional and Performance Safety Requirements addressing each failure or error identified as a potential cause of that Threat;
- *Service Safety Requirements imposed on the sufficiency of a set of Controls due to a potential impact on the set's parent Threat or Consequence.* Decomposition is achieved through the imposition of Functional and Performance Safety Requirements (a) across the Controls within the applicable Control set in order to ensure their effectiveness, and/or (b) specifying the establishment of new additional Controls as required;
- *Service Safety Requirements imposed on the sufficiency of a set of Controls due to a potential impact on a Control within the set.* Decomposition is achieved through the imposition of Functional and Performance Safety Requirements as per (a) and (b) above, as well as (c) addressing each failure or error identified as a potential cause of erosion of the impacted Control.

In each of these three cases, the decomposition of each Service Safety Requirement must continue until satisfaction of the resultant set of Functional and Performance Safety Requirements is considered necessary and sufficient to satisfy the parent Service Safety Requirements.

Continuing with the example above; functional failures F_1, F_2 & F_3 and Human Error E_1 have been identified through analysis of the change in question. The failures and errors have been linked to the potential baseline impacts to which they may contribute, as shown in Figure 8:

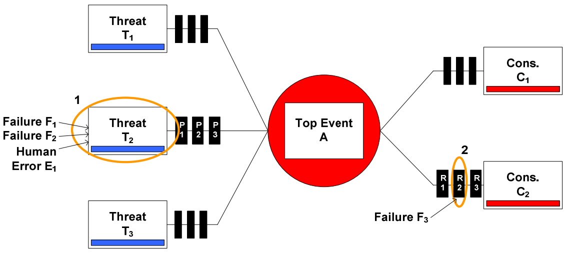


Figure 8: Linked Failure and Errors

The Service Safety Requirement, *The rate of occurrence of Threat T_2 shall be acceptable,* is a requirement imposed on the rate of occurrence of a Threat. Therefore, Functional and Performance Safety Requirements are imposed to address each of the linked failures and errors:

- *Failure F_1 shall occur no more frequently than Rate R_1;*
- *Failure F_2 shall occur no more frequently than Rate R_2;*
- *Failure F_2 shall not be a single point of failure leading to Threat T_2;*
- *Human Error E_1 shall trigger a system warning message.*

The Service Safety Requirement, *The Controls for Threat T_2 shall be sufficient in reducing the likelihood that the Threat causes Top Event A,* is a requirement imposed on the sufficiency of a set of Controls due to a potential impact on the set's parent Threat. Therefore, Functional and Performance Safety Requirements are imposed to ensure the effectiveness of the Controls:

- *Control P_1 of Threat T_2 shall be maintained;*
- *Control P_2 shall be more effective in preventing Aspect X of Threat T_2 through…;*
- *Control P_3 of Threat T_2 shall remain independent of Controls P_1 & P_2.*

Additionally, the Functional and Performance Safety Requirement specifying a new Control is imposed:

- *Control P_4 shall be added to manage the occurrence of Threat T_2.*

The Service Safety Requirement, *The Controls for Consequence C_2 shall be sufficient in reducing the likelihood that Top Event A causes the Consequence,* is a requirement imposed on the sufficiency of a set of Controls due to a potential impact on a Control within the set. Therefore, Functional and Performance Safety Requirements are imposed to ensure the effectiveness of the Controls and add additional Controls:

- *Control R_1 of Consequence C_2 shall be actively monitored;*
- *Control R_2 shall be maintained;*
- *Control R_3 shall be maintained;*
- *Control R_4 shall be added to Consequence C_2;*
- *Control R_5 shall be added to Consequence C_2.*

Additionally, Functional and Performance Safety Requirements are imposed to address each of the linked failures and errors;

- *Failure F_3 shall occur no more frequently than Rate R_3;*
- *Failure F_3 shall be annunciated on the HMI.*

In each of these cases, the decomposition would continue until the resultant set of Functional and

Performance Safety Requirements are considered sufficient to address each linked functional failure and human error and to ensure substantiation of the parent Service Safety Requirements.

5.4 Transition Safety Requirements

The process described so far has concentrated on the derivation and decomposition of Safety Requirements aimed at ensuring the system or service is acceptably safe under the applicable change, i.e. that the Service Safety Objectives are achieved. However, Transition Safety Requirements, at the service and then functional and performance levels, must also be derived and decomposed in order to ensure an acceptably safe transition, i.e. that the Service Safety Objectives are maintained during transition to the change.

This is achieved through the application of the same process, refocussed on the change transition. Operational Risk Baseline impacts must be re-examined to identify how these impacts may be potentially heightened during the transition, such as:

- Temporary exacerbation of a new Threat or further exacerbation of an existing Threat;
- Temporary further weakening, possibly to the point of full suppression, of an existing Control; or
- Temporary exacerbation of a new Consequence or further exacerbation of an existing Consequence.

These transition impacts are then used to derive the Service Transition Safety Requirements. Through identification and linking of the functional failures and human errors that can result in these additional impacts during transition, the Service Transition Safety Requirements can then be decomposed into a set of necessary and sufficient Functional and Performance Transition Safety Requirements through the process described above.

5.5 Satisfaction and Substantiation

The top-down process of derivation and decomposition of the objectives and requirements produces a logical flow of objectives and requirements (Figure 9) that can be satisfied and substantiated from the bottom-up.

Through the maintenance of necessity and sufficiency, the Functional and Performance Safety Requirements are the risk mitigation means by which the Service Safety Requirements are met and therefore the Service Safety Objectives are both achieved by the change and maintained during its transition. At this base level, satisfaction and substantiation of the specific and measurable requirements derived flows upward to demonstrate satisfaction of each Service Safety Requirement, which in turn flows up to demonstrate achievement and maintenance of each Service Safety Objective.

It is this demonstration of achievement and maintenance of the Service Safety Objectives that provides the basis for the argument that the service in question will remain acceptably safe during and under the applicable change.

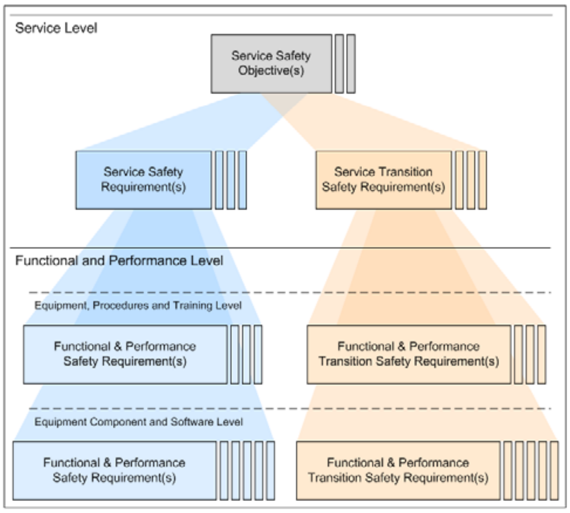

Service Level
Service Safety Objective(s)
Service Safety Requirement(s)
Service Transition Safety Requirement(s)
Functional and Performance Level
Equipment, Procedures and Training Level
Functional & Performance Safety Requirement(s)
Functional & Performance Transition Safety Requirement(s)
Equipment Component and Software Level
Functional & Performance Safety Requirement(s)
Functional & Performance Transition Safety Requirement(s)

Figure 9: Logical Flow of Objectives & Requirements

By using the Bowtie concept as the centre of the change management requirements definition process, qualitative and quantitative requirements can be established which relate specifically to the operational risk being managed. Depending upon the type of change and the in-service experience of the system, the focus for equipment, procedural, training or managerial requirements can be varied.

Focussing effort during the change on establishing the correct set of requirements and gaining agreement from all stakeholders on how to manage the operational risk is a value adding process, which is closer to managing operational safety than the development of a bespoke safety argument. Instead using this approach, the safety argument is embedded within the risk assessment.

6 Conclusions

6.1.1 Haddon-Cave's Nimrod Review

The approach documented in this paper is aligned with Haddon-Cave's Nimrod Review conclusions, in that during operations the focus should be on operational risk, rather than safety arguments. The concepts in this paper align to Haddon-Cave's recommendation that "A paradigm shift is required away from the current verbose, voluminous and unwieldy collections of text, documents and GSN diagrams to Risk Cases which comprise succinct, focussed and meaningful hazard analysis which stimulate thought and action".

Haddon-Cave's indentified attributes for "Risk Cases" remain appropriate:

- Succinct;
- Home-grown;
- Accessible;
- Proportionate;
- Easy to understand; and
- Document-lite.

Operational Risk Assessments in the Bowtie notation provide a means of achieving this. As well as these attributes, we would also include "in-service experience based" and "timely" within the list.

Instead of *Risk Case*, the term *Risk Assessment* is more appropriate as it implies that some action needs to be taken. An Operational Risk Assessment is not a static item. Instead it changes, based upon the continual evaluation of the risk baseline, in line with the process documented in the Safety Management System.

6.1.2 Bowtie

In this paper, we have shown how maximum value of safety processes such as risk management, change management and incident investigation is achieved if they are effectively integrated. Further to this, we have provided an overview of how the Bowtie concept provides a strong basis for integration both internally within an organisation and cross-organisationally.

Safety Cases have their place and certainly argumentation is essential. However, when demonstrating a top level claim for a safety document (Safety Case / Operational Risk Assessment), it is important to remember the context in which decisions will be made based on the document. There is a difference between the one-off decision to commission and the ongoing judgement to continue operating.

We contend the Bowtie concept provides a good framework for establishing Operational Risk Assessments and, as previously discussed, can connect all elements of a Safety Management System together. We encourage using concepts such as Bowtie as a strong approach for integrating all safety processes. Changing the focus within safety processes from Safety Cases to Bowtie Risk Management will improve end-to-end safety management.

7 References

The Australian (2011): First Qantas flight expected by 2pm – Joyce, 31 October 2011.

ABS Consulting (2012): www.absconsulting.com/thesis/, 2012.

Cullen The Hon Lord W. D. (1990): The Public Inquiry into the Piper Alpha Disaster, Department of Energy, London, HMSO, November 1990.

Eurocontrol (2006): Safety Case Development Manual, Edition 2.1, 13 October 2006.

Haddon-Cave QC, C (2009): The Nimrod Review - An independent review into the broader issues surrounding the loss of the RAF Nimrod MR2 Aircraft XV230 in Afghanistan in 2006, 28 October 2009.

Kelly T.P. (1998): Arguing Safety - A Systematic Approach to Managing Safety Cases, DPhil Thesis, Department of Computer Science, University of York, York.

Hurst, S. (2004): Lessons Learned from the Real World Application of the Bow-tie Method. *Risk World*.

ICAO (2009): Doc 9859 – Safety Management Manual, 2nd Edition.

ISO (2009): ISO 31000 Risk management -- Principles and guidelines, ISO 2009.

Origin Consulting (2011): GSN Community Standard Version 1, November 2011.

RTCA EUROCAE (1992): DO-178B Software Considerations in Airborne Systems and Equipment Certification, December 1992.

Sharif, S. (2011): The Bow Tie Risk Assessment Tool, Civil Air Navigation Services Organisation. http://www.canso.org/cms/showpage.aspx?id=2577

Contribution to the characterization and identification of human stability with regard to safety: application to guided transport systems

Vincent Benard[1] Philippe Richard[1] Frédéric Vanderhaegen[2] Patrice Caulier[2]

[1]IFSTTAR -ESTAS 20, Rue Elisée Reclus, 59666 Villeneuve d'Ascq, France

{vincent.benard,philippe.richard}@ifsttar.fr

[2]UVHC -LAMIH UMR CNRS 8201 Le Mont Houy, 59313 Valenciennes, France

{frederic.vanderhaegen,patrice.caulier}@univ-valenciennes.fr

Abstract

This paper presents an original contribution based on the concept of human stability by identifying the associated risks as part of the safety system assessment. The difficulties to take into account human factors in safety studies are first highlighted and definitions of new ways for the integration of human factors based on the existing concepts of stability and resilience are proposed. Although the stability concept is usually defined around a sustainable equilibrium point that induces a feeling of safety control during normal operation, it appears that the stable behaviour of a human operator can lead to risk in certain situations or contexts such as hypo-vigilance, inattention and so on. The core of this paper lays the foundation of human stability for risks assessment. Here, Human stability is defined as the ability of the operator to stay in a stable operating state under specified conditions. This concept is formalized and 3 modes of stability are developed (time, frequency and sequential modes) in order to identify states and change of states of the human stability. The concept of human stability is then applied in the framework of ERTMS/ETCS and shows that sequences of Human stability states and changes of Human stability states may be precursors of risk. Finally, some perspectives highlight the interest of human stability for the definition of risk indicators to assess system safety, by considering the Human operator as a safety/security multi-criteria sensor for the supervision of human-machine systems.

Keywords: Human stability, resilience, safety, transportation application.

1 Introduction

With an opening-up of borders, markets and exchange spaces, people and goods transportation is now a major economical and ecological problem for a large majority of countries. Through various research projects related to transportation systems, this issue is reflected by integrating new technologies, optimizing performances of these systems, but also by improving comfort and safety of passengers and goods. Although there have been technological innovations, the occurrences of incidents/accidents are significant. Statistically, it is highlighted that 30% of these occurrences of incidents/accidents are technical failures, while 70% of them are attributable to human factors (Amalberti, 2001). From this observation, this article aims to present a new concept of safety assessment focused on human operator: the human stability. This new concept is applied to guided transport systems.

This article is divided into 4 parts. The first part of the paper outlines briefly the main methods and tools usually used in dependability to assess guided systems safety and the interest to focus to other concepts like the resilience or stability systems. The second part of the paper justifies the orientation of the research works concerning the new concept of human stability and it proposes a formalization of this parameter. The third part of the paper is an application of this notion of human stability to an ETCS platform within the framework of rail driving. The final part of the paper explains how human stability could be a detector of human errors and risks to the system and presents some perspectives.

2 Safety of guided transport systems: emergence of new issues

The safety in guided transport is integrated throughout the system lifecycle, not only for the regulatory and normative aspects during the design and operation phases, but also for the decommissioning phase. With safety comes the development of operating, supervision and maintenance procedures.

2.1 Safety assessment

To meet the requirements of safety standards, guided systems key players can use a range of methods and tools from hazard assessment (see table 1) that are applicable a priori. Much of these methods and tools focus only on technical aspects of systems and infrastructures without really taking into account the human factors. These traditional tools and methods have shown their limits for the quantitative risk assessment with a growing complexity of systems, some experts suggest to explore new concepts like system resilience without focusing on existing hazard analysis tools (Ligeron, 2006).

In the following subsection, the concept of resilience is presented in more details.

2.2 Safety assessment

By studying resilience in several application fields, it appears that there is no formal definition of this concept and each application domain provides a definition focused on their problems (Goussé, 2005; Martin, 2005; Hollnagel and Woods, 2006; Poupon and Arnoult, 2006; Foussion and Linkowski, 2007; Zieba and Al., 2007; Morel and Al, 2009; Riana and Terje, 2011). It also appears that its terminology is shared with that of the stability concept. Some authors express however differences between the concepts of resilience and stability, although they are close. According to (Lundberg, 2006), the stability is the ability of the system to respond to regular disturbances or events while resilience focuses on unprecedented disturbances or events. Regular events are defined as well-known events (failure machine for example); irregular events are events that it is possible to imagine but, which are normally rare (earthquake for example); lastly, unprecedented events are so rare that normally no organized mechanisms for coping with them exist (flooding of New Orleans for example).

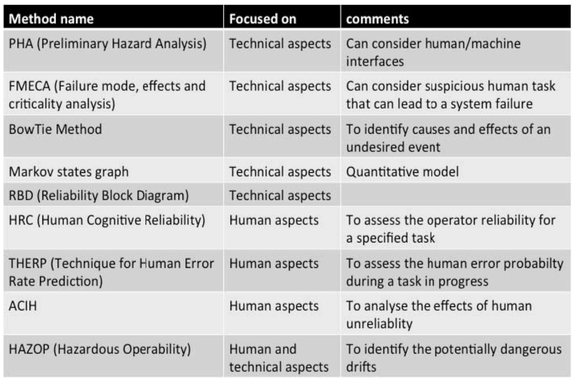
Based on these findings, a definition of the resilience is proposed in (Richard, 2012). Thus, the resilience could be the ability of a system to maintain or return to its original state or to an optimal area of stability. The resilience is able to manage the occurrences of disturbances (see figure 1) by responding:

- in a proactive way : the resilience aims to identify weak signals that may be causing an alarming situation and to correct this situation that might become catastrophic,
- in a reactive way : in this case, the unexpected event happened; the system or/and the operator must react in order to compensate this occurrence,
- or in a curative way : the incident or accident can not be avoided, but the system or the operator is able to limit the consequences of the event.

The system can absorb a disturbance, either by returning to its original equilibrium point after the event occurrence, or by determining a new equilibrium point and by reaching it after an unstable period.

2.3 Taking into account of human aspects

Whatever its level of sophistication and automation, a complex system, such as a guided transport system cannot produce optimum performances and avoid the risk of disastrous events, without the assistance of a human operator who is responsible for the system supervision. To understand the operator as a safety element of the system and not only as a disruptive element, it seems necessary to control the variables that characterize the human behaviour during a dynamic situation (Duquesne, 2005). Although the operator has various faculties and cognitive strategies for problem solving, his behaviour may cause unintended errors in certain circumstances such as deviation of his workload or the manifestation of a dissonance, if he does not evolve in another state. The concept of human stability as defined in the third part of this article aims to highlight and to understand this behavioural duality of the operator, which allows him to be both the weak element and an important element of the system.



Method name	Focused on	comments
PHA (Preliminary Hazard Analysis)	Technical aspects	Can consider human/machine interfaces
FMECA (Failure mode, effects and criticality analysis)	Technical aspects	Can consider suspicious human task that can lead to a system failure
BowTie Method	Technical aspects	To identify causes and effects of an undesired event
Markov states graph	Technical aspects	Quantitative model
RBD (Reliability Block Diagram)	Technical aspects	
HRC (Human Cognitive Reliability)	Human aspects	To assess the operator reliability for a specified task
THERP (Technique for Human Error Rate Prediction)	Human aspects	To assess the human error probabilty during a task in progress
ACIH	Human aspects	To analyse the effects of human unreliablity
HAZOP (Hazardous Operability)	Human and technical aspects	To identify the potentially dangerous drifts

Table 1

3 Human stability formalization and identification

This section allows the characterization of human stability in relation to criteria relating to the human

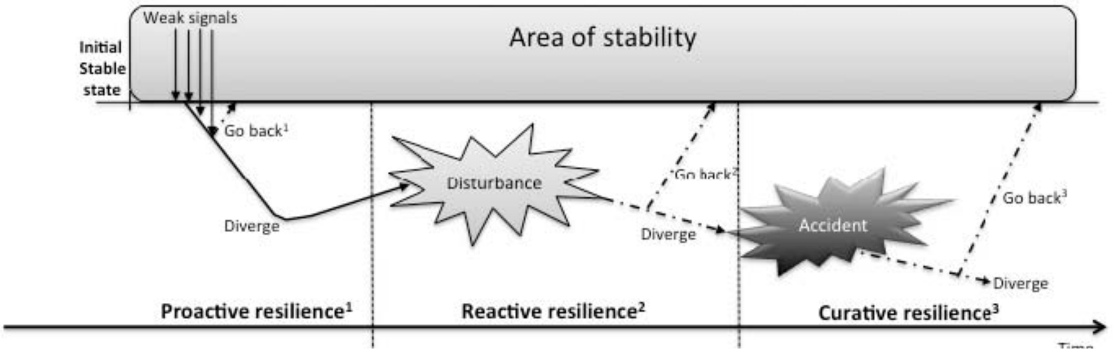


Figure 1 Resilience interpretation

operator. Afterwards, disturbances are regular events, as defined in the subsection 2.2; this hypothesis explains the choice of human stability terminology rather than human resilience terminology.

3.1 Definition

Human stability is defined as the ability of a human operator to be and to stay in a stable state in given conditions (environmental, organisational conditions) for one or more criteria (workload, task achievement, etc; see §3.4). This ability presents the state or the transitions between various states for the operator. These different states are described in the subsection 3.2.

3.2 Formalization

Based on some of the definitions of stability in automation, the human stability refers to a set of states and transitions between these different states (see figure 2):

- Stable state: for the studied criterion, the operator is in a stable state if and only if the value of the criterion is contained between two limit values (Bounded Input, Bounded Ouput principle). These limit values are subject to change depending on conditions in which the operator is (environmental, organizational). A state is considered as stable for the studied criterion if $x(t) < x(t_b)$- $\alpha_1\, x(t_b)$ or $x(t) > x(t_b)+ \alpha_2\, x(t_b)$
- Unstable state: one (or more) criterion of human stability diverges. A state is considered as unstable for the studied criterion if $x(t) < x(t_b)$- $\gamma_1\, x(t_b)$ or $x(t) > x(t_b)+ \gamma_2\, x(t_b)$
- Indeterminate state: It is an unspecified state.

The transitions between these different states can be formalized by:

- Leaps: these transitions represent the sudden and rapid transition from a stable state to another one.
- Breaks: these transitions represent the sudden and rapid transition from a stable state to an unstable state and vice versa.
- Indeterminate transition: the state of destination is indeterminate.

A transition can be identified if: $x(t) < x(t_b)$- $\beta_1\, x(t_b)$ or $x(t) > x(t_b)+ \beta_2\, x(t_b)$ with $\beta_1 > \gamma_1$, $\beta_2 > \gamma_2$

With $x(t)$, value of the studied criterion at time t; $x(t_b)$, value of the studied criterion at time t_b, α_1 and α_2 lower and upper limit of the stability state (these values are empirically determinate), β_1 and β_2 are the switching amplitude; γ_1 and γ_2 are the divergence amplitude.

Although the process of identifying states and change of state is classic in style for the Human stability, it is different in substance. In contrast to technical systems, « to stay in a stable state for a long time » for a human operator might be dangerous with regard to safety (for example, in monotonous context, the Operator may loose its vigilance).

3.3 Identification

The human stability parameter being formalized, the target is to monitor it during disturbed situation. The identification and detection of states and state changes are determined by the AT (time-dependent algorithm) and AS (sequential algorithm) algorithms.

3.3.1 AT algorithm

The text In order to detect a stable state, the AT algorithm checks at each sampling step that the value of the studied criterion remains around the first measured value. In order to detect an unstable state, the algorithm controls

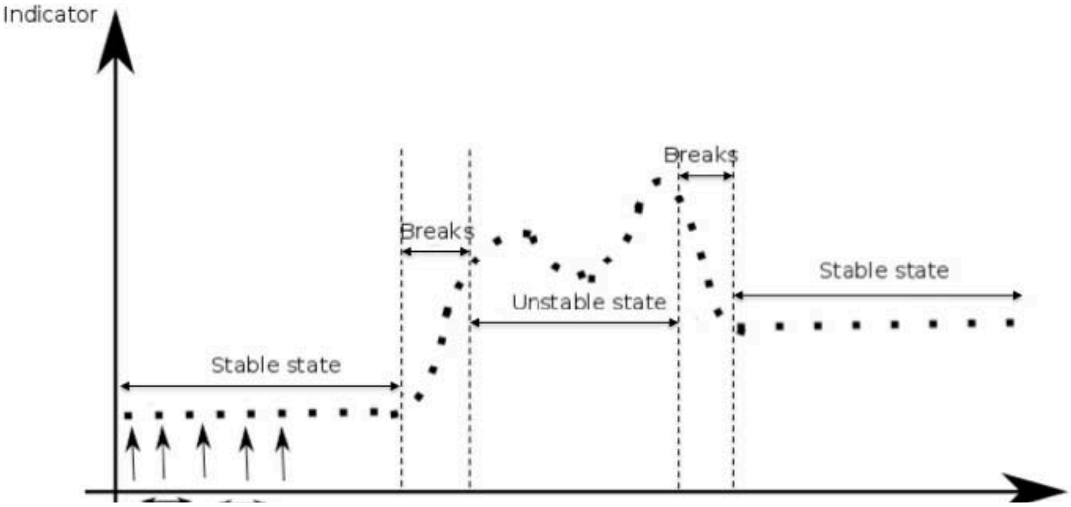


Figure 2 Graphic different states and the Human Stability

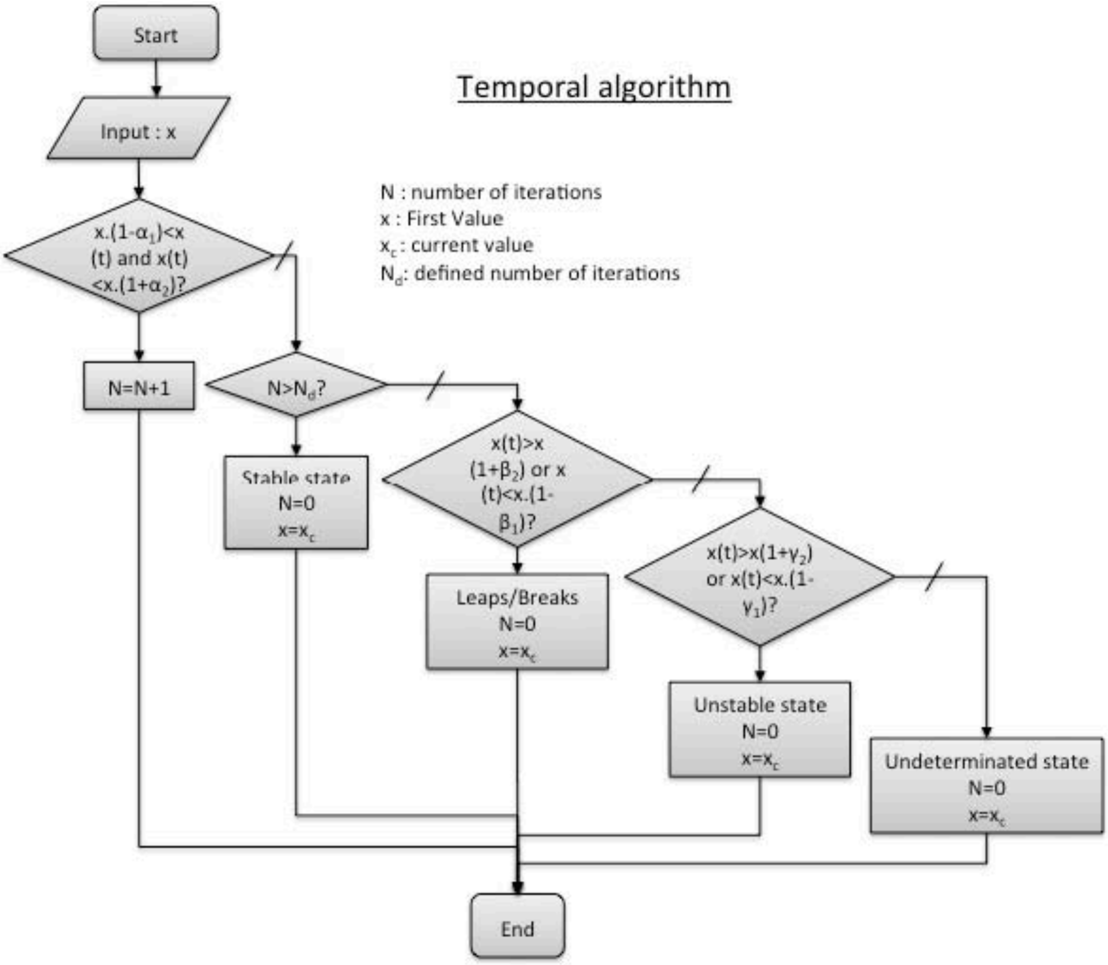

Temporal algorithm
Start
Input : x
N : number of iterations
x : First Value
x_c : current value
N_d: defined number of iterations
x.(1-α_1)<x(t) and x(t)<x.(1+α_2)?
N=N+1
N>N_d?
Stable state N=0 x=x_c
x(t)>x(1+β_2) or x(t)<x.(1-β_1)?
Leaps/Breaks N=0 x=x_c
x(t)>x(1+γ_2) or x(t)<x.(1-γ_1)?
Unstable state N=0 x=x_c
Undeterminated state N=0 x=x_c
End

Figure 3 AT algorithm diagram

the divergence of the criterion in relation with the measured values previously. For a transition between states, it aims to determine a brief divergence with high amplitude compared to the previous measurement. This AT algorithm is presented by figure 3.

3.3.2 AS algorithm

The AS algorithm aims to identify sequences of stability states or stability state changes concerning the operator. The goal is to highlight specific sequences (signatures) that can be correlated with disturbances affecting the system or its environment. The figure 4 illustrates the AS algorithm.

The identification algorithms of human stability are evaluated on different criteria from the operator behaviour. The representative criteria are described in the subsection 3.4. For example, when approaching an element of railway infrastructure (level crossing, tunnel or station), the train has to slow down. In a normal situation of this type, we can expect that the driver switches from one stable state to another stable state for the task "speed control" (i.e. to obtain a sequence prescribed or recommended by the designer such as stable/leap /stable).

During a usual situation, driving or supervision tasks are monotonous and repetitive. In the scope of guided transport systems, it is interesting to know both the behaviour and the performances of the operator as well as his intrinsic state for which these kind of tasks can lead to negative effects such as hypo vigilance, fatigue, inattention, etc.). It is mentioned in (Edkins, 2007) that a majority of accidents related to human error in rail transport are linked to attention criteria. In (Richard and Al, 2010), the criteria are classified according to three categories (see figure 5).

The category "state" is intrinsic and is not easily observable. It can assess for example the workload of operator (Sperandio, 1980). It is divided into three aspects:

- "Cognitive" aspect. : cognitive indicators represent the "degree of knowledge monopolized by the Human operator in his/her activity, which are the skill levels, rules and deep knowledge identified by (Rasmussen, 1980).
- “Psychological” aspect. Psychological indicators represent the human operator’s feelings: stress,

dissatisfaction, frustration, inhibition or even guilt.

- “Physiological" aspect. Physiological indicators give indirectly information to the mental work of Human operator: ocular activity (eye movements, gaze direction, blinks), facial recognition, heart rate and speech. Other categories are extrinsic and more easily measurable.
- The "behaviour" category focuses on the system parameters that are directly controlled by the human operator: speed and inter-distance for a guided transport system, for example.
- The “performance measures" category focuses on the compliance of the human operator with driving rules and safety standards and the quality of the product or service. Among the technical evaluation of these indicators include the sense of obligation indicators, the technical characteristics of the added task, or the analysis of the changes in operating behaviour (Spérandio, 1980).

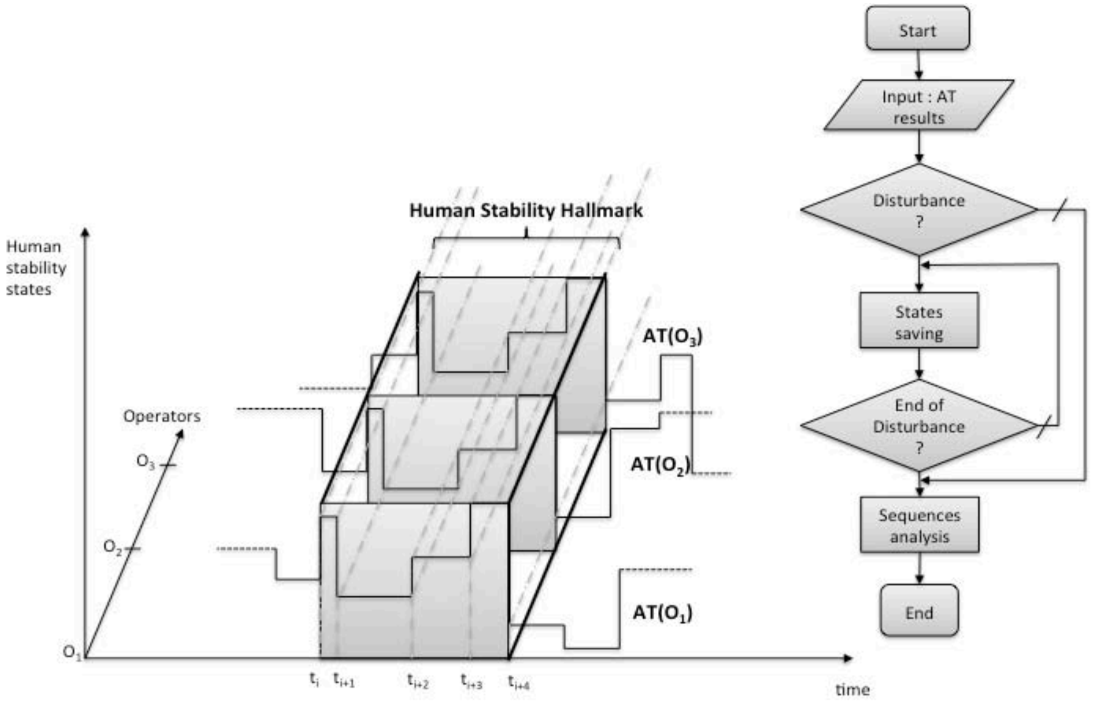


Figure 4 AS algorithm diagram

Finally, if this three-dimensional structure of human stability indicators shown in Figure 5 seems generic, the formulation of indicators can be answerable due to the nature of the system.

4 Application to ETCS system

This part presents the application of the human stability to the guided systems field. This application uses the new ETCS rail control system. Two scenarios have been defined: the first, in order to familiarize the 10 selected students, who had no knowledge in railway domain, with the ERTMS platform (http://www.inrets.fr/linstitut/unites-de-recherche-unites-de-service/estas/equipements-scientifiques/simulateurertms.html) during an ordinary driving operation. The second scenario proposes the same course in a disturbed-driving situation with the same selected students.

4.1 The ERTMS platform

The ERTMS platform is made up of various modules (traffic management module, driving module, 3D module to reproduce the driving environment) and is compliant with SRS 2.3.0d (European Railway Agency, 2008). The objectives of this platform are mainly to optimize the traffic management, to certify real railway components and software in a virtual environment, to test driving situation for different rolling stock configurations and to train drivers and maintain their knowledge.

4.2 Experimentations

The simulation put 10 students in a driving situation with a high-speed train and a 60 kilometres long track made up of various infrastructure elements (bridge, stations, tunnels, level crossing, etc). The traffic on this track is light and 6 events are positioned in the track at different milestones in order to disturb the drivers with a work area, a cognitive dissonance (contradictory data between on-board signals and external signalling: Authorization

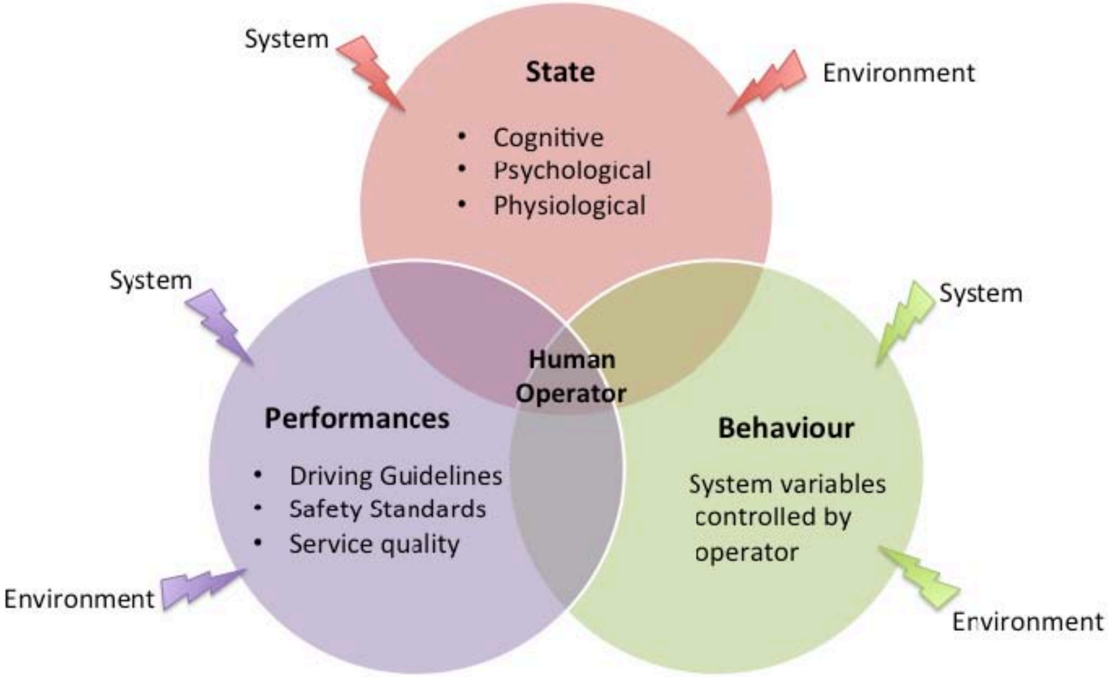


Figure 5 Criteria related to the human operator

by DMI to pass a red light signal), 3 changes of ERTMS level, and a change of driving mode (transition from full supervision to on-sight mode). The experimentation aims to identify via the AT algorithm the states and change of states natures for 3 studied criteria (to respect the speed instructions, to respect the train timetable, to ensure the passengers comfort) in order to extract via the AS algorithm some operator behavioural signatures during the scenario and in particular when disturbance occurs (see figure 6). These criteria are derived from the "behaviour" category defined in 3.4.

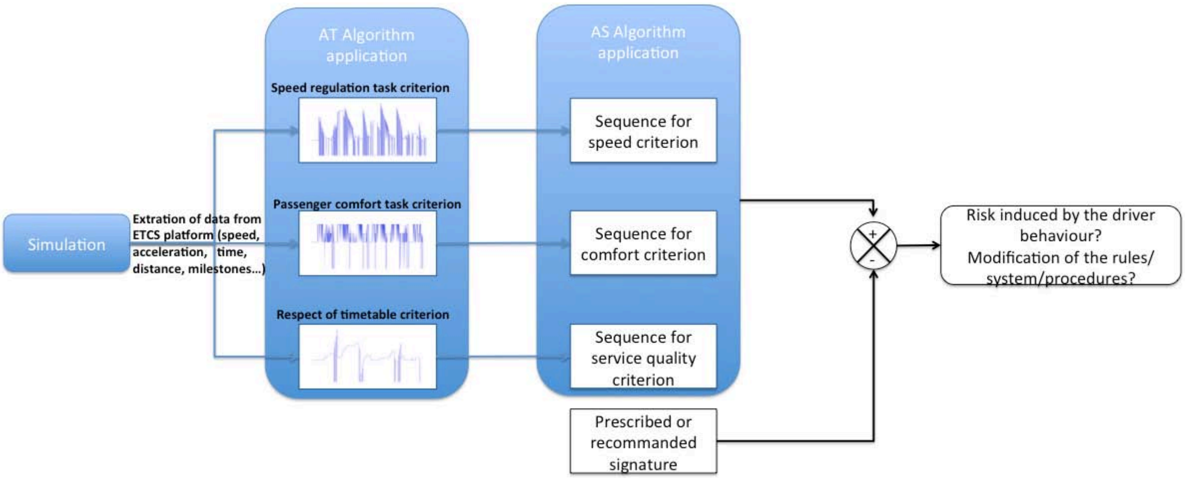
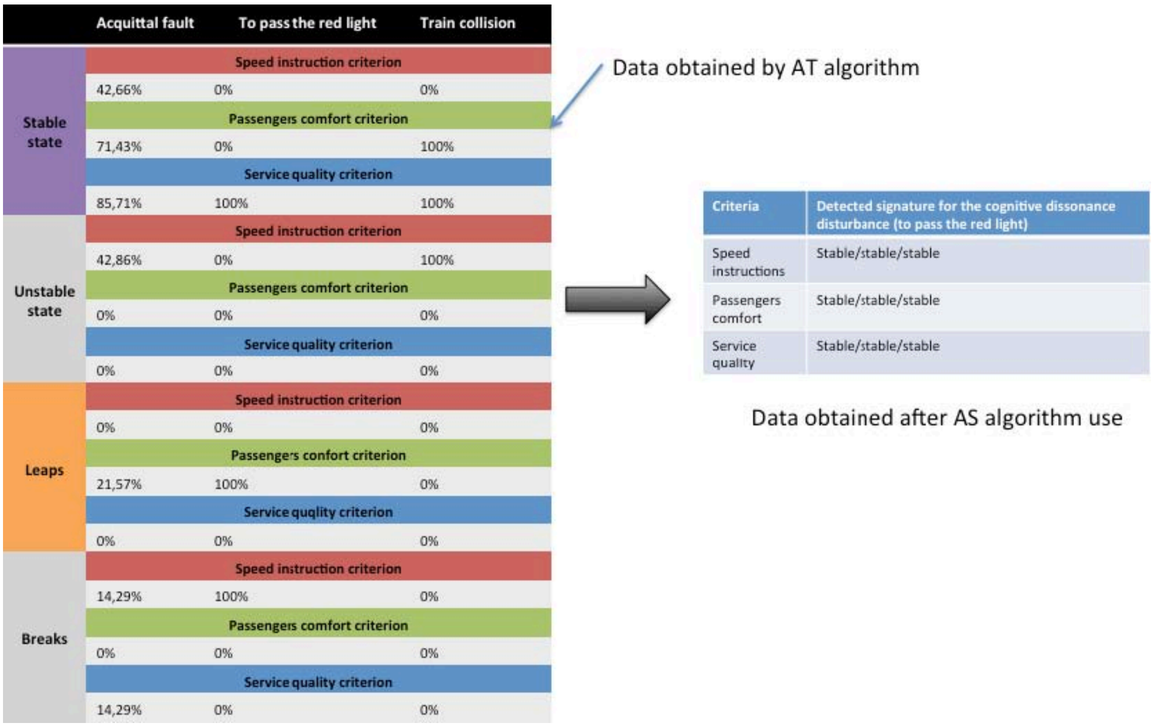

AT Algorithm application
Speed regulation task criterion
Passenger comfort task criterion
Respect of timetable criterion
Simulation
Extration of data from ETCS platform (speed, acceleration, time, distance, milestones...)
AS Algorithm application
Sequence for speed criterion
Sequence for comfort criterion
Sequence for service quality criterion
Prescribed or recommanded signature
+
-
Risk induced by the driver behaviour? Modification of the rules/ system/procedures?

Figure 6 Protocol of the experiment

4.3 Results

The first results obtained by the AT algorithm show the state of the operator when a disturbance occurs. It determines what state of stability or transition between states was concerned at the occurrence of the disturbance for the studied criterion. Once the states and transitions identified by criterion for each student, then, AS algorithm allows detecting the signature associated to each disturbance. For the case study here, a signature is considered as the sequence of 3 states or transitions (before the occurrence of the disturbance, at the time of the occurrence of disturbance, after the occurrence of the disturbance).

These signatures are sequences of stability states and can be recommended or risky. Recommended signatures entail a success in the disturbance management while risky signatures entail a failure in the disturbance management. Figure 7 illustrates results for some disturbances examples processed by AT and AS algorithms. It shows the occupation rate of the operator in the different states and transitions for each disturbance and the advised signature when the cognitive dissonance occurs.



	Acquittal fault	To pass the red light	Train collision
Stable state	Speed instruction criterion		
	42,66%	0%	0%
	Passengers comfort criterion		
	71,43%	0%	100%
	Service quality criterion		
	85,71%	100%	100%
Unstable state	Speed instruction criterion		
	42,86%	0%	100%
	Passengers comfort criterion		
	0%	0%	0%
	Service quality criterion		
	0%	0%	0%
Leaps	Speed instruction criterion		
	0%	0%	0%
	Passenger's confort criterion		
	21,57%	100%	0%
	Service quqlity criterion		
	0%	0%	0%
Breaks	Speed instruction criterion		
	14,29%	100%	0%
	Passengers comfort criterion		
	0%	0%	0%
	Service quality criterion		
	14,29%	0%	0%

Data obtained by AT algorithm

Criteria	Detected signature for the cognitive dissonance disturbance (to pass the red light)
Speed instructions	Stable/stable/stable
Passengers comfort	Stable/stable/stable
Service quality	Stable/stable/stable

Data obtained after AS algorithm use

Figure 7 Some results extracted from the AT and AS algorithms for some examples of disturbances

5 Conclusions and perspectives

This paper proposes the study of a new concept for evaluating the behaviour of a Human operator in the man-machine systems: the human stability. The experimentations discussed in this article assess different criteria independently. It allows identifying the nature of states and state changes linked to the parameter "human stability " and highlights recommended and risky signatures for each disturbance. Nevertheless, the mono-criterion study does not seem sufficient. The operator activity cannot be reduced to a single criterion, but may be influenced by a set of criteria from different categories (workload, stress, personal or physical problem...) together influencing the system. The multi-criteria study will require a proposal of a new formalism to improve the study of human stability. It will also be necessary to weight the criteria, i.e. to provide a level of importance for each or a set of criteria. This work suggests a diagnosis of human stability too: why the state of the driver behaviour was unstable at the time of the disturbance occurrence, was this a special case? It would be interesting to investigate deeper with more students in order to assess with an acceptable degree of certainty the signature at the time of the disturbance. It seems interesting too, to develop this concept with an objective of prediction. In (Richard et al., 2009), it is proposed a study on Human operator modelling by the dynamic hybrid system community. This type of model can take into account continuous and discrete components of the Human operator. Another perspective of these works is to extend the study of human stability to the others categories evoked in paragraph 3.4, in particular for the facial recognition (Luong, 2006). In this context, determination of states and transitions via facial recognition application can be done on line. Lately, by combining several criteria of different categories, this kind of application could be implemented in the driver cab in order to identify on line the human stability and to alert the operator when his/her behaviour seems risky.

6 References

Amalberti, R. (2001). The paradoxes of almost totally safe transportation systems., Safety science, pp 109-126.

Duquesne,L. (2005). Jugement multicritère d'acceptation individuelle de la signalisation routière variable.

Mémoire de Master Recherche AISIH, UVHC, Valenciennes, France, Juillet 2005.

European Railway Agency (2008), System Requirements Specification (subset 026), technical report.

Fossion, P. and Linkowski, P. (2007), La pertinence du concept de résilience en psychiatrie. In Rev Med Brux. Goussé, V. (2005), Apport de la génétique dans les études sur la résilience : l'exemple de l'autisme. Annales Médico-Psychologiques.

Hollnagel, E. and Woods, D. (2006) Resilience Engineering: concepts and precepts Epilogue: Resilience Engineering Precepts. Ashgate Publishing Ltd.

Ligeron, J.C. (2006), Le cercle des fiabilistes disparus ou critique de la raison fiabiliste. Editions préventique, 2006.

Lundberg, J. and Johanson, B. (2006). Resilience, stability and requisite interpretation in accident investigations. Proceedings of the Second Resilience Engineering Symposium, (Eds.) Hollnagel, E. & Rigaud, E., 191-198

Luong H.V. (2006), Reconnaissance multimodale de gestes de communication non verbale.

Martin S. (2005), la résilience dans les modèles de systèmes écologiques et sociaux, PhD thesis, ENS Cachan.

Morel G. et al. (2009). How good micro/macro ergonomics may improve resilience, but not necessarily safety . Safety Science, 47, 285–294.

Poupon, Y. and Arnould, P. (2006). Mécanique : choix d'un matériau polymère injectable.

Rasmussen, J. (1980). What can be learned from human error reports? In K. Duncan, M. Gruneberg & D. Wallis (Eds.), Changes in working life. Wiley: London.

Riana S., Terje A. (2011). A risk perspective suitable for resilience engineering. Safety Science, 49, 292–297.

Richard, P., Vanderhaegen, F., Benard, V. and Caulier, P. (2009). Proposition of establishment of a model of human-machine system by hybrid dynamic system model applied to guided transport. Proceedings of 8. Berliner Werkstatt Mensch-Maschine-Systeme, Berlin, Germany, 7-9 October 2009.

Richard, P., Benard, V., Vanderhaegen, F., Caulier, P. (2010), Towards the Human stability in transportation systems : concepts and objectives. In IFAC HMS, 2010.

Richard, P. (2012). Contribution à la formalisation et à l'identification de la stabilité humaine au regard de la sécurité : application aux transports guidés. Phd Thesis. Université de Valenciennes et du Hainaut-Cambrésis.

Zieba, S. et al. (2007), Resilience and affordances: perspectives for human-robot cooperation? European Annual Conference on Human Decision Making and Manual Control

Rapid Risk Assessment of Technical Systems in Railway Automation

Jens Braband
Siemens AG
Ackerstr. 22, 38126 Braunschweig, Germany
jens.braband@siemens.com

Abstract

The European Railway Agency (ERA) has the challenging task of establishing Common Safety Targets and Common Safety Methods throughout Europe. In this context, the harmonization of risk assessment methods is also discussed. The purpose of this paper is to present a new approach to risk assessment of technical systems in railway automation, which allows a rapid risk assessment while at the same time also allowing a rigorous check that the method is well constructed and robust. As a particular reference, a new German pre-standard, which lays out requirements for such semi-quantitative approaches, is taken into account. A particular method is constructed in this paper and the means by which compliance with legal and regulatory requirements can be demonstrated, is discussed. Although the paper deals with the European legal framework in railway automation, the approach can easily be generalized to other legal frameworks and other application domains.

1 Introduction

The European Railway Agency (http://www.era.europa.eu), established by European Regulation 881/2004, has the mission of reinforcing railway safety and interoperability throughout Europe despite continuing privatization. Central to its work on railway safety is the development of measures based on common safety targets (CST) and common safety methods (CSM), common safety indicators (CSI) and harmonized safety certification documents. For some work and problems related to the assessment of CST see Braband and Schaebe (2012).

The CSM describe how safety levels, the achievement of safety targets and compliance with other safety requirements are assessed in the various member states. As a first step, EC Regulation 352/2009 will finally come into force for the complete European railway sector by July 2012. In this regulation, a semi-quantitative risk acceptance criterion for technical systems (RAC-TS) similar to civil aviation has been introduced: *For technical systems where a functional failure has credible direct potential for a catastrophic consequence, the associated risk does not have to be reduced further if the rate of that failure is less than or equal to 10^{-9} per operating hour*.

This criterion is limited to those technical systems where failure can lead to catastrophic effects, e.g. train accidents involving many fatalities, and for which there are no credible barriers or substantial mitigating factors that will prevent this consequence from materializing. The criterion can be used for the utmost critical functions performed by technical systems on railways such as speed supervision, control of the switch position, complete and permanent loss of the brake system, or loss of the traction cut-off function. This means that formally RAC-TS is related only to potentially catastrophic accidents, similar to the criterion related to hull loss accidents in civil aviation. In order to apply it also to other severity categories, RAC-TS must be embedded in a risk analysis method.

In this paper we focus on semi-quantitative risk analysis methods, which are very similar to the rapid risk assessment method approach advocated by Johnson (2011). In fact one purpose of the paper is to motivate and demonstrate that semi-quantitative methods are in fact rapid risk assessment methods, but satisfy additional requirements.

The paper is organized as follows: after a discussion of problems related to risk analyses, an applicable standard is reviewed, from which the requirements are taken. These requirements are compared to the requirements for rapid risk assessment methods. Then a new risk analysis method is constructed and some arguments and examples concerning the validation of the method are presented.

2 Problems with risk analyses in railway applications

Risk is a combination of accident severity and accident frequency. Accident frequency may be calculated by hazard frequency and the probability of a hazard developing into an accident. This probability is derived by taking into account the effectiveness of barriers. Barriers are understood as any means to prevent, control, or mitigate undesired events or accidents. Barriers must be under the control of the organization operating the system as they have to be enforced during operation. They can be of different origin, e.g. human actions, operational barriers, technical barriers.

It is well known that risk acceptance is an intricate topic and that risk analyses in railways may be quite time-consuming and tedious, in particular when they are performed quantitatively, see e. g. Braband (2005) for an overview. There exist simpler semi-quantitative methods, e.g. risk matrix, risk graph or risk priority numbers;

however, they often lack justification and it is not clear whether the derived results are trustworthy. So, a major research challenge is to construct dependable semi-quantitative methods.

In particular, schemes based on risk priority numbers (RPN) are widely used in Failure Modes, Effects and Criticality Analyses (FMECA) although it is known that they have not been well constructed and that their use may lead to incorrect decisions, for the following reasons:

- The risk of different scenarios that lead to the same RPN may differ by orders of magnitude.
- Scenarios with similar risks lead to different RPN.

This has already been observed by Bowles (2003) and has now also lead to cautionary advice in the standards.

Risk matrices are a well-known tool in risk assessment and risk classification, and are also used in the railway domain (see for example EN 50126 (1999) or Braband (2005)). Some major problem of such risk matrices are:

- Risk matrices must be calibrated to their particular application.
- The results depend on the system level to which they are applied.
- The parameter classes must be concisely defined in order to avoid ambiguity and misjudgments.
- It must be defined which frequency is meant, e.g. accident or hazard frequency.
- It is not directly possible to take barriers or risk reduction factors into account in the risk matrix.

However, if these problems can be overcome, risk matrices are a well-accepted and easy-to-use tool, and can be useful for risk prioritization. When risk matrices are to be applied in the railway domain, they need to be applied in combination with a method which can additionally take into account the effect of barriers and their related risk reduction. Typical candidates for additional methods would be the fault tree analysis (FTA) in a quantitative analysis or semi-quantitative tables as used by risk priority numbers.

In conclusion for the railway domain rapid - in particular semi-quantitative - methods are very attractive and already widely used, but their justification is often questionable. Only a few approaches (see Bepperling (2008) and Milius (2010)) have been presented so far where semi-quantitative methods have formally been validated. A standard for the use of such methods, or against which methods can be validated, has been missing so far.

3 Construction of a semi-quantitative risk analysis method

3.1 DIN V VDE V 0831-101

Recently this German pre-standard DIN (2011)has clearly set out requirements for semi-quantitative risk analysis methods. It is now possible to construct a method and validate it with respect to these requirements. There are in total 28 requirements. Not all of these relate to construction of the method - some concern its application. Table 1 gives an informative overview of the requirements; the mandatory requirements appear in bold. For more details we have to refer to DIN (2011).

Construction	**A1**	**State reference units and application scope.**
	A2	**Be conservative in your assessment.**
	A3	Make sure parameter granularity is sufficient.
	A4	**Work out a user guide.**
	A6	**State clearly the applicable system level**
	A8	Allow for hazard classification.
	A12	**Assessment of accident severity**
	A13	**Assessment of accident frequency**
	A14	Description of all barriers
	A15	**The tables should be compatible.**
	A17	**Assessment of human reliability**
	A18	**Assessment of operational barriers**
	A19	**Assessment of exposition**
	A20	**Assessment of external barriers**
	A21	Assessment of technical barriers
	A22	**Take into account dependencies of barriers.**
	A23	**Calibrate the method (against a RAC).**
	A24/ A25	**Assure proportionality between risk and criticality.**
	A26	**Small changes lead to small changes.**
	A27	**A safety requirement has to be derived.**
	A28	**Give rules on how to derive the Safety Integrity Level**
Application	**A5**	**Justification of parameter choice**
	A7	**Identify hazards systematically.**
	A9	**Work out hazard scenarios.**
	A10	**Justify the choice of the relevant scenario.**
	A11	**Document results in a hazard log.**
	A16	**Identify safety-critical application conditions.**

Table 1: Summary of requirements

3.2 Requirements for rapid risk assessment methods

Johnson (2011) has gathered principles from leading application examples to define basic principles for rapid risk assessment methods:

1. Consistency between different 'analysts' looking at similar incidents;
2. Repeatability: the same 'analyst' should derive similar findings for similar incidents looked at over a period of time;
3. Economy: not more than one day's training in safety management or hazard analysis should be necessary;
4. Validity: Rapid risk assessment techniques should be confirmed and refined using all available information about previous accidents and incidents;
5. Applicability: should be applicable to operational tasks and must support everyday decision making.

3.3 Risk Score Matrix approach

In this paper, a semi-quantitative approach is proposed that fulfils all requirements of the German pre-standard DIN V VDE V 0831-101 and also Johnson's criteria. It is called the Risk Score Matrix (RSM) and consists of the application of a risk matrix and score tables for assessment of the barriers, similar to RPN schemes. The complete approach is shown in Figure 1, including additional and alternative steps. The final result consists of hazard rates (HR) related to the functional failures of the technical system and the assumptions on which the analysis rests, which may turn into safety-related application rules (SAR). This process is explained in detail in the following chapters.

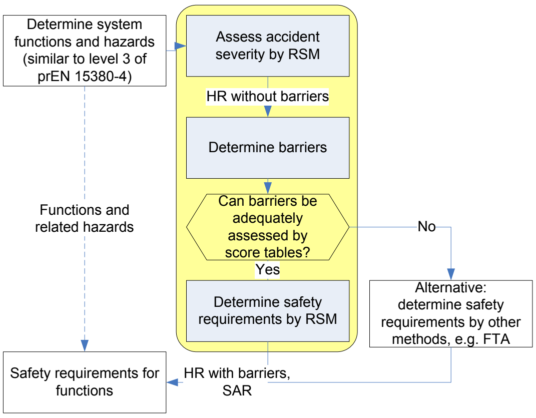

Determine system functions and hazards (similar to level 3 of prEN 15380-4)
Assess accident severity by RSM
HR without barriers
Determine barriers
Can barriers be adequately assessed by score tables?
No
Yes
Functions and related hazards
Determine safety requirements by RSM
Alternative: determine safety requirements by other methods, e.g. FTA
Safety requirements for functions
HR with barriers, SAR

Figure 1: Overview of the Risk Score Matrix model

4 Description of the approach

4.1 System definition

The discussion in this paper focuses on technical systems only. According to EU Regulation 352/2009, a technical system is a product developed by a supplier including its design, implementation, and support documentation. It should be noted that:

- The development of a technical system starts with its system requirements specification and ends with its safety approval.
- Human operators and their actions are not included in a technical system. However, their actions may be taken into account as barriers mitigating the risk.
- Maintenance is not included in the definition, but maintenance manuals are part of the product.
- Technical systems can be subject to a generic type approval, for which a stand-alone risk acceptance criterion is useful.

A function is defined in prEN 15380-4 (2010) as a "specific purpose or objective to be accomplished that can be specified or described without reference to the physical means of achieving it." A function level is a "level, to group functions of equal purpose". The distinction between levels is described informally as follows:

- First-level function: functional domain that encompasses a set of functions related to the same general focus or service for the considered (rolling stock) system.
- Second-level function: related to a specific set of activities that contribute to completion of the functional domain defined at the first level (at this level, it is not said how a second-level function is to be implemented).
- Third-level function: related to a specific activity out of the related set of activities, it encompasses a set of tasks (a function at least at level 3 should be supported as much as possible by one single subsystem).

It is proposed to use prEN 15380-4 (2010) which contains up to five hierarchical levels. Taking into account the definition of function level, level 3 seems to be the most appropriate for the application of RAC-TS. At least it does not seem reasonable to go into more detailed levels such as level 4 or 5. Table 2 gives a non exhaustive list of functions to which RAC-TS may be applied. Although prEN 15380-4 (2010) relates to rolling stock only, it can be extended to infrastructure functions quite easily, e.g. by identification of all interfaces of other functions to rolling stock. Some functions (or at least interfaces) are already defined. In Table 2, some examples of level 3 functions related to signalling are proposed.

Code	Function description
=LBB	Detect track vacancy
=LBC	Detect train at a particular spot
=LBD	Locate train
=LCB	Determine train description
=LDB	Provide diagnostics
=LEB	Supervise driver vigilance
=LEC	Automatic train stop
=LED	Supervise braking curve
=LEE	Supervise maximum train speed
=LFB	Optimize train running
=LGB	Monitor switch
=LGC	Lock switch
=LGD	Monitor derailer
=LGE	Lock derailer
=LGF	Monitor level crossing
=LHB	Provide signal information
=LJB	Provide cab radio
=LKB	Display state to driver
=LKC	Display state to dispatcher
=LKD	Transmit commands

Table 2: Examples of signaling functions

4.2 Risk matrix

A suitable risk matrix has already been proposed and justified in Braband (2011), see Table 3. The table shows intolerable and tolerable combinations in a frequency scaling of $\sqrt{10}$ and has been calibrated to match RAC-TS. Safety targets would be chosen at the boundary between these two regions (medium gray shading). This scaling is compatible with the common scaling for Safety Integrity Levels (SIL), as two classes form one SIL. Note that for higher severity levels a slight risk aversion has been taken into account and that there are no particular safety requirements for category A.

HR	B	C	D	E
n. a.	█	█	█	█
10^{-5}/h	▓	█	█ Intolerable	█
$3x10^{-6}$/h	░	█	█	█
10^{-6}/h	░	█	█	█
$3x10^{-7}$/h	░	▓	█	█
10^{-7}/h	░	░	█	█
$3x10^{-8}$/h	░	░	█	█
10^{-8}/h	░ Tolerable	░	▓	█
$3x10^{-9}$/h	░	░	░	█
10^{-9}/h	░	░	░	▓ RAC-TS

Table 3: Proposed risk matrix

The corresponding accident severities are defined in Table 4. Classification can be performed based on a qualitative estimate of the typical accident severity or based on statistical data (fatalities and weighted injury score (FWI)). Note that "typical" does not mean worst case; in a safety sense, it should be interpreted as a typical bad outcome, i.e. worse than average. When considering statistical data, it should be noted that railway accident severity statistics are often highly asymmetric and skewed, so that particular care has to be taken when evaluating such statistics.

ID	Combinations	FWI range	Typical FWI
E	Multiple fatalities	2≤FWI	5
D	Single fatality or multiple serious injuries	0.2≤FWI<2	1
C	Single serious injury or multiple light injuries	0.02≤FWI<0.2	0.1
B	Single light injury	0.01≤FWI<0.02	0.01
A	-	FWI<0.01	n. a.

Table 4: Consolidated severity categories

4.3 Assessment of barriers

The model generally takes into account the following types of barriers:

- possibility to avoid accident by human interaction (H)
- possibility to mitigate the hazard by an independent technical system (T)
- operational barriers (B)
- low demand frequency (D)

The presence and efficiency of these barriers together with the severity category determines the outcome of the assessment and thus the appropriate safety requirements that will have to be achieved for the technical system under evaluation. The assessment is carried out via a score scheme where scores are allocated to the barriers and then these scores are added to calculate the total risk reduction, starting from the risk matrix in Table 3. Since the scores for the barriers are added instead of multiplied, this means that the scores allocated are given in a logarithmic scale where each score represents a "risk reduction" with a factor of $\sqrt{10}$ and two scores represent a reduction of one order of magnitude (i.e. one SIL). It should be noted that the effectiveness of the barriers must be monitored in operation, typically as a part of the operator's safety management system.

The total risk reduction is then calculated as the sum of scores, possibly reduced by a score accounting for the level of independence of the different barriers present. This is to avoid adding several barriers that are functionally dependent on each other and that are likely to fail simultaneously.

It should be noted that such a semi-quantitative assessment method may not fit all particular problems; e.g. there may be rare cases when other barriers occur and need to be taken into account. Also, some of the tables may be overly conservative, e.g. the assessment of human reliability by parameter H. In such cases, it is advised to apply first the risk matrix (Table 3) without any barriers and evaluate the barriers by an alternative method, e.g. Fault Tree Analysis, Event Tree Analysis or Markov models, as appropriate for the particular problem.

For the sake of brevity, it is not possible to present and discuss all score tables. Instead, the focus will be on the assessment of human reliability to demonstrate the principle.

4.4 Assessment of human reliability

In some situations, it can be foreseen that there are still barriers present after the failure of a technical system due, for example, to the driver or staff observing the problem and acting correctly. Human interaction can also, in some cases, be carried out by passengers or third persons. Examples could be staff or passengers correctly using on-board fire extinguishers in case of fire or similar situations. Evaluation is based on three tables (5a, 5b and 5c) and calculates a combined score as the sum of the following sub-scores:

- type of task
- stress level at which the task is performed
- environmental conditions under which the task is performed

The approach is similar to simple screening techniques in human reliability assessment, e.g. Accident Sequence Evaluation Program (ASEP), e.g. Sträter (1997), or the approach validated by Hinzen (1993). Such approaches are known to be pragmatic and generally conservative. Note that also alternative assessment schemes could be transformed into similar tables. This assessment of human barriers does not pretend to give a deep and exact description of the human actions to be carried out and their reliabilities. It merely intends to give a conservative order estimate and does not replace further ergonomic studies, e.g. on the design of human-machine interfaces.

Pre-conditions for the application of this assessment are:

- Operators must be properly trained and have sufficient experience.
- There must not be any goal conflicts in performing the task, e.g. safety vs. performance.

A – score	Action type	Comment
4	Skill-based	Well-known and trained skill-based action
2	Rule-based	Rule-based action that has been appropriately trained and managed
0	Knowledge-based	But no routines or rules are defined.

Table 5a: Action type assessment

W – score	Work environment	Comment
1	Good conditions	The work is performed under normal conditions with regard to sight, noise, physical forces and weather.
0	Adverse conditions	The working conditions are adverse with regard to at least one factor: lighting, noise, physical forces (e.g. excessive vibrations) or adverse weather conditions (too cold, too hot, etc.).

Table 5b: Work environment assessment

ST – score	Stress level	Comment
1	Optimal	
0	Excessive demands	The work load is very demanding. The stress level is high, e.g. work under time pressure.
	Insufficient demands	The work performed is not very demanding and mostly routine.

Table 5c: Stress level assessment

The combined score is then calculated from Tables 5a, 5b and 5c as H= A + W + ST.

4.5 Assessment of barrier dependence

For every barrier that is taken into account, it must be analyzed whether its risk reduction is independent of the other barriers. If it is not, some scores will be subtracted from the score of the barrier, in accordance with Table 6b below. If the correlation is strong, the new barrier may reduce the risk only marginally.

Tables 6a and 6b can be justified on the basis of experience with conditional failure probabilities in human task analysis, e.g. Sträter (1997) and common cause analysis of technical systems.

The reduction of the barrier score is calculated by Table 6b, which gives the reduction of the barrier score Φ as a function of the original barrier score (top row) against the dependence of the new barrier with respect to all previous barriers.

Dependence class	Comment
Independence (I)	There is no functional dependence between the factors; no common causes for failures exist.
Low dependence (LD)	The barriers are statistically independent; no significant physical influence. Related to human tasks, the task is performed by a different person at a different location and in a different operational situation.
Medium dependence (MD)	The mitigating factors have a single common cause failure – if one barrier fails, there is a slightly increased chance that the other also fails. Related to human tasks, e.g. two of the following characteristics are the same: same person, same location or same operational situation.
High dependence (HD)	The barriers have more than one common cause. If one barrier fails, there is a significantly increased chance that the other also fails.
Complete dependence (CD)	Several common causes. The new barrier will not be taken into account.

Table 6a: Dependence classes

Φ	1	2	3	4	4+i
I	0	0	0	0	0
LD	0	0	-1	-1	-(i+1)
MD	0	-1	-1	-2	-(i+2)
HD	0	-1	-2	-3	-(i+3)
CD	-1	-2	-3	-4	-(i+4)

Table 6b: Dependence assessment

4.6 Validation of the Risk Score Matrix method

It is not possible to give all arguments concerning the requirements from Table 1 here, but it is possible to outline a few of the key arguments, whose fulfillment is quite obvious by the construction of the tables. For examples of the complete validation of semi-quantitative approaches, see Bepperling (2008) and Milius (2010).

The scope as well as the units of measurement are well defined by Table 2 and RAC-TS, so A1 and A6 can be fulfilled. As all tables are constructed conservatively, A2 is met. The granularity of the method is set to √10, which fits well to the SIL scale and is reasonable, so A3 can be fulfilled. As this scaling is used consistently throughout all tables, A15 is complied with. The tables shown in this section also meet the respective requirements A12, A13, A17, A18 and A22. The method is also calibrated appropriately against RAC-TS, so A23 follows. The method is monotone with respect to risk (A24), i.e. a higher risk gains a more demanding safety requirement. Also, small changes in the parameters lead only to small changes in the safety requirements (A26).

4.7 Is Risk Score Matrix a Rapid Risk Assessment Method?

We justify the construction of the Risk Score Matrix against the criteria defined by Johnson (2011)

1. Consistency: in particular requirements A1 and A4 would support this jointly with the requirements for justification A5 and A10.

2. Repeatability: this is supported by the harmonized function list from table 2 as well as the requirements for the construction of the tables and also A4 and A5
3. Economy: if the analyst is experienced with respect to the system and the application conditions then one day's training in the Risk Score Matrix method would be sufficient
4. Validity: the tables are based on experience and the method has been validated against all requirements of the standard DIN (2011)
5. Applicability: the method is applicable also to operational tasks, if they include use of technical systems, but they are not intended to be used in daily operations or missions. This is due to the different scope of risk assessment of technical systems in railways and military missions

Finally we conclude that RSM is indeed a rapid risk assessment method, although dedicated to a very particular purpose. It can also be observed that the standard DIN (2011) defines more particular and detailed requirements for semi-quantitative methods than Johnson (2011) does for rapid risk assessment methods. The major difference is that DIN (2011) has more detailed requirements on the construction of the method.

4.8 Examples

In some cases, like =LGB from Table 2, RAC-TS is directly applicable. The main hazard would be that the status of a switch would be determined wrongly so that a train may run over a switch which is set in an incorrect direction. If passenger trains at high speed ran over this switch, then ID E would be determined from Table 4 leading to a THR of 10^{-9} per operating hour per switch. Some human mitigation may be possible (e.g. at low speed) and there is also the possibility that the switch is not set in the branching direction (50% chance), so that the overall score (due to Tables 5a to 5c) may be assessed as 1, leading to a THR of $3x10^{-9}$ per operating hour per switch.

In another example, =LGF from Table 2, the main hazard would be that road traffic would not be protected by the level crossing and the consequence might be a collision at the level crossing, from which ID D as the typical accident severity would be derived from Table 4 leading to a THR of 10^{-8} per operating hour per level crossing. Additionally, human mitigation may be possible (e.g. at low speed or with good sight) by the road users, so that the score (due to Tables 5a to 5c) may be assessed as 1. However, this mitigation is not independent from the severity estimate. Additionally, it can be taken into account that level crossings are not allowed on high-speed lines and often avoided on lines with high traffic density. Thus, finally a score of 1 may be assessed, leading ultimately to a THR of $3x10^{-8}$ per operating hour per level crossing.

5 Conclusion

The risk acceptance and setting of THRs for technical systems can be based on a risk score matrix as explained in this document taking into account a set of typical barriers. This approach is compliant with EC regulations as well as with requirements of the relevant standards.

When using the new Risk Score Matrix approach, mutual recognition will also depend on the list of functions to which the risk matrix is applied. So, the use of a common risk score matrix will facilitate the mutual recognition process, but not lead to an automatic approval.

It has been demonstrated that the Risk Score Matrix is truly a rapid risk assessment method.

6 References

Bepperling, S. (2008). Validation of a semi-quantitative approach for risk assessement on railways (in German), PhD thesis, Technical University of Brunswick

Bowles, J (2003) An Assessment of RPN Prioritization in a Failure Modes Effects and Criticality Analysis, Proc. RAMS2003, Tampa

Braband, J. (2005). Risk analyses in railway automation (in German). Hamburg, Eurailpress

Braband, J. (2010). On the Justification of a Risk Matrix for Technical Systems in European Railways. In E. Schnieder (Ed.), FORMS/FORMAT 2010 (pp. 237-288). Springer

Braband, J. and Schaebe, H. (2012): Assessment of National Reference Values for Railway Safety - A Statistical Treatment, Proc. ESREL2012, Helsinki

CENELEC (1997) EN 50126 Railway applications –The specification and demonstration of Reliability, Availability, Maintainability and Safety (RAMS)

CENELEC (2010) prEN 15380 Part 4: Railway applications – Classification system for rail vehicles – Function groups

DIN (2011) Semi-quantitative processes for risk analysis of technical functions in railway signalling (in German), DIN V VDE V 0831-101

EC (2009) Regulation No. 352/2009 of 24 April 2009 on the adoption of a common safety method on risk evaluation and assessment as referred to in Article 6(3)(a) of Directive 2004/49/EC of the European Parliament and of the Council

Hinzen, A.(1993): The influence of human factors on railway safety (in German), PhD thesis, RWTH Aachen, 1993

ISO (2010) DIS 26262: Road vehicles – Functional safety

Johnson, C.: Using Rapid Risk Assessment Techniques to Combat Degraded Modes of Operation, Proc. ISSC2011, Las Vegas, 2011

Milius, B. (2010). Construction of a semi-quantitative risk graph (in German), PhD thesis, Technical University of Brunswick

Sträter, O. (1997) Evaluation of Human Reliability on the Basis of Operational Experience, PhD thesis, Technical University of Munich

Functional Safety based on a System Reference Model

Manfred Broy
Institut für Informatik,
Technische Universität München
D-80290 München Germany

broy@in.tum.de

Abstract

Ensuring functional system safety comprises four major tasks. First, all possible hazards and risks of incidents with respect to functional safety have to be identified. Second, the system requirements specification must be shown to be valid in the sense that it excludes all the hazards with sufficiently high probability. Third, it has to be shown that the requirements are implemented correctly. Fourth, it must be demonstrated that for the implementation all possible failures of subsystems that could lead to violations of the functional safety requirements systems are excluded with a sufficiently high probability. This way it has to be shown that the specification and its implementation lead to an acceptable risk in terms of probabilities of violations of safety requirements. For a proper engineering of functional safety we suggest the use of a rigorous modelling framework. It consists of: a system modelling theory that provides a number of modelling concepts that are carefully related and integrated; a system reference model; and a reference architecture structuring systems into three levels of abstractions represented by views, including a functional view, a logical subsystem view and a technical view. It is demonstrated how, in this framework, all kinds of safety issues are expressed, analysed and traced; and how, due to the formalization of the framework, safety problems are formally analysed, specified and verified.

Keywords: Functional Safety, Hazards, System Modelling, Requirements, Specification, Design, Architecture.

1 Introduction

It is well accepted by now that software intensive systems - due to their functional power, their tight integration with human machine interaction, their safety critical functionality, and their additional complexity - bring in essential challenges to guaranteeing functional safety. Functional safety of systems addresses the general requirement that there is only a bounded, calculable, and acceptable risk that the usage of the system may result in harm for the health and life of people or other assets.

We suggest a systematic concept to categorize incidents and a comprehensive modelling approach to support functional safety.

1.1 System Development Steps and their Relation to Functional Safety

The development of systems follows a simple and clear structure:

- REQU: elicitation, analysis, and documentation of the requirements and their validation
- SPEC: functional specification of the system, verification of the specification w.r.t. the requirements
- ARCH: design of the architecture by decomposition into subsystems called components and their specification, verification of the architecture
- IMPL: implementation of the components and verification according to their specification
- VEIN: integration and system verification

This structure is reflected in the tasks to guarantee functional safety properties as follows:

- REQU: elicitation, analysis, and documentation of the safety requirements and their validation
- SPEC: verification of safety requirements on the basis of the functional specification
- ARCH: Failure-Modes-and-Effect Analysis (FMEA) on the basis of the architecture – identification of expected failures for components and their probability, analysis of the effects of failures, calculation of probabilities of failures and resulting violations of safety requirements
- IMPL: implementation of the components according to their specification, validation and verification of probabilities as requested in the FMEA
- VEIN: integration and system safety verification

This shows how tightly issues of functional safety are embedded into general system engineering steps, in particular model-based engineering

1.2 A Systematic Approach to Safety Issues

In this section we classify hazards and incidents along the lines of [Gleirscher 11].

1.2.1 Hazards and Incidents

A *hazard* characterizes a potential situation in the usage of a system that represents a degree of threat to life, health, property, or environment. A hazardous situation that has happened in the operation of a system is called an *incident*. A system is functionally safe if it is free of hazards and therefore there is no risk of incidents.

There are two basic ways to define functional safety for systems: empirical and analytical approaches. In an empirical approach, we consider the statistics of systems under operation with respect to incidents; in an analytic approach we analyse a system with the goal to calculate the risk of incidents.

1.2.2 An Empirical View onto Functional Safety

There is obviously a clear pragmatic concept of functional safety in connection with systems and their operation. If we observe the operation of systems over a certain period of time we realize if and how often incidents happen and this way get an empirical assessment of hazards, risk of incidents, and functional safety.

In principle, in empirical approaches we do not need to identify hazards (possible situations that represent a degree of threat to life, health, property, or environment) in advance, but may identify, collect, and classify hazards as the result of empirical observations where hazards are identified via observed incidents. This is much more easy than to identify all hazards in advance, but cannot guarantee functional safety, but only monitor and evaluate functional safety during operation.

1.2.3 Analysing and Guaranteeing Functional Safety

When designing systems with the potential for hazards we have to exclude any unacceptable risk to come to the conclusion that there does not exist a safety problem with the system in operation. Clearly functional safety has the goal to avoid unacceptable risk and hazardous situations.

A systematic approach in avoiding unacceptable risk always consists of the following steps for a system under development:

1. Specification of the operational context (as part of domain modelling)
2. Identification of hazards
3. Specification of the system's functional behaviour excluding hazards
4. Analysis of possible defects and failures in the system and its subsystems leading to hazards
5. Measures to reduce the unacceptable risk in the system and its subsystems

If we follow such a systematic approach, we work with the following views:

1. Context behaviour as specified
2. System behaviour as specified
3. System behaviour as realized with defects both of systematic or probabilistic nature
4. Context assumptions and defects due to violations of the assumptions about the operational context

We consider the following classification of hazards (and related potential incidents) and their relation to specifications:

	Specification	Realization
Context	Hazard not identified and recognized in context specification	Violation of specification of operational context
System	Hazard not excluded by system specification	Violation of specification of system behaviour

All together we get the following classifications of reasons for incidents due to hazards:

Classification of incident	Cause of hazard
Hazard not identified in context specification	Errors in the analysis of the set of hazards and potential incidents; hazards that were not recognized in the elicitation of safety requirements
Hazard not excluded by system specification	Errors in the specification, either of the system or of the operational context, since the composition of the ideal system behaviour and the ideal operational context behaviour still allow for hazards
Violation of specification of system behaviour	Hazards and risk of incidents due to systematic or probabilistic failures in the system and its subsystems
Violation of specification of operational context	Hazards and risk of incidents due to violations of the idealistic assumptions about the context

A result of safety analysis should be probabilistically formulated bounds for the risk of hazards and incidents – bounds sufficient for the given safety requirements. If hazards and incidents happen during the operation of systems, we have to distinguish between: hazards and incidents, that are a result of remaining risks, just as analysed in the safety process; and hazards and risk of incidents, that have to be seen as a result of faults in the functional safety analysis.

Modelling techniques can help to analyse, in a systematic manner, functional safety issues. However, as we will show, we need a careful modelling of the system, its possible defects, the operational context, the assumptions about the operational context, and possible violations of assumptions about the operational context. The better such an approach is, the more reliable the safety analysis is.

What we demand and describe is along the lines of ISO 26262 which emphasizes:

> NOTE 2 There is a difference between
>
> to perform a function as required (stronger definition, use-oriented) and
>
> to perform a function as specified, so a failure can result from an incorrect specification.

This citation taken from ISO 26262 underlines the fact that a safety analysis falls short if it only shows the risk of hazards due to violations of the behaviour in terms of a function as specified; in contrast, a safety analysis also has to guarantee the absence of hazards in the empirical general sense as defined above. Note that there have been a number of serious incidents, for instance in air traffic, where systems reacted as specified, but the specifications were not adequate for functional safety since they did not match with the expectations of the pilots.

2 Modelling and Structuring Systems

In the following we introduce a short overview of system modelling techniques and architectural views. Fig. 1 gives a schematic illustration of a system and its operational context.

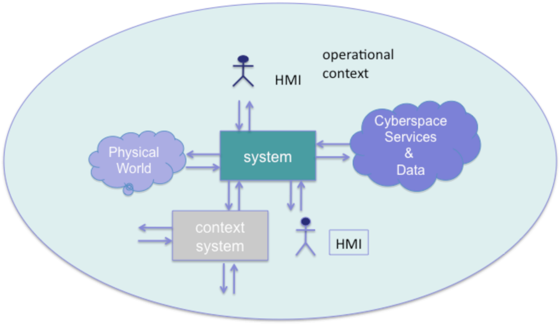


Figure 1: System and its Context

We use basically two frameworks for structured views onto systems

- modelling theory
- structuring of systems into adequate levels of abstraction

A key starting point is the fundamental concept of a system. The modelling and architectural framework has two parts:

- a family of mathematical and logical system modelling concepts for systems, addressing the notion of interface, state and architecture with two models of behaviour:
  - o logical model: a system described in terms of interface, architecture and state – we distinguish between the interface view (black box view) and a glass box view
  - o probabilistic model: a system described in terms of probabilities for its behaviours – more precisely probability distributions on sets of possible behaviours.
- a structured set of views – sometimes called comprehensive system architecture; it comprises the following views:
  - o context
  - o functional view (structured system interface behaviour):
    - hierarchy of system functions with modes of operation to capture their dependencies and their context
    - probability distribution on behaviours,
  - o subsystem architecture view: hierarchical structure of subsystems (in terms of "logical components"),
  - o technical and physical view: electronic hardware, software at design and runtime, mechanical and physical hardware, and their connections.

The two modelling frameworks are related and described in the following. We start by briefly introducing the modelling theory; for details see [Broy 12].

2.1 The System Modelling Theory

Our approach uses a specific notion of discrete system with the following characteristics and principles:

- A discrete system has a well-defined boundary that determines its interface.
- Everything outside the system boundary is called the system's environment. Those parts of the environment that are relevant for the system's operation are called the system's operational context.
- A system's interface describes the means by which the system interacts with its context. The syntactic interface defines the set of actions that can be performed in interaction with a system over its boundary. In our case syntactic interfaces are defined by the set of input and output channels together with their types. The input channels define the input actions for a system while the output channels define the output actions for a system.
- We distinguish between syntactic interface, also called static interface, which describes the set of input and output actions that can take place over the system boundary and interface behaviour (also called dynamical interface), which describes the system's functionality; the interface behaviour is captured by the causal relationship between streams of actions captured in the input and output histories. We give a logical behaviour as well as a probabilistic behaviour for systems.
- The interface behaviour of systems is described by: logical expressions, called interface assertions; by state machines; or it can be further decomposed into architectures.
- A system has an internal structure. This structure is described by in a state view by its state space with state transitions and/or by its decomposition into subsystems forming its architecture in case the system can be decomposed correspondingly. The subsystems interact and also provide the interaction with the system's context. The state machine and the architecture associated with a system are called its state view and its structural or architectural view respectively.
- In a complementary view, the behaviours of systems can be described by sets of traces, which are sets of scenarios of input and output behaviour of systems. We distinguish between finite and infinite scenarios.
- Moreover, systems operate in time. In our case we use discrete time, which seems, in particular, adequate for discrete systems. Subsystems operate concurrently within architectures.

This gives a highly abstract and at the same time comprehensive model of systems. This model briefly is formalized in the following.

2.1.1 Data Models – Data Types

Data models define a set of data types and some basic functions for them. A *(data) type* T is a name for a data set. Let TYPE be the set of all data types.

2.1.2 Interface Behaviour

Systems have syntactic interfaces that are described by their sets of input and output channels attributed by the type of messages that are communicated over them. Channels are used to connect systems to be able to transmit messages between them. A set of typed channels is a set of channels with a type given for each of its channels.

Definition. Syntactic interface

Let I be a set of typed input channels and O be a set of typed output channels. The pair (I, O) characterizes the syntactic interface of a system. The syntactic interface is denoted by (I▸O). ❑

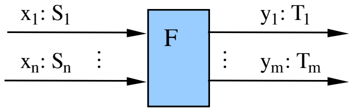


Figure 2: Graphical Representation of a System F as a Data Flow Node

Fig. 2 shows the syntactic interface of a system F in a graphical representation by a data flow node with its syntactic interface consisting of the input channels $x_1, \ldots, x_n$ of types $S_1, \ldots, S_n$ and the output channels $y_1, \ldots, y_m$ of types $T_1, \ldots, T_m$.

Definition. Timed Streams

Given a message set M of data elements of type T (M is also called the carrier set of type T), we represent a *timed stream* s of type T by a mapping

$$s: \mathbb{N}\backslash\{0\} \rightarrow M^*$$

In a timed stream s a sequence s(t) of messages is given for each time interval $t \in \mathbb{N}\backslash\{0\}$. In each time interval an arbitrary, but finite number of messages may be communicated. By $(M^*)^\infty$ we denote the set of timed infinite streams. ❑

A (timed) channel history for a set of typed channels C assigns to each channel $c \in C$ a timed stream of messages communicated over that channel.

Definition. Channel history

Let C be a set of typed channels; a (total) *channel history* x is a mapping (let $\mathbb{M}$ be the universe of all messages)

$$x : C \rightarrow (\mathbb{N}\backslash\{0\} \rightarrow \mathbb{M}^*)$$

such that x(c) is a timed stream of messages of the type of channel $c \in C$. $\vec{C}$ denotes the set of all total channel histories for the channel set C. ❑

For each history $z \in \vec{C}$ and each time $t \in \mathbb{N}$ the expression $z{\downarrow}t$ denotes the partial history (the initial communication behavior on the channels) of z until time t. $z{\downarrow}t$ yields a finite history for each of the channels in C represented by a mapping

$$C \rightarrow (\{1, \ldots, t\} \rightarrow \mathbb{M}^*)$$

$z{\downarrow}0$ denotes the history with empty sequences associated with each of its channels.

The behavior of a system with syntactic interface (I▸O) is defined by a mapping that maps the input histories in I onto output histories in $\dot{O}$. This way we get a functional model of a system interface behavior.

Definition. I/O-Behaviour

A causal mapping F: $\vec{I} \rightarrow \wp(\vec{O})$is called an *I/O-behaviour*. By $\mathbb{F}$[I▸O] we denote the set of all (total and partial) I/O-behaviours with syntactic interface (I▸O) and by $\mathbb{F}$ the set of all I/O-behaviours. ❑

Interface behaviours model system functionality. For systems we assume that their interface behaviour is total. Behaviours F may be deterministic (in this case, the set F(x) of output histories has at most one element for each input history x) or nondeterministic.

2.1.3 State Machines by State Transition Functions

State machines with input and output describe system implementations in terms of states and state transitions. A state machine is defined by a state space and a state transition function.

Definition. State Machine with Syntactic Interface (I▸O)

Given a state space Σ, a state machine (Δ, Λ) with input and output according to the syntactic interface (I▸O) consists of a set $\Lambda \subseteq \Sigma$ of initial states as well as of a nondeterministic state transition function

$$\Delta: (\Sigma \times (I \rightarrow \mathbb{M}^*)) \rightarrow \wp(\Sigma \times (O \rightarrow \mathbb{M}^*)) \;❑$$

For each state $\sigma \in \Sigma$ and each valuation a: $I \rightarrow \mathbb{M}^*$ of the input channels in I by sequences of input messages every pair $(\sigma', b) \in \Delta(\sigma, a)$ defines a successor state σ' and a valuation b: $O \rightarrow \mathbb{M}^*$ of the output channels consisting of the sequences produced by the state transition. (Δ, Λ) is a *Mealy machine* with possibly infinite state space. If in every transition the output b depends on the state σ only but never on the current input a, we speak of a *Moore machine*.

2.1.4 Systems and their Functionality

Systems interact with their contexts via the channels of their interfaces. We identify both systems by names. A system named k has an interface, consisting of a syntactic interface (I▸O) and interface behaviour

$$F_k: \vec{I} \rightarrow \wp(\vec{O})$$

The behaviour may be a combination of a larger number of more elementary sub-function behaviours. Then we speak of a *multifunctional* system.

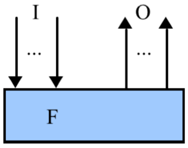


Figure 3 Graphical Representation of a Function Interface with the set of input channels I and the set of output channels O

Let SID be the set of system names. A system named k ∈ SID is called *statically interpreted* in a system model or in an architecture if only a syntactic interface (I_k▸O_k) is given for k and *dynamically interpreted* if an interface behaviour $F_k \in \mathbb{F}[I_k \blacktriangleright O_k]$ is specified for component k.

2.1.5 Architectures

In the following we assume that each system used in an architecture as a component has a unique identifier k. Let K be the set of identifiers for the components of an architecture.

Definition. Set of Composable Interfaces

A set of component names K with a finite set of interfaces $(I_k \blacktriangleright O_k)$ for each identifier $k \in K$ is called *composable*, if the following propositions hold:

- the sets of input channels Ik, $k \in K$, are pairwise disjoint,
- the sets of output channels Ok, $k \in K$, are pairwise disjoint,
- the channels in $\{c \in Ik: k \in K\} \cap \{c \in Ok: k \in K\}$ have consistent channel types in $\{c \in Ik: k \in K\}$ and $\{c \in Ok: k \in K\}$. ❑

If channel names and types are not consistent for a set of systems to be used as components we simply may rename the channels to make them consistent.

Definition. Syntactic Architecture

A syntactic architecture $A = (K, \xi)$ with interface $(I_A \blacktriangleright O_A)$ is given by a set K of component names with composable syntactic interfaces $\xi(k) = (I_k \blacktriangleright O_k)$ for $k \in K$.

$IA = \{c \in Ik: k \in K\} \backslash \{c \in Ok: k \in K\}$ denotes the set of input channels of the architecture,

$DA = \{c \in Ok: k \in K\}$ denotes the set of generated channels of the architecture,

$OA = DA \setminus \{c \in Ik: k \in K\}$ denotes the set of output channels of the architecture,

DA\OA denotes the set of internal channels of the architecture

$CA = \{c \in Ik: k \in K\} \cup \{c \in Ok: k \in K\}$ denotes the set of all channels

By $(I_A \blacktriangleright D_A)$ we denote the *syntactic internal interface* and by $(I_A \blacktriangleright O_A)$ we denote the *syntactic external inter*face of the architecture. ❑

A syntactic architecture forms a directed graph with its components as its nodes and its channels as directed arcs. The input channels in I_A are ingoing arcs and the output channels in O_A are outgoing arcs for that graph.

Definition. Interpreted Architecture

An interpreted architecture (K, ψ) for a syntactic architecture (K, ξ) associates an interface behavior $\psi(k) \in \mathbb{F}[I_k \blacktriangleright O_k]$ for the syntactic interface $\xi(k) = (I_k \blacktriangleright O_k)$, with every component $k \in K$. ❑

An architecture can be specified by a syntactic architecture given by its set of subsystems and their communication channels and an interface specification for each of its components.

2.1.6 Probabilistic Interface View

We provide a probabilistic model for systems along the lines of [Neubeck 12]. Given a set of typed channels C we define a probability distribution for a set $H \subseteq \vec{C}$ by the function

$$\mu: H \rightarrow [0:1]$$

Let $m[\vec{C}]$ denote the set of all probability distributions over sets $H \subseteq \vec{C}$.

Given a behaviour

$$F: I \rightarrow \wp(\vec{O})$$

its probabilistic behaviour is defined by a function

$$D_F: I \rightarrow m(\vec{O})$$

where for every input history $x \in I$ by

$$D_F(x)$$

we get a probability distribution for every input history $x \in I$

$$\mu_x: \wp(F(x)) \rightarrow [0:1]$$

We get a probability $\mu_x(Y)$ by the function μ for every measurable set $Y \subseteq F(x)$ of output histories. This shows that μ defines a probability distribution μ_x for every input history $x \in I$ on its set $F(x)$ of possible output histories.

2.2 Overall Structuring of Systems into Levels of abstraction

We choose a systematic structuring of systems and their contexts using the following categories.

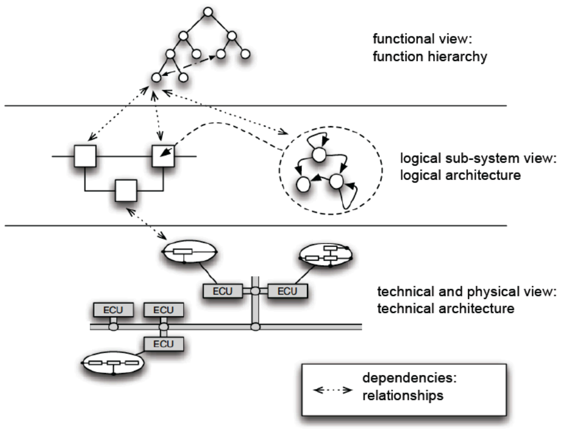


Figure 4 Levels of Abstraction Taken from [Broy et al. 08]

We structure the properties of systems into a number of views that are the result of viewpoints. We use three fundamental views:

- usage: function and context
- design: (logical) subsystem structure
- implementation: technical, physical, syntactical representation and realisation

Each view uses modelling concepts taken from a basic set of modelling elements

- interface and interface behaviour in terms of the interaction over the system boundaries
- architecture and architectural behaviour in terms of structuring a system into a set of subsystems

and their connection by communication channels and its interaction between the components and over the system boundaries

- state and state transition behaviour in terms of describing the state space of a system, its state transitions triggered by interaction.

For behaviour we distinguish

- logical behaviour in terms of the correct patterns of interaction
- probabilistic behaviour in terms of the probability of certain patterns of interaction.

These different aspects of behaviour apply to all three modelling concepts interface, state, and architecture.

3 Key Challenges for Functional System Safety

As we can see from the categorization of incidents, in hazard classification it is essential to analyse what can go wrong at the level of the specification and design, and what are the effects of failures of subsystems (as identified by FMEA). In particular, it is essential to make sure that, first of all, no potential hazards are overlooked in domain modelling and that the functional specification excludes all the hazards with sufficiently high probability. In particular, a very difficult task is to find out to what extent a particular system design may lead to failures in its operational context; this includes especially errors of humans operating the system.

There are quite a number of incidents, in particular in avionics and perhaps less spectacular and less well analysed also in the operation of other systems such as cars, boats and trains that are due to wrong reactions by their users, such as pilots. In these cases, the system functions were specified and implemented in such a way that users get confused and could not operate systems properly as expected in particular situations and as required by the identified user groups.

3.1 System Boundaries and Hazards

A hazard is due to certain critical events inside a system or in its operational context. Therefore we distinguish two categories of hazards for a system under safety analysis:

- Intrinsic hazards are hazards that result in incidents inside a system; as an example take a battery together with its control unit, which is a system that might explode or catch fire
- Extrinsic hazards are due to incidents that happen in the operational context of a system that are under the control of the system; for example, the explosion of a battery due to a fault in the control system is an extrinsic hazard from the perspective of the control system (where the battery is part of its operational context).

In safety analysis we have to capture and analyse and exclude both intrinsic and extrinsic hazards. Extrinsic hazards are related to the interface behaviour and the functionality of systems. Intrinsic hazards become extrinsic if we change the scope and focus the system under analysis such that the critical events are no longer part of the system. An example is the shift of the focus from a battery together with its control unit to the control unit with the battery as part its operational context. The change of scope is typically a result of design and system decomposition.

3.2 Domain Modelling

One particular important issue to find out about hazards is a very precise understanding, analysis, and modelling of the system's operational context. Typically incidents happen in the operational context. There are two difficulties that have to be mastered in domain modelling as basis and part of safety analysis.

3.2.1 Identifying and Modelling Hazards

First of all we have to understand what potential hazards are. So we have to carry out a careful analysis of the environment and operational context to find out about hazards. This is very much related to the task of requirements engineering.

The similarities between hazard analysis and identification and requirement analysis and identification are obvious. It is a difficult issue to find out about all the actual requirements. Forgetting a requirement leads to a system that does not fulfil the user expectations in some respects. In analogy, overlooking a possible hazard leads to a safety analysis in which no measures are undertaken to ensure that this overlooked hazard is not happening or that the probability of it happening is low enough. There are quite a number of practical examples where such a problem has happened (example: Titanic).

In both cases of requirements engineering and hazard analysis the completeness of a specification cannot be verified but has to be checked by validation. A careful validation of the result of the hazard analysis and identification is mandatory. [Gleirscher 11] discusses environment modelling for hazard analysis and for hazard-oriented derivation of scenarios for specification validation and system testing.

3.2.2 Relating Domain Specific Levels of Abstraction

A particular difficulty results from the different levels of abstraction for the formulation of safety requirements. Fig. 5 shows schematically four chunks of system properties from an example inspired by [Kondeva 12].

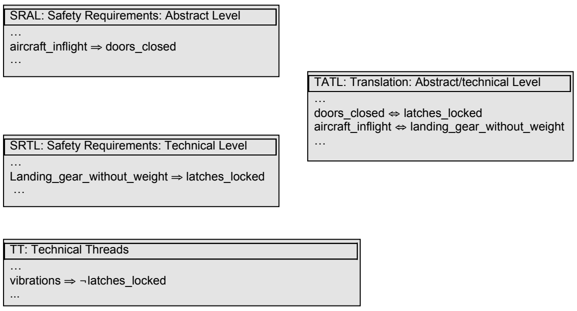

SRAL: Safety Requirements: Abstract Level
…
aircraft_inflight ⇒ doors_closed
…
TATL: Translation: Abstract/technical Level
…
doors_closed ⇔ latches_locked
aircraft_inflight ⇔ landing_gear_without_weight
…
SRTL: Safety Requirements: Technical Level
…
Landing_gear_without_weight ⇒ latches_locked
…
TT: Technical Threads
…
vibrations ⇒ ¬latches_locked
...

Figure 5 Safety Requirements: From Abstract to Technical Level and Threats at Technical Level

At the abstract level safety requirements are formulated in application domain oriented language addressing key concepts and notions of the application domain. At the technical level the same safety requirement is expressed in technical terms. This "translation" is not as simple as the one in the illustration. As stated above, it is an inference, based on the architectural structure and the behaviour of the subsystems in the structure. There is no chance to produce this manually (see [Struss, Fraracci 11], [Struss 11]). This has to be generated, and this is exactly what model-based prediction for the physical components has to deliver. We need a translation of the abstract safety requirements into the technical ones in terms of logical assertions that formalize this relationship. This relation is part of the domain model. We have to show

$$\text{SRTL} \wedge \text{TATL} \Rightarrow \text{SRAL}$$

(see the example in Fig. 5). Then, on the technical level, additional technical threats have to be and can be identified that are hard or even impossible to find at the abstract level. In the example in Fig. 5 we get some inconsistency and thus a contradiction to safety requirements in SRTL if we assume that there may vibrations while the aircraft is in flight that they in term might unlock the latches. Such inconsistencies can be checked and found by SAT solvers.

Of course, this change of levels of abstraction typically continues. At the technical level, there does not exist the signal "Landing_gear_without_weight". There exists: "no signal at the pin connected to the weight sensor". This technical view is essential in order to analyse the impact of a broken weight sensor, open wires and connectors between sensor and ECU, shorts of the wires, etc.

3.3 Modelling Context, HMI and Safety

Hazards can only be caused by a system in the interaction between the system and its operational context.

3.3.1 Hazards as Result of the Interaction between Systems and their Operational Context

Another issue is to understand how such hazards in the operational context are triggered by the system. This leads to the necessity to have a kind of a formalisation of the interaction between the operational context and the system.

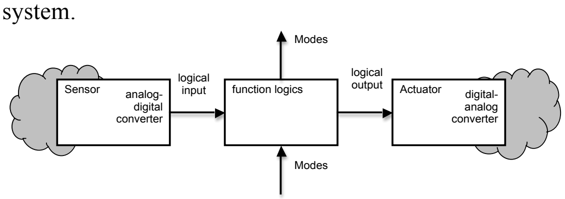


Figure. 6 Schematic split of a function in hybrid pre- and post-processing

More precisely we have to model the operational context including extrinsic hazards and incidents as they may appear in the operational context. Only if such modelling is done in a sufficiently formal way, we can start to get estimations of the bounds on the risk of incidents and hazards (see [Struss, Fraracci 11]).

3.3.2 Hazards as Result of the Interaction between Systems and Their Users

Clearly, such a modelling can be very difficult because the operational context, in particular, has to deal with the user interface and the way users operate a system. In principle, we have to look at issues such as: "what is the probability that a user presses a wrong button in a particular situation?" and to find out what is the psychological analysis that is needed to speak about those probabilities.

This shows that in the analysis of functional safety the logical and the technical user interface have to be looked at and analysed very carefully.

Second we have to deal with issues in approaches like use case analysis. One way to do this would be to identify intrinsic and extrinsic hazards and then to develop a number of anti-use-cases that describe scenarios of hazards happening and to analyse what are the possibilities to avoid these hazards. All this activity can be and must be done quite independently of the question of the necessary and additional FMEA to make sure that the system as specified with an operational context as modelled does the right thing. Today, in practice, functional safety analysis is often too much focused onto FMEA and vulnerability impact analysis with the danger to miss hazards that are not due to defects of subsystems.

3.4 Functional Modularity and Extrinsic Hazards

Note that strictly speaking, in terms of extrinsic hazards, it is not the system that is safety critical but its functions. In fact, in a safety analysis we have to identify the safety criticality of the functions. This goes hand in hand with modelling the operational context; we can see how the functions are connected to the operational context and which of the functions may cause hazards. In addition, we have to consider a number of failure assumptions for the functions that have to be related to FMEA and then find out which are the functions and the output provided by those functions as safety critical aspects. Then we can analyse which error deviations of functions we can tolerate and which error deviations we cannot tolerate and where we have to be sure that they can happen only with a certain sufficiently low probability.

Today we typically deal with so-called multi-functional systems. These are systems that introduce and offer a large number of different functions as pointed out in [Broy 10]. These functions have to be specified in a modular way, in spite of the fact that they are usually not logically independent. There are behavioural dependences between these functions. When mastering the specification of systems from a safety point of view, we have to deal with the different functions that are part of the functionalities.

As shown in [Broy 10] it is possible to identify and specify dependences between functions. If there is a function F that depends on another function F' and if the dependency of these functions may lead to hazards then the function F' (which, considered in isolation, is not safety critical) has to be treated as a safety critical function, if the dependency may lead to extrinsic hazards. More precisely, using these dependencies we can introduce a directed dependency graph with functions as nodes. Then we identify the functions that are safety critical. This way we to stick to a kind of propagation and inheritance of safety criticality levels such that a highly safety critical function F may pass on its safety level to functions F' that show dependencies to F.

3.5 Tracing and Safety#

Finally, in standards for functional safety, tracing is required for safety critical functions. Unfortunately, what we see as a foundation of tracing in the scientific literature so far is not sufficient. Based on the proposed modelling framework, a very rigorous approach to tracing is possible by representing all the properties of systems within a formalised logical framework. This way we can introduce a completely formalised concept of tracing.

In doing so we get a precise concept of what tracing is. In particular, we can study traces between general requirements, functional specification and architectural decomposition. Such an approach provides a firm framework for defining what traces are but at the same time it addresses the question of how dense traces are and how many traces we need. By the approach we see how difficult and complex tracing is. Here we need more research and also empirical studies.

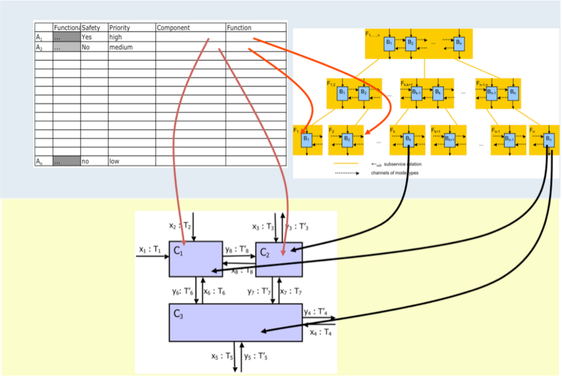


Figure 7 Tracing between Requirements, Functional Hierarchy, and Logical Subsystem Architecture

Recently, we have performed a number of empirical studies about dependencies between functions in trucks to find out about how many dependencies we can expect between those functions. Similar numbers are not available for dependencies between requirements, functional specification, and architecture.

4 Summary and Outlook

We have introduced and sketched a rigorous framework of modelling that allows us to capture logical and probabilistic properties at different levels of abstraction. We believe that such a rigorous framework allows for modelling that can be used both for system specification, design and implementation, for verification including test case generation, for safety analysis as well as for diagnoses.

Using a rigorous modelling approach we model the system as well as its operational context. We recommend to distinguish and to model intrinsic as well as extrinsic hazards. We, in particular, recommend validating the specification carefully to make sure that hazards are not implied by it. Doing so, we can apply all kinds of automatic analysis and verification techniques to deal with functional safety. In any case, the quality of functional safety analysis depends on the expressive power and the adequate application of the modelling techniques and methods.

Acknowledgements

It is a pleasure to acknowledge helpful and stimulating discussions with Mario Gleirscher, Antoaneta Kondeva, and Peter Struss.

5 References

M. Broy: The ‚Grand Challenge' in Informatics: Engineering Software-Intensive Systems. IEEE Computer, Oktober 2006, 72–80

M. Broy, I. Krüger, M. Meisinger: A Formal Model of Services. TOSEM - ACM Trans. Softw. Eng. Methodol. 16:1 Feb. 2007

M. Broy: Model-driven architecture-centric engineering of (embedded) software intensive systems: modelling theories and architectural milestones. Innovations Syst. Softw. Eng. 3:1, 2007, 75-102

M. Broy: Multifunctional Software Systems: Structured Modeling and Specification of Functional Requirements. Science of Computer Programming 75 (2010), S. 1193–1214

M. Broy: Software and System Modeling: Structured Multi-view Modeling, Specification, Design and Implementation. In: Conquering Complexity, edited by Mike Hinchey and Lorcan Coyle, Springer Verlag, January 2012, S. 309-372

M. Broy, M. Feilkas, J. Grünbauer, A. Gruler, A. Harhurin, J. Hartmann, B. Penzenstadler, B. Schätz, D. Wild: Umfassendes Architekturmodell für das Engineering eingebetteter Software-intensiver Systeme. Technische Universität München, Institut für Informatik 2008, TUM-I0816 (Technical Report)

M. Gleirscher: Hazard-based Selection of Test Cases. In: Proc. 6th ICSE Workshop on Automation of Software Test (AST'11), May 2011

M. Gleirscher: Behavioural Safety of Software-controlled Physical Systems. Ph.d. Thesis forthcoming 2012

A. Kondeva: Safety-based Requirements Engineering: Systematic refinement and specification of safety requirements in the avionic domain. Ph. D. Thesis forthcoming 2012

P. R. Neubeck: A Probabilitistic Theory of Interactive Systems. Ph. D. Thesis forthcoming 2012

P. Struss, A. Fraracci: FMEA of a Braking System - A Kingdom for a Qualitative Valve Model. In: 25th International Workshop on Qualitative Reasoning, Barcelona, Spain, 2011

P. Struss: Automated Failure-modes-and-effects Analysis of Embedded Software (Extended Abstract). In: 2nd International Workshop on Software Health Management, SHM-2011/4th IEEE International Conference on Space Mission Challenges for Information Technology (SMC-IT), Palo Alto, 2011

Safety Protocols: a New Safety Engineering Paradigm

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

The field of *system safety* looks on the surface to be a mature discipline based on everyday intuitions about safety risk. System safety looks at potential accidents that could arise due to system behaviour. It is based on the notion of *system hazard.* In this paper, we look at the theory and practice of system safety. We propose a model of system safety behaviour suitable for describing and evauating the goals and processes of safety engineering. We argue that the notion of hazard is not appropriate as the central pillar of safety engineering and that it can actually be misleading. Instead, we propose that safety engineering is better served by a focus on *safety constraints.* To illustrate the benefits, we consider an approach to "hazard analysis" that begins by simply identifying all the dangerous physical flows in the systems intended environment and proposing a *safety policy* for managing them. Safety engineering then proceeds with the elucidation of *safety protocols* that coordinate the various systems in the environment in operating safely within the proposed policy constraints. We illustrate our approach using a case study.

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

1 Introduction

The field of *system safety* — as described, for example, in Leveson's well-known textbook entitled "Safeware" (Leveson 1995) and by safety standards such as MIL-STD 882C (Department of Defense 1993) — involves an approach to safety engineering that is very familiar (especially in the USA) and apparently well-understood. It looks primarily at potential accidents that could arise due to system behaviour and its most basic tool is the pervasive and widely-used concept of "hazard". System safety purports to be the technical expression of everyday intuitions about safety: co-opting every-day terms (such as "hazard") and imbuing them with elaborate technical interpretations.

Although the field of system safety looks on the surface to be a mature discipline, experience with a number of Defence projects suggests that Safety Programs are deficient in their safety arguments with uncomfortable frequency. Many safety programs strive to follow the processes required by (say) MIL-STD 882C (Department of Defense 1993): a great deal of analysis is done and diligently reported. Unfortunately, on close examination, the safety arguments can sometimes boil down to little more than "a great deal of analysis was done and diligently reported." As observed by the Nimrod Review, "... the task of drawing up the Safety Case became essentially a paperwork and tick-box exercise." It seems that the process of system safety can be easily abused. What is the reason for this?

One basic reason is that the field of system safety still lacks clear agreement on basic terminology. The well-known text by Leveson (Leveson 1995) does (for the most part successfully) attempt to provide clear definitions — but these are not introduced until Chapter 9. As forums such as the High Integrity Mailing List (Kelly 2011) demonstrate, the definitions of fundamental concepts in system safety are still the subject of much debate.

A more serious issue is discussed in this paper. We argue that the notion of hazard is not a useful one and that it can actually be misleading. In place of the unending hunt for the hazard, we propose that safety engineering should be carried out through the positive proposal of safety policies for dealing with dangerous physical flows and of safety protocols that coordinate interactions between systems so as to implement said safety policies.

This paper is structured as follows. In Section 2 we look at the terminology of system safety. In Section 3 we discuss a range of issues relating to hazards and hazard analysis. Section 4 gives a brief overview of Leveson's more recent approach to accident modelling and hazard analysis. In Section 5 we describe briefly the approach taken by the recently published DEF(AUST)5679 (Department of Defence 2008*b*). Then in Sections 6–8 we describe the notion of safety protocols. We illustrate our arguments in Sections 10–12 using a case study. Finally, Section 13 presents some concluding remarks.

2 System Safety

The primary driving concept in system safety is that of the *accident.* For a given system, the first key step in safety engineering is to consider the possible accidents to which system behaviour could contribute. As defined by Leveson (Leveson 1995):

> **Definition**. An *accident* is an undesired and unplanned (but not necessarily unexpected) event that results in (at least) a specified level of loss.

Succinctly put, an accident is an undesired loss event. One may debate whether or not loss of equipment or capability — as opposed to harm or loss of life — should be included in system safety engineering, but this is a minor consideration. We do not believe that the notion of accident is controversial: it has a meaning in everyday life but also makes sense as a technical concept.

Also familiar is the notion of *accident severity*. This characterises the damage that may be done by the loss event and is often used in safety engineering to rate accidents, possibly with a view to allocating more effort into protecting against the more severe possibilities.

We now turn to the notion of *hazard*, which is more problematic. In everyday life the notion of hazard is commonly used and seems to be well understood. We are familiar with the following examples:

1. "Smoking in the toilets is a fire hazard and smoke detectors have been fitted" (aircraft safety announcement);
2. "Confined space. Hazardous Atmosphere. Check oxygen level before and during entry" (warning sign);
3. "Ice on road. Hazardous driving conditions." (road sign); and
4. "Tripping Hazard", "Biological Hazard", "Electrical Hazard", "Overhead Hazard" (other warning signs).

Intuitively speaking, a hazard is a situation from which it is sufficiently likely that an accident could arise. To take the first example, it is easy to imagine that a cigarette butt that is carelessly disposed of in the wastepaper bin in an aircraft toilet could quickly lead to a fire that would threaten the safety of the aircraft. The announcement both warns of this hazard and also states that this hazard will be quickly detected (and thus dealt with by the cabin staffor automatic systems).

Thus in common parlance the notion of hazard is usually associated with some dangerous physical substance or release of energy (*dangerous flow*). In developing the field of system safety, it has been thought essential to retain the hazard as a central concept. However, this common notion of hazard has generally not been thought to be sufficiently powerful. Systems may be involved in accidents in many ways, even if they do exhibit such dangerous flows. Just consider an air traffic control system: the physical dangers intrinsic to the actual system equipment pale in comparison to its potential to do harm in the wider air traffic environment. Thus, considerable effort has been made to expand the definition of hazard to encompass all forms of dangerous interaction with the environment.

As defined by Leveson (Leveson 1995):

> **Definition**. A *hazard* is a state or set of conditions of a system (or object) that, together with other conditions in the environment of the system (or object), will inevitably lead to an accident (loss event).

As Leveson points out, implicit in this definition is that hazards must be determined with respect to the particular environment of the system. Hazards occur at or within the system boundary, which must be well defined, and may interact with other systems in the environment, which remain vague, in causing an accident. Leveson's definition is similar to most definitions of hazard, of which we quote just two:

> **Definition**. A *hazard* is a physical situation or state of a system, often following from some initiating event, that may lead to an accident (Ministry of Defence 2007)
>
> **Definition**. *System Hazards* are top-level states or events from which an accident, arising from a further chain of events external to the System, could plausibly result (Department of Defence 2008*b*).

This expanded notion of hazard takes a central place in modern system safety practice (as, for example, described in Leveson's book). Much of the effort applied in a safety program is devoted to the identification and assessment of hazards. This effort is called *hazard analysis* and involves techniques such as *Fault-Tree Analysis*(FTA) or *Failure Modes and Effects Analysis* (FMEA). On the one hand, FTA analyses the causes of hazards by reasoning backwards from a given top-level state (or event), using Boolean logic to describe how low-level events (which can be normal events or failure events) combine to bring about the hazard. On the other hand, FMEA reasons forward from low-level failures to determine how they may lead to system hazards.

Leveson (Leveson 1995) defines failure (a concept familiar in reliability) as follows:

> **Definition**. *Failure* is the non-performance or inability of the system or component to perform its intended function for a specified time under specified environmental conditions.

In fact, most of the techniques used in hazard analysis are borrowed and adapted from reliability engineering and depend on the concept of failures at least as strongly as on the concept of hazards.

Also striking is the degree of low-level information required by existing hazard analysis techniques. The design of the system must be quite well progressed for such techniques to be truly effective.

The system safety effort is usually directed through some form of probabilistic risk assessment in which a "hazard risk index" (HRI) is determined by a combination of hazard frequency and accident severity. If an HRI is too high, the risk may be regarded as unacceptable, or acceptable only with further measures designed to build in safety. Although this notion of hazard frequency is a natural one for simple physical hazards, it is harder to understand for the generalised notion of hazard adopted in system safety. It seems generally acknowledged that it is not sensible for hazards related to software behaviour and the Joint System Safety Handbook (Department of Defense 2010), for example, recommends that the notion of Software Control Category be used instead.

3 The hazards of "hazard"

In this paper we pose the specific question: *is this generalised notion of hazard suitable as the central pillar of modern safety engineering?* It is our contention that it is *not*, and that we need something better to guide our thinking.

The most important deficiencies are the following.

What we want vs what we have. Hazard analysis leads us to confound two quite different issues: what the system (imagined but not yet built) is required to do versus what the system actually does (as built). On the one hand, hazard analysis aims to determine system safety requirements, acknowledging that safety must be "designed" in to the system. On the other hand, hazard analysis uses techniques that, to be effective, require deep knowledge of how the system actually works.

Failures are not the whole story. The fixation that can be seen on "failure" of system components or items of equipment as potential root causes of hazards clouds the distinction between reliability and safety and leads to an emphasis on what Leveson (Leveson 2011) calls *component* *failure accidents*. However, an accident can also arise from "dysfunctional interactions between components" even if the components are working reliably. Such accidents are called *system accidents* or *component interaction accidents*.

Failures distort the story. Safety engineering is unusual in that prime focus is given to what can go wrong: that is, what is *not* required of the system. Usually, engineering concentrates on what *is* required of the system. Often "what can go wrong" is a much bigger and more imaginative world than "what we want." Failures thinking can lead to consideration of behaviours that are conceivable but disallowed by existing design constraints; it makes it hard to decide how much hazard analysis is enough; and it can even lead to extended consideration of failures that have no safety impact. Such an approach is a contributing factor to "laborious, discursive, document-heavy" safety cases — a key deficiency identified by the Nimrod Review (Haddon-Cave 2009).

Hazard analysis tends to be inward looking. The failures-focus of hazard analysis techniques makes them inward looking, concentrating on how problems within the system may lead to accidents in the environment. This makes it hard to describe safety functionality in terms of the system interface; contributing to the notorious non-compositionality of safety cases. Compositionality fundamentally relies on "plug and play" interface specifications.

Hazards are more complex than they seem. Superficially, the notion of hazard seems simple enough, but a little thought about its practicalities shows that it is hiding a great deal of complexity. Exactly what systems states may contribute to accidents is context sensitive and interacts with the notion of causality in subtle ways. Obviously, dangerous flows from the system are hazards. Equally, any system flow that may directly cause a dangerous flow in the environment is a hazard. Moreover, any system flow that directly causes an event in the environment that directly causes a dangerous flow is a hazard and so on and so forth. This kind of reasoning may be iterated to arbitrary numbers of intermediate events. In algorithmic terms the definition of hazard is quite complex indeed.

When do we stop? The iterative nature of the definition of hazard can potentially lead to the elaboration of very complex accident scenarios that can be hard to understand and may be considered "unlikely" on no better grounds than the number of intermediate events required. It also offers no guidance when to stop, making it very hard to be confident that all hazards have been enumerated.

How do we find hazards anyway? The determination of hazards involves a search for chains of events in the environment that may result in accidents, but says little about how to structure this search. In practice, the search becomes something of an imaginative process, usually structured only by guide words suggestive of possible ways in which things could go wrong. Again this can make it hard to be confident that all hazards have been enumerated.

Logical hazards complicate things. For certain kinds of systems, such as command-support systems, air traffic control systems etc, more subtle "logical" hazards relating to information flow tend to dominate. Logical hazards are difficult to understand or even to define, as there may be many levels of causal indirection between the "hazard" and the dangerous physical or material flows that it may eventually trigger. Such non-physical hazards may be problematic for existing hazard analysis techniques because they will not always be amenable to guide word analysis.

Software is not stochastic. Since we are ultimately concerned with assessing safety risk, there may be a tendency to assign probabilities to hazards (or to lower-level states or events). Such probabilities are dubious in the case of software-intensive systems. As Leveson (Leveson 1995) points out:

> Risk assessment is currently firmly rooted in the probabilistic analysis of failure events. Attempts to extend current probabilistic risk assessment techniques to software and other new technology, to management, and to cognitively complex human control activities have been disappointing.

Experience with a number of Defence projects suggests that the role of software in system safety is not always treated with sufficient care.

What do fault trees mean? The elucidation of lower-level hazards via techniques such as FTA is "handraulic" and not amenable to tool support. Attempts to provide a formal semantics for fault trees have met with limited success (Schellhorn et al. 2002).

For the above reasons, although we acknowledge that the concept of hazard is widely used in system safety and that many are comfortable with it, we believe that the notion of hazard is neither a useful nor helpful concept when we are looking for fundamental notions in system safety. At best we can say that hazards are only a means to an end, and play a role only as an auxiliary concept used to facilitate thinking in safety engineering.[1]

4 The STAMP Approach

As observed above, some of the issues and problems with the conventional approach to system safety — such as the heavy emphasis on failures — have already been pointed out by Leveson (Leveson 1995). Leveson has written a soon to be published new textbook, currently available on her web site in draft form (Leveson 2011). In this book, Leveson has proposed a new model for accidents and a new way of thinking about system safety. Her accident model is based on systems theory and is called *System Theory Accident Modelling and Processes* (STAMP).

In STAMP, accidents occur when external disturbances, component failures, and/or dysfunctional interactions among system components are not adequately controlled. Safety is viewed as a control problem for an adaptive socio-technical system. In such a framework, understanding why an accident occurred requires determining why the control structure was ineffective.

In systems theory, control is always associated with the imposition of *constraints*, which play a vital

[1] It is interesting to note that the *OHS Act (Com) 1991* does not make fundamental use of the concept at all (except for the use of the term "high-hazard" facilities).

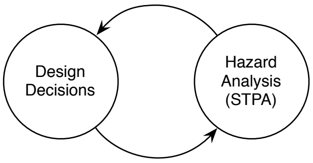


Figure 1: STPA-based design (Leveson 2011).

role in the STAMP approach. Accidents are considered to result from inadequate enforcement of constraints on behaviour at each level (e.g., technical, managerial, regulatory) of a socio-technical system. An example of a (physical) safety constraint is: "the power must never be on when the access door is open". Preventing accidents now and in the future requires a control structure that will enforce the necessary constraints.

Leveson describes a new approach to hazard analysis called STPA — which originally stood for STAMP-Based Hazard Analysis but has now been changed to System Theoretic Process Analysis. STPA is meant to extend conventional hazard analysis to cover new factors such as design error, software flaws, component interaction accidents and social and human processes. There is still the notion of system hazard and component hazard. System safety requirements and design constraints are also important concepts. However, the role actually played by hazards within the STPA approach is not very clear.

What is clear is that Leveson envisages STPA reaching deep into the system design process in a tightly coupled feedback loop as shown in Figure 1. This is one of the most puzzling aspects of STAMP, given Leveson's stated concerns (Leveson 2011, Ch. 6) over the tendency to isolate, misdirect, and delay safety efforts; what might be termed the too-much/too-late approach to safety. If essentially open-ended hazard-analysis efforts are required after every design decision — because hazard analysis eventually requires total knowledge of design detail — then it is little wonder that the tendency is to postpone safety efforts until the very end.

In later chapters, Leveson discusses the formulation of safety constraints using, as a worked example, the collision avoidance system TCAS II for aircraft (Leveson 2011). Leveson provides detailed specifications for the constraints (safety-related or not) and assumptions and limitations of the full socio-technical system in which TCAS II operates.

We are not going to discuss the STAMP and STPA approaches further. However, it is important to take away the following lessons: that Leveson thinks it desirable that the notion of hazard be de-emphasised and the notion of requirement or constraint be given a more prominent role.

5 DEF(AUST)5679

The "normal" approach to system safety — along with the heavy reliance on the notion of hazard — is reflected in most existing safety standards. DEF(AUST)5679 (now at Issue 2 (Department of Defence 2008*b*)) was written (at least in part) in an attempt to provide an approach to system safety driven more by safety requirements.

DEF(AUST)5679 provides requirements for the structure of the *safety case* (an evidence-based argument for safety). Safety case development is structured into three phases with associated reports (see Figure 2):

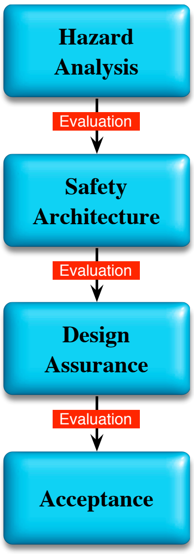


Figure 2: DEF(AUST)5679 Safety Case Development phases (Department of Defence 2008*b*).

Hazard Analysis – assess the danger (or threat to safety) that is potentially presented by the system;

Safety Architecture – demonstrate that the overall system is designed to be safe; and

Design Assurance – demonstrate that the components are designed to be safe.

In the hazard analysis phase, the system *interface* is defined in terms of inflows and outflows exchanged with the environment. Other systems present in the environment (the *operational context*) are described in appropriate detail. A hazard analysis then determines the ways in which the system, in its operational context, may contribute to an accident. The outputs of hazard analysis are as follows.

Accidents – external events that could directly result in death or injury.

Severities – a measure of the degree of seriousness of accidents in terms of the extent of injury or death that may result.

Hazards – states or events at the system interface from which an accident — arising from a further chain of events external to the system — could conceivably result.

Accident scenarios – a causally related mixture of system behaviours (*hazards*) and environment behaviours (*coeffectors*) that may culminate in an accident.

Thus, DEF(AUST)5679 adopts a fairly traditional notion of hazard with the conceptualisation of accident sequences taking a mandatory role in the hazard analysis phase. However, hazards turn out to play a limited role compared with much of current practice. Inward looking hazard analysis plays no role; hazard analysis is outward looking and is merely used to assist in determining what constitutes safety for the given system. Once this is achieved, the notion of hazard is no longer used.

The safety architecture phase begins with the development of a collection of *system safety requirements*. The system safety requirements are expressed

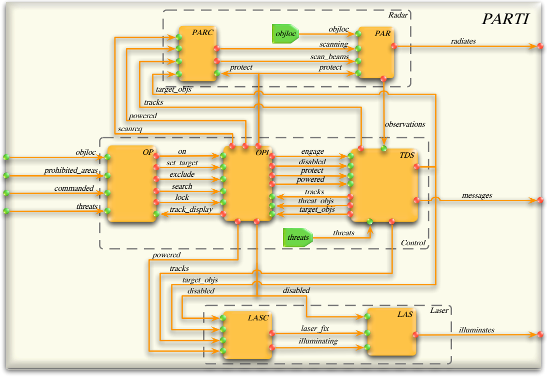


Figure 3: A safety architecture presented as a block diagram.

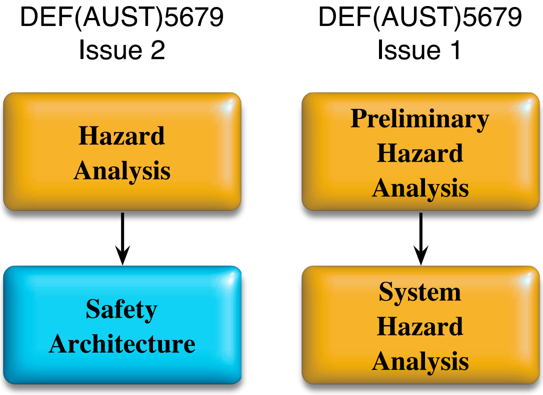


Figure 4: The hazard analysis phases of DEF(AUST)-5679-Issue 1 compared to Issue 2.

in terms of the system interface and collectively ensure that the system hazards do *not* occur. During subsequent system development, they are treated much like other system requirements.

So as to clarify the basic safety functionality of the system, the Safety Architecture decomposes the system into *components* (Figure 3 shows an example safety architecture (Mahony & Cant 2008)). The interaction between these components is described and they are assigned *component safety requirements* in order to discharge the system safety requirements. Finally, a *correctness* argument is made that shows how these component safety requirements ensure satisfaction of the system safety requirements (this is called *architecture verification*).

In the last phase of *design assurance*, the components are modelled to an appropriate level of detail and shown to satisfy their component safety requirements both through verification arguments on the models and through extensive testing.

In DEF(AUST)5679-Issue 2, *hazard analysis activities have no mandated role in either safety architecture or design assurance*. Instead, the safety requirements identified by hazard analysis are simply flowed down through subsequent phases. This is in sharp contrast to the approach adopted in Issue 1 of DEF(AUST)5679 (see Figure 4). Issue 1 mandated two hazard analysis phases: a Preliminary Hazard Analysis which looked from the system boundary out and a System Hazard Analysis which looked down into the system for dysfunctional interactions between components and for failures of individual components. Replacing System Hazard Analysis with Safety Architecture has the potential to make Issue 2 safety cases shorter, easier to understand and more convincing.

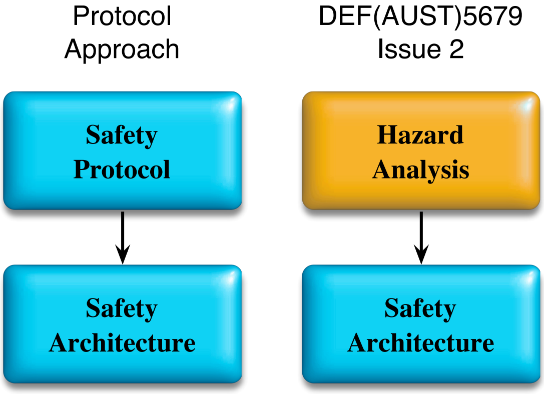


Figure 5: The protocol model compared to DEF-(AUST)5679-Issue 2.

The authors are currently carrying out a systematic application of DEF(AUST)5679-Issue 2 to the construction of the safety case for a real Defence system. In the course of this work, the essentially arbitrary nature of hazard analysis has become increasingly clear. This led to consideration of the potential feasibility of adopting DEF(AUST)5679's requirements flow-down model even earlier in the development cycle than safety architecture, replacing Hazard Analysis with Safety Protocol development as shown in Figure 5.

The goals of the Safety Protocol phase are essentially the same as those of Hazard Analysis, that is to identify potential accidents that may arise from the system operating in its intended environment and to propose system safety requirements that the system needs to satisfy to avert these accidents. However, instead of focussing on hazardous behaviours to be avoided, Safety Protocol development focuses on identifying safe behaviours to be adhered to.

In the following sections we briefly describe a model of system safety behaviour and then use it to describe the processes and outputs of Safety Protocol development.

6 A Simple Model of System Safety

We begin with a brief consideration of the setting in which safety engineering proceeds, describing a simple, generic model of system operation that is analogous to the system architecture model underlying DEF(AUST)5679 safety architecture. This model is used to structure the development of system safety requirements in much the same way as the architectural model structures the development of component safety requirements.

Suppose that we wish to engineer and operate a system S safely within a wider environment E. In general, the elements of E that will bear on safety include the following:

- the new system S;
- a collection of other systems (engineered elements) $\{S_1, \ldots, S_n\}$;
- a collection of humans (more generally protected elements) $\{H_1, \ldots, H_m\}$; and
- a physical *medium* (for example, the ocean or the atmosphere) M, in which these entities interact.

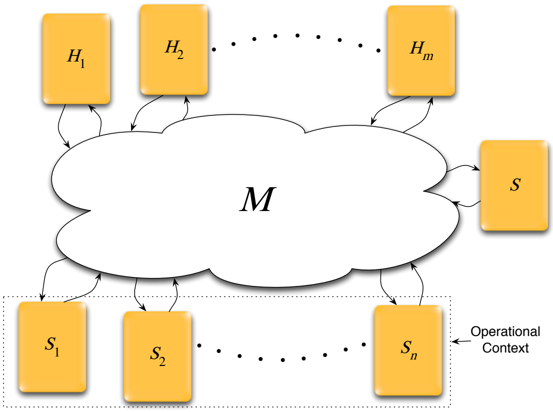


Figure 6: The structure of the environment.

Each element of E will have associated observable *inflows* and *outflows*: whether qualitative or quantitative; physical or logical; state or event based. We call these associated flows the *interface* to the element. This is a familiar concept for the engineered systems in the environment, but it is also readily applicable to the human elements and even the medium. The overall situation may be depicted in block diagram form as shown in Figure 6

Which humans should be included in the environment? Generally, the humans will comprise an undetermined number of potential bystanders. It may be convenient to aggregate such bystanders into a single representative block. However, some humans may have well defined roles for observing and/or controlling various systems in the environment. Such roles may be represented in the environment as specialised human blocks or separated out into system blocks that are implemented using human operators; depending on the nature of the safety functionality inherent to the role.

When considering the safety of human elements of E, the primary outflow of interest is the health status of the individual. The inflows of interest comprise the impact of *potentially* harmful energy on the individual. When these dangerous flows rise to *actually* harmful levels an accident can be said to occur, so we call the corresponding inflows accident flows and will generally represent them as event flows where the events represent the occurrence of accidents.

Which systems should be included in the environment? At a minimum, they should include all systems that S is intended to interact with, even indirectly, as all such interactions should be open to analysis for safety implications. In DEF(AUST)5679 parlance, this collection of systems is called the *operational context* of S. Ideally, the operational context would comprise some form of platform or system-of-systems which has an existing well-defined safety case and to which S is to be integrated.

When considering the safety of systems acting in E, the outflows of interest are those that may interact with the humans in the environment. The most obvious such flows are the dangerous flows that may emanate from the system, but control flows such as safety enclosures, interlocks, marked boundaries, warning signs, alarms, etc may be of significance.

If dangerous flows from one or more systems should be transmitted to a human element at a harmful level, then the corresponding accident event will be triggered. The precise mechanics of how dangerous flows from a number of systems are transmitted and aggregated through the environment is determined by the medium M. For example, there may be a number of radar sources in a given environment. The resultant radar intensity at each point in the environment is determined by the medium according to the power and direction of the signals from the source systems. An accident occurs when a human is positioned in the environment where the intensity of signal exceeds safe levels.

While the list of environmental flows that are known to be harmful is quite long: heat energy, kinetic energy, gravitational potential, poisons, explosive substances, etc; it is certainly finite and guidance can be found from many sources (Comcare 2007, Royal Australian Navy 2006, Department of Primary Industries 2007). Determining the dangerous flows for a given system is little more complex than running down a checklist. Once the dangerous flows have been identified, the “hazard analysis” part of our approach is over. Instead, we move our attention to the question of how safely to operate the various systems in the environment.

7 The Safety Policy

Since accidents are always associated with the presence of dangerous flows, achieving a safe environment is a matter of controlling the way in which humans interact with the dangerous flows in the environment. Typically, this is done by eliminating, containing or isolating the dangerous flows, thus protecting the humans from their harmful effects. We call the approach taken to controlling the dangerous flows in an environment the *safety policy*. It may be described by a collection of *safety constraints* (expressed in terms of the system and human interfaces) that, if adhered to within the given medium, will ensure a safe environment (no accidents).

Guidance on safety policy development can be found from many sources (Royal Australian Navy 2006, Department of Primary Industries 2007) . In general, a safety policy will take one of the following approaches to controlling the dangerous flows.

A dangerous flow may be eliminated or constrained below harmful levels. Many standards exist that offer guidance as to safe tolerances for exposure to potentially harmful substances and energies.

A dangerous flow may be isolated from the humans in the environment. This may involve active control, monitoring for human presence and directing dangerous flows away from them; or passive control, building barriers around the dangerous flow or removing it to a remote location.

Finally, the humans in the environment may be made resistant to the dangerous flow by restricting working hours, requiring the use of protective equipment, etc.

Policy development bears some similarity to hazard analysis. It requires an understanding of system interfaces and investigates the potential effects of dangerous flows in the environment. Both involve an outward search through the environment to find dangerous flows that may be influenced by the system S to cause harm. However, policy development is a more contained and positive activity than traditional hazard analysis. In particular, because it focuses on the direct interactions between humans and dangerous flows, it does not require nor promote the kind of analysis of complex causal chains of hazards and co-effectors that requires a deep understanding of the system and environment, both in nominal and failure modes.

Indeed, failure analysis cannot occupy its traditional central place in safety policy development as (quite deliberately) too little is known of the inner workings of the systems operating in the environment.

Only the interface flows of systems (and primarily the dangerous flows) are considered and the focus is on determining how these should be constrained to promote a safe environment. This is not to say that danger mitigation (reacting to policy breaches) and harm minimisation (reacting to accidents) should not feature in a safety policy, only that safety policy development can (and must) be addressed from the very earliest stages of system development.

As system development proceeds, as equipment choices are made and unmade, the collection of dangerous flows may change and perhaps even the medium itself. Any such changes will force a re-development of the safety policy. However, at each point, the scope of the safety policy remains the same: it does not creep inexorably into the deepest nooks and crannies of system design as does the traditional failures-oriented hazard analysis process.

In some cases there will be an existing safety policy applying the operation context and analysis may be concentrated on the ways that the new system may perturb the existing policy. For example, the addition of a new radar source may require further constraints on existing radar sources, perhaps reducing their maximum intensity, perhaps restricting their direction of signal.

In any case, the desired endpoint is a convincing, positive argument that the proposed safety policy will ensure a safe environment (when operated in the proposed medium). The purpose of subsequent safety engineering, is to enforce adherence to the chosen safety policy.

8 The Safety Protocol

In developing the safety policy, the focus is on the interaction between systems and humans, determining what constraints systems must satisfy in order to operate safely. The obvious next step is to move our focus onto the interactions between the systems themselves, placing a structure on those interactions that will enable adherence to the chosen safety policy. During this design process, safety constraints may be decomposed and/or strengthened and new flows introduced to serve as communications channels between systems. The eventual aim is to assign to each individual system a collection of safety requirements expressed solely in terms of the given system's inflows and outflows — in such a manner as to ensure that the aggregation of all the system safety requirements implements the safety policy. We call such an assignment of safety functionality across the various systems a *safety protocol*.

The safety protocol constraints on S must be expressed in terms of the various system interfaces so that they can then be adopted as system safety requirements as development moves into the system architecture phase. Safety protocol constraints on systems in the operational context should also act as safety requirements to their respective systems and may be needed as assumptions during safety architecture verification on S. In any case, it seems appropriate to treat them on a par with the constraints on S and express them solely in terms of their system's interface.

The safety protocol should also describe any safety mitigations present in the operational context. Such factors include redundant safety functionality, system isolation, safety monitoring and protective barriers: anything that may impact on the degree of reliance placed on S for ensuring the overall safety the environment.

The development of a safety protocol is a creative process, concentrating on the desired interactions between the systems in the environment. It may include many aspects of traditional hazard analysis, but without the usual focus on failures. It may involve a certain amount of trial and error, proposing safety constraints and challenging them with accident scenarios that show them to be inadequate. Equally, it may involve the adoption of standardised approaches to controlling specific dangerous flows or mathematical calculation of "worst-case" propagation of dangerous flows in the given medium. Many existing standards provide useful guidance on developing hazard control protocols (Royal Australian Navy 2006, Department of Defense 1993, Department of Primary Industries 2007).

In any case the desired endpoint is a convincing argument that the protocol actually implements the safety policy.

9 Methodological Matters

While we have spoken of the safety protocol *approach* in the preceding, what we have described is perhaps better considered a modelling framework from which to hang considerations about the fundamental nature and purpose of system safety engineering. We do not suggest that this framework provides even a significant portion of a viable methodology for safety engineering, but we do believe that it has shown its value in immediately clarifying some of the murky waters surrounding the foundations of system safety. This can be used to assist in evaluating and utilising existing methodologies. We do not go into any details of existing methodologies here, but raise some relevant points in the following.

Recognising the safety constraint as the primary focus of safety engineering effort has the potential to clarify the concept of hazard, as used in existing safety techniques. Current definitions of hazard are unsatisfying in that they conflate the notion of constraint violation with those of dangerous flow and equipment failure. This is particularly problematic when considering (or justifying the exclusion of) accident scenarios involving dangerous flows or equipment failures that are managed by protocol measures in the operating context. Both dangerous flows and equipment failures may legitimately occur within a system operating correctly within its safety constraints – this is after all the purpose of the safety constraints. It is important that practitioners clearly distinguish these three concepts and that safety methodologies should encourage them to do so.

The authors' primary interest in this system safety model lies in its potential to enhance the presentation and evaluation of safety cases. We have identified a simple three-level taxonomy of safety constraints. Policy constraints directly address the safe management of dangerous flows within the operating context. Protocol constraints address the safe interaction of systems in the operating context. Architecture constraints address the safe interaction of components within a system. This separation of concerns clarifies the structure of the safety engineering process and opens the potential for highly formal requirements tracing in support of high assurance safe cases. Again it is likely that practitioners will benefit by clearly distinguishing these three levels of safety constraint and, even if they do not require such structure, safety methodologies should, at least, be able to accommodate it. Moreover, a constraint focus allows safety engineering to be better integrated with the general engineering process, which is also requirements focused.

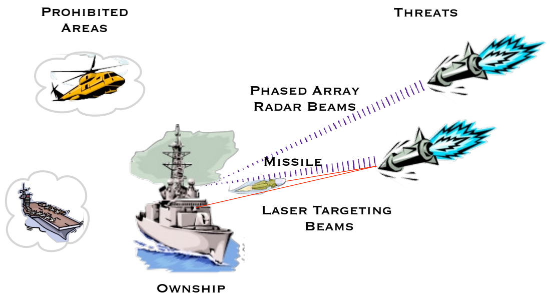


Figure 7: PARTI System Overview

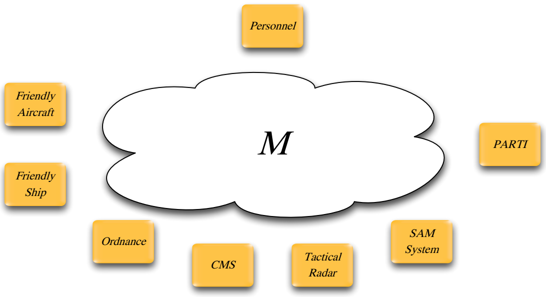


Figure 8: PARTI Environment

10 The PARTI Environment

We illustrate the safety protocol approach using a case study from DEF(AUST)10679 (Department of Defence 2008*a*, Mahony & Cant 2008). All of the information presented here is taken from that case study and the purpose is to contrast safety protocol development with the hazard analysis approach taken there.

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 elements of interest in the PARTI and environment are depicted in Figure 7.

Our approach begins by describing the environment in which the PARTI must operate in sufficient detail to allow the development of a safety policy governing the operation of the environment with the PARTI. Ideally there would be an existing fleet-level or theatre-level safety policy which we would be updating to allow the inclusion of one or more PARTI systems on ship platforms. Unfortunately, no such policy is likely to exist and developing such a policy is likely beyond the scope of the PARTI development. Instead we strive to describe only those aspects of the environment, safety policy and safety protocol relevant to the safe operation of the PARTI. Additionally, for the sake of brevity, we adopt a narrow focus on "system safety" issues — ignoring "OHS" issues — that would not be appropriate in a real safety case.

The PARTI system is to be installed on a class of frigate operating in a naval theatre of operations. The elements of safety concern in this environment are as follows.

CMS — The Combat Management System (CMS) provides command and control over all ship-based systems. The CMS has no relevant dangerous flows.

Search Radar — The CMS supports a conventional search radar for maintaining situational awareness and supporting an Identify Friend and Foe (IFF) functionality. The radar emits a HF radio signal.

SAM — The SAM System launches missiles against targets identified by the CMS. The missiles have target illumination home-all-the-way capability that is to be enabled by the PARTI. Missiles also have self-destruct functionality that will activate on order from the SAM system, on loss of target illumination and at mission expiry. The SAM system has no other ability to influence missile flight post-launch. The SAM system can deliver dangerous kinetic and explosive flows through its missiles.

PARTI — The PARTI provides precision tracking and target illumination in support of the SAM system. The PARTI can deliver dangerous HF radiation and laser energies.

Ordnance — Other systems include the harpoon missile, the 5-inch gun, the Nulka Anti Missile Defence system and a torpedo system. These ordnance can deliver dangerous explosive energies.

Friendly Aircraft — Helicopters may land or take-off from the frigate and/or from nearby ships. Other friendly aircraft may also be present. These aircraft can deliver dangerous kinetic, chemical and fire energies.

Friendly Ships — The frigate may be accompanied by other non-hostile surface vessels. These ships can deliver dangerous kinetic, chemical and fire energies.

The structure of the environment is depicted in Figure 8.

The medium M resolves the physical interactions between system outflows to determine the resultant system inflows. For the most part this is a straightforward resolution of positional interactions, in particular determining when humans actually come in contact with dangerous flows present in the environment: is the human present when a collision between aircraft and missile or terrain causes dangerous acceleration; is the human close enough to an explosion to be harmed; is the human in the path of a laser beam. Interactions with the (at least) two radars are also complicated by the need to resolve the superposition of the interacting wave forms.

11 The PARTI Safety Policy

The purpose of the safety policy is to describe safe interaction between the systems and the humans in the environment. The most desirable approach to ensure safe interaction is to restrict the release of energies to safe levels: prevent collisions and dangerous accelerations; prevent fires; etc. However, the PARTI environment contains a number of dangerous flows that might reasonably be termed "mission critical". The SAMs must fly energetically to their target and explode effectively. The PARTI must emit radar and laser signals at intensities suitable for the purpose of guiding the SAMs to destroy incoming threats. Instead of preventing the release of these energies, our policy is ensure that they are never released in the presence of the humans in the environment.

To achieve this, a region of the environment is set apart for the enacting of PARTI functionality, the SAM system and the PARTI agreeing to operate solely in this space and the humans in the environment agreeing not to enter it. This *tactical region* may vary according to the threat situation but will always exclude inhabited areas of ownship and other friendly surface vessels. The DEF(AUST)10679 case study (Department of Defence 2008*a*), adopts a policy of always setting the tactical region so as to exclude all friendly manned traffic in the environment's airspace. This protects friendly manned aircraft at the possible expense of making it impossible to effectively respond to an incoming threat. Thus, since the safety onus is placed on the PARTI, friendly aircraft and ships effectively have no safety responsibilities related to the PARTI system.

Overall, the safety policy constraints are as follows.

CMS — The tactical region is set and promulgated by the CMS, according to situational awareness and in response to command input. The tactical region must always exclude a suitable buffer zone around all manned friendly aircraft and surface vessels. The tactical region should always include only the essential volume(s) of space required to respond effectively to any identified threat(s).

Search Radar — The search radar shall always emit HF radiation within safety standards set for naval operations.

SAM — The SAM system shall always operate its missiles within the tactical region.

PARTI — The PARTI shall always emit its HF radiation and laser beams only within the tactical region.

Ordnance — All ordnance may only detonate within the tactical region.

12 The PARTI Safety Protocol

We now proceed to consider the potential interactions between systems in the environment so as to develop a protocol for their safe operation in the environment.

An obvious matter of concern in regard of ordnance and friendly vessels lies in the potential for radar and laser beams to damage these systems with consequent loss of safety control and release of dangerous flows. To avoid this threat, it is sufficient to ensure that radar and laser energies are never directed at any of these systems. This is essentially the purpose of the tactical region safety constraints and they serve to protect the systems as well as their human operators.

The primary matter of concern is the PARTI mission of providing target guidance for the SAM system. Since the SAM system has little control over missiles once launched, the PARTI must have primary responsibility in ensuring that missiles fly within the tactical region and therefore do not interfere with friendly traffic. Clearly, these systems must communicate effectively if they are to operate safely and this communication will be enacted through the CMS.

The CMS determines (in response to operator input) when the SAM and PARTI are operational.

The CMS develops situational awareness of the threat environment through an array of sensors, including the search radar and IFF. This situational awareness is transmitted to the PARTI as a list of tracks, some of which are tagged as threats.

The CMS determines (in light of its situational awareness and in response to operator input) the tactical region and factors this information to the PARTI.

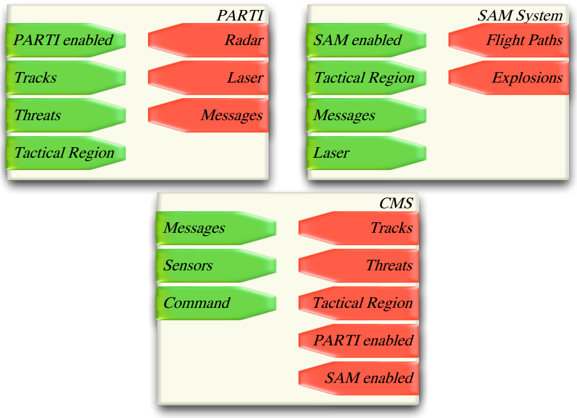


Figure 9: Protocol Interfaces

Based on this information from the CMS, the PARTI uses its PAR to acquire precision tracks on identified threats and then commences illuminating them with its targetting lasers. When target illumination is established it sends a (target) *acquired* message to the SAM system, along with the current position of the threat. The identified threat is then referred as a *target* until a (target) *released* message is sent or it is destroyed.

The SAM system configures a missile to acquire target lock on the identified threat position and launches it. The missile briefly flies a preprogrammed path ρ, within the tactical region, during which it must either attain target lock or self-destruct. Once target lock is attained it flies a line of sight path to the threat.

The PARTI maintains target illumination until the threat is destroyed or it becomes unsafe to maintain illumination. It is safe to maintain illumination provided the laser and the missile have safe *line of sight* to the threat. Line of sight is essentially the straight line path between objects, modulo the missile's flight navigation tolerances. The line of sight is safe provided it is entirely within the tactical region and there is no third object in line of sight. If line of sight becomes unsafe, the PARTI informs the SAM system and ceases illumination. The SAM system then transmits a self-destruct command to the missile, which will self-destruct, either because it has detected the loss of target illumination or because it has received the self-destruct command.

The resulting safety interface to the PARTI safety protocol is shown in Figure 9 (outflows are shown in red and inflows in green).

The protocol described above is summarised by the following safety requirements.

CMS_A – at all times the tracks presented form an accurate model of the objects moving through the environment to within allowed tolerances.

CMS_B – at all times the tactical region plus an allowed tolerance contains no friendly tracks.

SAM_A – if a missile is launched, the SAM system is enabled and there is a valid target.

SAM_B – when a missile is launched, it initially flies along an initial path ρ safely within the tactical region.

SAM_C – if a missile departs from its initial path ρ, it has either acquired target lock or self-destructed.

SAM_D – while a missile has target lock, it navigates a line of sight path to its target to within allowed tolerances.

SAM_E – if an in-flight missile does not acquire target lock promptly or loses target lock or flies beyond its mission deadline or the SAM receives a target release message, the missile self-destructs promptly.

PARTI_A – only tactical regions are irradiated.

PARTI_B – if an object is being illuminated then it is a target.

PARTI_C – each target acquired message gives the correct current position of a threat within the tactical region.

PARTI_D – if an object is a target then it is a threat, the PARTI is enabled and the object is being illuminated.

PARTI_E – if line of sight to an object is not safe then it is not a target.

PARTI_F – if the PARTI is not enabled then it is not radiating or illuminating.

The protocol constraints CMS_A and CMS_B clearly ensure satisfaction of the CMS safety policy constraint. Similarly, the PARTI_A, PARTI_B, PARTI_E and PARTI_F ensure satisfaction of the PARTI policy constraint.

Modulo the determination of correct tolerances, the protocol also ensures satisfaction of the SAM policy constraint. To see this, consider the path of a missile from launch to detonation or self-destruct. SAM_B ensures that some initial segment of the missile's path is safely in the tactical region. Now suppose that the missile has travelled safely in the tactical zone up until some point x on its flight path. Then the missile is safely within some fixed tolerance, say δ, of the tactical region boundary and regardless of the missile's current speed and direction, it will remain within the tactical region until it has moved a distance $\frac{\delta}{2}$ to a point y. Either the threat remains a target while the missiles moves to y or it does not. If the target remains a threat, then by PARTI_E there is safe line of sight to the target at all times and in particular the point y is safely within the tactical region. If the threat ceases to be a target while the missile moves, then by PARTI_B the threat is no longer being illuminated and by PARTI_E a target released message has been sent. Thus, by SAM_E the missile will self-destruct before it can leave the tactical region.

Note that SAM_C and SAM_D are not used in this argument as they are in fact redundant safety functionality. If either protocol constraint is violated, the PARTI will detect loss of line-of-sight, release the target and the missile will self-destruct.

13 Final Remarks

In this paper, we have discussed the traditional hazard-based approach to developing system safety requirements, highlighting a number of its properties that we see as short comings. In its place, we suggest a more contained investigation of the dangerous flows associated with a system and its environment, together with a more direct determination of a policy for safely constraining these dangerous flows. A safety protocol is then developed that describes a particular way of coordinating the interactions between systems in the operational context so as to implement the safety policy and therefore ensure a safe environment.

Why have we used the word "protocol" instead of "requirement"? This is a most interesting question. In the computer science field the term protocol is used to mean "a set of rules governing the exchange or transmission of data between devices". There are many familiar examples: communications protocols, such as TCP/IP, but also security protocols, such as Internet Key Exchange. In the field of safety, the term "protocol" has so far mostly been used in a narrow sense to denote procedural rules designed to promote safety, such as rules for food handling or procedures for operator behaviour in factory operations. We like the term in our situation because we wish to think that averting a dangerous flow is going to involve a kind of "agreement" or "handshake" between the various systems in the operational context. To use another familiar analogy, it is going to be a "contract" that represents agreement between the components. The notion of protocol is broad enough to cover both system design constraints and procedural obligations on humans.

References

Comcare (2007), Identifying hazards in the workplace, Guide booklet, Australian Government, http://www.comcare.gov.au/.

Department of Defence (2008*a*), Guidance Material for DEF(AUST)5679/Issue 2, Australian Defence Handbook DEF(AUST)10679/Issue 1, Australian Government.

Department of Defence (2008*b*), Safety Engineering for Defence Systems, Australian Defence Standard DEF(AUST)5679/Issue 2, Australian Government.

Department of Defense (1993), System Safety Program Requirements, Military Standard MIL-STD-882C, United States of America.

Department of Defense (2010), Joint Software Systems Safety Handbook (SSSH), DoD publication, United States of America.

Department of Primary Industries (2007), Guideline for hazardous energy control (isolation or treatment), MDG 40, New South Wales, http://www.dpi.nsw.gov.au/minerals/safety.

Haddon-Cave, C. (2009), *The Nimrod Review*, The Stationery Office Limited, UK.

Kelly, T. (2011), 'Safety critical mailing list', http://www.cs.york.ac.uk/hise/sc_list_arc.php.

Leveson, N. G. (1995), *Safeware: System Safety And Computers*, Addison-Wesley.

Leveson, N. G. (2011), 'Engineering a safer world', http://sunnyday.mit.edu/safer-world/index.html.

Mahony, B. & Cant, A. (2008), The PARTI architecture assurance, *in* 'Proceedings of the 13th Australian Conference on Safety-Related Programmable Systems', Australian Computer Society.

Ministry of Defence (2007), Safety management requirements for defence systems, part 1 requirements, Defence Standard 00-56, British Government.

Royal Australian Navy (2006), Navy Safety Systems Manual, ABR 6303, Australian Government.

Schellhorn, G., Thums, A., Reif, W. & Augsburg, U. (2002), Formal fault tree semantics, *in* 'Proceedings of The Sixth World Conference on Integrated Design & Process Technology'.

Practical Early-Lifecycle Application of Human Factors Assessment

Simon Connelly, Andrew Hussey, Holger Becht
Ansaldo STS Australia
11 Viola Place, Eagle Farm 4009, QLD

simon.connelly@ansaldo-sts.com.au

andrew.hussey@ansaldo-sts.com.au

holger.becht@ansaldo-sts.com.au

Abstract

Human Reliability Analysis (HRA) is often seen as a time consuming task, which requires significant expertise. This may lead to a reduced focus on the human in the loop, and a failure to consider both where human error and recovery may impact on system safety performance.

Through the use of a case study involving a Positive Train Control (PTC) driver interface, this paper aims to examine whether early system architecture phase task analysis can produce meaningful results with little time overhead or human factors expertise. The approach which has been used was to conduct a task analysis on a system sequence diagram, identifying the high order goals and the individual driver tasks, including alternate paths. Once this task analysis was completed, a tailored FMECA was conducted to identify human failure modes which may lead to system hazards and to thereby limit the scope of the subsequent HRA. The criticality analysis was performed via a HEART analysis to estimate error likelihoods, and which also identified risk factors in the HMI design and operating environment.

The outcomes of the case study were design requirements on the resulting driver interface, in addition to operating procedures, and training requirements. It is argued that the approach presented allows for an analysis to be conducted early in a system design lifecycle at low cost and with limited expertise, which adds to the overall safety argument for the end product.

Keywords: Human Reliability Analysis, HEART, Human Factors

1 Introduction

This paper provides a method for analysing and assessing safety-related human-machine interfaces. The method provided extends on the existing methods currently used for such assessment. Four key contributions and advances over current learning within the domain of analysis and assessment of safety-related human-machine interfaces are argued. These contributions/advances are summarised as follows:

1. Analysis is conducted early in the development lifecycle, before significant effort has been expended on developing the HMI, so that the development work can be guided from the outset by the analysis;
2. The method requires minimal human-factors expertise and can be performed by engineers with minimal (re)training. This is not to say that later more detailed expert analyses would not be conducted, but such small-footprint analysis means that it can be done as part of the normal safety engineering process and before time has been expended developing options that later may turn out to be unsuitable in terms of overall risk;
3. The method utilises and combines well understood safety-analysis techniques with HMI-analysis methods, meaning that safety engineers are extending and building on their existing knowledge; and
4. The method uses a quantified analysis to make comparative assessments of risk, to guide overall development direction. The intent is not to make a formal/precise assessment of quantified error likelihood, but to enable different design options to be compared within a broad risk framework.

The remainder of the paper is structured as follows:

1. Section 2.1 provides an overview of the literature concerned with HMI analysis techniques applied in the paper.
2. Section 2.2 discusses methods for assessing the risk of human error.
3. Section 3 discusses the analysis method used in this paper, which combines elements of both existing HMI-analysis techniques, as well as commonly used safety-engineering methods.
4. Section 4 gives an overview of the Case Study used in the paper, as well as the significant outcomes of the analysis, in terms of the specific HMI under analysis.
5. Section 5 summarises and concludes the paper, linking the contributions/advances listed above with the method and outcomes discussed in Sections 3 and 4.

2 Literature Survey

2.1 Analysing Operator Error

Operators are an integral part of any interactive system, working with safety-critical machines via operator interfaces to achieve task goals. The safety argument for an interactive system should provide confidence that hazardous operator error rates have been minimised by analysis of operator characteristics (e.g., skill level and training) and the characteristics of the workplace of which the operator is a part (e.g., noise levels, lighting and the task itself).

Hussey & Atchison [Hussey00] presents a generic method for operator safety case preparation. Per Hussey & Atchison, there are four key steps to analysing hazards arising from operator error:

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

2.1.1 Task Analysis

Tasks are goal-directed activities to transform some given initial state into a goal state. A task can be decomposed into sub-tasks unless the task is itself composed only of elementary actions.

"Knowledge Analysis of Tasks" (KAT) [Johnson 92] is a form of Cognitive Task Analysis (CTA) and divides a task analysis into four main parts:

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

CTA and similar techniques are well established, e.g., Carroll and Rosson have used scenarios as design representations [Carroll90].

Task models enable identification of requirements and analysis of designs for new requirements and user training needs [Johnson90]. Task models examine the knowledge or competence required to operate a system [Hoppe90].

For the purpose of safety-critical systems, the task analysis may describe procedures for normal operation of the system, maintenance procedures and also procedures for emergency situations [Redmill97]. The description of procedures for normal operation and maintenance should include any recovery steps by which errors of the user are detected and corrected to avoid an accident [Kirwan92]. Task Analysis may be conducted within the context of an overall Cognitive Work Analysis (CWA) [Vicente99]. The CWA informs the task analysis process and provides a functional model of the workplace within which tasks will be performed.

In this paper, to maintain the simplicity of the method that is presented, only the basic task analysis techniques, such as KAT, are considered. More advanced workplace models could be constructed to further inform both the task and human error analysis. However the extension of the methods in this paper to consider such workplace models is outside the scope of this paper.

2.1.2 Human Error Analysis

HAZOP Studies (e.g., [Std00-58]) and FMEA (e.g., [StdIEC-1025]) are the predominant techniques for analysing human error based on a task analysis as the model of the system. HAZOP Studies have been used by e.g., [Chudleigh93], [Kirwan94] and [Leathley97]). Because it is possible to categorise human error types and mechanisms, FMEA is the basis of many current methods for human error analysis including HEART (Human Error Assessment and Reduction Technique) [Williams86] and THERP (Technique for Human Error Rate Prediction) [Swain83]. THERP has been further developed and specialized for the nuclear plant industry via the SPAR-H method [Byers00].

HEART and THERP are both "first-generation" HRA methods. First generation techniques use a simple pattern-matching of the error situation with related error identification and quantification whereas second generation techniques are more theory based in their assessment and quantification of errors. One of the more widely used second-generation techniques is CREAM [Hollnagel98]. CREAM uses performance criteria (both positive and negative) in combination with a model of cognitive demand to determine overall error probabilities.

Only unintentional errors are considered in this paper. Categories of error include omissions, substitutions and repetitions (the latter two are commission errors) [Senders91]. Example error categories documented by Redmill [Redmill97] include: Action or check made too early or too late; Right action or check on wrong object; Wrong information obtained.

Reason and Embrey [Reason86] and Whalley [Whalley88] have summarised the common causes of human error:

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

Norman's [Norman90a] model of human-machine interaction is referred to as the "execution-valuation" model (also refined by Rasmussen [Rasmussen83] in his "step-ladder" model of decision making from automatic activation and execution through to conscious interpretation and evaluation). Norman categorises errors into two types; slips and mistakes. The same distinction has also been made by Reason [Reason90]. Slips are concerned with automatic behaviour at the physical execution level. Mistakes are the result of conscious deliberation; a "wrong" procedure is formulated. Errors arise when decision makers take short-cuts in the decision process, e.g., using rule-based routines when knowledge-based decision is demanded by the novelty of a situation [Reason86]

2.1.3 Error reduction

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

1. Prevention by hardware or software changes: automation of tasks and use of interlock devices and behavioural "forcing functions" to prevent error. Norman [Norman90a] calls features that prevent slips or mistakes "forcing functions" because they force a user to choose a safe sequence of actions. Whilst automation of functions is necessary for tasks that exceed an operator's physical capabilities [Mill92], the automation must not leave residual tasks that are outside the operator's capacity (e.g. during emergency situations) [Bainbridge87].
2. Enhanced error recovery: provide feedback, checking procedures, supervision and automatic monitoring of performance.
3. Reduce errors at source: improve procedures, training and interface design.

2.2 Risk Assessment

2.2.1 Qualitative Assessment

Error Producing Conditions (EPCs) are associated with characteristics of the operator interface, the individual, human cognition generally and the organisation [Rasmussen82]. Redmill [Redmill97] has produced a categorised list of EPCs including: Task demands and characteristics; Instructions and procedures; Environment; Displays and controls; Stresses; Individual capabilities; Social and cultural influences.

2.2.2 Quantitative Assessment

Human reliability quantification techniques aim to quantify the Human Error Probability (HEP) which is defined as: number of errors per number of opportunities for error. EPCs similar to those given by Redmill [Redmill97] appear as a factor in most of the available estimation methods, e.g., HEART, THERP [Kirwan90].

HEART uses a combination of generic task categories, coupled with nominal human unreliability assessments (HRAs) (as performance ranges), as well as EPCs that are used to refine those nominal HRAs to generate an assessed nominal likelihood of failure or Human Error Probability. HEART requires that the assessor judge the effect of each EPC (in terms of assessed proportion of possible effect, APE, between 0 and 1).

More recently, the Rail Safety and Standards Board (RSSB) issued a rail-specific HRA method based on HEART, which incorporated a rail-specific taxonomy of human error [RSSB04]. The RSSB-HRA method is oriented however toward existing rail technologies, whereas the example case study application considered in this paper is novel in its approach to railway operations. For this reason, we chose to use HEART rather than RSSB-HRA for our case study.

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

3 Methodology

3.1 Analysis Process

The analysis takes as an input the functional requirements and an understanding of the HMI design (an early prototype or design proposal is sufficient). User Goals are identified based on the functional requirements. A separate set of goals should be generated for each user group and system function.

For each goal we identify the tasks to be completed to achieve the goal, including alternate paths based on choice points (e.g. confirm correct vs. reject incorrect input) – a system architecture specification is generally useful for this step, however it isn't necessarily required, merely an understanding of the interaction sequences between the system and the user which are required to achieve the goal.

Once the task analysis is complete for all goals, conduct a modified FMECA on the output, analysing each step as a separate "component". Guidewords or SME advice may be used to determine the valid failure modes for each task step. To enable this analysis the "standard" FMECA process has been tailored as shown in Table 1. The event tree for the task analysis can be represented directly in the FMECA table to enable the task analysis and safety analysis to be combined.

3.2 Intent

The method discussed in this paper provides four key advantages over existing approaches to HMI-analysis of safety-related systems. The advantages are discussed in the following subsections.

3.2.1 Early lifecycle analysis

The method used in this paper enables engineers to conduct an early lifecycle analysis with reduced need for expert HF input, and provide early design advice on the suitability of a proposed HMI design. By conducting such analyses early in the product lifecycle it is possible to achieve customer / end-user buy-in of safety related interface designs, and also to determine where specific workflows may need to be enforced to achieve system level safety requirements.

Identification of critical tasks may also enable efficiencies to be designed into the workflows with limited impact on safety performance.

3.2.2 Minimal human-factors expertise

As the conduct of FMECAs is well understood within the RAMS and Engineering communities, it is envisaged that a broad range of resources could apply this approach, without the need for detailed HF knowledge.

It is important to note that this isn't a full HEART analysis as such a full analysis would require much more time than is intended here, and significant support from skilled HF resources.

Column Title	Description
ID	Row identifier
Goal	Top level goal identified from analysis of functional requirements
Main Task	Step in the main task sequence
Alternate Path(s)	Possible alternate sequence steps broken off at each choice point. Can rejoin at the next main step or reference an earlier or later main sequence step
Failure mode(s)	Possible failure mode for this task, where a task has more than one valid failure mode, each should be examined. Failure modes may be based on SME advice, or guidewords
Local Effect	impact on the current task, including implications for future task steps (be they main or alternate)
System Hazard	Definition of any possible hazards the local failure effect may present
HEART task Category	HEART category selected for this task failure
Category nominal unreliability	The nominal human unreliability allocated to the selected Heart category
Error Producing Conditions	A summary of the applicable EPC(s) from the HEART table, and the Assessed Proportion of Effect (APE) for each.
HEP	Calculated based on the HEP for the task failure
External Triggers / Conditions	Defines what is required for hazard to become an accident, including pre-existing mitigations
Adjusted likelihood of hazard	Modified HEP taking into account the impact of the external triggers and conditions
Severity of accident	Severity of worst credible accident, calibrated to match the risk matrix or other technique, in use.
Risk	Calculated risk
Recommended mitigations	Mitigations to reduce risk where required, taking into account the hazard mitigation hierarchy. Noting that the focus of this analysis technique is to reduce either the category of the Task, Or to remove EPCs / reduce their APE.
Post-mitigation Likelihood	Likelihood following implementation of mitigations
Post-mitigation Severity	Severity following mitigation
Post-mitigation Risk	Calculated Risk, once all mitigations implemented
Comments	Any further comments which relate to this failure mode.

Table 1: Analysis structure

3.2.3 Well-understood techniques

The approach focuses on conducting an early breakdown of User Tasks, based on Functional requirements of the system, and performing a FMECA style analysis on failures of each user task. This FMECA is calibrated based on a very quick HEART based analysis (guided directly by the tables). The intention is to remove subjectivity in the analysis, by using a structured calibration such as HEART it allows the team conducting the analysis to compare like with like.

3.2.4 Comparative assessment

The analysis is not intended to form a quantitative risk analysis, rather it allows for a comparative assessment of the different HMI hazards presented in the proposed system.

As the analysis is comparative in nature, it is argued that it is less prone to individual risk rating criteria. As long as the engineer conducting the analysis applies the HEART method consistently, it does not matter if they have a more or less risk averse strategy.

This will enable the system designers to either eliminate hazards, or develop the system in such a way to reduce these hazards SFAIRP. The key outcomes will be the critical functions, and the magnitude of achieved risk reduction from the selected mitigation strategies.

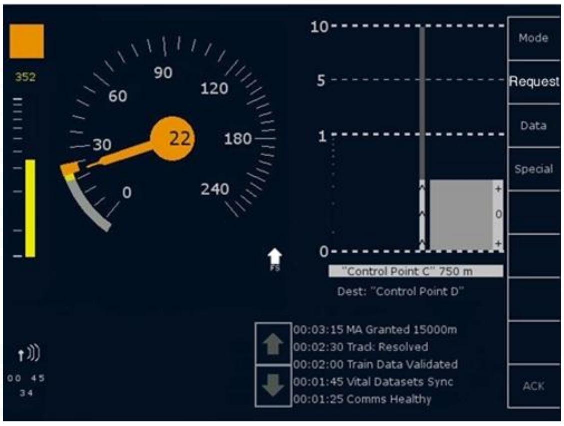
4 Case Study

To demonstrate an application of the methodology a case study is provided in this section. The selected case study examines a proposed design for a Positive Train Control (PTC) Driver Machine Interface (DMI). No specific technology is referenced, as the purpose is to examine the interface only, rather than compare different PTC solutions.

4.1 PTC Screen

The proposed PTC under examination provides supervision of a train against defined allowed speeds (both temporary and permanent) and defined Limits of Authority (LoA) within a rail network. To enable this supervision the PTC needs to be configured with the network geography to be covered and the specific configuration of each train to be supervised.

Two separate interfaces are provided to support the configuration and operation of each installed PTC. To configure the track database, and generic information about the network and train types a Maintenance interface is provided, which is portable and shared between the fleet. Each locomotive is also fitted with a DMI which allows for driving advice to be provided to the driver, and to seek configuration specific to a given train, or driver confirmations.

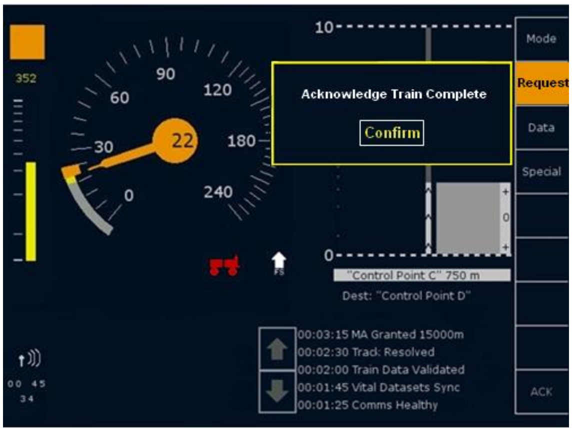


"Control Point C" 750 m
Dest: "Control Point D"

00:03:15 MA Granted 15000m
00:02:30 Track Resolved
00:02:00 Train Data Validated
00:01:45 Vital Datasets Sync
00:01:25 Comms Healthy

Figure 1: Example PTC DMI

An example DMI configuration is provided in Figure 1. The DMI has a touch sensitive screen which is used to receive input directly from the driver. Data entry is managed through selecting menu items from the right, and entering data, or confirming data as shown in Figure 2.



Acknowledge Train Complete

"Control Point C" 750 m

Dest: "Control Point D"

00:03:15 MA Granted 15000m
00:02:30 Track Resolved
00:02:00 Train Data Validated
00:01:45 Vital Datasets Sync
00:01:25 Comms Healthy

Figure 2: Confirmation overlay.

It is the second interface that this paper is concerned with. In summary the following functions are provided by the DMI:

1. Display LoA information;
2. Display Maximum allowed speed, and upcoming speed changes;
3. Display warnings of impending violation, and notification of enforcement;
4. Display driving mode (i.e. whether the train is under active supervision or not;
5. Receive train configuration particulars; and
6. Receive driver confirmation of internal data; and
7. Confirmation for significant alerts.

Items 5, 6 and 7 in the above list have been selected to demonstrate the analysis methodology. These functions were selected as they involve user input and multiple interaction steps, which present immediate opportunities for error. Functions one to four involve general information gathering and situation awareness. Such functions are subject to latent understanding failures and require a more thorough understanding of the context of use than is feasible to provide in this paper.

4.2 Use Cases

To demonstrate the simple task analysis three functions have been selected. To provide context, a brief summary of the requirements is provided, followed by the use case sequences. Whilst the sequences could be represented as either sequence diagrams or UML Use cases, a tabular format is provided in Table 2 to provide an example of a simple, yet effective representation.

1. Track Selection: At the commencement of a mission the PTC system may not be able to accurately resolve which track a train is on in multiple tracking areas. The PTC shall request Driver selection of current track occupancy from a list of available tracks.

2. Enter Train Data: At the commencement of a mission the PTC shall default to "worst case" train configuration, i.e. most restrictive braking enforcement calculations assuming maximum length and weight. To allow for reduced headways the PTC shall allow the driver to enter the current configuration of the train. The driver shall be required to confirm train configuration prior to any modification of the braking calculations.

3. Confirm Integrity: Should the PTC detect a loss of train integrity (defined to be a significant change of detected length, or an unexplained loss of brake pressure) it shall immediately alert the driver to the loss of integrity, and notify the central authority server (In the ERTMS concept this is referred to as a Radio Block Centre, or RBC) to prevent roll-up of protection behind the train. The PTC shall allow the driver to "acknowledge train integrity" (despite the system detection of loss of integrity) thereby removing the alert and allowing for roll-up behind the train.

ID	Goal	Main Task	Alternate Path(s)
1	**1. Track selection**	1. DMI displays candidate list of tracks received from DB	
2		2. scroll and select track	
3			2a.1. scroll
4			2a.2. close window
5	**2. Enter data**	1. Select data menu	
6		2. Select driver ID/train ID/train data	
7		3. Enter data via keypad	
8			3a.1. navigate text via arrow keys
9			3a.2. delete text via delete key
10			3a.3. return to 3
11		4. observe entered data on confirmation window	
12		5. confirm or reject entered data	
13	**3. Confirm integrity**	1. Observe Loss of integrity	
14		2. select Request menu	
15		3. select Train Integrity menu	
16		4. select Acknowledge	

Table 2: Example Sequences

4.3 FMECA

To demonstrate the application of FMECA to the task sequences identified in Table 2 an analysis of Function 3 (Confirm Integrity) is provided in Table 3. In the interests of space only Steps 1 and 4 are shown. The outcome of failures to perform steps 2 and 3 were determined to be equivalent to failure to identify a loss of integrity from a system point of view.

The table shows possible error modes for IDs 13 and 16 in the task analysis. There are two error modes for ID 16 and these are shown as 16a and 16b.

Comparing the pre-mitigation risk likelihoods of line item 13 with line item 16b it is clear that item 16b is more critical (several orders of magnitude), even taking into account the subjective nature of the failure rate calibration. As such it is reasonable to apply further risk reduction to incorrect confirmation of integrity.

ID	13	16a	16b
Goal	**3. Confirm integrity**		
Main Task	1. Observe Loss of integrity (B7)	4. select Acknowledge	
Alternate Path(s)	-	-	-
Failure mode(s)	Fail to observe loss of integrity, proceed on mission without conducting rest of task	Fail to acknowledge (cancel request)	Acknowledge that train is complete when wagons have been left behind or are being dragged.
Local Effect	Fail to confirm integrity, RBC prevents roll-up behind train	Fail to confirm integrity, RBC prevents roll-up behind train	Fail to initiate safeworking procedures to protect following trains
System Hazard	Obstruction left on track / track damage	See line 13	Obstruction left on track / track damage
HEART task Category	F (taken to be top of the band as no external checking)	-	F (taken to be top of the band as no external checking)
Category nominal unreliability	0.007	-	0.007
Error Producing Conditions	8. Channel capacity overload 6, APE = .2 there is a lot of information on the display, only a small icon indicates integrity loss, audible alert should draw TD attention	-	4. A means of suppressing or overriding information or features which is too easily accessible 9, APE = 1 as the interaction for confirming loss is the same as confirming completeness and the warning will be removed from the DMI.
HEP	0.014	-	0.063
External Triggers / Conditions	Integrity failure must lead to wagon left behind or track damage (ARTC input) RBC Failure to protect following traffic (SIL 4)	-	Loss of integrity leads to wagons left on track (or track damage sufficient to cause derail)
Adjusted likelihood of hazard	4.424E-13	-	1.26E-05
Severity of accident	5	-	5
Risk	M	-	H
Recommended mitigations	No further reduction required	-	Recommend system allows for confirmation that integrity is lost, and design support in other system to protect following rail traffic. Additionally, require drivers to enter actual train length at train creation to reduce false positives. Results in moving Nominal Unreliability to the bottom of Band F (8E-4) and reduce APE to .2
Post-mitigation Likelihood	4.424E-13	-	4.16E-07
Post-mitigation Severity	5	-	5
Post-mitigation Risk	M	-	M
Comments	Assume design shall be updated to include Audible alert on detection of loss of integrity Assume integrity is lost once every 1000 train hours.	-	Design is such that drivers are not required to enter train length on train creation, leading to default (worst case) figures being used) Assume that confirmation of integrity involves procedural checks such as walking the train length or seeking confirmation from crossing trains. Assume integrity is lost once every 5000 train hours. (comparable railway experienced 30 coupler separations in 150000 train hours)

Table 3: Example Sequences

To determine the most effective mitigation the engineer must examine the selected EPC(s) and determine to either remove them, or reduce their APE. As the currently proposed PTC DMI only allowed for operator confirmation that integrity had not been lost it was identified that to allow the driver to confirm that the train had lost integrity would reduce the likelihood of inadvertently suppressing the information. By extension, this also prevents the driver form unwittingly allowing following traffic into an area where there may be significant track damage (dragging wagons) or standing vehicles (where part of the train has separated). Furthermore it was identified that requiring the driver to enter the train length at train creation would reduce the number of "false alarms", leading to reduction in learned behaviour of ignoring integrity alerts.

By allowing the engineer to identify these design clarifications it was determined that this would reduce the nominal unreliability as well as the APE. Note that the Error Producing Condition was not completely removed as it is still possible to incorrectly suppress the alarm.

4.4 Outcomes

In terms of Norman's [Norman90a] model of human-machine interaction, the design of the system is prone to "slips" whereby the Driver confirms integrity when the train is not in fact whole. Similarly, using Reason and Embrey's [Reason86] list of human error mechanisms, the most prominent cause for item 16b is Stereotype takeover. The error reduction strategy chosen is a simple form of prevention by software changes, as discussed by Kirwan [Kirwan90]. In terms of the comparative assessment, the nominal likelihood calculated moves from 1.26E-05 to 4.16E-07. The reduction in likelihood is modest but may be sufficient where there are other factors necessary for an accident to occur. The analysis and proposed design solution indicates that risk has been reduced, but that the risk for 16b remains significant, and further attention may be necessary for risk to be reduced sufficiently in accordance with the overriding SFARP principle, as required by the Rail Safety Act.

4.5 Limitations

It is noted that the methodology presented in this paper is limited to those instances where qualitative analysis is appropriate. This is due to the comparative nature of the analysis; if human error contribution to overall system performance requires quantification, then a formal HRA will need to be conducted. The methodology also assumes that the task model is simple enough to be presented in the FMECA format discussed here. As such the level of abstraction must be selected carefully.

5 Conclusions

This paper has discussed a new approach to early-lifecycle analysis of HMIs, to determine risk and assess possible design mitigations. Four key contributions and advances over current learning within the domain of analysis and assessment of safety-related human-machine interfaces. These contributions/advances have been demonstrated in the paper as follows:

1. Analysis is conducted early in the development lifecycle, before significant effort has been expended on developing the HMI, so that the development work can be guided from the outset by the analysis. This is demonstrated by the analysis in the case study, which is based on a functional task model of the system and does not require detailed design mockups or storyboards;
2. The method requires minimal human-factors expertise and can be performed by ordinary engineers. The authors of the paper are not HMI experts, but have used the combined method demonstrated in the paper to propose changes to the case study HMI that would be later tested to show that they improve the overall safety of the system;
3. The method utilises and combines well understood safety-analysis techniques with HMI-analysis methods, meaning that safety engineers do not need to retrain, but instead are extending and building on their existing knowledge. The case study shows how we have combined FMECA, a commonly used safety-analysis technique, and a simplified version of HEART, an accepted HMI-analysis method;
4. The method uses a quantified analysis to make comparative assessments of risk, to guide overall development direction. This has been demonstrated by section 4.4 in the case study, which shows how we have used comparative assessments to claim that the revised HMI is safer than the original proposal.

To further demonstrate the effectiveness of this methodology it has been proposed for application to analysis of a train control system. This will allow us to refine the approach and integrate it into our safety lifecycle.

6 References

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

[Byers00] J. C. Byers, D. I. Gertman, S. G. Hill, H. S. Blackman, C. D. Gentillon, B. P. Hallbert, and L. N. Haney. Simplified Plant Analysis Risk (SPAR) Human Reliability Analysis (HRA) Methodology: Comparisons with other HRA Methods.. Idaho National Engineering and Environmental Laboratory, 2000

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

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

[Hollnagel98] E. Hollnagel. Cognitive Reliability and Error Analysis Method – CREAM. Oxford: Elsevier Science.

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

[Hussey00] A. Hussey and B. Atchison. Hazard Analysis of Interactive Systems, *TR-0018, Software Verification Research Centre, School of Information Technology, The University of Queensland*, May 2000.

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

[Johnson 92] P. Johnson. *Human Computer Interaction - Psychology, Task Analysis and Software Engineering*. McGraw-Hill Book Company, 1992.[Kirwan90] B. Kirwan. Human Reliability Assessment, In J. Wilson and E. N. Corlett, editors, *Evaluation of Human Work*, chapter 28, Taylor and Francis, 1990.

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

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

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

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

[Norman90a] D. A. Norman. The Design of Everyday Things. Doubleday, 1990. [Norman90b] D. A. Norman. The 'problem' with automation: inappropriate feedback and interaction, not 'over-automation', *Philosophical Transactions of the Royal Society of London, Series B*, 327(1241): 585-593, 1990.

[RSSB04] Rail Safety and Standard Board (RSSB) Rail-specific HRA technique for driving tasks. Final report, 2004.

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

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

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

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

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

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

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

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

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

[Vicente99] K. H. Vicente. *Cognitive Work Analysis: Towards safe, productive, and healthy computer-based work*. Lawrence Erlbaum Associates, 1999.

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

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

Risk Based Safety Assurance: towards a defensible and practical methodology

C.B.H. Edwards[1] M. Westcott[2]
[1]AMW Pty Ltd
PO Box 468, Queanbeyan, NSW 2620
Email: Chris.Edwards@amwps.com

[2]CSIRO Mathematical and Information Sciences
GPO Box 664, Canberra, ACT 2601
Email: Mark.Westcott@csiro.au

Abstract

The estimation of safety assurance for complex systems using risk based methodologies requires ranking of accident risk and (implicitly) the estimation of acceptable residual risk following system deployment. The paper describes a methodology for estimating and comparing risks, based on combining the derived Probability Density Function (PDF) of accident probability (Likelihood) with actual accident result (Outcome). The paper also suggests a method for calculating the residual risk after the completion of hazard mitigation activities, together with a possible methodology for implementing this approach. The effect of uncertainty on risk estimates, the importance of risk interpretation and the difference between risk and safety targets are also discussed.

Keywords: Hazard Risk Analysis, Residual Risk, Risk Uncertainty, Risk Interpretation, Marked Point Process, Stress-Strength Model.

1 Introduction

The use of Probabilistic Risk Analysis (PRA) is widespread throughout industry and government. An overview of PRA is provided by NASA (2002). A special case of this methodology is used in the safety evaluation of complex systems and is often known as Hazard Risk Assessment (HRA).

The efficacy of a risk based system safety program for systems being deployed in their intended environment follows from an arguable reduction in safety risk resulting from that safety program. The essential outcome of a HRA based safety program is called the Safe Residual Risk (SRR), a measure which purports to provide information about the risk of an accident after preventative measures are taken to mitigate an occurrence of the accident. This paper addresses practical issues of estimating risk and residual risk and also considers the uncertainties of those estimates.

2 Risk Concepts for Complex Systems

The concept of risk for a complex system assumes that the risk of a particular outcome is based on a relationship between outcome likelihood and the severity of that outcome. The acceptability of that risk is then dependent on the context of the risk taker and on whom the risk is imposed. For example, acceptable risks for driving on public roads are qualitatively different to those accepted by the aviation industry. Typically, an outcome is deemed to be acceptable if the likelihood of the outcome is below a predefined value, or if the nature of the outcome is considered to be relatively trivial. There can also be a cost/benefit comparison, where the cost of reducing risk is weighed against the benefits that could result from the reduction. This is one facet of the widely-used (and widely-discussed) ALARP criterion (see for example Melchers, 2001).

Comparison of risks often finds expression in the construction of a Hazard Risk Matrix (HRM) where the Severity of Outcomes and Likelihoods are tabulated in a two dimensional matrix, e.g., MIL-STD-882C. The limitations of risk matrices in a more general context have been widely documented for more than a decade with Cox (2008) providing a useful summary of the position. In the context of assessing the safety of complex defence systems Edwards *et al* (2009) discussed theoretical limitations when attempting to interpret an HRM and proposed a revised methodology aimed at improving the development of system safety assessment, mainly focussing on informally quantifying the Severity of Outcomes of the HRM, but also proposing a mathematical transformation of the estimated accident likelihood. Edwards and Westcott (2010) reviewed and extended the approach to the second HRM component, i.e., estimating the Likelihood of the occurrence of a mishap or accident and addressed the treatment of accident likelihoods by HRA based safety standards, where the resulting accident Probability Density Function

(PDF) was used to provide an empirical estimate of the accident likelihood.

3 Computation of Risk

3.1 Assumptions and Notation

We begin by noting that as far as possible, Safety terminology in this paper is based on that provided by the Royal Society (1992). We also note that while the term *likelihood* has a specific technical statistical meaning, in the HRA domain it is often used interchangeably with the word *probability*.

The occurrence of death or injury involving complex systems usually results from a series of (often unlikely) events. In the HRA domain such a series of events is often called the realisation of a hazardous state. Consistent with the ICAO definition we call the realisation of a hazardous state an Incident.[1] In combination with external coeffector(s)[2] an Incident can lead to an Accident. We thus distinguish between the incident sequence and the accident process with the aim of combining both concepts into a single estimate of risk.

For example, a speeding driver encounters water on the road resulting in the loss of control of the car. Here the water on the road is the external coeffector which, in combination with the high speed, leads to the hazardous state (the loss of control) or the incident. This does not necessarily mean that someone is killed or injured, but that there is now a non-zero probability that an accident might occur. Estimation of the accident outcome probability is a separate exercise from estimating the incident probability.

Any attempt to create a taxonomy of risk quickly reveals the width, depth and complexity of the topic. In this paper we confine our focus to physical systems whose deployment has been approved by an authorised technical regulator. We make no comment on the methodology used in the design and construction of such systems.

Implicit in all that follows is the assumption that a properly endorsed system boundary has been established and that system hazards have been identified by a Hazard Analysis (HA) process. We further assume that each of the events in the incident sequence can be identified and understood *a priori* to the occurrence of an accident. The importance of the system boundary and the hazard analysis are discussed later.

3.2 Risk Distributions and Uncertainty

A particular problem associated with the current approach to HRA is the uncertainty associated with the estimation of risk. There are two sorts of randomness that lead to uncertainty about the interpretation of risk estimates, i.e., stochastic variability resulting from the use of known statistical processes (such as a Poisson process), and the representation of ignorance about the actual process. These are called *aleatoric* and *epistemological* uncertainty respectively. As the Royal Society report (1992) notes, “both of these [probabilities and consequences] are subject to uncertainties, related to lack of precision in models or random variation”. NASA (2002) also contains a useful discussion of this topic.

In the case of complex defence systems that are either in the acquisition phase or newly fielded, there is likely to be a good deal of uncertainty about risk estimates. Such uncertainty results from the necessarily Bayesian approach to estimating probabilities used in estimating risk. The accommodation of this uncertainty occurs in the wide bounds placed on the distributions of events leading to hazard realisation, and on the estimates of parameters associated with the characterisation of those distributions. Vose (2008, Section 4.3) provides an excellent guide to those choices.

This paper largely uses probability distributions as a way of expressing uncertainty. There are a number of non-probabilistic ways to treat uncertainty, such as sensitivity analyses, ‘what-if’ scenarios, model envelopes and model averaging. Our approach is in line with the philosophy expounded by Sir David Cox (1988), where he says “probability provides a mathematical framework for describing uncontrolled variability and, in a different role, a basis for measuring uncertainty. The philosophical interest and importance of the subject stems from this dual claim to be able to study and analyse random variability and also to be able to come to terms with uncertainty, to recognise its existence, to measure it and to show that advancement of knowledge and vigorous action in the face of uncertainty are possible and rational”.

3.3 Proposed Approach to Calculating Risk

We now propose an alternative approach to the computation of risk which, while not original, could be of interest to practitioners concerned with estimating risk associated with deploying a complex system. The proposal is based on the notion that, while uncertainty about the Likelihood (the Incident probability) is best expressed as a Probability Density Function (PDF) (Edwards and Westcott, 2010), Outcome (the Accident) severities associated with complex systems may also have uncertainty which can be expressed in the form of a statistical distribution. For example, random events leading to an accident are often modelled as a Poisson process. Here the concept of using a statistical model is a simple extension of the idea that accident severities are not deterministic in nature and have a natural variability. For example, an accident described as “multiple loss of life” clearly can take many states ranging from zero to the maximum number possible. In this case a careful consideration of the physical situation would provide an empirical distribution of deaths in the event that a particular accident has occurred. If only a single death was possible the accident PDF would reduce to a binary

[1] Incident - An occurrence, other than an accident, associated with the operation of a system which affects or could affect the safety of operation. (ICAO Annex 13)

[2] Coeffector - External states or events that may contribute to an accident. (DEF(AUST)5679).

distribution characterised by a single probability. In fact most accident severities (such as the cost of legal liability) possess intrinsic variability, which can be described by some form of statistical distribution.

3.3.1 Incident Model

There is a simple stochastic model of incidents and their consequences that provides a helpful interpretation of the Likelihood/Consequence description of risk. Suppose that incidents occur at random times $t_1 < t_2 < \dots$ in some interval of interest [0,*T*], and that with the incident at t_i there is associated a random consequence X_i. This forms a *marked point process*, where the times are the point process and the consequences are the marks. The times are often assumed to be a Poisson process but this is not conceptually necessary. If the consequences add, so that the total consequence over [0,*T*] is the sum of the relevant X_i, we have

$$\text{Consequence}(T) = \sum_{i=1}^{N(T)} X_i$$

where $N(T)$ is the number of incidents in [0,*T*]. Suppose that the *incident rate* (expected number per unit time) is constant over [0,*T*] , and is equal to λ, say. If the incident times and the consequences are statistically independent (and in some other cases as well), and the consequences all have the same expected value, μ say, the *expected consequence* over $N(T)$ is $\lambda T.\mu$. So the *expected consequence per unit time* is $\lambda.\mu$, a product of a Frequency/Likelihood λ and a 'consequence' μ. If the accidents are rare, so that effectively there will be at most one incident in [0,*T*], then $N(T)$ is effectively a 0-1 random variable and so $E\{N(T)\} = \lambda T \square \Pr\{N(T)=1\}$. Thus the expected consequence over [0,*T*] can be interpreted as the probability of an incident times the expected consequence, which is another commonly used calculation method for risk.

This is of course very basic. In practice, 'time' should be replaced by 'exposure'. Incidents will often have associated random characteristics which influence the consequence (vehicle speed affects the level of damage or number of deaths), so the marks can be multivariate. For large *T* the incident rate and mean consequence are likely to vary. For serious incidents, an occurrence might lead to system modifications which might change the model parameters. Nonetheless, we feel the basic conceptual model of a marked point process provides a rationale for the product combination of Likelihood and Consequence and can be helpful in thinking about other matters in the field of Risk.

3.3.2 Incident Probability

A statistical method for estimating the probability of an incident was described in Edwards and Westcott (2010) where we noted (inter alia):

'Since we are simulating probabilities, we need a distribution on [0, 1] or a subinterval thereof. Two obvious choices are the triangular and beta distributions (Johnson et al, 1995; Vose, 2008).'

Both in the earlier paper and in this one we have chosen in the example to use a triangular distribution determined by the triple (a, b, p_{nom}), where a and b are initially 0.0 and 1.0 respectively and where p_{nom} is the nominal event probability. This provides a very conservative, or 'precautionary', estimate of the distribution and reflects the degree of ignorance about the event process. Clearly, if better information about the particular event was available then the bounds on the distribution could be tightened and/or the value of p_{nom} modified.

3.3.3 Possible Refinements to the Incident Sequence Model

We mentioned earlier that the mark associated with an incident could be multivariate, with the consequence (one of the variables) being affected by the values of the other variables. How might this be realised in the simulation approach?

One possibility is to expand the incident sequence to allow a degree of severity at each step in the incident sequence. In keeping with the spirit of our approach, this will in general be a random variable. Formally, the outcome at each step of the incident sequence is expanded from a binary yes/no result to a real number, *y* say, which we call the *severity level* of the step. Note that in DEF(AUST)5679 (2008) 'severity level' is applied to the impact of any resulting accident, e.g., multiple loss of life. The value of *y* could be a mixture of a binary and continuous response. For example, if the step is whether a door is left open, possible outcomes are 0 (the door is fully open), 1 (the door is closed) or a real number between 0 and 1 (the proportion of the doorway blocked by the door).

What might be the effect of *y* in the model? The simplest possibility is that it does not affect the occurrence of any of the subsequent steps in the incident sequence, so it just gets carried through to the end as an extra output variable. It then, however, affects the consequence. This could be a functional dependence if we use a deterministic consequence model, or as a parameter of the distribution of consequences if the consequence is random (for example, the mean of a Poisson distribution for the number of deaths is a function of *y*). In more complex models, it might affect the occurrence of subsequent steps, and perhaps their severity levels as well. Clearly this could rapidly lead to a very complex and interdependent probability model.

To simulate such a model, a distribution for *y* must be chosen at each step where a severity level is included. If the probability of occurrence of a step and its severity level are affected by severity levels from previous steps, appropriate conditional probabilities and distributions are required. In the spirit of this paper, these should reflect uncertainty in the forms of these functions. Again, the complexity of such a model could rise very rapidly. The issues involved arise also in fields such as Bayesian Belief Networks (BBN), where transition probabilities between nodes in the network are treated as random variables. One simplification used there is to dichotomise continuous responses, which has its own set of issues (see for example Kuhnert and Hayes, 2010).

We give another example of a severity level in Section 6, which is an expanded discussion of the accident sequence model used in Edwards and Westcott (2010).

3.3.4 Accident Model

The accident severity in general is also a random variable. Here it is more obvious that outcomes are the result of some inherently random process, but we should also allow for uncertainty in the specification of this process. One simple way of representing these sources of randomness is to assume that the severity has a probability distribution $F(x;\theta)$ that depends on a (possibly vector) parameter θ and then take θ to also be random, in the same way as we made the hazard realization parameter random on 3.4.1.

As an example, which we use later in the paper, the number of deaths in an accident might reasonably be assumed to be a Poisson-distributed random variable with mean θ. To determine θ, one possibility is to decide on a plausible value for the probability q of no deaths and then take $\theta = -\log(q)$, because for the Poisson distribution $q = e^{-\theta}$. To deal with uncertainty in θ, one possibility is to choose q from a distribution on [0,1] (or a subset thereof) as in 6.1.1.

Note that there will be situations, particularly when safety cases are being made, where there is only one consequence of interest. In such situations, the 'outcome severity' is effectively a 0-1 variable; does the outcome occur or not? In such cases, the consequence can be regarded as the last step in an accident sequence in which there is already a 0-1 variable for each step in the sequence. Such situations are covered by the discussion in Edwards and Westcott (2010); the present paper is a refinement where the yes-no consequence of the final step is expanded to a more general set of possible values.

3.3.5 Combination of Incident and Accident Probabilities

Once the idea that both the Outcome and Likelihood components of Risk can be described statistically is accepted, it is a small conceptual step to the idea that a Risk Function (RF) can be generated by 'combining' the two statistical distributions. Such a combination is similar to that discussed by Brooker (2004) in the context of aviation safety, though his use of the word 'convolution' for this combination is non-standard. While the derivation of such a combination may not yield to theoretical analysis, it can be readily generated by computer simulation techniques similar to those described below. Importantly, the RF is not merely a distribution of probabilities, but rather a distribution that includes a measure of consequence. This is consistent with the notional interpretation of a conventional qualitative HRM (likelihood combined with outcome) and is necessary if both dimensions of risks are to be used with ranking risks in a hazard mitigation programme.

If the probability that an incident occurs is p and the accident severity is X, the risk R will be calculated as $R{=}p.X$, where both p and X are random variables, assumed independent. In principle the distribution of R can be calculated theoretically but the answer is unlikely to be mathematically tractable. An alternative approach is to use simulation, extending the approach in Edwards and Westcott (2010) which effectively looked at simulating the distribution of p alone. In the simplest situation, this simulation would be augmented with a simulation of values of X from an appropriate $F(x;\theta)$, with an added simulation for the values of θ as discussed above.

3.3.6 Risk Metrics

In providing a quantitative assessment of the risk of a particular outcome we need to provide some summary metrics based on the RF. Following NASA (2002) we suggest in the example below a number of arbitrary, but potentially useful metrics.

3.4 Comparing Risks and Safety Targets

3.4.1 Risks

In principle risks can be compared and ranked by comparing metrics from the derived RFs. However such a comparison only has meaning if the measure of the outcomes is the same, e.g., deaths or financial loss. If different sorts of outcomes can be related by a statistically valid calibration then it becomes possible to compare the risk of different outcomes. For example, the cost of death or injury of personnel could be related via actuarial tables.

Noting that the RF contains all the uncertainties built into the incident and accident PDFs comparison of the RFs becomes a relatively straightforward process. Metrics which allow satisfactory comparisons to be made were provided in our previous paper and are also discussed later in this paper.

3.4.2 Safety Targets

The availability of RFs allows various risks to be compared in quantitative terms, thus assisting the process of resource allocation following mitigation activities. However there is often a requirement to determine if the probability of death or injury is below a specified value. These are typically called Safety Targets.

Safety targets are usually couched in terms of reduced probabilities of death or injury, and are often required by safety authorities in response to the demands of a particular safety standard. Comparison of single value probabilities with a probability level in a safety standard has little meaning unless the uncertainty of this probability is also considered. The methodology discussed above (see also Edwards and Westcott, 2010) allows the computation of the estimated probability of zero deaths and hence provides a mechanism for assessing whether a particular safety target has been met. This approach is illustrated in the example below.

4 Residual Risk

4.1 Background

The first comprehensive statement on Residual Risk was provided by Asquith (1949):

'..that a computation must be made in which a quantum of risk is placed on one scale and the sacrifice, whether in money, time or trouble, involved in the measures necessary to avert the risk is placed on the other. If it be shown that there is a gross disproportion between them, the risk being insignificant in relation to the sacrifice, the person upon whom the duty is laid discharges the burden of proving that compliance was not reasonably practical.'

Another definition is provided at http://www.businessdictionary.com/definition/residual-risk.html

'Exposure to loss remaining after other known risks have been countered, factored in, or eliminated.'

At paragraph 2.27 AS/NZS ISO 31000 (2009) notes that:

'residual risk (2.1) remaining after risk treatment (2.25)'

and that

'Residual risk can contain unidentified risk.

Residual risk can also be known as "retained risk".'

The concept of residual risk is often discussed in safety standards. Such standards usually demand the provision of qualitative estimates of residual risk as part of the safety assurance process. The lack of a measure of residual risk makes comparisons of the effectiveness of mitigations problematic and hence leaves an inherent uncertainty about the effectiveness of a safety program based on such qualitative estimates.

4.2 Safe Residual Risk

Estimating safe residual risk associated with the deployment of a complex system is focused on estimating safety risk associated with the behaviour of the system in its operating environment. Computation of SRR follows from the assumption that, while possible outcomes identified by the HA remain after system deployment; the estimated probability of such outcomes will have been reduced by design or other mitigation strategies. This process is likely to be repeated several times until the SRR is reduced to a level acceptable to the appropriate Technical Regulatory Authority (TRA). Reductions in residual risk are likely to be asymptotically diminishing while associated costs increase, suggesting that the asymptotic limit is similar to the As Low as Reasonably Practical (ALARP) value. As such the final risk function represents a statement of the system residual risk as far as a particular accident is concerned.

5 Estimating the Effect of Mitigation Activities

5.1 The Stress-Strength Relationship of a Safety Program

Stress-Strength models have found application in a wide diversity of domains, with Kotz *et al* (2003) providing a comprehensive review of the topic. In its simplest form the model is described as $P(X < Y)$, where X is the stress and Y is strength of a system. We now suggest that the Stress-Strength concept be applied to the competing demands of a safety program, where "strength" is represented by the effectiveness of the various mitigation techniques available to the safety engineer, while the cost of such mitigation represents the "stress". The following is provided as an illustration of how such a model might be used.

Consideration of both the Stress and Strength of a particular mitigation has the potential to provide the TRA and Program Managers with information to guide the choice of mitigation technique. Noting that it is likely that the cost of retrospectively implementing a strong safety mitigation will be higher than one of lower strength, it is important that the strength of implementing various mitigation strategies be calibrated with the associated cost and be able to be related to the requirements of risk-based safety standards.

5.2 Cost of Hazard Mitigation Strategy

In practise the choice of mitigation strategy depends on the demands of accident probability reduction made by risk based safety standards such as MIL-STD-882C. While the mitigations are discussed qualitatively in these standards they do not provide guidance as to the quantitative reduction in accident probability to be achieved from various mitigation strategies. For example, a relatively weak mitigation may not reduce the incident probability to an acceptable level, thus requiring a stronger (and usually more expensive) retrospective mitigation to be implemented.

Relating the reduction in incident probability to the mitigation Strength is thus required in order to provide a quantitative estimate of that probability. Table 1 provides *suggested* guidance on the degree of reduction in incident realisation probability following particular hazard mitigation strategies. Entries in the table are the magnitude of the reduced *nominal event probability* provided in response to the application of a particular hazard mitigation strategy. Thus for example, the use of procedural mitigations provides only one order of magnitude reduction in the probability that a *particular event* leading to the incident will occur. In making this comparison we are aware of debate over the effectiveness of implementing procedural mitigations which require operator intervention to mitigate a hazard. Sandom (2007) provides a useful discussion of the issue. Despite the debate we believe that for complex systems procedural solutions currently provide the lowest strength hazard mitigation.

Hazard Mitigation Strategy	**Hazard Mitigation Strength (reduction in event probability)**	**Safety Program Stress (cost)**
Procedural	10	Low
Output Checking	100	Moderate
Mechanical Interlocks	1000	Medium
Module Redesign	10000	High
Formal System Redesign	100000	Very High

Table 1. Suggested Stress and Strength of Hazard Mitigation Strategies

Although in this paper we do not comment generally on system design and construction issues, we note that the most effective reduction in incident probability is achieved through integration of safety requirements with other system requirements and subsequent formal system design prior to system deployment. Here potential hazards identified during the system hazard analysis are mitigated by careful design prior to system development and testing. This process is well documented in DEF(AUST)5679 (2008).

Balancing the choice of mitigation strategy is the need to minimise safety program stress. Table 1 also provides guidance on the degree of program stress induced by the various strategies. Calibration of stress with cost depends on issues particular to a specific program and should be considered as part of any development of system requirements.

6 Example

The example is based on a missile system developed for the RAN and is discussed in Edwards and Westcott (2010). The accident sequence was developed by an independent safety contractor and focused on the injury or death of personnel involved in removing an expended missile canister from the launch system. The process is shown in Figure 1. Probability values, including conditional probabilities, assigned to each transition by the safety contractor are also shown.

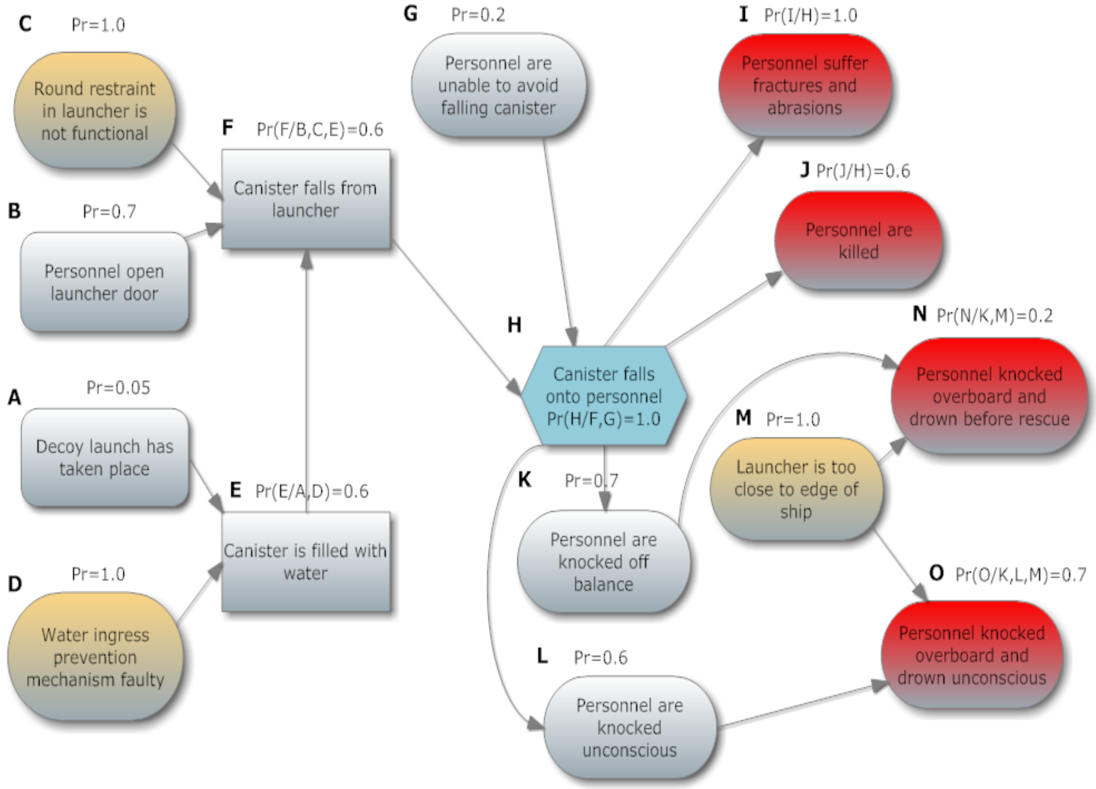

C Pr=1.0
Round restraint in launcher is not functional
G Pr=0.2
Personnel are unable to avoid falling canister
I Pr(I/H)=1.0
Personnel suffer fractures and abrasions
F Pr(F/B,C,E)=0.6
Canister falls from launcher
B Pr=0.7
Personnel open launcher door
J Pr(J/H)=0.6
Personnel are killed
H
Canister falls onto personnel Pr(H/F,G)=1.0
N Pr(N/K,M)=0.2
Personnel knocked overboard and drown before rescue
A Pr=0.05
Decoy launch has taken place
M Pr=1.0
Launcher is too close to edge of ship
E Pr(E/A,D)=0.6
Canister is filled with water
K Pr=0.7
Personnel are knocked off balance
D Pr=1.0
Water ingress prevention mechanism faulty
O Pr(O/K,L,M)=0.7
Personnel knocked overboard and drown unconscious
L Pr=0.6
Personnel are knocked unconscious

Figure 1. Example Accident Sequence

For purposes of illustration we now focus on Outcome J i.e., Personnel are killed.

6.1 Derivation of the Density Functions

6.1.1 Incident Sequence

In accordance with our previous paper (Edwards and Westcott, 2010) the derived PDF of the probability of the incident sequence leading to death from a falling canister is shown in Fig. 2 below. Note that this sequence not include event J, as that accident process is now modelled separately, and that each probability value provided by the independent safety contractor is used as the modal value of a triangular distribution on [0,1]. Alternatives to this choice are discussed in Edwards and Westcott (2010).

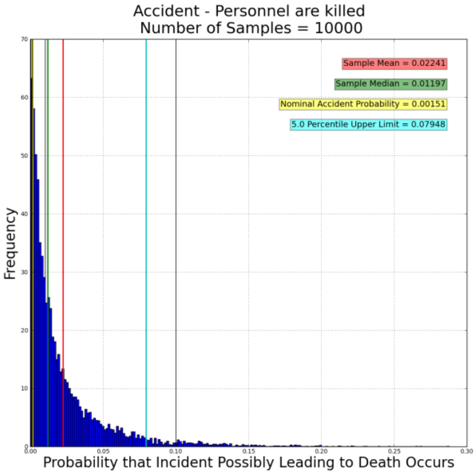


Figure 2. PDF of Incident Probability

6.1.2 Accident Severity

We now turn to the need to estimate the associated accident severity PDF. Clearly the number of personnel killed could vary from zero to the maximum number involved in opening the launcher door. The outcome severity is modelled by a Poisson distribution.

As discussed in 3.4.2., if q is the independent contractor's estimate that no personnel would be killed by the falling canister, then the mean θ of the Poisson distribution is $-\log(q)$. From Figure 1 it is seen that in the case of Event J, $1-q=0.6$, so $\theta=0.9163$. A simulation of the resulting Outcome PDF is shown in Figure 3. For fixed θ these values are of course analytically known, but if q (and hence θ) were also taken as random, to represent the uncertainty in the nominal value, simulation would almost certainly be required. An example of this is discussed later.

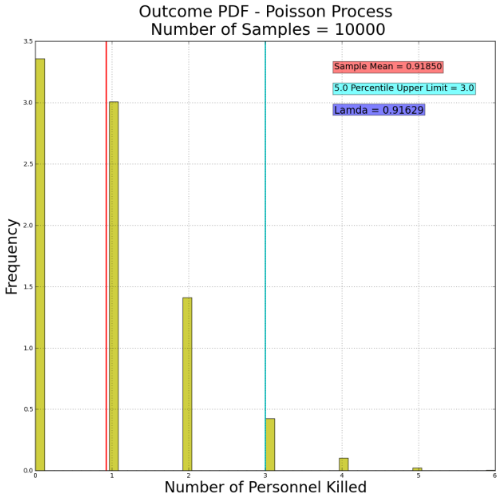


Figure 3. PDF of Accident Outcome

6.1.3 Risk Function

In order to estimate the Risk Function we now combine the Incident and Accident models. Figure 4 shows the resultant Risk Function of the risk that personnel are killed by a falling canister. We observe that upper 5th percentile value of the distribution is 0.09236. We stress that this value is not a probability but is a risk value, since the actual number of personnel killed are combined with the accident probability in the simulation. Our contention is that this is a possible metric for ranking the risk of personnel being killed with other risks identified in the scenario shown in Figure1.

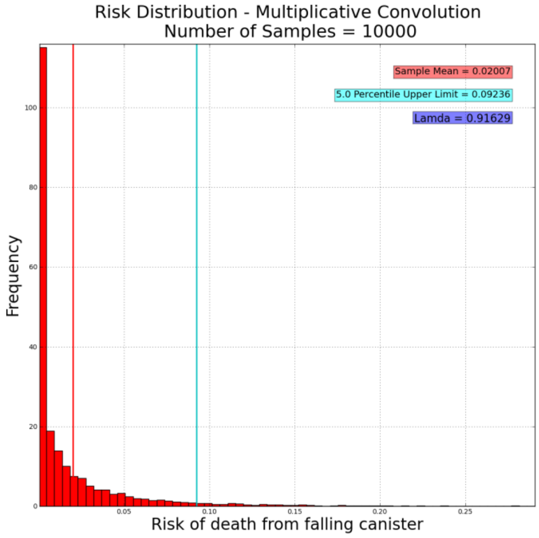


Figure 4. Risk of Death from Falling Canister

6.2 Comparing the Risks

As noted above comparison of the risk of the outcomes can be made if the measure of the outcomes is the same. So it would seem reasonable that we could compare the risks of Events J, N and O shown in Figure 1, as each outcome involves the loss of life. We thus obtain for 10,000 simulations the following table:

Risk Comparison Metric	**Event J**	**Event N**	**Event O**
Mean	0.02007	0.00298	0.01448
Median	0.00356	0.00000	0.00359
Upper 5th Percentile Limit	0.09236	0.01798	0.06489

Table 2. Comparison of Risks for Events J, N and O

From the above table it can be seen that the risk of Event J is significantly greater than Event O. Assuming Figure 1 represents the physical reality of the system, Event J should be the first focus for mitigation.

6.3 Residual Risk Following Hazard Mitigation

We now focus on Event J and examine possible mitigation strategies. For illustrative purposes we assume that the system developer and TRA agree that the installation of a mechanical interlock on the round restraint mechanism is both feasible and affordable. Consistent with the stress/strength model in Table 1 the nominal probability of Event F is reduced to 0.0006 and the triangular distribution of Event F is now characterized by the triple (0.0, 0.1, 0.0006). Here the upper 5th percentile value of the Risk Distribution is reduced to 0.00647. Repeating the process for events N and O with the hardware interlock in place yields the following:

Risk Comparison Metric	**Event J**	**Event N**	**Event O**
Mean	0.00130	0.00019	0.00092
Median	0.00014	0.00000	0.00015
Upper 5th Percentile Limit	0.00647	0.00095	0.00414

Table 3. Comparison of Risks for Events J, N and O with Hardware Interlock

The result of the installation of a hardware interlock is that the overall risk of at least one death from any of the events has been reduced by at least one order of magnitude. Additionally, Event J continues to carry the greatest risk of causing death.

6.4 Safety Targets

The probability of at least one person being killed by a falling canister (Event J) is now computed as in Edwards and Westcott (2010); see the discussion in 3.4.2 on one consequence of interest. Table 4 shows the change in the probability of at least one death before and after the mitigation provided by the hardware interlock.

	No Hardware Interlock	Hardware Interlock
Mean	0.01212	0.00076
Median	0.00581	0.00029
Upper 5th Percentile Limit	0.04533	0.00316

Table 4. Event J - Safety Target Probabilities

In this case the probability of at least one death has been reduced by at least an order of magnitude. Acceptability of such a reduction would depend on the target demanded by a particular standard. For example, in the context of MIL-STD-882C, the mitigation would likely be accepted by the TRA as an 'Unlikely' probability, i.e., "less than 10^{-3} and greater than 10^{-6}" for a defined period of deployment.

6.5 Modified Incident Model

As suggested above an enhancement to the incident model could be achieved by incorporating a degree of severity for each event in the incident sequence. For illustrative purposes we now focus on Event E "Canister is Filled with Water", and assume that rather than either being filled or empty, the canister can be partially filled. This has a practical interpretation as it seems intuitive that a 10% filled canister is much less likely to cause a serious accident than a 90% filled canister in the event that the canister were to fall on a person.

In order to model Event E we now assume that the degree to which the canister is filled with water is a random variable represented by a rectangular distribution between [0.0 , 1.0]. For each simulation we now sample from the distribution to obtain the degree to which the canister is filled. We call this variable p_W. Its value is now used to determine the value of θ (and hence the accident Poisson distribution as discussed in 6.1.2) by the following linear relationship

$$\theta = \theta_0 + \left(\theta_1 - \theta_0\right) p_W$$

where θ_0 is the value of θ for an empty canister and θ_1 is the value of θ for a full canister. In this example θ_0 has been set to 0.01 as even an empty canister possesses some risk, while θ_1 is set as 1.0. The resulting RF is shown in Figure 5.

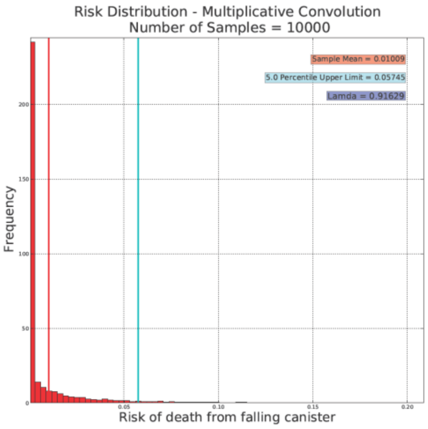


Figure 5. Risk of Death from Falling Canister – Canister Randomly filled with Water

In this example the result of assuming a uniform distribution for the degree to which the canister was filled with water is a 50% reduction in the 5% upper percentile risk value. In practise the relationship between the percentage of water in a canister and the accident model would need to be validated by the systems designers and operators.

7 Discussion

7.1 System Boundary

The importance and associated difficulty of defining the system boundary is discussed in Edwards (2008). In order to avoid future programmatic arguments about the scope of a safety program it is important that the system boundary be established at the outset of the safety program. This can be a deceptively difficult task, but is critical in establishing a holistic approach to understanding system risk.

The identification of interfaces to external systems on the system boundary is of particular importance because data flows across the interface may have the propensity to contribute to an incident sequence. For example, consider a flight control system that depends on airspeed data from an external sensor for correct operation. Such a system might require airspeed data of a higher resolution than that provided by a currently installed wind sensor in order to operate safely. If the wind sensor was considered to be outside the system boundary it is possible that the need to modify the pitot static system, or to ensure the flight control system operated safely in the presence of aberrant data, might be conveniently ignored (e.g. for cost and/or schedule reasons) during system development. The end result could then be that an incident occurs after the system has been accepted into service.

The establishment of an appropriate system boundary thus requires careful consideration, should include consideration of system integration issues, and should result from a consensus between the various regulatory authorities. In developing an agreed system boundary care needs to be taken to ensure that technical and not schedule and cost issues drive the process.

7.2 Hazard Analysis

The importance of conducting a planned and considered HA cannot be overemphasised. This process aims at identifying hazards at the system boundary. Cant and Mahony (2011) note:

> *'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.'*

The strategic importance of conducting a HA is well-documented in Issue 2 of DEF(AUST)5679 (2008). There is however no agreed benchmark against which to judge the quality an HA. While various bespoke techniques have evolved over the years there is still no satisfactory way of measuring the completeness or internal consistency of a HA. We suggest that this is an area requiring further study.

7.3 The Role of the Technical Regulator is Assessing Risk

The TRA plays an important role in assessing system risk. Bouleau (2011) notes:

> *'True risk analysis necessarily involves understanding interpretations. That's much more difficult [than a mathematical representation], not least because it is sensitive to the information that is available to concerned social groups, and to their imagination, not in the sense of dreaming or delirium, but in their ability to perceive the field of possibilities.'*

The role of the TRA is to provide the interpretation of risk in a particular context which complements the contribution of the risk analyst. Thus both the analysis and interpretation of a particular risk form the complete assessment of risk provided to the decision maker. In this regard the degree of independence of the TRA within an organisation is of some importance. If the TRA reports to a middle manager there is a danger that the presentation of risk assessments will be reinterpreted before being placed in front of a decision maker higher in the chain of command.

7.4 Statistical Issues

7.4.1 Choice of Risk Metric

Throughout this and our earlier paper (Edwards and Westcott, 2010), we have used the upper 5th percentile of a density function as a conservative measure to characterise accident and incident distributions. It is worthy of note that the choice is deliberately arbitrary and may not suit a particular regulatory regime. Just as electrical engineers in the USA and Australia chose different voltages and frequencies for the distribution of electrical energy, so the various TRAs might want to use different metrics to characterise risk. In the example we have suggested a number of metrics that might be useful. Alternatively, the expression that "on balance the risk is worth taking" could be interpreted to imply that the median value of the RF might be an appropriate metric in some circumstances.

7.4.2 Logical Structure of the Event Sequence

When estimating the probability of an incident it is first necessary to estimate probabilities of individual events leading to that condition and then combine them to provide the final estimate. As noted in Edwards and Westcott (2010),

'Frequently, the accident sequence will contain logic dependencies. So, for example, Event C will depend on Event A AND Event B occurring. In this case issues of estimating conditional probabilities arise as well as questions about the stochastic independence of events.'

It is important that the process of estimating the probability of an incident takes into account any logical structure imposed on the event sequence leading to the incident. This in turn leads to an inescapable requirement to estimate conditional probabilities associated with events. Such a requirement is often used to support a uniquely qualitative approach to estimating risk, an approach which abrogates any implied contract for intellectual responsibility between the safety community and the public.

7.4.3 Probability Elicitation

The use of single value probabilities in risk assessment has recently received much attention. For example in the Nimrod Review (Haddon-Cave, 2009) it is noted (Section 22.43):

'(8) Care should be taken when using quantitative probabilities, i.e. numerical probabilities such as 1 x 10-6 equating to "Remote". 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. QRA is an art not a science. There is no substitute for engineering judgment. As the HSE emphasised to the Review: "Quantitative Risk Assessment has its place, but should never be used as an absolute measure of safety."

(9) Care should be taken when using historical or past statistical data. The fact that something has not happened in the past is no guarantee that it will not happen in the future. Piper Alpha was ostensibly 'safe' on the day before the explosion on this basis. The better approach is to analyse the particular details of a hazard and make a decision on whether it represents a risk which needs to be addressed.'

In making safety claims about complex systems that have not been deployed it is important to ensure that the claim is not based on subjective judgement for which there is little evidence. Sandom (2011) notes:

'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.'

In a more general vein Cant and Mahony (2011) note:

'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".'

In response to the above we note that we too have been concerned about the current state of quantitative risk assessment techniques - as exemplified for example in MIL-STD-882C. However, contrary to the bulk of opinion in this area we believe that sound application of probability elicitation and quantitative statistical techniques in the limited field of deployed complex physical systems, complemented by associated qualitative arguments and the provision of contextual interpretation by a TRA, offers the potential to enhance the understanding of the nature of the risk and its assessment for practical systems.

There are well-documented pitfalls in the process of probability elicitation. These include issues such as overconfidence, representativeness, anchoring, affect, hindsight bias and linguistic uncertainty. The importance of elicitation techniques has been discussed previously in Edwards and Westcott (2010). Kuhnert et al (2009) provides a useful description of issues associated with the elicitation of probabilities in a number of different contexts, and additional references.

We again emphasise the importance of the provision of traceable and well-documented arguments to support the use of probabilities in developing safety arguments, particularly for complex systems under development or those modified to operate in a new environment.

7.4.4 Uncertainty of Risk Estimates

In this paper we have commented on two sources of uncertainty; uncertainty in the model and uncertainty in the data. We have concentrated on the former and shown one way of representing this uncertainty, namely making the model parameters random variables. In a recent comprehensive report, Hayes (2011) gives further discussion of these issues, but also mentions two other types of uncertainty; decision uncertainty and linguistic uncertainty. The former refers to policy analysis after risks have been estimated. The latter comes from imprecise and subjective use of language. One specific example is the use of terms such as 'likely' or 'improbable' in qualitative risk assessment exercises such as traditional risk matrices. These issues were discussed in Jarrett (2008) and Edwards *et al* (2009).

7.5 Implications for Safety Standards Based on Hazard Risk Assessment

In Edwards and Westcott (2010) we noted that:

'The use of HRA to assess system safety is widespread throughout industry and government and there is a natural reluctance to question the efficacy of processes and standards that have become the accepted norm. This is because such standards impose a lesser burden of diligence and, it must be said at lower cost, than might otherwise be the case. Thus while governments continue to accept an unquantified residual risk resulting from HRA safety assessments, industry has a vested interest in maintaining the status quo. This is particularly the case with respect to MIL-STD-882, whose development and use continues to be supported by defence industry and government as partners in an unhealthy symbiosis.'

The concept of a HRM was developed in response to a need to provide a systematic approach to the provision of assurance that complex systems were safe to use. MIL-STD-882, which was first issued in the 1970s, epitomises this approach. It now appears that the concept of a HRM was a product of a 'qualitative' approach to the problem of system safety developed in the 1970s and is an artefact that has outlived its usefulness. We suggest that a more rigorous mathematical modelling approach to the problem is required and that the example used in this paper illustrates one possible approach. We do not claim it is the only approach that might be used and have previously suggested that the approach of Jarrett (2008) in constructing a modified HRM has merit.

In commenting on the use of MIL-STD-882C to assess system risk we do not suggest that the standard be discarded. Rather we believe that the kernel arguments used to assess system risk need to be upgraded and then integrated with the various deliverables required by the standard. Such an upgrade should provide a manageable process allowing for acceptance of a revised standard by both industry and governments.

A common argument for the continued use of a conventional HRM in the safety domain is that it provides a convenient way to present a summary of system risk to senior management. We do not agree with this argument and assert that a conventional HRM can too easily provide a facile summary of an often complex analysis, which in turn has the propensity to misdirect the allocation of mitigation resources.

8 Towards a Practical Application of Risk Functions

8.1 Process Outline

Before risk functions can be applied to guide resource allocation for risk mitigation some experimental testing of the process will be required. In particular the process of choosing the risk metric and the calibration of the stress-strength hazard mitigation strategies will need to be exercised and the results subjected to a thorough examination. Nevertheless an outline of a practical application methodology is clear. The steps are:

a. Define the system boundary;
b. Conduct a thorough hazard analysis that identifies potential hazards, particularly those realised on the system boundary;
c. Model the event sequences leading to incidents;
d. Elicit probabilities for the incident sequences;
e. Model accidents that potentially can follow identified incidents;
f. Combine the probability density functions for the incidents and the accidents;
g. Rank the identified risks;
h. Determine required mitigation strategies; and
i. Determine if the residual risk is acceptable and if safety targets are met.

Clearly this process must involve some sort of feedback loop when problems in the process are detected. However, we have chosen not to suggest a process flow chart at this stage because the methodology has not been tested in practice.

8.2 Stress-Strength Model for Safety Programs

The stress-strength hazard mitigation model suggested earlier provides a method for relating the impact of hazard mitigation to programmatic issues of interest to managers and other decision makers. By developing a linkage between hazard mitigation and programmatic stress the model offers an opportunity to develop an understanding between the safety community and the builders and users of complex systems. Clearly this concept requires further development and would necessarily need to be the subject of further research and discussion between both communities.

8.3 Required Disciplines

Membership of safety teams will vary depending on system and context, but experience suggests that such teams need to be multidisciplinary and include (inter alia) members competent in systems analysis and statistical modelling together with members experienced in conducting hazard analyses. Both NASA (2002) and DEF(AUST) 5679 (2008) discuss membership of safety teams in general terms but do not specify the requirement for particular skill sets.

9 Conclusion

We conclude that the current approaches to assessing and managing risk in the domain of complex system safety are both process intensive and inherently unsound. We believe that careful application of statistical techniques offers an avenue for advancing the discipline and provide a basis for further development of risk based system safety standards.

Acknowledgements

The authors would like to thank Dr Neale Fulton for advice on the application of Stress-Strength models and for other constructive comments.

10 References

Asquith, Lord Justice (1949): Edwards v National Coal Board, 1 KB 704; 1949 1 All ER 743 p712 and p747, a case on the interpretation of S 102 (8) of the Coal Mines Act, 1911.

AS/NZS ISO 31000 (2009) Risk Management - Principles and Guidelines. Canberra: Standards Australia.

Bouleau, N. (2011): Finding the true meaning of risk. New Scientist 210 (2818): 30-31

Brooker, P. (2004): Airborne separation assurance systems: towards a work program to prove safety. Safety Science 42: 723 – 754.

Cant, T. and Mahoney B. (2011): Urgent operational requirements: Impact on the safety case. In T. Cant (ed.) Conferences in Research and Practice in Information Technology (CRPIT) 133: Proceedings of the Australian System Safety Conference, 25-27 May 2011, Melbourne, Australia.

Cox, D.R. (1988): Of human numbers and human needs. Address to 22nd Convocation of Indian Statistical Institute, 14 January 1988. Extract reprinted in *A Bouquet of Remembrances*, Indian Statistical Institute, May 2008.

http://www.isical.ac.in/~isiaa/ISIAA_Bouquet-of-Remembrances.pdf (accessed 29 August 2012)

Cox, L.A. (2008): What's wrong with risk matrices? Risk Analysis 28: 497-512.

DEF(AUST)5679 (2008): Commonwealth of Australia Australian Defence Standard, Safety Engineering for Defence Systems, Issue 2. At http://www.defence.gov.au/dmo/dmo/function.cfm?function_id=60 (accessed 4 January 2012)

Edwards, C.B.H. (2008) The role of the evaluator inAustralian Defence Standard DEF(AUST)5679. In: T. Cant (ed.) Proc. Thirteenth Australian Conference on Safety-Related Programmable Systems (SCS 2008), Canberra, Australia. CRPIT, **100**. pp. 27-35

Edwards, C.B.H., Westcott, M. and Fulton, N.F. (2009): The application of hazard risk assessment in defence safety standards. In: A. Tuffley (ed.) Proceedings of Improving Systems and Software Engineering Conference (ISSEC), Canberra, August 2009, 135-146. ISBN 978-0-9807680-0-8.

Edwards, C.B.H. and Westcott, M. (2010): Estimating accident likelihood. In: A. Tuffley (ed.) Proceedings of Improving Systems and Software Engineering Conference (ISSEC), Brisbane, August 2010, 41-60. ISBN: 978-0-9807680-1-5

Fan, D-Y. (1991): The distribution of the product of independent beta variables. Comm. in Statistics – Theory and Methods 20: 4043 – 4052.

Haddon-Cave, C. (2009): The Nimrod Review. An independent review into the broader issues surrounding the loss of the RAF Nimrod MR2 Aircraft XV230 in Afghanistan in 2006. London: The Stationery Office. At http://www.official-documents.gov.uk/document/hc0809/hc10/1025/1025.asp (accessed 4 January 2012)

Hayes, K.R. (2011): Uncertainty and uncertainty analysis methods. CMIS report EP102467. At http://www.acera.unimelb.edu.au/materials/endorsed/0705a_final-report.pdf (accessed 21 December 2011)

ICAO - International Standards and Recommended Practices, Aircraft Accident and Incident Investigation. Annex 13 To the Convention on International Civil Aviation. At http://www.iprr.org/manuals/Annex13.html (accessed 21 December 2011)

Jarrett, R. (2008) Developing a quantitative and verifiable approach to risk assessment. CSIRO Presentation on Risk, August 2008

Kotz, S., Lumeiskii, Y. and Pensky, M. (2003): The Stress-Strength Model and Its Generalizations. Theory and Applications. New Jersey: World Scientific, Inc.

Kuhnert, P.M., Hayes, K., Martin, T.G., McBride, M.F. (2009): Expert opinion in statistical models. In R.S. Anderssen, R.D. Braddock and L.T.H. Newham (eds) 18th World IMACS Congress and MODSIM09 International Congress on Modelling and Simulation Society of Australia and New Zealand and International Association for Mathematics and Computers in Simulation, July 2009, pp. 4262-4268. ISBN: 978-0-9758400-7-8. At http://www.mssanz.org.au/modsim09/J2/kuhnert_J2.pdf (accessed 4 January 2012)

Kuhnert, P.M. and Hayes, K. (2010): How believable is your BBN? In R.S. Anderssen,, R.D. Braddock and L.T.H. Newham (eds) 18th World IMACS Congress and MODSIM09 International Congress on Modelling and Simulation Society of Australia and New Zealand and International Association for Mathematics and Computers in Simulation, July 2009, pp. 4319-4325. ISBN: 978-0-9758400-7-8. At http://www.mssanz.org.au/modsim09/J3/kuhnert_J3.pdf (accessed 4 January 2012)

Melchers, R.E. (2001): On the ALARP approach to risk management. Reliability Engineering and System Safety 71: 201-208.

MIL-STD-882C (1993) Department of Defense, System Safety Program Requirements, ASMC Number F6861. At http://www.everyspec.com/MIL-STD/MIL-STD+(0800+-+0899)/MIL_STD_882C_965/ (accessed 4 January 2012)

NASA (2002): Probabilistic Risk Assessment Procedure Guide for NASA Managers and Practitioners, V1.1. At www.hq.**nasa**.gov/office/codeq/doctree/pra**guide**.pdf (accessed 4 January 2012).

Royal Society (1992): Risk, Analysis, Perception and Management. Report of a Royal Society Study Group. London: The Royal Society

Sandom, C. (2007): Success and failure: human as hero – human as hazard. In T. Cant (ed.) Conferences in Research and Practice in Information Technology (CRPIT), 57: Proceedings of the 12th Australian Conference on Safety Related Programmable Systems, August 2007, Adelaide, Australia

Sandom, C. (2011): Safety assurance: fact or fiction? In T. Cant (ed.) Conferences in Research and Practice in Information Technology (CRPIT), 133: Proceedings of the Australian System Safety Conference, 25-27 May 2011, Melbourne, Australia.

Vick, S. G. (2002): Degrees of Belief; Subjective Probability and Engineering Judgment. Reston, VA: American Society of Civil Engineers Press

Vose, D. (2008): Risk Analysis – A Quantitative Guide, 3rd edition. Chichester: Wiley.

Software and System Safety: Promoting a Questioning Attitude

Terry L. Hardy
Great Circle Analytics, LLC
1238 Race Street
Denver, Colorado, USA

thardy@gcirc.com

Abstract

System safety is an accepted approach to help understand and manage hazards and risks in complex systems in order to prevent accidents. Many different industries use system safety analyses and methods to help reduce the potential for harm to people, property, and the environment. When used correctly, system safety methods can provide tremendous benefits, focusing resources to reduce risk and improve safety in complex systems. Because computing systems are increasingly being used to control critical functions and supply safety decision information, software may directly or indirectly contribute to an accident. Therefore, software must be included as part of an organization's system safety efforts to manage hazards and risks. However, for many organizations, software is not effectively incorporated into the system safety process, and questions are not asked about whether the analyses are appropriate for complex, automated systems. This paper will summarize several accident reports and use those reports to illustrate potential failures in the system safety process with respect to software and computing systems. Lessons learned will be discussed, and some essential questions in software safety will be presented. This discussion is intended to provide insights to help promote a questioning attitude that can improve software safety and system safety efforts.

Keywords: software safety, system safety, lessons learned.

1 Introduction

As more advanced technology and automation are used, transportation systems, energy production systems, medical devices, manufacturing processes, and many other systems continue to increase in complexity. These complex systems create safety risks to their operators and to the communities they serve. System safety is an approach to help manage hazards and risks in complex systems.

System safety is often implemented through a system safety process. A typical system safety process includes the following components:

- Safety planning
- Hazard identification
- Hazard risk assessment and risk decision making
- Risk reduction and hazard controls
- Risk reduction verification
- Hazard tracking, anomaly reporting, and change management

Of particular concern with regard to system safety are the risks related to software and computing systems. Software and computing systems may be safety-critical if they:

- can cause a hazard (for example, if a software command or automated system can inadvertently create the potential for harm),
- control a hazard (for example, if software is needed to prevent a mishap),
- are used in critical calculations or analyses (for example, output from models and simulations),
- are used to test critical systems.

Software includes computer programs, procedures, scripts, rules, and associated documentation and data pertaining to the development and operation of a computer system. Software can be developed by the organization implementing the system or may be purchased as Commercial Off-The-Shelf (COTS) software. Software safety encompasses not just the software but also the computing system. A computing system includes the software and supporting hardware, sensors, effectors, humans who interact with the system, and data necessary for successful operation. Examples of computing systems include Programmable Logic Controllers (PLC) and Supervisory Control and Data Acquisition (SCADA) systems.

In spite of the fact that software is such an important part of complex systems, the analysis of hazards and risks from software has been inconsistent across industries. Hazard and safety analyses have historically been hardware-focused. Therefore, many analysts may not understand how to incorporate software into their system hazard analyses, and evaluators of those analyses may not understand what should be assessed. Organizations may be focused on compliance to regulations, which often do not address software, and therefore those organizations may not properly assess or mitigate software-related risks. As a result, organizations need to increase the attention given to addressing the potential for hazards related to software and computing systems.

2 Lessons Learned: System Safety Process Failures

Although the system safety process is an accepted approach to reducing risk in complex systems, there are a number of ways this process can fail to prevent an accident, especially in systems where software and computing systems are used. The sections that follow present potential failures in implementation of the system safety process, based on review of hundreds of software-related accidents and incidents (Hardy 2012). These sections will use findings from accident reports to provide lessons learned on those process failures. Note that in discussing these accidents this paper does not intend to oversimplify the events and conditions that led to the mishaps. Rarely is there only one identifiable cause leading to the accident. Accidents are usually the result of complex factors that include hardware, software, human interactions, and procedures. The descriptions here are meant to provide examples of where the system safety process failed in some way and to show how software and computing systems can play a role in those accidents. Readers are encouraged to review the full accident and mishap investigation reports to understand the often complex conditions and chain of events that led to each accident discussed here.

2.1 Failing to plan

System safety efforts must be planned, like any other engineering activity, and then that plan must be followed to be effective. Safety planning includes the planning for the management of system safety and emergency planning in the case where something could go wrong. It is not enough for a plan to exist – the plan must also be effectively implemented, updated, and followed.

On October 26, 1992, the London Ambulance Service (LAS) introduced a new computer aided dispatch (CAD) system to automate call taking, resource identification, and resource mobilization tasks. The automation was intended to improve emergency medical services for the city. At the time, the LAS provided ambulance service to 6.8 million people living in a 600 square mile area, making it the largest ambulance service in the world. The LAS received 2000-2500 calls per days, of which 1300-1600 were emergency calls. Just a few hours after the new computer system was introduced problems began to surface. The system was unable to keep track of ambulances and their locations. Multiple ambulances were sent to the same location in some cases. The system could not keep track of duplicate calls. And the system began to generate so many exception messages that the dispatchers became overwhelmed, and calls were lost. As the system became bogged down the LAS was forced to partially switch back to the manual system. Eight days later the computer system quit working and the LAS had to resort to a completely manual operation. Some estimates stated that as many as 46 people died as a result of the service failures.

An investigation into the incident found multiple causes to the system failures.

- The vendor chosen to build the system was selected primarily on the basis of price, and the vendor's cost estimates were unreasonably low.
- An unrealistic schedule of 11 months from start of development to deployment was placed on the vendor.
- At the time the system went live there were 81 open, known issues and no load testing had been performed on the system.
- Dispatcher training was inadequate.
- The system did not function well when given invalid or incomplete data on positions and statuses of ambulances.
- The user interface was poorly designed and did not respond properly to incorrect user entries.
- A memory leak in a small portion of the code led to the failure of the system eight days after deployment.
- Software requirements were developed without input from key users of the system, including dispatchers and ambulance operators.
- No quality assurance was performed on the software, and configuration management processes were lax.
- The system was overly complex.

The failure of the LAS CAD system was therefore a combination of errors related to safety planning, organizational priorities, safety management, process quality, product design, and product verification (Finkelstein and Dowell 1996).

2.2 Failing to accurately identify what can go wrong

Identifying what can go wrong, also known as hazard identification, is arguably the most important part of the safety analysis effort. One could think of the hazard identification step as defining the problem to be solved. If one does not properly identify the problem then it becomes difficult to assess the risk or postulate solutions. Describing what can go wrong can be difficult in complex systems, and identifying hazards takes persistence and creativity. In addition, complex systems using software can fail in complex ways, and some conditions and environments are difficult to postulate.

On February 11, 2003, an employee of the Southern Clay Plants & Pits in Gonzales, Texas was fatally injured while performing maintenance on a reaction tank. The U.S. Mine Safety and Health Administration (MSHA) determined that the cause of the accident was a failure to close and secure a manual gate valve for a steam line and a failure to place the batch PLC in the stop mode. The company was a surface clay mill that purchased clay and blended, refined, milled and processed the material into products used in paints, inks, and grease. On the day of the accident the employee had been informed that there had been a product change in one of the batch processing systems. The employee was assigned to perform cleanup duties on a reactor tank. Two valves controlled steam entry into the tank: a manual gate valve and a butterfly valve with an automatic pneumatic actuator. The PLC controlled the functioning of the batch system based on sensors that monitored material flow. At the time of the accident the PLC was in "slurry hold" mode. In this mode the system was programmed to actuate the steam valve when the clay slurry level reached 5.5 feet. An aluminum extension ladder used by the employee caused the level sensor to falsely sense that slurry was in the reactor, which resulted in the PLC sending a command to open the steam valve. Because the manual valve had been left open, steam at 350°F then entered the tank, fatally burning the employee (U.S. MSHA 2003a).

2.3 Underestimating risk

After the hazard has been identified there needs to be an understanding of the significance of the potential problem to facilitate safety decision making. Risk assessment helps to understand potential problems and their significance, and helps to prioritize resources to fix the problems identified. The concept of risk includes an understanding of both the severity of the consequences and likelihood of the event. Without a proper analysis of both severity and likelihood it is possible that the risk could be underestimated. A number of accidents involving software and computing systems has shown that risk is frequently underestimated or misunderstood in these systems.

On January 19, 1995, an X-31 U.S. government research aircraft was destroyed when it crashed in an unpopulated area just north of Edwards Air Force Base while on a flight originating from the NASA Dryden Flight Research Center, Edwards, California. The crash occurred when the aircraft was returning after completing the third research mission of the day. The pilot safely ejected from the aircraft but suffered serious injuries, including two fractured vertebrae and a broken ankle and rib. A mishap investigation board studying the cause of the X-31 accident concluded that an accumulation of ice in or on the unheated Pitot-static system of the aircraft provided false airspeed information to the flight control computers. The resulting false reading of total air pressure data caused the flight control system to automatically misconfigure for a lower speed. The aircraft suddenly began oscillating in all axes, pitched up to over 90 degrees angle of attack and became uncontrollable, prompting the pilot to eject. The mishap investigation board also faulted the safety analyses, performed by Rockwell and repeated by NASA, which underestimated the severity of the effect of large errors in the Pitot-static system. Rockwell and NASA had assumed that the flight software would use the backup flight control mode if this problem occurred, and this in itself would reduce the risk. The mishap investigation board noted that probability and severity were confused in this safety analysis; just because the risk assessment concluded that the probability of total pressure being lost was low did not mean that the consequences were any less severe. This risk assessment resulted in a failure to recognize the safety-criticality of the Pitot tube and thus a failure to perform testing using both nominal and off-nominal conditions. (Haley 1995).

2.4 Overestimating the effectiveness of safeguards

If we simply identified the hazard and assessed the risk we would do little to improve safety. It is the implementation of safeguards (hazard controls) and designing safety into the system that reduces the risk. However, these controls must be appropriate for the hazard considered and they must be effective. Ineffective controls may provide a false sense of security, and may not work when needed. Automated systems may have weaker controls than thought, especially if human interaction is required. In addition, hazard controls themselves could introduce new, unforeseen hazards.

On February 18, 2009, an employee was fatally injured at the Ravensworth Coal Preparation Plant reject waste bin in the Hunter Valley region of New South Wales, Australia. The accident occurred when 10 tons of waste rock were inadvertently released from the reject bin and fell onto the cabin of the employee's truck. At the Ravensworth Coal Preparation Plant, raw coal was extracted from the mine and usable coal was separated from waste rock. The waste rock was transferred approximately 2 kilometres on conveyers to the reject bin. The waste rock was then loaded from the reject bin onto trucks and hauled away. The process of loading the trucks with waste rock was controlled by a PLC system. The PLC system included truck detection sensors, traffic lights, bin capacity sensing, and remote control, hand-held transmitters used by the truck drivers. On the day of the accident the truck driver drove his truck under the reject bin delivery chute. A signal was sent from the handheld remote control to command the chute to open. The accident report stated that it was not clear whether the signal was sent inadvertently or intentionally. Opening the chute required that two of three lines of truck detection sensors be blocked in addition to a command from the remote control to assure that the truck was in the correct location. Each sensor line contained three sensors, and all three sensors had to be blocked for the entire line to be considered as blocked. At the time of the accident the truck was obscuring one line of sensors, and a second line of sensors was obscured by dirt on the lenses and therefore was not working correctly. Because two of the sensor lines were blocked and the remote control signal had been sent, the PLC automatically opened the reject bin chute door and dropped 10 tons of material on the truck cab before the driver had safely cleared the chute, resulting in the fatal injury (State of New South Wales 2010).

2.5 Failing to verify that safeguards actually work

Once the control strategy has been identified and implemented, those controls should be validated and verified. Validation determines that the correct system is being built and verification determines that the design solution has met all the safety requirements. Verification normally includes analysis, test, inspection, and demonstration. Experience has shown that verifications that are performed using improper assumptions or are conducted under conditions that are different from those in operation can lead to an underestimation of risk. Of special concern in software is the failure to test using sufficient off-nominal conditions and considering hardware failures and improper inputs.

On November 16, 2000, the Space Technology and Research Vehicles (STRV) microsatellites STRV 1-C and

STRV 1-D were launched on an Ariane 5 launch vehicle. STRV 1-C was intended to perform accelerated life testing of new components and materials in the high radiation environment of geosynchronous transfer orbit. STRV 1-D carried additional experiments. Two weeks after launch STRV 1-C displayed control problems; STRV 1-D exhibited the same problems a few days later. Eventually, both spacecraft lost communications with the ground. Investigations after the loss of the spacecraft found that a software error provided continuous current, instead of a short pulse, to latching relays. The continuous current heated the relays and degraded their insulation, which resulted in a short circuit that disabled the main receiver. A secondary receiver existed for redundancy, but this secondary receiver had been isolated by a trip switch. The trip switch required a ground command to be reset, and this could not be done without communications through the primary receiver. The problem was traced to a software specification that did not incorporate a requirement to command the relays by pulse. The problem was not found on the ground because the test software drove the relays with pulsed signals (Harland and Lorenz 2005).

2.6 Inadequate hazard tracking and anomaly reporting processes

Accident analyses often show that clues existed before the mishap occurred. Such clues frequently take the form of anomalies observed during the life cycle of a project. Therefore, learning from failure is critical to improving safety and preventing accidents. Anomalies discovered in the life cycle development must be properly reported to learn from those problems. In addition, a closed loop root cause and corrective action process must be in place to translate the documented anomalies into safety actions. That process must assure that hazard reports are re-evaluated as problems are found.

On August 12, 1998, the Titan IV A-20 launch vehicle lifted off from Florida. The rocket was carrying a classified National Reconnaissance Office payload. Approximately 40 seconds into flight the launch vehicle pitched down and began to break up, then automatically destroyed itself when the Inadvertent Separation Destruct System initiated the destruct sequence as soon as one of the solid rocket motors separated from the core booster. The payload was lost, although there were no injuries as a result of the accident. The accident investigation board found that exposed wires shorted during flight, causing an intermittent outage of the Missile Guidance Computer (MGC), which in turn lost the signal to the Inertial Measurement Unit (IMU) used to guide the rocket. The MGC recovered power, but the IMU then provided a false indication that the launch vehicle had pitched up and to the left (it had in fact been flying on the correct course). To compensate for the perceived pitch up, the MGC commanded the launch vehicle to pitch down and to the right. The aerodynamic stresses from these movements exceeded the structural margins of the launch vehicle and the rocket began to break up, ultimately destroying itself. The accident investigation board did not identify the source of the wire damage leading to the short circuit. However, the board reviewed historical records and identified hundreds of wiring faults and defects at the factory that were later discovered by inspection, and found previous incidents of short circuits while in flight. The board noted that the guidance system design was a causal factor because the timing signal from the MGC to the IMU was unable to withstand power transients that could reset the computer (U.S. Air Force 1999).

2.7 Failing to adequately manage change

While change is a normal part of the engineering process, there is no such thing as a minor change with respect to software safety. All changes to safety-critical systems must be evaluated because even minor changes can have major safety impacts. This typically means that organizations must have robust change management and configuration management systems, and changes must be factored back into the hazard analysis.

On October 24, 2002, a grinder exploded at the Foreman Quarry and Plant in Foreman, Arkansas. An operator was killed when flammable waste fuel covered him and ignited. The operator had started the pump for solid waste fuel processing when the accident occurred. The U.S. MSHA stated that the cause of the accident was that the safety monitoring system designed to shut off the waste fuel system pump had not been maintained so that it functioned properly. The Foreman Quarry and Plant, operated by Ash Grove Cement Company, mined limestone and processed it for use in Portland cement. Kilns were used in the processing, and these kilns were heated by burning coal, natural gas, and liquid waste fuel. The liquid waste fuel was delivered by truck or railcar and pumped into large storage tanks. From the storage area it was pumped through a grinder to reduce the particle size of the solids in the fuel. Two independent systems monitored and controlled the waste fuel delivery. A Foxboro Intelligent Automation Distribution Control System (I/A DCS) monitored and recorded normal operating parameters. The Foxboro also issued audible and visual alarms that were available at the plant control room. A PLC provided basic start up and shutdown of the system and responded to commands from the Foxboro. On the day of the accident the Foxboro sensed that the fuel delivery pressure was low, apparently due to blockage in the line. As designed, the Foxboro sent a command to the PLC to shut down the pumps. However, the PLC failed to respond and the pumps kept running. Three months prior to the accident this PLC had been installed; this was supposed to be a simple replacement of an older PLC of similar capability. However, the Foxboro had not been connected to the newer PLC, and the connections remained to the older non-functioning PLC. The system had never been tested with the new PLC. A test had been scheduled three days prior to the accident but had been aborted when a pump failed during the test; the test had never been rescheduled. The accident report stated that the blockage may have broken free just prior to the accident. With the pumps running, the pressure elevated significantly and a “water hammer” effect caused overpressurization in the system at the grinder. The grinder was torn loose from its base, spraying fuel and pulling loose a 480-volt cable that ultimately served as an ignition source (U.S. MSHA 2003b).

2.8 Weak safety culture

Most accidents are the result of a confluence of factors, and not just the result of failures of components or systems. Since the greatest threats to safety often originate in organizational issues, many industries have begun to realize that making the system safer requires improvements in the organization's safety culture. However, not all organizations have been successful in improving and maintaining organizational safety.

On April 21, 2010, the chief engineer on the container ship *Ever Excel* died when he became trapped between the top of the ship's passenger lift and the edge of the lift shaft. According to the U.K. Marine Accident Investigation Branch (MAIB), at the time of the accident the ship was undergoing a routine compliance inspection in Kaohsiung, Taiwan. The second engineer was unable to open the lift shaft doors to complete the inspection. The chief engineer tried to solve the problem and entered the lift car, climbed through an escape hatch, climbed on top of the lift car, and closed the hatch. The second engineer incorrectly believed that the chief engineer had set the controls to manual mode to take control of the lift car. Therefore, the second engineer released the emergency stop button then turned the reset key attached to the lift door. By closing the emergency hatch door the chief engineer had disabled the first safety barrier, an interlock that would not allow the lift to operate with the door open. The second engineer removed the second safety barrier, the emergency stop, by releasing the emergency stop and resetting the system. As a result, the lift returned to its normal automatic operating mode, and the lift automatically moved upwards, trapping and asphyxiating the chief engineer. The MAIB report noted that the crew had failed to follow manufacturer-suggested procedures in performing lift maintenance. The report also stated that the crew was unable to release the chief engineer after the accident and damaged the lift because they had not practiced emergency operation of the lift. In addition, the report identified a weak safety culture in the organization, stating, "It was evident that completing the task was considered more important than working safely." The report went on to state that communications were poor, risk assessments were not completed, there was little feedback provided to the crew on safe procedures, the company did not make use of previous accident and incident reports, and auditing was ineffective (U.K. MAIB 2011).

3 Overall Software Safety Lessons Learned

The accidents and incidents described here illustrate that there are significant challenges in the software safety discipline, and that organizations often fail to perform effective software safety efforts as part of an overall system safety approach. Some broad lessons learned that emerge from the examination of hundreds of accidents (Hardy 2012) include the following.

- *Decisions made in the acquisition and planning phases of development can profoundly affect safety***.** Planning typically involves trade-offs between many different facets of the program, including cost, schedule, performance, and safety. Poor planning can lead to unexpected safety consequences, and many safety decisions are actually made in the planning and acquisition phase. However, software safety personnel are often not included in early phases of a program when those critical decisions are being made. In addition, adequate resources may not be allocated to the software safety effort. This can result in a failure to perform hazard analyses and identify safety requirements early in the program when these activities provide the most impact.
- *Communication barriers between software engineers, hardware engineers, safety personnel, and management are common*. No one person can fully understand a complex system, especially one with software. Therefore, multiple individuals and organizations must interact and trade information to effectively reduce risk. This means that different parts of the organization must learn to speak each other's language. Communications between customers and suppliers must also be open and frequent. Misunderstandings and miscommunication are often contributors to accidents. Some of those misunderstandings come from inadequate requirements management efforts.
- *Software hazard causes are oversimplified or focused only on failures***.** Review of a number of hazard reports from different organizations has shown that software hazard causes and controls often do not provide sufficient detail or clarity. Software causes may be generically stated as "software error," instead of defining specifically the software functionality that can lead to an undesirable outcome. The focus is often on failure of the functionality to work, but other causes, such as inadvertent operation, may be ignored. Interfaces, especially those between software and hardware, may be misunderstood, and interactions between components are not explored. The software hazard analyses may not pay enough attention to those cases where the software works exactly as intended, but the implemented functionality is unsafe.
- *Risks may be underestimated and optimistically evaluated***.** Risk assessments allow organizations to make decisions about uncertain futures given existing knowledge. Assessing the risk of software-related systems presents challenges in large part because the evaluation of the likelihood of the hazard is difficult. Instead of using that limitation as an opportunity to carefully consider many different risk factors, organizations instead may create optimistic projections of what they want to happen. Or they may equate past success with low risk, ignoring the fact that testing and operations cannot feasibly consider all combinations of possible inputs.
- *Hazard controls may rely on good software processes and testing*. System safety efforts should follow the design order of precedence, where the first approach is to try to design out

the hazard or minimize the risks through design selection. Software is no different in this regard. Yet organizations may still focus on quality control and quality assurance efforts, such as focusing on good software processes or extensive unit testing, to prove that the design is safe. However, software processes and testing will not prevent an accident if the software design is flawed with respect to safe system operations.

- *There may be a failure to ask "what if the hazard controls don't work?"* Organizations may implement what appear to be effective controls, but then do not take the analysis any further. While organizations certainly understand that those controls may fail, they do not take the next step and ask what happens if they do not perform their function or perform the function incorrectly. Organizations make optimistic assumptions about the ability of the system, including hardware, software, and humans, to come to the rescue when the undesired event happens.
- *Testing tends to focus on functional operation and not off-nominal conditions.* Testing can be expensive, and most organizations are limited in the resources they can apply to testing. Therefore, the focus is naturally on making sure the system meets the requirements. This is necessary, but not sufficient. Many accidents have shown what can happen if testing does not include off-nominal scenarios and abnormal conditions. Testing should not just address what is required but also include what can go wrong.
- *Testing may not provide information on subsystem and component interactions.* Software and computing system accidents occur most often because of unanticipated interactions, not because the software was poorly coded. A number of accidents have occurred when no component failed in the conventional sense, but the interaction of components caused a system failure. Therefore, a significant focus of safety testing must be on a fully integrated system, with testing of end-to-end events. That testing must include stressing of the software, and should include interactions of the software with hardware, humans, and environments. Many verification efforts however fail to perform sufficient integrated system testing, or include operator interaction in that testing.
- *Anomalies may not be factored into the design or hazard analysis.* Learning from failure and problems is essential to safety. These problems provide clues of accidents yet to come. Therefore, software problem reports, like those of hardware, should be part of a larger root cause and corrective action system. These problem reports should include issues found during actual operation. Yet organizations do not always take these problems seriously or use those problems to look for issues that could lead to system failure. Software does not have to be perfect to be safe, and not all errors impact safety. But errors in safety-critical functions should be investigated and corrected. Conversely, organizations may incorrectly assume that a lack of anomalies or mishaps implies that the system is safe; in fact, latent errors could exist, and these errors may contribute to an accident.
- *Software change management and hazard analyses processes may not be integrated.* Engineering by its very nature is an activity that requires change, and changes occur in the hardware, software, processes, and organizations throughout development and into operation. While a number of organizations may have strong configuration and change management practices, those practices do not always integrate with the hazard analysis process. Hazards may fall through the cracks if those processes are not integrated.
- *Human-software interactions have significant safety implications that are often underestimated.* Humans interact with hardware and software in positive and negative ways. Organizations may not understand the importance of human-software interactions or pay as much attention as they should to displays and control panels. In addition, they may make changes to user interfaces and information flow on critical systems without adequate assessment. Organizations may count on operators saving the day when a bad day occurs in complex, software-intensive systems, but they may not provide proper tools and training to enable operators to perform those safety-critical functions.
- *Support software may be as critical to safety as control software but may not be included in safety analyses.* The focus of most software safety efforts is naturally on software that directly controls an operation. But software and computing systems show up in many different parts of the system, and this support software may turn out to be safety-critical. Support software, including models and simulations, may be just as hazardous as controlling software, but it is often not thoroughly examined.
- *Hazard analyses and safety systems may not be updated using operations and maintenance experience.* It is usually during the initial operating phases that the most is learned about the system. However, organizations may fail to feed what is learned in operations and maintenance back into their safety analyses.

4 Promoting a Questioning Attitude in Software and System Safety

It is important to promote the use of system safety methodologies and analyses. It is difficult to understand and then decrease the risk of complex technologies without the use of a structured approach to identifying and controlling hazards. However, as discussed above, lessons learned from past accidents and experiences point to the importance of cultivating and encouraging a questioning attitude toward all aspects of the system safety process, especially where software and computing systems are important for safety. Implementation failures can occur in any of the system safety process steps. We should use lessons learned such as those described in this paper to help us understand how previous efforts failed to prevent accidents, and how our own efforts might be similar. We should require compelling evidence before concurring with the analysis.

Most importantly, we should ask critical questions about the overall software and system safety process. By asking focused questions we can challenge assumptions. Such questions can stimulate thinking and get people to open up about the risks. Good questions allow us to view the system holistically, rather than just as the sum of its parts. Examples of such questions include the following:

- Do plans reflect how business is really done? Are plans reviewed? Do plans have unrealistic schedules or resource allocations? Is software part of that planning? Poor or unrealistic plans may reflect an organization that does not truly place a priority on safety activities.
- Is there a convincing story that the safety analysis is complete and thorough, and that software's contributions to hazards have been identified? Did the analyst use multiple tools (fault tree, hazard analysis, etc.) to perform the analysis? Were checklists, accident reports, previous experience, or a combination of those employed? Failure to show that the problem is being looked at from multiple perspectives could be an indication that there are holes in the analysis and that significant problems may not be identified.
- Are the reports detailed enough? Are causes descriptive? Does the logic make sense and is it complete? Do controls match up with the causes, showing a one-to-one or many-to-one relation? Lack of detail could be an indication of insufficient knowledge of the system, or lack of information on the system.
- Are the hazard controls primarily procedural rather than design changes, safety features or devices? Is there an overreliance on humans and software to "save the day"? Overreliance on operational controls may indicate a weak safety design.
- Can the control strategy actually be implemented and verified? Is the control strategy so complex that it will be impossible to determine whether it will work when needed? Is the control truly effective? Are controls truly independent? Complex controls or overlapping control strategies may be an indication of a weak safety design.
- Has the risk assessment truly considered the worst case? What is the basis for the likelihood levels? Has the risk assessment considered lower severity but higher likelihood cases? Is the risk analyzed by cause and by phase? Failure to provide good answers to these questions indicates a potential misunderstanding of the risk.
- Are problems found in test and design included in the hazard reports and factored into the design? Failure to incorporate problems and corrective actions is an indication of the potential to miss serious design flaws.

These questions help to identify whether the system safety process is robust. However, we must also ask questions related specifically to the use of software and computing systems in complex systems. The best questions come from real-world examples of accidents where software has been a contributor. Some examples of questions are as follows, and others can be found in Hardy (2011).

- Have safety-critical software, commands, and data been identified?
- Do hazard controls for software-related causes combine good practices and specific safeguards?
- Do standards exist for software peer reviews and other design reviews?
- Is software and system testing adequate, and do tests include sufficient off-nominal conditions?
- Is the computing system design overly complex?
- Is the design based on unproven technologies?
- What happens if the software locks up?
- Are the sensors used for software decisions fault tolerant?
- Has software mode transition been considered?
- Has consideration been given to the order of commands and out of sequence inputs?
- Will the software and system start up and shut down in a known, safe state?
- Are checks performed before initiating hazardous operations?
- Will the software properly handle spurious signals and power outages?

These are by no means all the questions a decision maker should ask, and positive answers to these questions provide no assurance that an accident will be prevented. These questions should encourage critical thinking and generate additional safety questions to provide further insight on system risk. A failure to ask these questions could mean that the potential for an accident is higher than we had assumed.

We also have a responsibility as system safety practitioners to share our doubts and questions with decision makers to allow them to understand what we do not know and where uncertainties exist. In particular, we should:

- *Avoid oversimplifying the potential hazard causes*. Identifying hazard causes in complex, automated systems can be a difficult process, and decision makers should be made aware of the challenges in performing this activity.

- *Do not downplay uncertainties, especially with likelihoods*. Obtaining credible reliability estimates for software may not be possible for new systems, and qualitative risk assessments should be supplemented with analyses of other factors such as complexity, maturity, degree of system testing, and so on.
- *Do not self-censor, especially with respect to hazard controls*. When safety practitioners are aware that a contract has been issued which limits the choices of hazard controls, it is natural to eliminate options from consideration. However, the decision maker should be aware that such choices are being made.
- *Provide alternatives, but discuss the tradeoffs in risk*. Rather than simply saying "no" to an activity, safety practitioners should provide the decision maker with options, then clearly describe the risk of each option.
- *Discuss the limitations of the testing and verification efforts*. It is practically impossible to test every possible combination of software inputs, or test every possible hardware or software configuration to be used. Decision makers should be made aware of these limitations.
- *Be clear about the effects of failures and changes during development and the potential for increased risk*. Problems discovered during development and in operation, and changes resulting from problem fixes and upgrades, can have major impacts on safety.
- *Use accidents and incidents to provide support for safety conclusions*. Decision makers will respond more favorably to our conclusions if there is concrete evidence to back up our claims. We should use available accident and incident reports to provide that evidence. These "stories" will also resonate better than statistics with decision makers in making our case.

It is up to all stakeholders to look for those conditions that could lead to an accident and to recognize that the worst can happen. This means we should all express concerns about safety management and engineering when necessary based on our knowledge, experience, and judgment, and based on lessons learned from accidents. We must ask questions to understand the potential for harm, to understand the steps taken to assure that the risks have been reduced, and to assure that there is proof that hazard controls are effective. And we must openly and honestly communicate what we do not know. We will never eliminate risk, nor do we want to. Without risk there is no reward. But it is up to all of us to promote and encourage a questioning attitude to ensure that we are knowledgeable of those risks and to assure that the risks have been appropriately reduced.

5 Summary

System safety can provide immense benefits to any industry, especially those designing, building, and operating complex systems using software and computing systems. By proactively identifying hazards, assessing and characterizing risks, and taking actions to reduce those risks, organizations can prevent accidents and reduce the potential for death, injury, property damage, and environmental impacts. However, poor system safety analyses can result in precious resources being used on low risk activities while larger risks are ignored. When applied inappropriately, system safety methods can lead to overconfidence and result in an underestimation of certain important risks. System safety efforts should be promoted and advocated, but we should also promote a questioning attitude to further the discipline. We should understand the ways that these analyses can provide misleading results, especially in software-intensive systems, and we should examine the ways in which risk can increase by the actions we take. Lessons learned in the form of accidents and experiences in implementing the system safety process should be used to fuel those questions. It is through a questioning attitude that system safety and software safety efforts can accomplish their main goal -- preventing accidents.

6 References

Finkelstein, A., and J. Dowell (1996): "A comedy of errors: the London Ambulance Service case study," 8th International Workshop on Software Specification and Design.

Haley, D. (1995): "Ice Cause of X-31 Crash," National Aeronautics and Space Administration Dryden Flight Research Center, Edwards, California, NASA Press Release 95-203.

Hardy, T.L. (2011): *Essential Questions in System Safety: A Guide for Safety Decision Makers*, AuthorHouse.

Hardy, T.L. (2012): *Software and System Safety: Accidents, Incidents, and Lessons Learned*, AuthorHouse.

Harland, D.M., and R.D. Lorenz (2005): *Space System Failures: Disaster and Rescues of Satellites, Rockets, and Space Probes*, Praxis Publishing.

State of New South Wales (2010): "Fatality involving David Hurst Oldknow Ravensworth Underground Mine Coal Preparation Plant Reject bin 802 18 February 2009," May 2010.

United Kingdom Marine Accident Investigation Branch (2011): "Report on the investigation into the fatal accident to the chief engineer in the lift shaft on board Ever Excel in Kaohsiung, Taiwan on 21 April 2010," Report No 6/2011.

United States Air Force (1999): "Titan IVA-20 Accident Investigation Board Summary," January 15, 1999.

United States Mine Safety and Health Administration (2003a): "Report of Investigation: Fatal Other Accident (Steam Burns), February 11, 2003, Southern Clay Plants & Pits Southern Clay Prod. Inc., Gonzales, Gonzales County, Texas," Mine I.D. No. 41-00298.

United States Mine Safety and Health Administration (2003b), "Report of Accident: Exploding Vessels Under Pressure Accident, October 24, 2002, Foreman Quarry and Plant, Ash Grove Cement Company, Foreman, Little River County, Arkansas," Mine I.D. No. 03-00256.

Applying System Safety Methodologies to Consumer Product Safety

Zhuojun LIU[1] Yongguang ZHANG[1] Peng YU[1] Huina MU[2]

[1]Academy of Mathematics & System Sciences of Chinese Academy of Sciences, Beijing 100190 of China
{zliu@mmrc.iss., yzhang@iss., yupeng@amss.}ac.cn

[2] Beijing Institute of Technology, Beijing 100081 of China
muhuina@bit.edu.cn

Abstract

Harm from consumer products is an increasingly serious problem that people have to face. There are too many consumer product related injury or fatality events occurring each year. To avert or mitigate the harm from consumer product related events, system safety methodologies can be used in the consumer product domain. This paper will focus on the perspective of identification and analysis problems in relation to causal factors with respect to consumer products. For a selected consumer product, we use a case study, standard matching and warning information to prepare a PHL to list the potential hazards – including causal factors leading to consumer product related injuries. Furthermore, based on the typical injury scenarios, we combine the FTA and FEMA together to form a composed method, which can be used on accident causal analysis within the consumer product safety domain.

Keywords: Preliminary Hazard List (PHL), Fault Tree Analysis (FTA), Failure Mode and Effects Analysis (FMEA), Consumer Product Safety, Injury Scenarios.

1 Introduction

A consumer product is generally defined as any tangible good for sale that is used for personal, family, or household (non-business) purposes; a product which usually is intended to satisfy consumers' living-need. The determination as to whether a tangible good is a consumer product depends on the view of national regulatory authorities on a case-by-case basis. This basis may vary from one jurisdiction to another. For example, the U.S. Consumer Product Safety Commission (CPSC) lists more than 15,000 different types of consumer products.

Although there are some differences in demarcation of consumer products among different countries, we all agree that consumer product safety is an important issue. We know the U.S. Consumer Product Safety Act (CPSA) requires manufacturers, importers, distributors and retailers to notify the CPSC immediately if they obtain information that reasonably supports the conclusion that a product distributed for commerce (1) fails to meet a consumer product safety standard or may be subject to a banning regulation, (2) contains a defect which could create a substantial product hazard to consumers, (3) creates an unreasonable risk of serious injury or death, or (4) fails to comply with a voluntary standard upon which CPSC has relied under the CPSA.

On one hand, people cannot live without consumer products; on the other hand consumer products-related injuries happen often. According to the U.S. National Electronic Injury Surveillance System (NEISS) [01], more than 10,000,000 people visit hospitals each year because of consumer product related injuries; Table 1 summarizes the consumer product injuries in America in recent years.

Time (year)	National injury estimate	National fatalities estimate
2003	12 720 963	8 054
2004	13 096 938	7 120
2005	13 096 983	6 259
2006	13 232 263	5 440
2007	13 232 338	5 439
2008	13 456 353	5 825
2009	13 966 353	5 379
2010	14 694 928	5 686

Table 1: U.S. NEISS Data

Note: Information collection based on https://www.cpsc.gov/cgibin/NEISSQuery/home.aspx

In Europe, the Injury Database (IDB) is an internet database set up by DG SANCO (Directorate General for Health and Consumer Affairs), a European Public Health Alliance, in 1999. Although the IDB as yet does not cover all countries in Europe, it can still provide useful injury information (see Table 2).

As yet there is no accurate data available on consumer product related injury events in China. We therefore need to make use other data sources to identify the extent of consumer product related injury resulting from products made in China. We refer to the RAPEX reports and CPSC recall reports.

RAPEX is the European Union Rapid Alert System that facilitates the rapid exchange of product safety information between Member States and DG SANCO on

Countries and Injury Situation		2003	2004	2005
Austria	Injuries cases	563000	581000	589000
	Incidence Rate	70	72	72
Denmark	Injuries cases	468000	443000	438000
	Incidence Rate	87	82	81
France	Injuries cases	5793000	6885000	9723000
	Incidence Rate	94	110	155
Netherlands	Injuries cases	617000	596000	599000
	Incidence Rate	38	37	37
Portugal	Injuries cases	654000	595000	578000
	Incidence Rate	62	57	55
Sweden	Injuries cases	520000	521000	508000
	Incidence Rate	58	58	56

Table 2: Injury Status based on IDB (Injury Database)

Note: Information from https://webgate.ec.europa.eu/idb/index.cfm?fuseaction=idbnetwork
Incidence Rate: Injuries per 1000 inhabitants

measures taken to prevent or restrict the marketing or use of products posing a serious risk to the health and safety of consumers. According to the Annual Reports [02, 03] on the operation of the Rapid Alert System for non-food consumer products, the total number of unsafe products notified through RAPEX in 2010 was 1963, marking a 15.5% increase over the previous year. Among them, the number of products notified coming from China was 1134, which accounts for some 57.8% of total number of notifications. Table 3 lists the total number of banned products and the number of banned products from China, by European countries in recent years.

	2005	2006	2007	2008	2009	2010
Notified number of Chinese products	346 (49.4%)	440 (47.6%)	689 (50.8%)	869 (56.2%)	990 (58.3%)	1134 (57.8%)
Total number of notifications	701	924	1355	1545	1699	1963

Table 3：Number of RAPEX notifications 2005 - 2010

Note: Information collection based on RAPEX Annual Reports

In United States market, from the recall reports [04] issued by CPSC, the number of recalled products from China exceed over 50% of total number of recall reports in recent years, see the Table 4.

	2005	2006	2007	2008	2009	2010
Number of recall reports for China	156 43.2%	172 47.8%	302 61.4%	227 51.7%	218 50.1%	218 54.1%
Number of recall reports for U.S.A	81 22.4%	66 18.3%	66 13.4%	67 15.3%	74 17%	69 17.1%
Total number of recall reports	361	360	492	439	435	402

Table 4：Unsafe products recalled by CPSC

Note: Information from http://www.recalls.gov/ or http://www.cpsc.gov/

According to the data sources above, we can easily see that consumer product safety is a serious problem in China. However we need to highlight that not all the notified or recalled China made products for the Europe and USA markets can be attributed to China, as some of the unsafe products were made in China by non-Chinese companies. So, to improve the safety of consumer products requires global commitment. Whilst this is a major challenge as evidenced by the yearly notified and recalled products, this is an even bigger challenge for China. In fact, the Chinese government has taken measures to improve the consumer product safety by funding research, such as the research which is the subject of this paper.

2 The Approach

In order to avert or mitigate the loss from consumer product related injury events, we apply System Safety Methodologies to consumer product safety. One important thing is that we need to consider how to identify the hazard -- including causal factor to consumer product related injuries events. In this paper, we use a combination method of case study, standard matching and warning information to prepare a PHL to list the potential hazards. We also develop an S-FFA method by combining FTA and FMEA together to analyse the reasons. Here, S-FFA means System Fault and Failure Analysis. We share some results achieved under the research program "Research on the impact factors of quality safety for consumer products and its standard development" supported by the Ministry of Science and Technology of the People's Republic of China.

We know that the concept of hazard is fundamental in the system safety domain. It is "a condition that is prerequisite to a mishap" [05]. In more detail, it is "a potential condition, or set of conditions, either internal and/or external to a system, product, facility, or operation, which, when activated transforms the hazard into a series of events that culminate in a loss(an accident)." For consumer product safety, we also talk about hazards in general. However we specially emphasize the hazards which are directly from the product itself. We consider them as *causal factors* which could possibly lead to an injury event taking place. In order to determine the causal factors, we designed a two-stage identification procedure.

2.1 Identification and PHL

At the first stage, we collect the potential hazards, mainly through three channels, a case study, standard matching and the RAPEX notification reports. To some extent, we can consider the potential hazards decided at this stage as the content of PHL. We insist on including much more information in the PHL to assist in the identification of the key or confirmed causal factors at the next stage.

Now, for a selected consumer product, we try to identify and analyse those causal factors which could have possibly led to an injury event that has occurred.

2.1.1 RAPEX and notification

RAPEX was developed to comply with the European Directive 2001/95/EC [05]. The Directive imposes a general safety requirement on any product put on the market for consumers or likely to be used by them, including all products that provide a service but excludes second-hand and antiques. Through a weekly notification report, RAPEX advices information on potential unsafe consumer products found in the Europe market in order to give customers the necessary alert quickly. Products notified through the RAPEX system pose a serious risk to the public. A serious risk is defined as one which requires rapid intervention by the public authorities, and it includes risks with effects that are not immediate. Therefore, from the notification reports, one can find many messages about the potential causal factors. For example, RAPEX Report 6 [06], published on 10-02-2012 described one brand of children's sweatshirt that has possible strangulation dangers due to the presence of a drawstring in the hood and neck area, which does not comply with the relevant European standard EN 14682. The action stated "*withdrawal from the market ordered by the authorities*". It is natural for us to therefore consider the inappropriate drawstring to be a potential causal factor leading to an injury. So, we add it to the PHL.

2.1.2 Standard matching

Based on the notification report above, we cannot assert that the sweatshirt will actually result in an injury. What we know is that the product does not comply with a European standard, EN 14682. This suggests that we should look for all standards relevant to the selected product when we consider its safety. We call this procedure *standard matching*. By looking for conflicts and omissions with standards, we can see whether some attribute of the product is not compliant. In practice, one useful way to find the potential causal factor information is through standard matching.

2.1.3 Case study

Compared to standard matching and product notification, a case study can give much real information about the causal factors. Let us analyse real accident cases for similar products. From January 1985 through January 1999, CPSC received reports of 22 deaths and 48 non-fatal incidents involving the entanglement of children's clothing drawstrings. Once the actual causal factor, namely the drawstring, had been determined (see Figure 1), CPSC issued a guideline to help prevent children from being strangled or getting entangled by the neck and waist drawstrings of upper outerwear garments, which impelled the ASTM (ASTM International, formerly known as the American Society for Testing and Materials) to adopt the Standard Safety Specification for Drawstrings on Children's Upper Outerwear. It is easy to see that through a case study we can not only learn the causal factor but also enrich the safety standard for consumer products.

All findings at this stage can be collected as the potential causal factors for the selected consumer products.

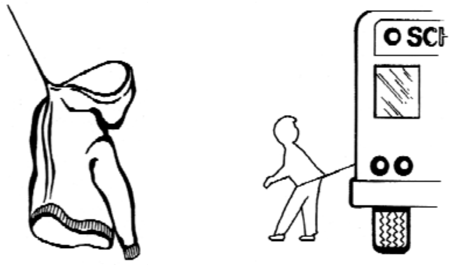


Figure 1: Information source from CPSC website
http://www.cpsc.gov/CPSCPUB/PUBS/208.pdf

2.2 The S-FFA Method

At the next stage, we continue the identification using the potential causal factors, and specially propose some injury scenarios, which can be thought as the sequence of events from which injuries can arise from that causal factor. In general, injury scenarios can be a comprehensive sequence description based on some real injury events that have actually occurred, or some hypothetical, but plausible sequence of events by experts, possibly refined through some experimental analysis. Sometimes, the information of potential causal factors is also useful to suggest an injury scenario.

It is necessary to consider the consumer for whom the product is intended and how the consumer uses the product in identifying the injury scenario, assuming that the consumer follows the user instructions or, if there are none, the expected normal handling and use for the product. Furthermore, other scenarios should be developed that include vulnerable consumers, slight or more pronounced deviations from normal use, unfavorable conditions of use, such as the situation shown in Figure 1.

There are many system safety methodologies; we will adapt some of these to make them more suitable for consumer product safety. Here, our purpose is to find the key "confirmed" factors that relate to the injury scenario for a selected product. In order to do it, we propose the S-FFA (system fault and failure analysis) method. S-FFA combines the FMEA and MFTA (Modular FTA) methods into one, (see Figure 2).

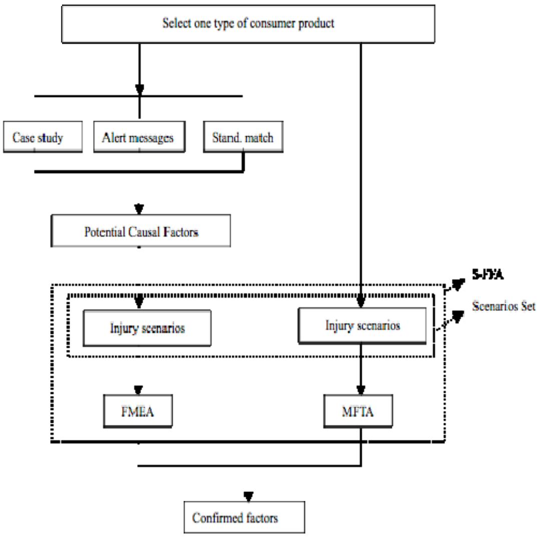


Figure 2: Two-stage identification procedure

At the end of the first stage, we have established a PHL to identify the many potential causal factors. We then need to find underlying the key factors, the confirmed factors that relate to the many causal factors. The aim is to undertake improvement measures for the information gained about these key factors. At the second stage, we make use of the S-FFA method to find the confirmed factors from the potential causal factors. In other words, we applied S-FFA to do the analysis in two ways; one from injury scenarios derived from the Potential Causal Factors; the second from posturing likely injury scenarios purely from a user perspective.

Since FMEA is a methodology used in product development for the analysis of potential failure modes within a system, thus enabling classification by the severity and likelihood of the failures, it can be adopted to do bottom-to-up analysis here. For a selected potential causal factor, if FMEA analysis leads to an injury scenario, or a situation description relevant to a consumer product injury event, the selected factor can be considered as a confirmed factor.

On the other hand, we can use MFTA to do the top-down analysis. MFTA (Modular FTA) is a variant of FTA that better suits consumer product safety. In general, consumer products have many attributes. However, when considering the safety attribute, we need only consider the structure of the product, the components and parts that compose the product, and the inherent (factor) property of the product. This allows us to simplify the FTA procedure for the analysis of consumer product safety. For a selected product and starting with one injury scenario, we get into the *component* level analysis to identify which *component* is the cause of the injury event. Then we further decompose the component to the *part* level, to identify the problem *part* within the *component*. Finally, at *factor* level, we need to determine what inherent property of the product causes the *part* to be the problem. This kind of FTA procedure can usually be achieved in three levels. We call this three-level analysis procedure an MFTA. The output of MFTA would contain the findings about the causal factors to the injury scenario; they can be considered to be the confirmed factors. It is easy to use for people who are non-experts in system safety, because the routine procedure is relatively simple. Figure 3 is an illustration of this three-level MFTA for a textile-bruise scenario, i.e. a bruise injury scenario due to a textile.

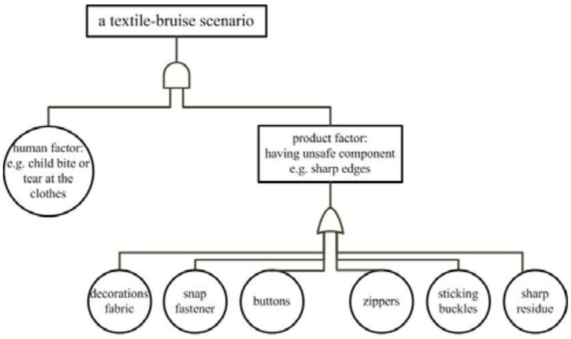

a textile-bruise scenario
human factor: e.g. child bite or tear at the clothes
product factor: having unsafe component e.g. sharp edges
decorations fabric
snap fastener
buttons
zippers
sticking buckles
sharp residue

Figure 3: Bruise analysis for textile

For the bruise analysis, we focus mainly on two reasons: one is the consumer's behavior, for example, children are apt to bite or tear at the clothes, which is shown in the left branches of the tree at the top level; the second is from the product itself, for which more analysis is required. Past experiences with similar products are useful here. Buttons, decorative articles, zippers, buckles etc would be causal factors to the bruise injury, and are shown at the bottom of the tree in Figure 3.

Between 2008 and 2010, we undertook an R&D project titled "Research on the causal factors of quality safety for consumer products and its standard development" and supported by the Ministry of Science and Technology of the People's Republic of China. We found that consumer product safety is a serious problem in China and a complex issue to understand the mechanism resulting in consumer product injury. We proposed a description of the mechanism using a 3-branch and 3-level tree-like diagram shown in Figure 4.

3-branches

It is easy to see the three branches for analysing an injury event or establishing an injury scenario; personal (human) factors (left branch), the product itself (middle branch), and usage conditions (right branch). The personal factors may contain information about the vulnerability of the people, predictable misuse of the product etc; the usage conditions usually relate to the environment e.g. high temperature, high pressure, high voltage and extreme weather etc. We are going to put more emphasis on the product itself, because it is here where we can take measures to reduce or mitigate the harm, for example, by changing the design, by improving the production process and by enhancing the quality.

3-level analysis for consumer product

Our main purpose is to analyse the (key) causal factors to an injury event related to some consumer product. Any method for this purpose should be practical and pragmatic. The 3-levels for consumer product are: the *component* level of the product, the *part* level of the product (or the *component*), and the *factor* which is the inherent property of the product which causes the product injury event to occur.

From the mechanism discussed above in the way a consumer product injury occurs, we can use the S-FFA method to do the causal factor analysis. The pictorial representation of our integrated approach is shown in Figure 4 above. For the safety analysis of consumer products, the simpler three-level thinking is reflected by the middle branch in Figure 4. Of course, we also must consider the personal factors and the usage conditions.

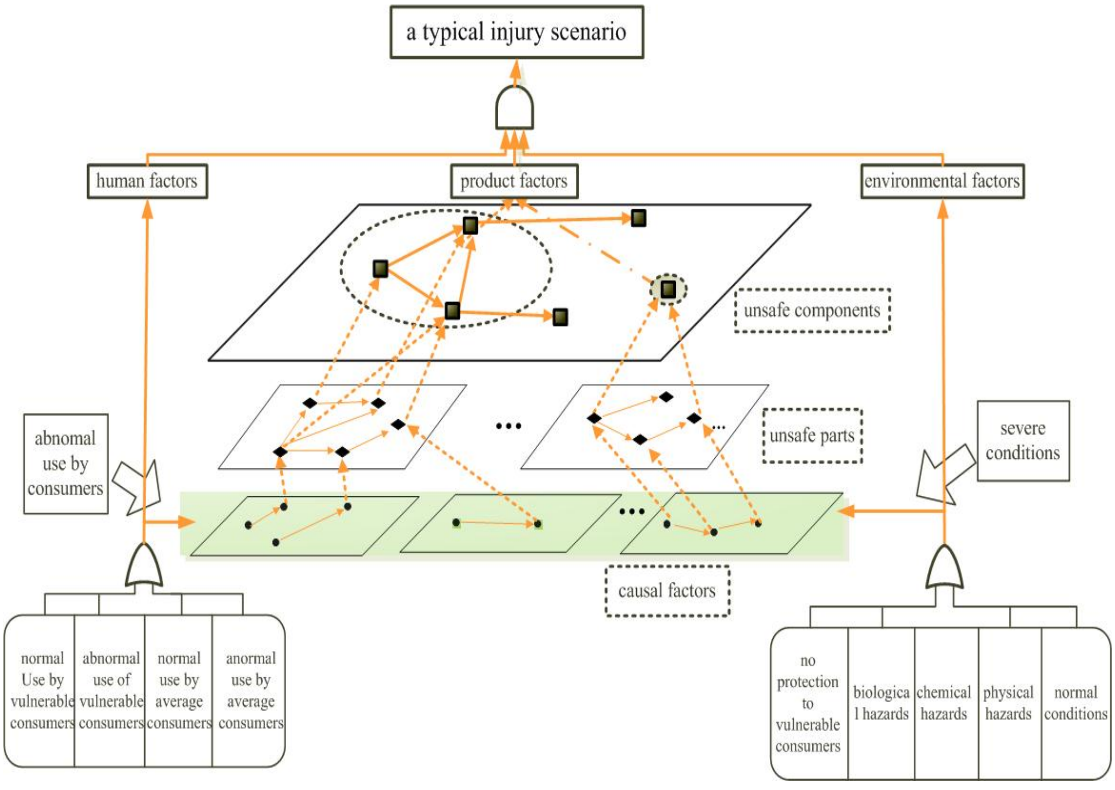


Figure 4: Three-level top-to-down analysis

3 Conclusion

Clearly, consumer product safety is important. We discussed how to apply the System Safety Methodologies to Consumer Product Safety and demonstrated the usefulness of System Safety techniques. The tools and methods we developed have been utilized to identify the injury causes for textile products and electrical appliance products. Based on our findings on the causal factors for the selected products, we are making effort to raise the safety level through improving design of the product, revising the related product standards, even remolding production processes.

For consumer product safety, the identification of the causal factor is just a first step; we then need to undertake further analysis and (risk) assessment. We are expecting to have our own (consumer product related) injury data base like NEISS and an early alert system like RAPEX such that we are at a position to do more quantitative analysis, to improve decision making with respect to consumer product safety. In addition to, we need to also encourage safety education, emphasizing both the responsibility of the government and enterprises.

The authors would like to thank the anonymous reviewers for their detailed comments and suggestions.

4 References

[01] NEISS (2000): The National Electronic Injury Surveillance System – A Tool for Researchers http://www.cpsc.gov/neiss/2000d015.pdf. Accessed now.

[02] Rapex Annual Report (2010): Keeping European Consumers Safe – 2010 Annual Report on the operation of the rapid alert system for non-food dangerous products.

http://ec.europa.eu/consumers/safety/rapex/docs/2010_rapex_report_en.pdf, Accessed now.

[03] Rapex Annual Report (2009): Keeping European Consumers Safe – 2009 Annual Report on the operation of the rapid alert system for non-food dangerous products.

http://ec.europa.eu/consumers/safety/rapex/docs/2009_rapex_report_en.pdf, Accessed now.

[04] RECALL HANDBOOK (1999) U.S. Consumer Product Safety Commission Office of Compliance Recalls and Compliance Division.

http://www.cpsc.gov/businfo/8002.html, Accessed now.

[05] Stephenson, J. (1991) System Safety 2000 – A Practical Guide for Planning, Managing, and Conducting System Safety Programs, Van Nostrand Reinhold, New York.

[06] Roland, H. E. and Moriarty, B. (1990) System Safety Engineering and Management (Second Edition), John Wiley & Sons, Inc.

[07] GPSD (2001): General Product Safety Directive: Directive 2001/95/EC of the European Parliament and of the Council on General Product Safety, Brussels, Belgium, 3 December 2001.

http://europa.eu/legislation_summaries/consumers/consumer_information/l21253_en.htm, Accessed now.

[08] Weekly overview of RAPEX notification Report 6, 2012 (No. Ref. 07, 0209/12)

http://ec.europa.eu/consumers/dyna/rapex/create_rapex.cfm?rx_id=423, Accessed now.

5 Biography

Zhuojun LIU, Ph.D., Research Professor, Academy of Mathematics and System Sciences, Research & Consultation Center for System Safety, Chinese Academy of Sciences, Beijing 100190 of China, zliu@mmrc.iss.ac.cn

Yongguang ZHANG, Research Professor, Academy of Mathematics and System Sciences, Chinese Academy of Sciences, Beijing 100190 of China,

yzhang@iss.ac.cn

Peng YU, Ph.D., Doctorate Student, Academy of Mathematics and System Sciences, Research & Consultation Center for System Safety, Chinese Academy of Sciences, Beijing 100190 of China,

yupeng@amss.ac.cn

Huina MU, Ph.D., Assist Professor, Beijing Institute of Technology, Beijing 100081 of China,

muhuina@bit.edu.cn

Risk Assessment in the Wild

Andrew Rae and Richard Hawkins
Department of Computer Science
University of York
Heslington, York, United Kingdom

andrew.rae@york.ac.uk

Abstract

This paper introduces an empirical research method for systems engineering based on the examination of work products. To illustrate the method we describe an investigation of safety risk assessment as it is actually recorded, rather than the standards, forms and procedures used to guide risk assessment. A body of risk assessments was collected via a combination of public search, freedom of information request, and private request. The risk assessments are from multiple domains, for multiple purposes, and follow diverse formats – the one thing that they have in common is that they are genuine work products.

Due to the necessarily arbitrary selection process, the collection cannot resolve quantitative hypotheses about the distribution of phenomena. However, it provides an opportunity to explore assumptions and suspicions about the real-world conduct of risk assessment that cannot be examined by looking at academic literature or guidance documents.

The paper makes contributions in three areas:

- Our early findings about the characteristics of the risk assessment collection
- Our experiences with the exercise itself, and the lessons learnt which may be helpful in future similar research
- Observations on the relationship between theoretical and applied system safety, and the methods that may be applied to answer important questions in each sphere.

Keywords: Risk Assessment, Empirical, Research Methods.

1 Introduction – The Empirical Approach to System Safety Research

System safety is a relatively young engineering discipline. Whilst concern about accidents is long-standing (ASSE 2011), the start of modern system safety engineering is generally dated to the invention of Fault Tree Analysis in 1965 (Dhillon 1982). The body of knowledge in system safety, typical of many young disciplines, is populated by beliefs and techniques drawn from experience or constructed based on plausible theories. As we showed in a study of current practice (Rae et al. 2010) there is a limited basis of evaluation evidence for supporting or pruning the body of knowledge. Progress in an academic discipline is characterised by refining and replacing knowledge in the light of new evidence, and system safety currently lacks strong mechanisms for testing the knowledge we have. System safety is inherently a "soft science", populated by researchers drawn mainly from a "hard science" engineering background. Faced with questions requiring soft science research methodologies, the field has concentrated on activities not involving empirical study. It is possible that discomfort with research methods seen as "unscientific" has led to a failure to recognise the developing body of work in the traditionally soft sciences aimed at tackling empirical difficulties.

The traditional distinction between "hard" and "soft science is discussed by Howard under the criteria of empirical cumulativeness and predictive accuracy (Howard 1993). Empirical cumulativeness is the reliability with which experiments produce results which are consistent with each other (Hedges 1987). Predictive accuracy describes how well a theory can predict the outcome of a real-world interaction. For example, a theory in psychology might correctly predict the outcome of an event 70% of the time – this shows low predictive accuracy. On the other hand, different experiments to validate the theory might consistently produce this 70% result – this shows empirical cumulativeness. Howard argues that soft sciences may have low predictive accuracy due to the fact that the phenomena being studied have a large number of interacting causes, making it difficult to comprehensively account for variation in observations.

On the criterion of predictive accuracy, system safety engineering is inevitably a soft science. The safety of a system emerges from a large number of interacting causes, and the precise characterisation of these causes and interactions is far beyond the state of the art. Consider, for example, the measurement of "safety culture" (Guldenmund 2000) . Even if safety culture could be fully characterised (it cannot) or reliably measured (it cannot), culture would be only one among many factors determining accident rates. This does not make safety culture an unscientific concept. If we could establish empirical cumulativeness by finding a repeatable measure of safety culture, and establishing a reliable correlation between safety culture and accidents, then the fact that the correlation is not 100% does not make the relationship less real.

System safety engineering is growing in scope and importance (see for example the recent introduction of

ISO 26262 (ISO 2011)). It is important that the body of knowledge continues to grow in parallel. This growth requires empirical foundations.

In Section 2 of this paper we discuss the ways in which system safety research can be supported by empirical research, and introduce risk assessment as a running example of a topic needing empirical support.

In Section 3 we introduce the "Safety Menagerie" as a research method. In Section 4 we show how the research method can be applied to questions about risk assessment, and provide indicative findings. The contribution of the paper is not these findings, but the conclusions reached about the research approach itself, which are provided in Section 5 and discussed in Section 6.

2 Observation and Measurement

Alexander (2010) discusses the range of system safety research goals, and the suitability of various research methods for addressing these goals. Broadly speaking, there are two main categories of goals which can be best supported by empirical research:

1. Evaluation of methods and techniques; and
2. Observation and measurement of current practices.

Each of these goals requires knowledge to be shared across what Alexander refers to as the "research/practice boundary". In evaluation research, the techniques are transferred to industry, with information about efficacy returned across the boundary. In observation and measurement research, the challenge is to gain an accurate view of industry practice, to provide grounding for development of new theories and techniques. Trevelyan (2007) describes important discrepancies between the way engineers describe their work and the actual practice of that work. Thus, instruments such as surveys are seldom suitable for acquiring the necessary insight.

The use of social science methods in studies of engineering practice has received considerable recent attention (Ahmed 2007). This is particularly the case in software engineering, which has many features in common with safety engineering. The performance of a software project is influenced by many poorly understood factors, making it difficult to isolate single factors for systematic study (Wohlin et al. 2003).

Where there is a close relationship between the effect to be studied and the environment in which it occurs, case studies are considered the most appropriate research method (Creswell 2007). The most common forms of engineering case study are participant observation and action research. These methods have produced interesting results, but are limited in scope to a small number of workplaces, creating external validity problems for many questions of research interest.

In order to refine existing safety engineering methods, design new methods, and improve education, it is desirable to understand how safety engineering is currently practiced. This is particularly the case if there is divergence between "best practice" as described in the literature, and "industry practice" occurring in real-world organisations.

This paper focuses on the practice of risk assessment. For this practice, there are research questions of interest that can only be answered by observing the real world. The high level questions concern the value-add of risk assessment as an activity.

1. How often does risk assessment lead to implementation of improvements to a system or operations?
2. To what extent are the outcomes of risk assessment predetermined or expected before the assessment is conducted?
3. Are the hazards of a system better understood after risk assessment?

We may also be concerned with what makes a good risk assessment.

4. Are there elements of a risk assessment currently considered important which do not influence the outcomes?
5. Are some methods of risk assessment more effective than others?
6. How does the practice of risk assessment vary between industries? Between different types of risk? Between different system technology?

In designing guidance and education, we may be interested in the practical shortcomings of risk assessment.

7. To what extent do risk assessments document their assumptions?
8. What types of uncertainty are treated well or poorly in risk assessments?
9. Are internal inconsistencies common in risk assessment?
10. Do risk assessments commonly cite evidence in support of estimates used?
11. Is the theoretical division between risk assessment and risk acceptance preserved in practice?
12. Are mitigations selected systematically or arbitrarily?

Further, risk assessment may reveal beliefs and attitudes held by those who perform the assessment.

13. Is risk aggregated, or is each source of risk treated atomically?
14. What types of risk are considered in scope and out of scope?
15. Is risk identification part of risk assessment, or are the important risks considered to be already identified?
16. What language is used in talking about risk and in drawing conclusions?

Finally, we may be interested in what risk assessment reveals about those performing the risk assessment.

17. Do risk assessments for systems involved in accidents look different from risk assessments for other systems?

18. Is the style and language of risk assessment an indication of safety culture?

3 The Safety Menagerie Method

3.1 Purpose

In the work reported in this paper we are trialling a method of observational research based on work products. Document analysis is frequently used in ethnography to extend and add detail to interviews and observations (Creswell 2007). Such analysis typically focuses on the way culture is revealed through features of the document. For our present research we are interested in documents as records of practice. This is not as direct as actually observing practice, but covers many more situations for the same research effort. Rather than examining one work situation through a clear lens, we observe many situations through a foggy window.

The results reported in this paper are preliminary. Throughout the work we were as much concerned with testing and improving the research methods as we were with the research questions. The main question addressed by this paper is "Can existing safety work products be used as research objects to learn about and improve the practices of system safety?"

The specific work products the paper is concerned with are risk assessment reports. Risk assessment is a natural starting point for exploring real world safety engineering practices because:

- it is widely practiced;
- it is typically well documented in a single report; and
- risk assessment reports are often treated as public or non-confidential documents.

3.2 Data Collection

For convenience of reference, the data set for this project is referred to collectively as the "System Safety Zoo – Risk Assessment Reports" (SafeZoo-RAR). Each item in SafeZoo-RAR is self-described as a report of a risk assessment activity. Exactly what is meant by "risk assessment" varies between items, as is discussed further below. SafeZoo-RAR has been assembled by a non-systematic search process combining solicitation and search-engine approaches. The items have been made public by a number of mechanisms:

- regulations which require publication of risk assessments;
- Freedom of Information requests;
- government or local authority information policy;
- publication in support of press-releases;
- provision in response to informal requests for information; and
- publication for no apparent deliberate purpose.

The combination of search method and publication methods means that SafeZoo-RAR is not systematically representative of all risk assessment reports. It is likely to be biased towards industries and organisations with an interest in public disclosure, and in many cases the knowledge that the reports could become publicly available may have influenced their content.

A known bias in the sample is that it excludes industry groups with policy directly requiring secrecy of risk assessments. Specific examples are major hazardous facilities (where revealing risk assessments is considered to compromise national security) and medical devices (where risk assessments are considered proprietary information).

3.3 Composition of SafeZoo-RAR

SafeZoo-RAR consists of approximately one hundred risk assessments. The exact size is fluid – new risk assessments are added to the collection as they are obtained. A permanent method of open access to SafeZoo-RAR has not yet been found. Most of the reports have not been formally published, so there is no reliable method for other researchers to recreate the data set from the names of the reports. However, we do not have license to redistribute the individual reports.

Access to data is an important issue for the research methods we are trialling. Unlike case study research, where replication can be achieved through comparable case studies, researchers attempting to replicate any of our results will need access to the original data set. Whilst in theory a new data set could be assembled, this will only be possible if there are many reports which are readily findable but not found by the SafeZoo-RAR search. For the purpose of current publication we have summarised the reports described in this paper in Table 1 and will provide access on request.

Thirty of the reports within SafeZoo-RAR have been classified according to questions of interest. The details captured for each report are:

1. Title or identifier
2. Purpose
3. Jurisdiction
4. Source of Harm
5. Size
6. Whether the report includes a quantitative assessment of risk
7. Whether the report includes risk identification, or is based on previously identified risks
8. Whether the report discusses uncertainty
9. Whether the report discusses risk acceptability
10. Whether the report documents assumptions
11. Whether the report recommends actions
12. Whether the report discusses rejected actions
13. What targets of risk the report considers

The full SafeZoo-RAR has not been classified, to allow tentative conclusions drawn from this initial set to be tested on further reports.

3.4 Data Analysis

The process of analysis is based on iterative test and improvement of models. Firstly, models are created. These models reflect how we as researchers "expect" that risk assessment is conducted. From these models testable hypotheses are formed – questions that can be asked of the SafeZoo-RAR and answered in the affirmative or negative. These questions are then applied to a subset of SafeZoo-RAR. The result is a set of surviving hypotheses, as well as insights gained from the falsified hypotheses. These are used to form new models and the process is repeated. Several illustrations of this process are provided in Section 4.

4 Application of the Method

Section 2 discusses a range of questions which can only be addressed with real-world data about risk assessment. Among these questions is the relationship between theoretical models of risk assessment and actual risk assessment practices. In this section we show how this relationship can be explored using the SafeZoo method. We take two theoretical models for risk assessment, identify measurable features of the models, and search for those features within a subset of SafeZoo-RAR.

4.1 The Red Book Model

"Risk Assessment in the Federal Government: Managing the Process" (Committee on the Institutional Means for Assessment of Risks to Public Health, National Research Council 1983), informally known as the "Red Book", describes risk assessment as a scientific process which is conceptually and managerially distinct from the political process of risk treatment or acceptance. This is a theoretical model of how risk assessment is conducted. If the model matches reality, there are several hypotheses which would be confirmed. For example

a) Risk assessments would not contain statements of risk acceptability
b) Risk assessment conclusions would be reported in a way which did not imply acceptability or unacceptability
c) Risk assessment would contain statements about uncertainty which indicate whether the conclusions are certain enough to allow decisions about acceptability

These hypotheses can be tested to see whether the model fits each item in a subset of SafeZoo-RAR. Because of the inherent bias in the set, we cannot draw quantitative conclusions, but the overall usefulness of the model can be explored.

When the hypotheses were applied to twenty-three risk assessment reports, ten contained explicit statements of risk acceptability. A further two reports strongly implied acceptability in their conclusions. Three reports quantitatively compared risk to pre-determined benchmarks. Of the remaining reports, six recommended actions in response to the risk, implying that residual risk would be acceptable if the actions were taken. In only two cases was the risk assessed without any implied judgement of the acceptability of the risk.

Nine of the reports discussed uncertainty. In three cases it was explicitly stated that the conclusions could or could not be relied upon based on the amount of uncertainty. In the other cases causes of uncertainty were discussed without making judgements on the acceptability of the uncertainty.

From these findings, two conclusions can be tentatively reached. Firstly, the model of separation between assessment and acceptability is not generally applicable. Secondly, where the model might apply, knowledge about levels of acceptability is often available, informing (and arguably influencing) the risk assessment.

4.2 ALARP Model

"As Low as Reasonably Practicable" (ALARP) is the principle applied in order to meet the United Kingdom (Health and Safety Executive 2001) legal benchmark for risk reduction. At the heart of any practical application of ALARP is consideration of alternative risk reduction strategies (Redmill 2010). Whilst ALARP is only required for certain legal jurisdictions, it is applied more widely, and it is appropriate to consider the extent to which it is used within SafeZoo-RAR.

For assessments which include recommended actions, the ALARP model predicts that the reports would include discussion of risk control measures that are <u>not</u> recommended. This is because in order to determine that risk is ALARP the report must explain why further risk reduction is not practicable.

To test this hypothesis, thirteen reports containing recommendations were considered. Of these reports, only two discussed rejected options. One of these reports was written for the primary purpose of making a selection from several options.

From this result, it can be concluded that ALARP is not a generally applied method of choosing which mitigations to recommend. It would not be appropriate due to the small number of UK reports in the sample (five) to conclude that ALARP is generally not correctly applied in jurisdictions where it is a legal requirement – this would require a larger set of reports all from the same jurisdiction.

5 Strengths and Limitations of the Approach

The approach described in Section 3 and 4 has some inherent strengths and weaknesses described here. Whilst the strengths and weaknesses are apparent in our use of the approach so far, we have insufficient evidence to support or reject claims about the overall efficacy or efficiency of the research method.

For any given risk assessment report, there are objective questions which can be answered. We can explore the methods used to conduct the assessment, the scope of the assessment, whether the report contains common features that undermine risk assessments, and the way the assessment is reported. We may also be able to explore more subjective questions about the values and attitudes reflected in the language of the report and its conclusions.

There are also questions which we cannot answer about each report. Unless specifically mentioned, we cannot know about preparation or training for the risk assessment, and context such as procedures or norms that guided the assessment. We cannot know what decisions were supported by the assessment, or even if the report is an accurate representation of the assessment itself.

To extend the validity of findings beyond the scope of a single report, it is necessary to find patterns within the reports, and then to test these findings on further reports. Without evidence that the data set is representative, there will be a need for more systematic investigation of models that have passed this initial attempt at falsification.

The main strength of the approach is that it provides insight into safety methods as they are practiced rather than as they are academically described. As researchers who are heavily engaged in safety teaching, we are equally interested in evaluating what constitutes good practice, and the weaknesses of current practice.

Beyond individual techniques, we have the opportunity to examine a snapshot of the decision making processes of organisations attempting to manage risk. The existing body of work on sociology of organisations in the lead-up to accidents (Pidgeon 1991) suggests that leaders are forced to apply a form of `bounded rationality' when they think about risk. They cannot pay attention to everything, so it makes sense to devote resources to what they see as important. If the resources are mis-allocated, it appears as if the leaders were wilfully blind to some hazards. Through study of the risk assessments we can see what different organisations consider to be important risks, and how they discuss risks of different types. We can see the basis on which they choose to filter risks, prioritise risks, and determine the adequacy of risk mitigation.

6 Discussion and Observations

There is a large volume of safety work products held within organisations. Each item taken separately may seem of limited research value, but together they provide a cost-effective way of examining safety engineering practice. One fault tree is just a fault tree, but ten fault trees may provide a description of the way fault trees are used, and twenty fault trees may explain the mistakes commonly made in fault trees, and lead to better guidance.

Throughout this work we have been pleasantly surprised by the amount of material we have been able to access. Freedom of Information enquiries have been on occasion refused, and more often simply ignored, but most direct requests for examples or documents referred to in the media have met with positive responses.

7 Further Work

The research approach has proved practical, but has not yet yielded significant results. It is reported here for peer review of the method, and to provide encouragement to others to engage with empirical system safety research.

Our immediate ongoing work is exploring the representation of uncertainty in risk assessments. Initially, we were surprised by the fact that more than half of the reports, including all of those which present quantitative risk data, discuss uncertainty. Prior to this finding we expected that uncertainty would be ignored in most reports. Uncertainty, however, is invariably discussed only in terms of source data. Methodological uncertainty, including fallibility of the risk assessors themselves, is invariably omitted. This is only a tentative conclusion, but we are working on further comparisons of ideal treatment of uncertainty with the sample of reports.

8 References

Ahmed, S., 2007. Empirical research in engineering practice. *J. of Design Research*, 6(3), pp.359 – 380.

Alexander, R.D., Rae, A.J. & Nicholson, M., 2010. Matching Goals and Methods in System Safety Engineering. In *IET System Safety*.

ASSE, 2011. A Brief History of the American Society of Safety Engineers. *American Society of Safety Engineers*. Available at: http://www.asse.org/about/history.php [Accessed May 17, 2011].

Committee on the Institutional Means for Assessment of Risks to Public Health, National Research Council, 1983. *Risk Assessment in the Federal Government: Managing the Process*, Washington, D.C.: The National Academies Press.

Creswell, J.W., 2007. Qualitative inquiry & research design: choosing among five approaches, Sage Publications.

Dhillon, B.S., 1982. Systems safety: A survey. *Microelectronics Reliability*, 22(2), pp.265–275.

Guldenmund, F.W., 2000. The nature of safety culture: a review of theory and research. *Safety Science*, 34(1-3), pp.215–257.

Health and Safety Executive, 2001. *Reducing Risk Protecting People*, HSEBooks.

Hedges, L.V., 1987. How hard is hard science, how soft is soft science. *American Psychologist*, 42(2), pp.443–455.

Howard, G.S., 1993. When psychology looks like a 'soft' science, it's for good reason. *Journal of Theoretical and Philosophical Psychology*, 13(1), pp.42–47.

International Organization for Standardization, 2011. *ISO 26262 Functional Safety*

Pidgeon, N.F., 1991. Safety Culture and Risk Management in Organizations. *Journal of Cross-Cultural Psychology*, 22(1), pp.129 –140.

Rae, A.J., Nicholson, M. & Alexander, R.., 2010. The State of Practice in System Safety Research Evaluation. In IET System Safety. Manchester.

Redmill, F., 2010. ALARP Explored, CS-TR 1197. Available at: http://www.cs.ncl.ac.uk/publications/techreports.

Trevelyan, J., 2007. Technical Coordination in Engineering Practice. *Journal of Engineering Education*, 96(3), pp.191–204.

Wohlin, C., Höst, M. & Henningsson, K., 2003. Empirical Research Methods in Software Engineering. In *Empirical Methods and Studies in Software Engineering*. pp. 7–23. Available at: http://www.springerlink.com/content/UFKGYWVMM VBTPC4M [Accessed August 5, 2010].

Table 1- Risk Assessment Reports

Name	Purpose	Jurisdiction	Source of Harm
Dalgetty Bay Radium Contamination	React to concern	UK	Contaminated Land
Health Risk Assessment Report - Permanente Plant	Risk characterisation	USA	Emissions
Product Safety Assessment Report – Powercode	Risk communication	EU	Toxic component
Installation and Operation of the SV60 Supavac Solids Pump at the Newstan Colliery	Risk Control	USA	Mechanical Equipment
Report on the risk assessment of BZP	Risk characterisation	EU	Drug
Cornhill Caring Community Risk Assessment Report	Risk Control	USA	Human Behaviour
Risk Assessment Report: tert-butyl methyl ether	Risk characterisation	EU	Toxic Chemical
Human Health Risk Assessment Report (West Valley Water District)	React to concern	UK	Toxic Chemical
Oil Spill Risk Assessment for the Coastal Waters of Queensland	Risk Control	AUS	Oil Spill
Afton Wind Farm: Peat Slide Risk Assessment Report	React to concern	UK	Peat Slide
Cadmium (oxide) as used in batteries	Risk characterisation	EU	Toxic Chemical
Risk Assessment for Air Monitoring Results, Hopewell Virginia	Risk characterisation	USA	Toxic chemicals
Whatcom County Ferry	Risk characterisation	USA	Seagoing Vessel
West Louiseville Air Toxics Study	Risk characterisation	USA	Toxic Chemicals
Mercury Spill in Northern Peru	Risk characterisation	USA	Toxic Chemical
Boughton Village Green and Pond	Risk Control	UK	Water (drowning)
Raytheon Company Facility St Petersberg	Risk characterisation	USA	Toxic chemicals
Thames Coast Flood Risk	Risk Control	NZ	Water (flooding)
ACG Oil Spill	Contingency Planning	USA	Oil Spill
Foods Derived from Cloned Cattle and Pigs Produced by SCNT and their offspring	Risk characterisation	Japan	Novel food
Ergonomic Risk Assessment for Naval Medical Branch Office	Risk Control	USA	Ergonomics
Installation and Operation of the SV60 Supervac Solids Pump	Risk Control	USA	Equipment
Feasibility Study for the Protection of the Entrace to Kilkeel Harbour	Selection of Mitigation	UK	Exposed Harbour
Crindau, Piling Risk Assessment Report	Risk Control	UK	Contaminated Land
Geotechnical Risk Assessment for Galore Creek	Risk characterisation	Canada	Mining Structures
Adams Arrow DP2 FMEA	Risk characterisation	UK	Ship Control System
Quantitative risk assessment (QRA) of the BSE risk posed by processed animal proteins (PAPs)	Risk characterisation	EU	Unsafe food

Risk Assessment to Humans Posed by the Dingo Population on Fraser Island	React to concern	AUS	Wild Animals
Flam Store Risk Assessment	Risk Control	UK	Flammable Material
Analysis of Re-entry Survivability of UARS Spacecraft	Risk characterisation	USA	Spacecraft re-entry

CONTRACTING FOR ASSURANCE
OF MILITARY AVIATION SOFTWARE SYSTEMS

Squadron Leader D.W. Reinhardt
Royal Australian Air Force
Research Student
University of York
derek.reinhardt@defence.gov.au

Professor J.A. McDermid OBE FREng
Head of Department of Computer Science
University of York
United Kingdom
john.mcdermid@cs.york.ac.uk

Abstract

Contracts are instruments which provide a legally binding agreement for the purchase/exchange of goods or services. While both civilian and military aviation software systems are acquired by contract, in the military circumstance the contract has an additional regulatory and safety assurance role.

Military contracts typically achieve the regulatory and safety assurance outcome by ensuring that relevant contract clauses reference applicable regulations and safety standards. However, industrial practice suggests several key factors that influence the effectiveness of the contracting approach to achieving safety. For military aviation software systems, these factors seem to be particularly prevalent.

The paradigm of the standard (i.e. goal-based, prescriptive or combinations thereof) is a factor as it influences the perspectives and behaviours of suppliers and acquirers with respect to evidence provision to the regulatory authority. Another prevalent factor is the extent to which the standard guides the effective establishment and execution of a contract through providing certainty in both product and evidence delivery. Standards may also have a substantial impact on achieved product safety.

This paper examines these factors and aims to assess their effect on military aviation software system contracts. The paper sets out a framework for relating evidence to safety objectives. The framework also provides an approach for identifying, analysing and evaluating the tolerability of limitations (e.g. incompleteness) in evidence for assuring safety. A fictional example is presented to demonstrate application of the framework to the contracting process. Observations on evaluation of the framework are presented to provide support to their validity in industrial practice.

Keywords: Architecture, Assurance, Aviation Systems, Contracts, Fault Tolerance, Safety, Software Assurance, Software Safety, Tender.

1 Introduction

Contracts are instruments which provide a legally binding agreement for the purchase/exchange of goods or services. A contract normally consists of terms and conditions, and is supported by technical annexes to define the requirements for goods/services and scope of work. For aviation systems, contracts are used for the acquisition and/or modification of these systems between the developer/manufacturer (i.e. supplier) and the owner or operator (i.e. acquirer). While both civilian and military aviation systems are acquired by contract, there are key differences in the role of contracts between the military circumstance and the civilian circumstance. Specifically in the military circumstance, the achievement of regulatory and safety assurance functions has to be enabled through the contract. This is because the regulations and safety requirements established by military regulators are not legally enforceable onto a supplier unless the contract enables this. This is very different to the civil case (e.g. the Federal Aviation Administration (FAA) or Civil Aviation Safety Authority (CASA)) where the responsibility to promulgate and enforce regulations on suppliers is enshrined in law.

Military contracts typically achieve the desired regulatory and safety assurance outcome by ensuring that relevant contract clauses reference the applicable regulations and safety standards. However, on its own, this may be insufficient. The authors' practical experience suggests several key factors that influence the effectiveness of the contracting approach to achieving safety regulation. For example, the clarity within the nominated standard of the requirements for evidence provision from supplier to regulator seems to be a major factor. For military aviation software systems, these factors seem to be particularly prevalent. This paper is focussed on military software systems, however due to the unavoidable coupling between software and its system, where relevant this paper may take the perspective of the system, software or software system. For ease of discussion, we assume that certification is based on the delivery of (safety) arguments and supporting evidence to the acquirer.

1.1 Standards Paradigm: Goals-based or Prescriptive.

The paradigm of the standard (i.e. goal-based, prescriptive or combinations thereof) is a crucial factor for achieving regulation through contracts as it influences the perspectives and behaviours of suppliers and acquirers regarding the provision of evidence to the regulatory authority. For example, a goal based standard might set high level safety objectives and permit substantial flexibility for designs, which gives benefit in defining effective products. However it may have limitations with respect to establishing contractually enforceable benchmarks for evidence provision; and this will impact suitability and sufficiency of both evidence and argument. Similarly, resolution within the contract, of evidence and argument shortfalls might be equally limited, depending on the supplier's attitude and perspective.

On the other hand, a prescriptive standard may set clear benchmarks for evidence and activity completion that are straightforward to enforce through contractual mechanisms, but have limitations in relevance to achievement of product safety objectives. This means that, depending on the supplier and acquirer's bias in worldviews (see [McR12]), the paradigm choice will affect behaviours, and these behaviours will ultimately affect the level of safety (not just evidence provision) achieved through the contract.

The question of paradigm is further complicated for complex aviation systems involving technologies (e.g. software) where failures are (predominantly) the consequences of systematic faults. This is because, across academia and industry, there is still limited consensus (refer to [JTM07], [McD07], [McK06], [NTS06], and [Wea03]) as to how to provide assurance that these faults do not lead to unacceptable aircraft failure conditions. All that can be concluded from this lack of consensus is that current approaches to providing safety assurance of software in military aviation systems have limitations. Thus, as neither paradigm is without its limitations in this context, it is likely that the more effective approach may be a compromise between both paradigms. This raises the question, what combination of goal-based and prescriptive standards elements is necessary to minimise these limitations and enable effective safety regulation via contracts?

1.2 Integrating the Standard's Lifecycle with the Tender/Contract Lifecycle

Another important factor is the way the standard integrates with the contractual lifecycle. Ideally the standard should assist in reducing uncertainty about the delivered product, argument and evidence prior to the establishment of a contract. This is important because both acquirer and supplier will be seeking confidence that the contract will be successful prior to entering into the contract. Similarly, the standard should assist during contract execution. Should safety issues emerge during the contract, then timely and cost effective resolution will be a goal for both supplier and acquirer. The contract and standard should support the resolution of safety issues, and not hinder it by contributing to dispute.

An inspection of contemporary safety standards reveals that integration between the standard lifecycle and contract lifecycle varies significantly between standards. For example ARP4754 and RTCA/DO-178B make no mention of integration with contracts as the means of evidence provision. However, they effectively achieve some potential contract integration through certification authority liaison and artefact requirements within these standards. UK Defence Standard 00-56 Issue 4 makes numerous mentions of contracts and requirements on contractors, but doesn't provide requirements for contracts relating to the provision of arguments or evidence across the contracting process. Whereas, MIL-STD-882C and D deals explicitly with contract integration, include specific references to contract clauses, tender processes and data requirements.

It is evident that the requirements of the standards have a substantial effect for the integration of the standard across the tender/contract lifecycle. This raises the question, what elements of standards, and their implementation in contracts provides appropriate certainty (regarding product and assurance evidence) for acquirers and suppliers? Is it possible to define requirements for safety and assurance standards to achieve effective contract process integration?

1.3 What Does This Mean for Standards and Contracts?

Ultimately, it is vital that the regulatory and safety assurance paradigm used be compatible with the contracts used for military acquisitions, without impairing or detracting from the achievement of system safety. Success is dependent on perspective and worldview. Contracts which provide cost and schedule certainty are preferred by both suppliers and acquirers. Suppliers will also have a vested interest in profitability and acquirers in value for money. Suppliers will generally strive to achieve safety, and the acquirer's regulatory authorities will strive for achievement of an acceptable level of safety (or risk) without significant out of scope rework to treat risks, or without the retention of intolerable risks. How to do this for the assurance of military aviation software systems is still very much a challenge.

This paper further examines the different standards and contracting paradigms, and aims to assess their effect on military aviation software system acquisition. This paper articulates more generic principles learned as a result of defining a framework [ReM11] by which to contract for architectural assurance [ReM10], and to provide claims and evidence assurance [RMc10] for aviation systems.

2 Why Military System Acquisition Contracts are Different

In civil aviation the regulator responsible for airworthiness is a government agency (e.g. the FAA). The regulator is a legally recognisable independent entity from the supplier and acquirer of aircraft and aviation systems. Regulations established by the regulator are indoctrinated in law and are legally enforceable. However, in the military aviation domain, the regulator is typically part of the same high level organisation as the acquirer. For example in the Australian Defence Force, both the Directorate General Technical Airworthiness (regulator) and the Defence Materiel Organisation (acquirer) are part of the Commonwealth of Australia – a single legal entity in the eyes of the law. In the United Kingdom, the Military Airworthiness Authority (regulator) and the UK Ministry of Defence (acquirer) report to the Secretary of

State for Defence and are part of the Crown – again a single legal entity in the eyes of the law. The same can also be said for the relationship between military regulators and acquirers in the United States of America. This relationship between the acquirer and regulator roles has several implications for the way airworthiness is regulated; of which one significant factor highlighted in Section 1 is the impact on contracts between suppliers and acquirers. Regulatory enforcement is enabled by the contract rather than via laws for the military circumstance. The following subsections elaborate several impacts for contracts.

2.1 Enforcement of Design Requirements

In civil aviation, the supplier is required to supply aircraft and aviation systems that meet the applicable airworthiness design requirements promulgated by the regulations. For example, the civil airworthiness regulations (e.g. [14CFR25]), and their supporting guidance in the form of advisory circulars, orders and notices, define a substantial set of design requirements for their applicable aircraft category. These are usually supplemented by additional design requirements agreed between the supplier and regulator throughout the certification process. Design requirements are typically in the form of product requirements and assurance requirements (which includes evidence, verification, etc.). However, in military aviation, the airworthiness design requirements (or requirement to establish and agree them) must be included in the contract if they are to apply to the development. This means that the contract Statement of Requirement (SOR) should include or reference applicable airworthiness design requirements, including safety assurance requirements, and that the Statement of Work (SOW) must include activities to ensure elicitation and agreement of any additional airworthiness or design requirements relevant to the design. This is no simple task, as the set of potentially applicable airworthiness design requirements may be large and complex. In the context of military aviation software systems, the subset of applicable design requirements includes assurance requirements, in addition to a range of 'product' design requirements, depending on the system application; these assurance requirements are the main focus of this paper.

2.2 Obtaining Assurance Evidence

In civil aviation, the regulator obtains evidence required for certification from the supplier as required by the regulations. The regulations will require the supplier to provide the regulator with plans, artefacts (inspection, analysis and test documentation), access for the purposes of audit, access for the purposes of witnessing / participation / conduct of tests, etc. However, in military aviation, the regulator (as part of the acquirer) obtains these types of evidence required for certification from the supplier via the contract. This means that the contract SOW must include applicable activities for the generation of relevant certification evidence, including assurance evidence. Delivery versus access to evidence is usually dictated by intellectual property considerations, and will be evident from the artefacts listed in the Contract Data Requirements List (CDRL), and supporting Data Item Descriptions (DIDs).

2.3 Resolving Shortfalls in Assurance Evidence

In civil aviation, if there are shortfalls in the supplier provision of evidence to the regulator for certification, then the onus is on the supplier to resolve the shortfalls. If the supplier doesn't resolve the issue then they don't achieve certification, and they can't sell their product. However, in military aviation, resolving the shortfall in evidence will very much depend on whether it is in or out of scope of the contract. In many respects the acquirer can be considered to have already purchased the product once the contract is signed. If the issue is within scope, then the onus is on the supplier, but if there is any ambiguity regarding scope of the contract pertaining to the issue, then the onus for resolution is shared by the acquirer. If the supplier and acquirer can't agree that it is wholly within the scope of the contract, then the issue may be the subject of contractual dispute. Ramifications of a contractual dispute can include cost and schedule implications, a requirement to elevate beyond project staff, a requirement to negotiate over contractual interpretation and compliance, etc. These issues potentially have the impact of degrading the effectiveness of safety regulation achieved through the contract, particularly where projects must seek additional funding from Government (an onerous process) to resolve the safety shortfalls via contract change proposals.

3 Impact of Uncertainty at Contract Signature

Section 2 has identified several responsibilities of contracts if safety regulation is going to be effective via the contract. Uncertainty in any of these may increase the risk of the contract being unsuccessful. Signing a contract, in some respects, involves a gamble. It is a wager for both supplier and acquirer that the supplier can provide a system that the meets the acquirer's requirements within the cost and schedule dictated by the contract. The odds (for or against) depend on the uncertainty in factors important to either supplier or acquirer. Therefore, any sensible gambler (and one that abides by causality) will acknowledge that the contract success risk is a function of the uncertainty at contract signature. Lots of uncertainty, and the odds could be dramatically against success; lesser uncertainty, and the odds might favour success. Fortunately the normal processes for getting to contract signature such as project definition and tender phases provide the contract authority with a means of seeking important information prior to contract signature. This information, if sought and used effectively, can reduce uncertainty, and thus reduce potential contract risks.

How to seek the right information and effectively evaluate it with respect to safety for military aviation software systems is still very much a challenge. Furthermore the existing standards and contracting approaches offer limited guidance on how this might be achieved effectively. Industrial experience involving project overruns and cancellations due to safety assurance concerns suggests that the current approaches are also insufficient, although mostly the evidence is anecdotal.

To further understand the implications of uncertainty at contract signature for safety it is necessary to establish where this uncertainty might exist. To elicit this, consider the factors outlined in Sections 2.1 through 2.3 with respect to a military aviation software system and safety. In this context, uncertainty might exist with respect to the following:

- Will the design requirements proposed by the acquirer be adequate to achieve the safety objectives? Specifically, from a safety assurance perspective, will:
  - the software and system architecture, including the use of redundancy, diversity, and fault avoidance/tolerance likely permit achievement of the safety objectives?
  - the architecture provide adequate protection against systematic faults and failures?
- Will compliance with the design requirements and safety objectives be compelling based on the evidence provided? Specifically, will:
  - the behaviours of the system and its software be sufficiently understood and valid under both normal and failure circumstances?
  - these behaviours be appropriate with respect to safety?
  - the evidence support the safety assurance claims made by the supplier about these behaviours?
  - any limitations in evidence be tolerable?
- Will limitations in evidence be resolvable within the scope of the contract? Specifically, what is:
  - within scope?
  - out of scope, requiring a contract change?

Whenever there is uncertainty with respect to these questions throughout the contract lifecycle, then the contract risks relate to the following issues. The first is that the uncertainty might undermine the acquirer's aspiration to establish if the software system will likely be acceptably safe (if this supplier were to be chosen to contract with). Thus the supplier might be eliminated during the tender evaluation based on perceived uncertainty in suitability. The second, and ultimately more serious, issue is that if this design solution is contracted for, and it turns out the design has unsuitable behaviours; in this case there is risk that the acquirer may not be able to complete safety certification within the scope of the contract. Worse still, it may require the acquirer to retain risks, due to uncertainty, and these risks prove to be intolerable in practice.

If we extrapolate these factors alone, then the result is easy: have the supplier provide full disclosure to the acquirer during the tender process. However, the realities of the commercial business environment quickly show the impracticality of this aspiration. In domains where developmental and novel systems are more common-place, it is uneconomical to require suppliers to complete their development lifecycle to the point that answers to the above questions become entirely certain during the tender process. As only a small percentage of tender responses are actually successful, and tenderers already invest substantial resources in preparing them, the acquirer must be cognisant of the need to avoid deterring potentially suitable tenderers due to the level of effort required to tender. Therefore, in establishing the level of detail required in the tender response the solution must provide for sufficient disclosure and understanding, but while ensuring the minimum imposition on tenderers. This is a difficult balance.

Acquirers and suppliers enter into the tender and contracting activities with a set of motivations, aspirations and perspectives which are a unique dynamic contrast between goals for specific project success, mixed with broader commercial goals and commercial restrictions. Each of these will vary between every acquirer, supplier and circumstance. The most obvious motivations for the acquirer and supplier with respect to safety are that the solution will achieve the safety objectives, and that the evidence will show this. But it is the additional motivations that vary the perspective on achievement of this between supplier and acquirers. Acquirer motivators include:

- credibility of supplier cost and schedule forecasting,
- satisfying capability requirements,
- avoiding contract changes,
- costs of solutions falling within notional budgets, and
- delivery within capability scheduling requirements.

Supplier motivators include:

- providing a competitive tender cost/schedule,
- preservation of profit margins within the contract price,
- avoidance of contract penalties,
- ensuring that out of scope work requires a contract change (to protect the profit margin with the contract), and
- delivery of a broadly satisfactory product with minimal application of resources.

These motivators are intrinsically linked because cost and schedule are required to produce evidence, and evidence is required to show the provided solution meets safety objectives (and capability requirements). Because of this dependency, some of these motivators will work against each other, and this will cause divergence in supplier and acquirer motivations, and thus behaviours. Emergent (commercial) behaviours when issues arise that expose the polarisation between these motivators very much depends on the relationship between supplier and acquirer, the seriousness of the safety concerns or cost impacts, and the supplier's and acquirer's worldviews regarding assurance.

Given these contracting motivators, and assuming that any serious incompatibility between them for a given contract will result in limitations in successful outcomes for the contract: how might a framework be established to ensure that uncertainty at the time of contract signature can be bounded? I.e. what is the compromise between these motivators that enables the appropriate design solution to be identified during tender processes, and this solution to be achieved during contract execution?

The remainder of this paper examines how an approach might be established. Illustration of the benefits of the approach will be via an artificial but realistic example.

Consider an upgrade of an analogue flight control system to a digital flight control system for a military helicopter. The flight control system provides automatic flight functions and stability augmentation, and is mixed to the existing mechanical control system between pilot controls and control actuators. The objective of the acquirer is to achieve this upgrade, including the safety regulatory functions on behalf of the acquirer's regulatory authority, through a contract. The following sections examine how this can be effectively achieved.

4 Bounding Uncertainty Prior to Contract Signature – Successfully Using the Tender Process

It has already been mentioned that the tender phase provides a means for the acquirer to seek important information prior to contract signature. This information, if sought and used effectively, can reduce uncertainty, and thus reduce potential contract risks. How much the uncertainty has to be reduced is an important question, and this introduces the concept of bounding uncertainty.

Firstly, it is important to elaborate what is meant by bounded uncertainty, in this context. Put in engineering terms, it is establishing limits (upper bounds) on the cost of producing a safe product and an acceptable safety case. Bounds can be narrowed by the provision of information to the acquirer from the supplier during pre-contract phases (e.g. tender phase) balancing the motivators identified in the previous section. The limiting factor on information provision will be the affordability, for a tenderer, of conceptual and preliminary phases of requirements and design lifecycle phases within the resources that are commercially viable given the gamble of winning the tender.

In Section 3 a set of questions were introduced based on the three identified roles for contracts with respect to safety regulation: enforcement of design requirements, obtaining assurance evidence, and resolving shortfalls in assurance evidence. These questions were further refined into the context of military aviation software systems to seek information the regulator would require to be informed about safety assurance. These questions were holistically centred on three main topics: architecture, behavioural arguments and evidence provision/suitability.

Therefore, an approach to breaking this problem down further would be to examine how to bound uncertainty across each of these three topics. I.e. to effectively determine how much the regulator should know about each of these topics during the tender phase to be satisfied of a likely positive outcome, should the project go to contract.

Returning to the artificial flight control system example, let's assume that the contract authority for this project has determined that an open tender is the most suitable form of acquisition strategy for this project. The aircraft original equipment manufacturer has no off-the-shelf solution available, and various contractors have expressed interest in developing a solution.

The remaining sections of this paper will now describe how this tender may be prepared and evaluated, the most suitable option identified, and a contract established and executed for this option. Section 5 of this paper will consider the architectural topic, what information is required to inform acquirers about architectural suitability and how this information can be elicited in the pre-contract signature phases. Section 6 of this paper will consider the behavioural arguments and evidence topics, what information is required to inform about sufficiency, and how this information can be elicited by the pre-contract phases. Section 7 will then examine how issues arising as a result of the remaining uncertainty are identified and resolved post contract signature.

The example being used within this paper assumes a single phase tender process. However, this process may not always be the most suitable. Where the acquisition or modification is of substantial complexity, then the single phase tendering process may not incentivise suppliers to invest a level of effort to develop their solution to a level that permits effective evaluation. This may particularly be the case for an entire aircraft development. In these cases a two-phase tender may be more suitable. The first phase would identify holistic solutions that accord with the safety objectives of the program and use a normal tender construct. The second would be a partially funded tender phase, where funding is provided to a restricted set of tenderers to further develop the tender artefacts supporting evaluation against the framework. The second phase would be more synonymous with a Restricted Tender, but include provision for funding so that tenderers can invest a level of effort which they are compensated for. Such options are available where the acquirer is not satisfied that the tenderer is incentivised to offer competitive solutions, or to resolve the uncertainty to a level consistent with the constraints on acquirer funding. These multi-phase tender processes won't be directly addressed in the example used in this paper, but the concepts illustrated herein can be applied to those circumstances also.

5 Obtaining Solution Architectural Certainty

Obtaining architectural certainty from the tender phases and prior to entering into a contract is important as it enables early insight into potential architectural shortfalls. It also forces supplier consideration of architectural suitability including fault avoidance and fault tolerance; this is important as there is evidence in industrial practice that this is sometimes overlooked. A four step process is proposed for obtaining solution architectural certainty, as follows:

1. Set measurable benchmarks for architectural suitability
2. Inform architectural suitability using the tender process
3. Evaluate architectural suitability during the tender evaluation, and
4. Provide architectural assurance during contract execution.

The following sub-sections elaborate the four step approach to achieving this for the flight control system example and outline some of the benefits.

5.1 Setting Benchmarks for Architectural Suitability

The first step to obtaining architectural certainty is to set some benchmarks for solution architectural suitability. The benchmarks should not be specifying solutions so they do not stifle novelty or limit flexibility; they should instead set measurable criteria against which different solutions can be evaluated. In this way, benchmarks allow the acquirer a way of measuring solutions against each other from a safety perspective. Benchmarks also provide a way of specifying to a supplier what attributes their software system design should have.

A review of the literature reveals that there is very little published guidance on explicit benchmarks for architectural suitability, particularly with regards to systematic faults and failures. Some standards permit assurance levels on specific system components to be reduced based on architecture, but this is not a measure of the overall architectural adequacy. Therefore, new approaches are required to achieve this if architectures are to be effectively evaluated during tender evaluations. One possible approach has been developed by the authors that introduce the concept of an Architectural Safety Assurance Level and Layered Fault Tolerance Requirements [ReM10]. The core idea with the Architectural Safety Assurance Level is that it provides a measure of how many layers of defence an architecture provides against systematic faults. The layering defences against faults concept is synonymous with the 'defence in depth' principle often referred to in security manuals. It also derives from the 'Fail Safe Design Criteria' from [AC25.1309]. The 'layers of defence' concept is a useful measure because it is independent of specific solutions, emphasises architectural handling of faults between architectural components, and provides a notional level of confidence based on the number of layers of defence against each fault type.

To set the benchmark for the supplier, clauses could be developed for both the tender and contract SOR to communicate these benchmarks. The clauses should communicate the solution properties regarding the requisite number of layers of fault tolerance and avoidance/detection and handling requirements. The following is an example of a generic SOR clause to achieve this:

The [System Name] architecture and mechanisms for achieving fault avoidance and fault tolerance, against each type of credible systematic fault, shall meet the requirements for layers of fault avoidance and fault tolerance, where the number of layers is commensurate with the worst credible failure condition, as specified at {reference a Table in the SOR detailing the benchmark numbers of layers for each failure condition severity}

A specific instantiation of this clause for the Architectural Safety Assurance Level approach is described at [ReM11].

5.2 Informing Architectural Suitability

To reduce architectural uncertainty at the time of contract signature, the tender phase requires a mechanism to be informed of the architecture. This implies that a tender deliverable needs to include information about the suitability of the proposed architecture. Since the information will be used by the acquirer to evaluate the suitability of the architecture against the benchmarks, it is useful to ensure the information directly addresses the benchmarks set in Section 5.1.

One possible approach would be to require the tenderer, through the tender SOW, to provide a *Conceptual System and Software Architecture Suitability Documen*t. The document would describe how the system's architecture and mechanisms for achieving fault avoidance and fault tolerance against systematic faults would meet the benchmarks established above. The intent is to provide a description of the architecture at a level of fidelity that the acquirer can evaluate against the benchmark, without forcing the supplier to completely design and implement the system before contract signature. For a largely mature design, the document can focus on what already exists, and whether or not it requires supplementation; for a developmental design it provides a framework for the supplier to cost the architectural elements of their system with improved accuracy. The following is an example of the generic Tender SOW clauses to achieve this:

Total Layers of Defence. The [Tenderer] shall prepare a [Conceptual System and Software Architecture Suitability Document] per TDRL XX to describe how the [System Name] architecture and mechanisms for achieving fault avoidance and fault tolerance, against each type of credible systematic fault, is proposed to meet the {reference to SOR's requirements for number of layers of fault avoidance and fault tolerance to systematic faults}.

Adequate Constraints. The [Tenderer] shall prepare a [Conceptual System and Software Architecture Suitability Document] per TDRL XX to describe how each proposed constraint (i.e. absence/detection and handling mechanism) is proposed to achieve the architecturally layered fault tolerance requirements as defined by the SOR {reference the SOR requirement}.

A specific instantiation of these clauses for the Architectural Safety Assurance Level approach is described at [ReM11].

For the flight control system example, let's assume that each of the proposed options provides a *Conceptual System and Software Architecture Suitability Document*, for which the proposed architecture is briefly summarised as follows:

- Option A
    - Quad redundant digital flight control system incorporating two flight control computers with two independent channels per computer.
    - Dual sensors including air data systems, attitude/heading reference systems and triplex actuators and actuator sensors.
    - Incorporation of software fault tolerance within each computer.
- Option B
    - Quadruplex digital flight control computers incorporating a single channel per computer.
    - Incorporation of software fault tolerance within each computer.

- Option C
  - Quad redundant digital flight control system incorporating two flight control computers with two independent channels per computer.
  - Sensors include a single air data system, dual attitude/heading reference systems and dual actuators and actuator sensors.
  - Design is based upon a flight control system from a fixed wing military aircraft, and adapted for this application.
- Option D
  - Simplex digital control system, single control panel, and single sensors including air data system, attitude and heading references, and actuator position sensors.

Note that these architectural descriptions are deliberately brief. They are intended to be illustrative for the purposes of making a point about how contracting processes can be used to inform their suitability. A more detailed example, which includes a more thorough architectural analysis, is to be documented within the first author's PhD thesis.

5.3 Evaluating Architectural Suitability

The purpose of the tender requesting this information is to permit evaluation of the extent to which the holistic safety and software architecture requirements are costed into the tender response. The retrospective incorporation of constraints to treat systematic failure modes is rarely straightforward, particularly when architectural change is required. Therefore, it is in the acquirer's interests to establish the extent to which the contractor has determined an architecture based on the types of constraints required to meet safety objectives. While it is recognised that many sub-system architectures may not be well defined for large system acquisitions, the absence of this information in a tenderer's response will permit the acquirer to adjust the contractor's proposed costing by a risk figure based on the amount of uncertainty (or extent of suitability) in the tenderer's proposed architecture to provide a normalised evaluation of tenderer's responses that do include the relevant information.

As can be seen from the differing architectures proposed by Options A through D, the complexity of each solution differs notably. Using the benchmarks set for the architecture, each option is evaluated. The evaluation results are summarised as follows:

- Options A and B – Treatments to all classes of systematic fault use layers of fault avoidance and fault tolerance mechanisms. Architecture is suitable.
- Option C – Treatments relating to omission and value failures of the air data system sensor rely on fault avoidance via absence arguments only. There is limited software fault tolerance proposed for these failures. Therefore the architecture is deemed to contain weaknesses against these systematic faults and thus would require changes to adequately treat. Architecture is potentially unsuitable, and is flagged for further consideration once evidence provision is evaluated.
- Option D – Treatments relating to omission and value failures of sensors and flight control computers rely on fault avoidance from absence arguments only. This is assessed to provide grossly inadequate defences against these classes of systematic failures. Architecture is deemed unsuitable, and option is eliminated from the tender.

5.4 Providing Architectural Assurance

Once the preferred tenderer has been identified, and any uncertainties regarding the architectural assurances are tolerable (assuming in this case that it will end up being either Option A or B because of their superior architectural suitability), then it is possible to develop a contract between the supplier and acquirer.

Under the contract, the acquirer will need to achieve two things. The first is that they will need to maintain the benchmarks for product suitability by inclusion of SOR clauses similar to those defined in Section 5.1, but for the contract. Further the acquirer will require means to establish if the final 'as-delivered' architecture meets the prescribed benchmarks. This can be achieved by requiring the contractor to deliver (via appropriate SOW contract clause) a *System and Software Architectural Assurance Document*. The document would describe how the system's architecture and mechanisms for achieving fault tolerance against systematic faults actually achieves the benchmarks established above. The following is an example of the generic Contract SOW clauses to achieve this:

Total Layers of Defence. The [Contractor] shall prepare a [System and Software Architectural Assurance Document] per CDRL XX to describe how the [System Name] architecture and mechanisms for achieving fault avoidance and fault tolerance, against each type of credible systematic fault, meets the {reference to SOR's requirements for the number of layers of fault avoidance and fault tolerance to systematic faults}.

Adequate Constraints. The [Contractor] shall prepare a [System and Software Architectural Assurance Document] per CDRL XX to describe how each proposed constraint (i.e. absence/detection and handling mechanism) achieves the architecturally layered fault tolerance requirements as defined by the SOR {reference the SOR requirement}.

A specific instantiation of these clauses for the Architectural Safety Assurance Level approach is described at [ReM11].

The Contract Data Requirements List (CDRL) should require that various iterations of the document be delivered at relevant system engineering milestones to permit the acquirer to monitor the evolution of the architecture under the contract. This monitoring is important because it allows the acquirer to measure the progression of the architecture throughout the contract lifecycle, and to respond early if there are divergences to acquirer understanding and assumptions from the tender evaluation.

Obviously Data Item Descriptions (DIDs) will be required for all the deliverables listed in the CDRL (or TDRL mentioned in the previous section). DIDs are generally structural, and could be developed to provide a specific heading framework to support provision of the relevant information. However the SOR clauses setting benchmarks for the product, and the SOW clauses requiring provision of the information are the means by which the adequacy of the architecture is enforced. DID compliance is only a means of ensuring potentially relevant information has been provided in a structure that is understood by the acquirer.

6 Obtaining Argument and Evidence Certainty

Obtaining argument and evidence certainty from the tender phases and prior to entering into a contract is important because it enables early insight into potential argument and evidence shortfalls. It also forces explicit context specific agreement between acquirer and supplier on the measures of argument and evidence sufficiency for which there is no agreed universal approach. A four step process is proposed for obtaining argument and evidence certainty, as follows:

1. Set benchmarks for argument and evidence suitability
2. Proposal of argument and evidence using the tender process
3. Evaluate argument and evidence suitability during the tender evaluation, and
4. Provide argument and evidence assurance during contract execution.

The following sub-sections elaborate the four step approach to achieving this for the flight control system example and outline some of the benefits.

6.1 Setting Benchmarks for Argument and Evidence

The first step to obtaining argument and evidence certainty is to establish how to set benchmarks for argument and evidence sufficiency. In keeping with the notion of a compromise between goal-based and prescriptive standards, the benchmarks should not specify specific techniques or methods for evidence generation, but instead provide a coherent framework for how evidence will be related to safety properties, and provide a set of criteria for establishing when evidence generation is completed. In this way, benchmarks allow the acquirer a way of measuring evidence sufficiency from a safety perspective.

A review of the literature reveals that there is very little literature in the public domain that sets explicit benchmarks for measuring argument and evidence sufficiency. The generalised goal-based approaches provide flexible argument structures [Kel98], and the development of patterns and anti-patterns has provided some reusable argument structures that might provide the basis for argument agreement [KeM01]. Argument assurance [Wea03] and assurance deficit approaches [SSEI09] provide an approach, but they lack detail on evidence sufficiency benchmarks sufficient to reach a consensus before contract signature. Less generalised goal-based (or objective-based) approaches such as RTCA/DO-178B provide a detailed framework of sub-objectives that would form part of an argument structure, but unfortunately stray into prescription in some limited areas [Rei08]. In contrast, prescriptive standards provide very clear measures of evidence completion, but are lacking in justification for evidence sufficiency in a given context. Therefore, new approaches are required to achieve this if arguments and evidence are to be effectively evaluated during tender evaluations.

6.1.1 Benchmarks for Argument

First, we address the question of argument. Having an entirely flexible argument is useful in that it does not constrain design solutions, the claims that can be made about them, and does not limit novel approaches to arguing safety. Further, this approach means that the argument has the flexibility to present evidence that is important to the argument, rather than producing evidence because the standard requires it (as with the prescriptive standards). But the drawback is that it is very difficult to communicate acquirer expectations to the supplier if the overall approach doesn't provide a way for the supplier to measure the suitability of their design solution and argument. It should also be apparent when inappropriate design solutions are proposed and inappropriate claims used to defend them.

To bound the uncertainty such that the acquirer can be confident in the supplier's intended argument approach, a means is required to convey the attributes of acceptable arguments to the supplier through the tender and contract documents. The purist goal-based approach doesn't achieve this. On the other hand an entirely prescriptive standard provides a set of evidence that the supplier should produce, but the argument relating the evidence to the behaviours of the product and the safety claims may be either implicit, missing in part or missing entirely. Thus a move to activity and technique prescription doesn't address the need of contracts either. So how can these approaches be combined without undermining their advantages, while ensuring their usefulness as a contracting benchmark?

Let's consider any argument as consisting of some holistic claim about a property of a product with respect to safety, and a strategy for showing the credibility of this claim. This emphasises two key points: the claim and the strategy. This could be considered analogous to the relationship between the Goal and its Strategy in Goal Structuring Notation (GSN) as described by [Kel98].

Consider the claim first. At the architectural level, architectural assurance is based upon the presence of layers of fault avoidance or fault tolerance, such as detection/handling mechanisms. Let's call the requirements that define the specific fault avoidance or fault tolerance behaviour at the relevant layer a 'constraint', as a generalised term. Therefore it follows that an argument is required for the suitability of each 'constraint' and that each 'constraint' needs to be assured commensurate with its impact on safety. The architectural suitability elements of Section 5 provide a means for establishing the collective suitability of 'constraints', and how their behaviours combine to provide the requisite architectural defences against systematic faults. Thus we are left with providing evidence that each individual 'constraint' is assured, and we need to turn our attention to the strategy to achieve this. Note that the 'avoidance' constraint amounts to correctness, e.g. of control algorithms).

Consider this; what if the general evidence types used to support claims about the 'constraint' were categorised in with respect to software lifecycle products for which there is consensus. For example, current standards almost universally agree that there should always be:

- requirements at the system level,
- one or more design decompositions and refinements of these requirements (e.g. high level software requirements, abstract software requirements, low level software requirements),
- source code, and
- executable object code.

These are real software lifecycle products, and they exist as some form of physical document or electronically for virtually all developments. When they are lacking, it is not because they are inappropriate, it is because there is a gap in evidence. Further, since they appear in all of the existing software assurance standards, we can utilise the consensus this provides. There is some dispute that contemporary methods such as model-based software engineering undermine these general categories. However, consider this perspective. Model-based software engineering simply changes the sources of evidence for these products from human centric processes to tools. The evidence still exists; it just takes a different form depending on the construct of the tool. Such product benchmarks also provide a rationale for the types of evidence model-based software engineering tools should produce as their output, and this may help with establishing criteria for the qualification of such tools. Hence this paper argues that the categories of life-cycle products should still exist; it's just the source of evidence that changes (i.e. human to tool).

Examining the strategy in more detail, why not structure a set of generic sub-claims around 'attributes' of the aforementioned software lifecycle products (high level requirements, low level requirements, source code, executable object code, etc.). For example an attribute of a low level requirement might be its 'traceability' to higher more abstract level requirements. Numerous attributes (e.g. accuracy, consistency, traceability, compliance, verification coverage, etc.) can be defined which represent the extent of properties appropriate to the software lifecycle product.

Each 'attribute' would describe a distinct property of the evidence, such that collectively the properties would provide measurable knowledge in the claims made from the software lifecycle product. Further, instead of making the starting point of requirements entirely general (as is done in most software assurance and safety standards), ensure that they are examined with respect to real product behaviours that affect safety - in this case each specific 'constraint'. Effectively, we are explicitly annotating the 'attributes' of each software lifecycle product, with respect to the claims about the specific 'constraint'. This provides a generic universal approach to linking software lifecycle products (i.e. the real world evidence) with the properties of the software we are trying to make safety claims about.

One possible approach has been developed by the authors' (see [RMc10]) that introduces the concept of a Claims Safety Assurance Level (CSAL), and a set of generic arguments centred around the 'attributes' of lifecycle products of specified 'constraint' level requirements and applicable abstract level requirements, low level requirements, source code and executable object code. Since not all 'constraints' provide an equal contribution to the architectural level defences, and thus not all 'constraint' arguments are equal, a framework is also included that assists in determining the importance of satisfying each particular argument.

6.1.2 Benchmarks for Evidence Sufficiency

Turning our attention now to addressing the question of benchmarks for evidence. It has already been described that the goal-based approach allows flexibility in evidence, and that this is desirable. However the drawback is that a means of measuring and justifying the sufficiency of evidence has to be incorporated into each and every argument. This may be repetitive, and detract from the focus on the product aspects of the argument. On the other hand, the prescriptive approach lacks flexibility in evidence, and it does not help to group evidence in ways such that the 'so what?' can be answered from this evidence. However, the strength of the prescriptive approach is that it is very clear to suppliers trying to determine activity costs for inclusion in the tender response. So how can these approaches be combined without undermining their advantages, while ensuring usefulness as a contracting benchmark?

Consider this; what if the following assumptions are made:

- The set of evidence supplied is never infinite (because we don't have infinite time or money), thus the assurance it provides is never absolute; so there will always be limitations in the totality of evidence.
- The evidence produced from each method or technique will always have some limitation with it, and complementary evidence from one or more methods or techniques will usually be required to resolve the limitation.
- As there will always be limitations in the evidence; why not change the focus to determining if the limitations are tolerable in the specific context?

Further, a generic framework could be provided for determining the tolerability of the limitation in evidence for each argument that is going to be made. Since evidence is best presented at the sub-claim level, this is the best place to immediately assess the impact of tolerability of limitations of evidence. Once assessed with respect to the specific 'constraint' the (in)tolerability can then be evaluated in the context of the impact on architectural assurance, and thus provide meaningful insight into product safety risks.

The framework could take into account the generic properties of evidence (refer [Wea03]) including:

- *Relevance* of the evidence (as produced by method or technique X) to the sub-claim (e.g. compliance of the source code with the applicable low level requirements for constraint Y),
- *Trustworthiness* of the evidence based on who and how it was produced, and
- *Results* of the evidence, including where the results provide counter evidence.

This is advantageous because the supplier can be required to identify the limitations with each type of evidence proposed with respect to these properties of evidence. The supplier can also be required to identify how they will resolve any limitations through provision of additional evidence. The approach is generic because it reflects generic properties and limitations of evidence. The techniques and methods used to the produce the evidence are entirely within the supplier's control. The better the techniques and methods they propose, the fewer the limitations they will have to address; but this is a choice for the supplier. Further, the concept provides a means for the supplier to think critically about what techniques and methods they are proposing and provides a means for measuring the adequacy of each technique and method. Finally, when they've worked out their techniques and methods, they can cost these into their proposal, and thus the supplier can be confident in their proposal costing for the provision of evidence.

One possible approach that uses these principles has been developed by the authors. It introduces the concept of a Evidence Safety Assurance Level (ESAL) and 'Tolerability of Limitations' [RMc10]. The remaining sub-sections discuss how these principles can be incorporated into tenders and contracts to bound uncertainty.

6.2 Proposal of Argument and Evidence

To reduce uncertainty about the intended safety argument at the time of contract signature, the tender phase requires a mechanism to be informed of the argument. This implies that it is useful to know which generic claims are going to be applied to each architectural 'constraint'.

One possible approach would be to require the tenderer, through the tender SOW, to provide a *Software Assurance Plan* to describe which set of claims are going to be demonstrated for each 'constraint'. To ensure consistency in tenderer responses it is advantageous to align where possible the claims to the generic software lifecycle products and the generic attributes of each. The following is an example of a generic Tender SOW to achieve this:

The [Tenderer] shall prepare a [Software Assurance Plan] per TDRL XX to propose the attributes that will be assured, for each software lifecycle product, for each constraint described in the [Conceptual System and Software Architecture Suitability Document].

A specific instantiation of these clauses for the Claims Safety Assurance Level approach is described at [RMc10]. [RMc10] provides a systematically established set of attributes for each lifecycle, that provides confidence in its completeness of attributes for generic software behavioural claims.

To reduce uncertainty about the intended limitations in evidence for each of the aforementioned attributes at the time of contract signature, the tender phase also requires a mechanism to provide information on the likely scope of the body of evidence and its potential limitations.

One possible approach would be to require the tenderer, through the tender SOW, to provide two things:

- a *Software Development Plan* to describe which methods and techniques are going to be applied across the development, and
- a *Software Assurance Plan* to describe how any limitations in the evidence produced from the methods and techniques described in the software development plan are tolerable with respect to relevance, trustworthiness and results.

Software Development Plans are already routinely in use within projects; and this should be no surprise to any reader. However the key contribution this paper is proposing is a sister document (the Software Assurance Plan) that presents the analysis and justification for the adequacy of the Software Development Plan, with respect to the tolerability of limitations in evidence concept. By requiring each tenderer to explicitly justify the adequacy of their software development, then suppliers are provided a consistent set of expectations for costing their software development programs. This is important when it comes to establishing which of two or more software developments programs is most adequate with respect to evidence provision.

For the purposes of clarity the Software Assurance Plan is quite different from more conventional deliverables such as Software Verification Plans. A Software Verification Plan will usually provide the description of activities used to demonstrate requirements satisfaction. The Software Assurance Plan presents the analysis and justification for the adequacy of the Software Development Plan, by describing the claims and justifying the evidence proposed for each type of 'constraint'. Conventional plans such as verification plans, test plans, etc. are still envisaged being companion documents to the Software Assurance Plan.

The following is an example of a generic Tender SOW clause to achieve production of the Software Development Plan and Software Assurance Plan:

Software Development Plan. *The [Tenderer] shall prepare a [Software Development Plan] per TDRL XX to describe the methods and techniques proposed to be used throughout the software development lifecycle, including description of techniques or methods used prior to this development but for which evidence is relevant.*

Software Assurance Plan. *The [Tenderer] shall prepare a [Software Assurance Plan] per TDRL XX to describe how the evidence produced from the application of the [Tenderer] proposed methods and techniques is proposed to assure tolerability of limitations in evidence with respect to relevance, trustworthiness and results, for each attribute of each software lifecycle product, for each constraint described in the [Conceptual System and Software Architecture Suitability Document].*

A specific instantiation of these clauses for the Evidence Safety Assurance Level and Claims Safety Assurance Level approach is described at [RMc10].

For the flight control system example, let's assume that each of the proposed options provides a *Software Development Plan* and *Software Assurance Plan*, for which are briefly summarised as follows. Note that for the purposes of brevity within this paper this in only an illustrative summary without the corresponding justification. It doesn't represent the full content of these plans.

- Option A
  - ARP4754 system safety program with software assurance to RTCA/DO-178B Level A.
- Option B
  - DefStan 00-56 Iss 4 system safety program with software assurance to DefStan 00-55 Iss 2 SIL4, including the application of formal methods.
- Option C
  - MIL-STD-882D system safety program, with new software developed to RTCA/DO-178B Level A, and reused software developed to MIL-STD-498.
- Option D
  - MIL-STD-882D safety program, with software developed to MIL-STD-498.

6.3 Evaluation of Argument and Evidence

The purpose of the tender requesting this information is to permit evaluation of the extent to which the holistic evidence requirements are costed into the tender response and to establish if they meet the acquirer's expectations. The retrospective supplementation of evidence is rarely straightforward, particularly when it results in a change to one or more of the lifecycle products such as requirements, design or code. Therefore, it is in the acquirer's interests to establish the extent to which the contractor has proposed a sufficient set of evidence. While it is recognised that the evidence would not yet exist at the time of tender, clear insight into:

- the techniques and methods proposed,
- what evidence will be produced?,
- how this evidence will combine?, and
- what limitations in the evidence might be intolerable?;

will permit the acquirer to adjust the contractors proposed costing by a risk figure based on the amount of uncertainty (or extent of suitability) in the tenderers proposed evidence set. This would provide a normalised evaluation of tenderers responses compensating for tenders that do include the relevant information.

Considering the examples proposed in the previous section, it is evident that the evidence set proposed by Options A through D varies substantially for each proposal. Using the benchmarks set for the argument and evidence, each option is evaluated. Assume, for the sake of illustration, that the evaluation results are summarised as follows:

- Option A – There appears a limitation with the extensiveness of normal and robustness verification proposed against low level requirements relating to time-dependent properties, including synchronisation, of the flight control laws in relation to fault tolerance to jitter (early and late) related effects on sensor inputs. Tenderer is requested to clarify their proposal.
- Option B – There appears a limitation of the extensiveness of the application of analytic and empirical verification of behaviours relating to fault tolerance to value failures of air data system and attitude/heading reference system sensors. This is due to fault tolerance mechanisms being incorporated into device drivers which can only be verified in the Systems Integration Laboratory but for which there is no means with the current toolset to inject these fault conditions for the purposes of verification. This limitation is flagged for clarification with the tenderer.
- Option C – Limitations in evidence for reused software are substantial with respect to low level requirements, low level requirements verification, and coverage of implementation from requirements based verification. These limitations are assessed to be intolerable.
- Option D – Already eliminated based on architectural evaluation.

Options A and B require further clarification with the Tenderers, and this will be sought. Option C is eliminated from the tender evaluation due to intolerable evidence limitations, and Option D was already eliminated based on architectural shortfalls. Clarification with Options A and B reveals the following additional information for the evaluation:

- Option B – the limitation remains as the tenderer claims that low level verification undertaken prior to integration verification will provide sufficient evidence in this regard. Therefore verification of these requirements on the target computer with credible fault conditions is via inference only. These limitations are assessed to be intolerable. Option B is eliminated from consideration.
- Option A – the extensiveness of normal and robustness verification has been adequately clarified and is acceptable.

Therefore, Option A is selected as the winning Tenderer, and negotiations are commenced to progress to contract signature.

Note that in reality there are many other selection criteria for a product, and so it is common for capability, force integration, and political factors amongst others to affect selection. Hence, these other factors may sometimes require compromise on the ideal safety solution. However, this does not invalidate the process proposed in this paper. Instead, the process in this paper enables the acquirer to be informed about the safety assurance aspects such that it is possible to make informed trade-offs between safety assurance and other selection criteria. For example, it may be possible to choose Option B, make decisions regarding risk treatment or retention, because other benefits out-weigh the impact of its limitations.

6.4 Providing Argument and Evidence Assurance

Once the preferred tenderer has been identified (in this case Option A); and any uncertainties regarding the claims and evidence assurances are tolerable, then it is possible to develop a contract between the supplier and acquirer.

Under the contract, the acquirer will require a means to establish if the final 'as-delivered' claims and evidence meets the prescribed benchmarks. This can be achieved by requiring the contractor to deliver (via appropriate

SOW contract clause) a *Software Assurance Summary Document*. The document would describe how the assurance of the 'attributes' of software lifecycle products actually achieves the benchmarks established during tender processes. The following is an example of the generic Contract SOW clauses to achieve this:

Achievement of Claims and Attributes of Software Lifecycle Products

The [Contractor] shall prepare a [Software Assurance Summary] per CDRL XX to describe the attributes that have been assured, for each software lifecycle product, for each constraint described in the [System and Software Architecture Document].

Assessing the Evidence

The [Contractor] shall prepare a [Software Assurance Summary] per CDRL XX to describe how the evidence produced from the application of the [Contractor] proposed methods and techniques has assured the tolerability of limitations in evidence with respect to relevance, trustworthiness and results, for each attribute of each software lifecycle product, for each constraint described in the [System and Software Architecture Document].

A specific instantiation of these clauses for the Architectural Safety Assurance Level approach is described at [ReM11].

7 Resolving Issues after Contract Signature

Despite best intentions, whenever there is uncertainty there is potential for it to lead to an undesirable outcome as development progresses. The sections prior to this have largely been focussed on trying to bound the uncertainty in areas that really affect the case for safety. However, once a contract is commenced, if issues do arise with respect to architecture, claims or evidence, then it is useful to establish in advance the approach for resolution of these issues.

Considering the ongoing example of Option A, and let's assume that during preliminary design review several issues are identified as follows:

- Issue 1 – Proposed treatments to value failures of air data system airspeed data are identified to be inadequate under conditions of transition to the hover. A revised treatment is proposed requiring an adaptation to flight control law transition criteria to provide an improved fault tolerance against this fault.
- Issue 2 – Verification and validation of the accuracy of the software requirements relating to discrete implementation of the legacy analogue control laws is identified to contain shortfalls relating to the reuse of modelling from the previous implementation. Additional modelling of the discrete implementation is viewed as required by the acquirer.

There are two main options for providing contract scope for the work to resolve unforeseen issues that arise: either within the original contract, or through a contract change. Both are discussed in the following sub-sections.

7.1 Resolution within Contract Scope

Resolution within the contract scope is entirely dependent on the supplier openly acknowledging the requirement to resolve the issue and perhaps do extra work. However, when profit margins are at risk, and there is risk of schedules being affected, it is not uncommon for suppliers to argue work is out of scope.

Consider the two issues identified our example:

- Issue 1: This treatment is deemed in-scope of contract because it was a contractor oversight during the conceptual design proposal. Evidence is provided commensurate with previously identified attributes, lifecycle products and constraints.
- Issue 2: Acquirer and supplier enter into contractual dispute regarding the provision of additional evidence modelling the discrete implementation, because the supplier claims their limitations in the modelling are tolerable.

One way to address Issue 2 is to make absolutely explicit this requirement for limitations to be resolved to the satisfaction of the acquirer through a statement of work line item. This line item can then be costed and suppliers will be empowered to resolve such issues. An example of how this might be achieved is as follows:

Intolerable Limitations in Evidence, Claims or Architecture

Where the [Acquirer]'s certification evaluation establishes that the [Contractor] has not achieved the requirements of the {reference applicable SOR and SOW clauses relevant to architecture, argument and evidence}, or there are shortfalls in the 'Tolerability of Limitations' of evidence versus the criteria specified by this contract, then the [Contractor] shall undertake one or more of the following remediation actions to resolve the shortfalls to the satisfaction of the certification authority:

- *engineering change to architectural constraints,*
- *engineering change to implementation of architectural constraints, or*
- *additional analysis, verification and validation by further or supplementary application of methods or techniques.*

The [Contractor] shall amend all relevant deliverables per the CDRL to incorporate the engineering changes and additional evidence.

Note to Contractors

The above clause provides the means for the certification authority to address shortfalls against architecture, argument and evidence expectations. While this clause may be interpreted to result in unbounded programmatic risk for the contractor, the intent is to focus both acquirer and contractor efforts at establishing unambiguous consensus during the tender process and contract negotiations. The contractor should not sign the contract if they believe there remains substantial uncertainty regarding the provision of evidence against the framework, and instead request further clarification during contract negotiations.

The aim of this approach is to ensure that the tender phases and contract negotiation phases have systematically identified, disclosed and evaluated the intended body of evidence and that all intolerable shortfalls have been included within the contract. Thus the example clauses would only come into effect if an issue remains, and this would be less likely and less serious because the evidence planning was systematic in the first place.

The drawback to this approach is that suppliers may interpret this as a very risky statement of work line item and cost it commensurately. However there is a positive to the behaviour this generates for tender evaluation. If the acquirer evaluates the cost attribution against this line item from each tenderer, and there are notable differences in the costing, then the acquirer can use this to establish the tenderers confidence levels in their own cost estimates for achieving architectural, claims and evidence assurance. This is a very useful tool during tender evaluation, and something that is not easily gauged by other means. Even if the clause is removed during contract negotiations due to supplier concerns, its inclusion during the tender process is extremely revealing about supplier confidence in their proposals and costing.

7.2 Resolution Outside Contract Scope

Resolution of shortfalls outside the contract scope is easy from the perspective of defining the scope of work; as usually the analysis to determine that the architectural changes, design changes or evidence supplementation will be clear from the analysis done to demonstrate it is outside the original contract. If there is contingency funding to fund the contract change, then it will also be relatively straightforward for the acquirer.

However, if contingency funding is not available this is a very challenging path as it usually involves the allocation of additional funding to a project from Government. Most Government committees responsible for funding of military aviation system acquisitions are not sympathetic to issues that emerge late in the project lifecycle which were not forecast with original costing, allocated as contingencies, or articulated as program risks.

For the purposes of this paper, the approach described at Section 7.1 is preferred at least at the tender phase, so that the likelihood of additional out of scope work is well understood during the tender phase, and minimised in the contract phase.

8 Evaluation

As the concepts introduced in this paper differ substantially from existing approaches, evaluation of their effectiveness is required. However, because it is often difficult to apply novel approaches to real projects at the initial proposal of these approaches, evaluation by experiment is not straightforward. For this reason, literature [Van07] regarding the design of studies for participative research was examined to establish that preliminary evaluation of the concepts outlined in this paper was suited to survey questionnaire of suppliers, acquirers and related stakeholder agencies (e.g. regulators). A series of targeted workshops is being used to complement the survey questionnaires.

8.1 Description of Evaluation

A detailed survey questionnaire was prepared using the principles for questionnaire design from [Opp01] and [BAN86]. The questionnaire asked a mix of open and closed questions regarding the concepts and application thereof presented in this paper and the supporting literature. The questionnaire was provided to representatives of a range of supplier and acquirer agencies representing a cross section of the following:

- Military Regulatory/Certification Authorities
- Supplier Contractors
- Acquirer Agencies
    - Australian – Defence Materiel Organisation
        - Sustainment System Program Offices
        - Acquisition Projects
    - United Kingdom Ministry of Defence
    - United States Air Force
    - Defense Contract Management Agency
- Contractors to Defence (Professional Service Providers)
- Science and Technology Organisations supporting Defence Acquisition

8.2 Results of Evaluation

The evaluation is on-going; however analysis of results has been undertaken on 15 completed surveys. The surveys represent a cross-section of the above listed stakeholders from Australia, Canada, New Zealand, the United Kingdom, and the United States of America . The evaluation undertaken to date has provided the following feedback:

Framework

- Acquirers and Certification Authorities indicated that the approach may have helped to avoid several historical (and current) project issues where architectural safety shortfalls were responsible for project cancellation or significant project delays and cost increases. However, they noted that correlation in retrospect is easier than in reality.
- Some respondents were deterred by the notion of yet another assurance framework, while others noted that current approaches had limitations, and this approach seems compatible and extends some current approaches.
- Some respondents were deterred by the complexity of the inter-related assurance concepts and contracting mechanisms, although several of these indicated that the concepts were less complex than many of the systems to which they would apply. This would perhaps provide natural selection of suppliers that cope with complexity.
- Some respondents were positive about the concept of defences and 'constraints' although they had reservations that supporting methods as yet wouldn't enable them to model the relationships effectively. Extension to existing methods might be required.
- Many respondents indicated that the 'tolerability of limitations' concept appeared useful in that it provides some inherent rules for providing and measuring supplier justifications. There was some support for developing the rules even further, and providing examples.
- The majority of respondents indicated that one or more worked examples, of both a tender costing based on the proposed tender clauses, as well as of implementing the underlying ASAL/CSAL/ESAL frameworks from [RMc10] and [ReM10], would be beneficial.

Tender Evaluation and Contract Negotiation

- Acquirers and suppliers indicated that the proposed approach does provide product and evidence focus during the tender phase that appears beneficial, although until they actually apply it, this is only speculation.
- There was positive response to knowledge of architecture during tender processes, although some suppliers were concerned about how they might progress their design processes to that point for some tenders, particularly those involving sub-vendors.

Contract Execution

- There was consensus that knowledge of architecture and knowledge of evidence at tender would reduce the difficulty of contract execution.

Risk Evaluation

- Regulators and operational representatives indicated that knowledge of product behaviours and remaining defences would help with planning operational treatments, and with developing emergency procedures.
- Regulators indicated that they were still unclear how evidence/assurance shortfalls correlated to risks, and suggested developing the framework further to address risk measurement.

Cost and Schedule

- Suppliers expressed reservations about being able to resolve issues they haven't costed within contract scope, although praised that the underlying frameworks would potentially provide improved knowledge of product and evidence requirements during tender phases and thus reduce the opportunity for issue resolution within contract.
- Some suppliers and acquirers expressed concern that this would increase the cost of tender processes, and potentially deter some tenderers.
- Some suppliers had reservations about the perceived paradigm shift, and how they would cost effectively educate their staff on how to work within such a framework.

Further distribution to an increased sample size of the aforementioned organisations is presently being undertaken. Final results will be published within the aforementioned PhD thesis, and may also be targeted for journal publication.

8.3 Analysis of Evaluation Results

Analysis of the surveys received indicates the following:

- There is correlation between the respondent comments and the motivating issues. This indicates that the motivating issues are probably valid.
- A cross section of prescriptive versus goal-based 'world views' were evident in responses to motivating issues and general principles revealing that there is diversity in 'world views', although the results don't directly suggest a resolution.
- There was not direct correlation between 'world views' and position/negative comments indicating that there are issues of 'world view', education, paradigm shaping, and framework limitations involved.
- There is correlation between respondent comments on feasibility and usefulness and the general principles on which the framework is based. This indicates that the general principles may be widely agreeable, even if their opinions on the implementation differ.
- Suppliers focussed strongly on cost and schedule implications, and competitiveness with respect to other suppliers. The level of knowledge on the topic of safety assurance varied substantially between suppliers, acquirers and regulators.
- While supplier sentiment was that regulations are already too constraining for their businesses to be innovating, there was acknowledgement of the problems with the current approaches to assurance.
- Acquirers focussed on successful tender processes leading to successful contract execution. The level of knowledge on the topic of safety assurance varied substantially between acquirers and regulators.
- Views of safety and risk varied between respondents and warrants further clarification.

It is hoped that through the on-going conduct of the evaluation, and publishing of results, that consideration be given to apply these concepts to a real world system acquisition. This would overcome the limitations of the constructed environment of a survey and workshop.

9 Conclusion

This paper has examined factors affecting the provision of safety assurance evidence for military aviation software system contracts including the impact of the standards paradigm, integration of the standard with the contract lifecycle, enforcement of design requirements, obtaining of assurance evidence and resolution of shortfalls in product and evidence.

Acquirer (and regulator) certainty in the software systems behaviours and fault tolerance, the inherent argument in the claims and framework used to relate evidence to safety objectives, and the approach used for identifying, analysing and evaluating the tolerability of limitations in evidence are identified as particularly important. The impact of uncertainty in these topics at the time of contract signature has been examined with respect to the potential for a successful contractual outcome. Approaches have been proposed for obtaining assurances and bounding uncertainty by pre-contract and throughout the contract. An example was used to illustrate the benefit in the approach.

Observations on preliminary evaluation results conducted with respect to a framework based on these certainty motivators have been presented to provide support to their validity in industrial practice. Based on these initial observations, further evaluation in industry and acquirer communities is recommended.

10 References

[14CFR25] Title 14 Aeronautical and Space, Code of Federal Regulations Chapter I Federal Aviation Administration, Department of Transportation, Subchapter C – Aircraft, Part 25 *Airworthiness Standards: Transport Category Airplanes*

[AC25.1309] Federal Aviation Administration, Advisory Circular, AC25.1309-1A System Design and Analysis, 21 Jun 1988.

[BAN86] D.R. Berdie, J.F. Anderson, M.A. Niebuhr, *Questionnaires: Design and Use*, Second Edition, The Scarecrow Press, Inc. Metuchen, N.J. USA, 1986.

[DO178B] RTCA Inc., *RTCA/DO-178B: Software Considerations in Airborne Systems and Equipment Certification*, Washington D.C.: RTCA Inc., 1992.

[JTM07] D. Jackson, M. Thomas, L Millet, Editors, *Software for Dependable Systems: Sufficient Evidence?,* Committee of Certifiably Dependable Software Systems, National Research Council, National Academy of Sciences, USA, 2007.

[Kel98] T.P. Kelly, *Arguing Safety – A Systematic Approach to Managing Safety Cases*, PhD Thesis, Department of Computer Science, University of York, 1998.

[KeM01] T.P. Kelly, J. McDermid, *Safety Case Patterns – Reusing Successful Arguments*, Rolls-Royce Systems and Software Engineering, University Technology Centre, Department of Computer Science, University of York, Heslington, York, 2001.

[McD07] J.A. McDermid, *Risk, Uncertainty, Software and Professional Ethics*, 20 August 2007.

[McK06] J. McDermid, T. Kelly, *Software in Safety Critical Systems: Achievement and Prediction*, Nuclear Future, Volume 03, No. 03, 2006.

[McR12] J. McDermid, A. Rae, *Goal-Based Safety Standards: Promises and Pitfalls*, presented at the Safety Critical Systems Symposium, Springer, Bristol, February 2012.

[NTS06] National Transportation Safety Board, *Safety Report on the Treatment of Safety-Critical Systems in Transport Airplanes*, Safety Report NTSB/SR-06/02, Washington, D.C., USA, 2006.

[Opp01] A.N. Oppenheim, *Questionnaire Design, Interviewing and Attitude Measurement*, New Edition, Continuum, London, Great Britain, 2001.

[Rei08] D.W. Reinhardt, *Considerations in the Preference for and Application of RTCA/DO-178B in the Australian Military Avionics Context*, presented at the Australian Safety Critical Systems Association Conference, Aug 2008

[ReM10] D.W. Reinhardt, J.A. McDermid, *Assuring Against Systematic Faults Using Architecture and Fault Tolerance in Aviation Systems*, presented at the Improving Systems and Software Engineering Conference (ISSEC), Aug 2010.

[ReM11] D.W. Reinhardt, J.A. McDermid, *Contracting for Architectural, Claims, and Evidence Assurance for Military Aviation Systems*, Departmental Technical Report, Department of Computer Science, University of York, Oct 2011.

[RMc10] D.W Reinhardt, J.A. McDermid, *Assurance of Claims and Evidence for Aviation Systems*, presented at the 5th IET Conference, Oct 2010.

[SSEI09] R. Hawkins, J. McDermid, Software Systems Engineering Initiative, SSEI-TR-0000041, *Software Safety Evidence Selection and Assurance*, Issue 1, University of York, October 2009.

[Van07] A.H. Van De Van, *Engaged Scholarship, A Guide for Organizational and Social Research*, Oxford University Press, Oxford, Great Britain, 2007.

[Wea03] R.A. Weaver, *The Safety of Software – Constructing and Assuring Arguments*, PhD Thesis, Department of Computer Science, University of York, 2003.

Do You Get The Picture? Situation Awareness and System Safety

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

carl@iSys-Integrity.com

Abstract

Studies of the dependency between complex, dynamic systems and their human operators often focus on human-computer interactions without considering the emergent properties of human-machine systems in use. As systems become more complex, and typical operating environments more dynamic, the role of the operator has typically changed from providing manual to cognitive control. An understanding of human cognition in context is thus central to the design of human-machine systems and this is particularly pertinent in safety-related systems when the elimination of hazards is a principal concern. This paper will argue that operator situation awareness is an important, safety-related phenomenon and that it can be used to examine human cognition in context in order to add value to system safety. The paper will examine the dominant theoretical perspectives on situation awareness and a model of this critical phenomenon is presented. The paper will show how the proposed model of situation awareness can be used as a framework for the analysis and identification of hazards relating to operator awareness in the context of system use. It is also suggested here that modelling situation awareness is useful in identifying areas of interface design where safety and usability are mutually exclusive. An illustration of the use of this technique is provided to show how the model can inform the design of interactive systems and how it can be used to generate evidence to support system safety claims.[.]

Keywords: cognition, context, hazard analysis, situation awareness, safety, usability.

1 Introduction

Studies of safety-related systems have in the past considered safety predominantly from a technical perspective. Such studies have typically been limited to addressing hazards that could arise through hardware and software failures, yet human factors are becoming increasingly important in the design and evaluation of safety-related systems (Sandom 2007). This change in perspective has revealed a complex set of 'human' problems that are extremely challenging. The hazards associated with human failures are very different from those that have historically been the concern of safety engineers since they arise directly from the *use* of the system and therefore require some understanding of the cognition of users. The identification of interaction hazards arising during system use may help designers to improve the system interface and interactions such that the associated risks are mitigated or even eliminated. However, in order to study these interaction hazards, appropriate research constructs are required to help designers to understand the user's cognition during system use.

The dominant cognitive paradigm in Human Computer Interaction (HCI) research has been based on the human information processor as characterised by the seminal work of Card *et al.* (1983). Although the information processing model has been extremely useful, there is a growing awareness that there are a number of limitations associated with this reductionist paradigm for human cognition (Nardi 1996, Hutchins 1995, Suchman 1987, Winograd and Flores 1986). A key limitation with this model is that it has neglected the importance of how people work when using computer systems situated in the real world (Landauer 1987).

Making the context of the user-system interaction more central in understanding the cognition of the user and the resulting action is a key facet of a perspective referred to as 'situated cognition'. Here, in contrast to the information processing view, it is argued that the cognitive state that leads the user to exhibit 'purposeful, situated action' can only be fully explained in the specific context in which that action takes place. This suggests that an understanding of human cognition requires a holistic approach through careful consideration of the social, organisational and political aspects of HCI in the context of use.

This brief discussion suggests that a comprehensive understanding of situated human cognition is central to the design of interactive systems, and this is particularly pertinent when the elimination of hazards in safety-related contexts is a principal concern. In order to select and develop appropriate research constructs to look at such hazards, it will be useful to briefly consider the nature of the hazards themselves.

2 Human Factors and Systems Safety

Human factors are repeatedly mentioned as a major contributing factor or even the direct cause of accidents or incidents. For instance, an analysis of causal factors contributing to a situation in which the safety of aircraft was compromised show that 97.7% of incidents in UK airspace during 1996 were caused by human error (calculated from CAA 1998a and CAA 1998b). Human errors often occur when there are interaction problems between the user and the system.

By their nature, safety-related systems present unique hazards arising from the interactions between the user and the system and a safety case is usually required to provide a clear and comprehensible argument that a system is safe to operate. A safety case generally consists of claims about a system and evidence which is used as the basis of a safety argument to support those claims (see Figure 1). The safety case provides the assurance that a system is adequately safe for a specific application in a given context. For example, in the UK, National Air Traffic Services are required to produce safety cases for air traffic control systems to satisfy the air traffic control service regulators.

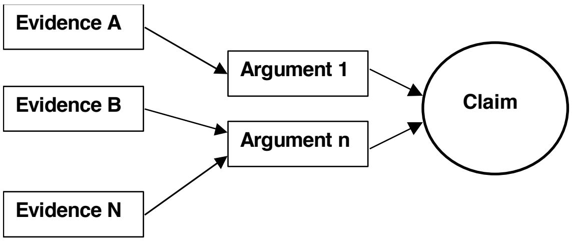


Figure 1 – Safety Claim Structure

Safety arguments, particularly those relating to system hardware components, are often based on evidence taken from reliability data and historical trends. However, it is often much more difficult, if not impossible, to derive accurate reliability evidence to support safety claims relating to many human factors issues such as those associated with the interaction between the system and the operator in a given context (Sandom 2011).

The reliability of the user-system interaction in hazardous situations is extremely important. If the user's interaction is inappropriate, there is the potential for catastrophic consequences. To examine these issues, safety engineers need user-centred ways of evaluating safety-related systems. If designers are to identify interaction hazards associated with the human operator and design mitigating features into the system to reduce the likelihood of the hazards being realised, it is crucial that designers have ways of understanding why users take particular actions in particular circumstances.

The user may act inappropriately because they have problems making sense of what they are doing at a given time. There are several human-centred constructs that may help us to understand these issues, an important one being the idea that people have 'pictures' of what is going on in their interaction with the system. This is often referred to as the user's Situation Awareness (SA). If users make errors in using systems, it may be because their SA is incorrect. A highly usable system may, for example, be so transparent that the users do not correctly develop their 'pictures' of the system interaction as the situation develops. Where users form incorrect or inappropriate 'pictures' of the situation, there is great scope for error, implying that SA has a significant impact upon system safety (Endsley 1995a). Finding ways of assessing and understanding the awareness of the situation held by users will be useful in helping identify areas where users form incorrect awareness and where, as a result, there are hazards. Consequently, situation awareness is an important, safety-related phenomenon that can be used to examine human cognition in context in order to add value to system safety (Sandom and Harvey, 2004).

3 Situation Awareness

In order to develop suitable ways of understanding and assessing SA, it is important to consider the existing research in the area. It is widely accepted that a user must have an appropriate awareness of their situation for the safe operation of any complex, dynamic system (Sarter and Woods 1991). However, SA is a complex concept and it is difficult to find an accepted definition of the term (Charness 1995, Hopkin 1995). Despite this, the widespread interest in SA, particularly within the field of aviation and other similarly complex domains suggests its potential contribution to interface and interaction design (Harris 1997,Garland and Endsley 1995).

In the context of human-machine interaction, current definitions are generally based on opposing views of SA as either a cognitive phenomenon or as an observer construct. The cognitive perspective is the prevalent view, seeing SA as a cognitive phenomenon that occurs ‘in the head’ of the user – though even within this broad perspective there are differing interpretations and emphases. In contrast, if seen as an observer construct, SA becomes an abstract concept located ‘in the interaction’ between user and environment. Despite the differences that exist in theoretical stance, a more detailed discussion will show that there are conceptual similarities between the different perspectives of SA. A detailed study can then be used to help to understand SA in the context of safety-related systems and to make use of it in informing their design.

3.1 Cognitive Perspective

Proponents of a cognitive perspective of SA view it as a phenomenon that occurs ‘in the head’ of an actor in a similar fashion to the dominant cognitive framework of the human as an information processor (Card *et. al.* 1983). Indeed, some even suggest that SA is yet another ‘black box’ component, or (sub-) process, within the human information-processing model (Endsley 1995b). The process-oriented view sees SA as being acquired and maintained by the user undertaking various cognitive activities (Sarter and Woods 1991). Cognitive definitions of SA also generally provide a rich description of key elements of decision making activities in complex systems such as perception, comprehension and projection (Endsley 1995b). There is another view of SA within the cognitive perspective, which sees SA as a product – a state of awareness about the situation with reference to knowledge and information (Endsley 1995a). Some researchers have even integrated the process and product perspectives (Isaac 1997).

Whilst the conflicting views may signify an apparent lack of coherence within the cognitive perspective, Endsley's theoretical model of SA (Endsley 1995b), based on the role of SA in human decision making in dynamic systems, has been widely cited and highly influential in cognitive science research. This model represents a typical cognitive perspective and it proposes three different levels of SA which are relevant to this paper:

Level 1 SA *Perception* of the status, attributes and dynamics of relevant elements in the environment.

Level 2 SA *Comprehension* of the situation based on a synthesis of disjointed Level 1 elements to form a holistic 'picture' of the environment.

Level 3 SA *Projection* of the near-term future of the elements in the environment.

The different levels suggest that SA is based on more than simply perceiving information about the environment, which is often the perceived definition of the phenomenon. Many cognitive accounts of SA suggest that after information concerning relevant elements is perceived, a representation of the situation must be formed before a decision can be made based upon current SA.

This leads to another common notion that is particular to the cognitive perspective with SA often considered synonymously with mental models (Isaac 1997) an area of long time interest for HCI. Seeing the mental model as a subjective awareness of the situation which includes what has happened, what could happen and what a user predicts will happen based on their goals and objectives (Kirwan *et. al*. 1998) suggests that this representation is the 'picture' that the user has (Whitfield and Jackson 1982). Despite making an explicit link with mental models, the models of SA proposed within the cognitive 'school' do not have iterative dimensions to reflect the dynamism of acquiring SA over time. Instead they propose models which capture or explain SA at any given instant in time.

3.2 Developing Perspectives

When seen as an observer construct, SA is explained as an abstraction that exists only in the mind of the researcher. From this perspective, SA is considered as a useful description of a phenomenon that can be observed in humans performing work through interacting with complex and dynamic environments (Billings 1995,Flach 1995a). The description is developed by considering observable behaviour in the environment – what the user does, how the system performs – but is not concerned with directly relating these things with cognitive states of the user. In one sense this might be associated with traditional behavioural psychology. A behavioural stance may simplify the discussion of SA by removing (or at least marginalising) interest in the user's mental state in favour of a reliance on observable action. A behaviourist stance is however much less rich as a research perspective, since no attempt will be made to relate action to intention on the user's part. In moving the SA debate forward, then, and looking for rich models to explain SA, identify hazards and ultimately inform the (re)design of safety-related systems, we would suggest that cognitive views of SA are more useful.

Yet, there are competing views of SA which do not fit neatly into the information-processing stance predominantly taken by the cognitive school, but which might be useful in developing an informed stance on SA. Smith and Hancock (1995), for example, propose a view of SA as adaptive and externally directed consciousness, arguing that there is currently an artificial and contentious division evident within the literature relating to general perspectives of SA as either exclusively knowledge (i.e., cognitive state, or product) or exclusively process. From this view, SA specifies what must be known to solve a class of problems posed when interacting with a dynamic environment. Smith and Hancock (1995) also criticise the lack of dynamism exhibited in the cognitive perspective, contending that SA is a dynamic concept that exists at the interface between a user and their environment. Moreover, they argue that SA is a generative process of knowledge creation and informed action taking as opposed to merely a snapshot of a user's mental model.

There are merits in many of the competing views of SA and the range of views that exist highlight the complexity of SA and the general immaturity of research in the area. The mental state of the user is important in trying to understand the awareness that the user builds up of a situation. Yet researchers often have only observable interaction data on which to draw, tempting them to marginalise mental state as a concern and focus on explaining SA without reference to the user's cognitive processes.

3.3 Situated Cognition Perspective

A helpful, synthetic and pragmatic perspective of SA sees it as a measure of the degree of dynamic coupling between a user and a particular situation (Flach 1995b). This view attaches importance both to the user's cognitive state and to the context or situation in which they are acting, reflecting a move away from traditional information processing models of cognition towards the situated cognition (and situated action) perspective introduced in Section 1 as a developing movement in HCI.

Reflecting this stance, a tangible benefit of SA research is the focus on the inseparability of situations and awareness (Flach 1995b). From this perspective, discussions of SA focus attention on both what is inside the head (awareness from a cognitive perspective) and also what the head is inside (the situation which provides observable data) (Mace 1977). Generally, this stance suggests that the user's current awareness of a situation affects the process of acquiring and interpreting new awareness from the environment in an ongoing cycle.

This view is similar to Neisser's Perception-Action Cycle (1976) which has been used to model SA (Smith and Hancock 1995, Adams *et. al.* 1995) in an attempt to capture the dynamic nature of the phenomenon. Central to this view of SA is the contribution of active perception on the part of the user in making sense of the situation in which they are acting. Such active perception suggests informed, directed behaviour on the part of the user.

As we have seen, one of the problems in making use of SA is the conflicting theoretical perspectives from which SA has been described and researched. Whilst theoretical debate is both healthy and necessary, a pragmatic stance which critically reviews the different perspectives and attempts to synthesise common elements may be a more immediate way of contributing to systems design. A useful outcome of such an approach would be a model that helps designers understand SA and its usefulness in designing interfaces to, and interaction sequences and dialogues within, safety-related systems.

4 Dynamic Situation Awareness Model

As the preceding discussions have highlighted, there are competing and sometimes confusing views on SA and its relation to people and the situation in which they are acting. There is significant on-going research to further these debates and refine the perspectives. Whilst such research is of long-term value in contributing to the maturity of the field and refining explanations of SA, this paper takes a more pragmatic approach, arguing that an attachment to a particular perspective can cause problems. Where there is contention between opposing perspectives, research can tend to become dogmatic which in an immature area may lead to opportunities for furthering our understanding being missed as researchers endeavour to strengthen their particular perspective. This paper is more interested in considering the focus of our research in the area and synthesising constructs from the existing perspectives that may help us make sense of the situations, which we are studying.

This paper will now draw themes, which we see as important to our work in SA, from the theoretical perspectives that we have discussed, and frame them as a dynamic model of SA based upon Neisser's Perception-Action Cycle (1976). We will then use this model to help us analyse and understand SA.

4.1 Awareness

As our discussion of the competing perspectives highlighted, the term SA is often used to describe the experience of comprehending what is happening in a complex, dynamic environment in relation to an overall objective or goal. Regardless of theoretical perspective, it is generally accepted that this experience involves both acquiring and maintaining a state of awareness (Endsley 1995b, Smith and Hancock 1995). This view is shared by Dominguez (1994) who, in an attempt to define SA as both a process and a product, compared 15 definitions and concluded that the perception of expected information in the environment occurs in a continual cycle which is described as 'continuous extraction'. To be useful therefore, a model of SA should reflect the equal importance of both the continuous process of acquiring and maintaining SA and the state of SA itself.

4.2 Situated Action

An area that we see as important, but on which there is much disagreement, is consciousness. Compare, for example, the description of Endsley's (1995b) model of SA with that prescribed by Smith and Hancock (1995). This tension reflects the broader 'cognitive' debate in HCI introduced earlier. Whilst the information-processing view within the cognitive paradigm has contributed substantially to psychology-oriented research, there is a growing view that it is limited and presents a constraint to the advancement of theory in the area. If research in SA is to take a broader perspective than that offered by the information-processing model, it will have to concern itself with issues which reflect deliberate action on the part of those being studied in the specific context in which they are acting. A model informed by this stance, would have to acknowledge the existence of consciousness and its contribution to situated action (Suchman, 1987) (or 'purposeful action'), and reflect that an individual's awareness of a situation consciously effects the process of acquiring and interpreting new information in an continuous, proactive cycle.

4.3 Context

The positions taken in themes I and II reflect the importance of the individual making sense of situations in a particular context, and frame SA in this light. Any model of SA should explicitly reflect this, showing that accurate interpretations of a situation cannot be made without an understanding of the significance of the situation within a particular context. In other words, the context in which an individual is acting has to be understood in order for us to appreciate the importance of particular situations and their likely relation to SA. This coupling of situation to context is suggested as a key issue, and is one which, as we have seen, has emerged as a theme of increasing importance in cognitive science and HCI (Nardi 1996, Hutchins 1995, Suchman 1987, Winograd and Flores 1986).

4.4 Dynamism

When an individual is making sense of the situation in which they are acting, their understanding is informed by them extracting relevant information from their environment. This information is temporal; the same information at different times (and therefore in different situations) may mean different things to an individual. The continuous information extraction process in which the individual is engaged implies that SA requires individuals to diagnose past problems and provide prognosis and prevention of future problems based on an understanding of current information. This suggests that a model of SA must be inherently dynamic, reflecting the development of SA over time, and that it must be responsive to environmental changes, for example in the information available to the individual.

4.5 Dynamic SA Model

The four themes have raised issues which can be used to frame a model of SA (see Figure 2). The model encapsulates the inherent dynamism of proactive extraction (founded on the user's awareness), the significance of context (reflecting the situations in which an individual is acting) and the contribution of both of these themes to 'situated action' in SA.

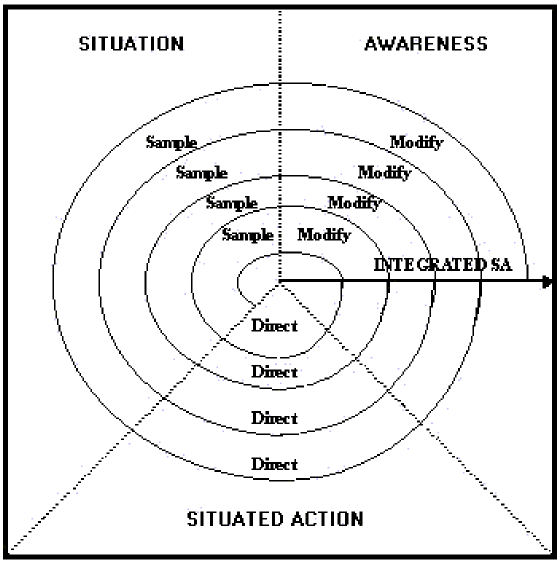


Figure 2 – Dynamic SA Model

The model of SA shown in Figure 2 is adapted from Neisser's Perception-Action Cycle (1976). Neisser's model portrays the adaptive, interactive relationship between an actor and their environment. Pictorially, this model owes much to Boehm's Spiral Model of the software development life-cycle (1988) which is also centrally concerned with issues of iteration and dynamism. It also depicts how awareness information is continuously extracted from a real-world situation and how this is integrated into an individual's awareness to form a mental representation upon which decisions are based and exploratory actions are taken. This model of SA addresses some of the key conflicts between opposing views of SA as either process or product as it encompasses both views. The model shows the inseparability of the SA acquisition process and the resulting (product) state of awareness that recursively direct the selection of relevant situation information in a continuous cycle.

It is worth noting that Norman's well cited action model (1988) appears very similar to Neisser's Perception-Action Model. An important difference, however, is that Neisser maintains that knowledge (or awareness) leads to anticipation of certain information that directs the sampling strategy and increases an individual's receptivity to some elements of the available information. In contrast, Norman's model does not expand on the how information is perceived other than passively and therefore concerns itself only with the process of action.

In Figure 2, the three terms sample, modify and direct are used. In Neisser's model, these terms are related to the environment, knowledge and action respectively. In the adapted model of Figure 2 the terms relate directly to the areas of situation, awareness, and situated action. For the purpose of using Neisser's model in the context of SA, the terms 'situation' and 'awareness' are substituted for 'environment' and 'knowledge' to imply that only a subset of elements of the environment and knowledge relevant to a specific task are considered. This is consistent with the view of SA espoused by Endsley (1995b).

As the individual begins to interact in their environment, they can be considered as moving along the spiral in the model from the central point. An individual may start anywhere in the cycle as, for example, a routine may take over to provoke initial action. Starting arbitrarily, the individual will sample the situation, building a perception of it by extracting and interpreting information content. This may lead the individual to modify their awareness, developing their subjective mental representation of the situation in which they are interacting. Changes in the individual's interpretation of the situation cause them to consciously direct their action (including what/where to sample next), anticipating future states in which they might find themselves and acting accordingly. The 'sample–modify–direct' cycle which the individual can be thought of as having passed through will have developed their awareness in a particular way. As time progresses the individual will cycle through these phases building an integrated awareness that grows with each iteration.

4.6 The Model in Action

In order to illustrate the potential usefulness of the model further, we can consider a specific example. A recent empirical study of a military command and control system revealed that the system displayed many different alerts to the operator. This system required individual alerts to be acknowledged or cancelled using a multiple key switching sequence. However, the vast majority of the alerts were deemed by the operators to be irrelevant and were therefore cancelled using a switching sequence which was consistent for all alert types. It was observed that this alert-cancelling action was carried out so frequently that it had become automatic for the operator. The problem was that the operators also cancelled some alerts containing safety-related information as they carried out the now automatic switching sequence on a screen of multiple alerts – despite the fact that these safety-related alerts were highlighted in a different colour.

We can use the proposed dynamic model of SA to analyse this observed human-computer interaction. In this example, we have based our appraisal of the situation on only observable data; we are talking about SA here as an abstraction that exists only in the mind of the observer.

We could carry out data collection using qualitative methods to probe the users in an attempt to construct a view of their cognitive state, which might enable us to develop a view of the SA of the user, 'defined' in terms of their mental state. In this sense, the type of data to which we have access in a particular instance drives our definition of SA as observer construct of a cognitive phenomenon.

In this example, the sampled situation reveals to the operator that numerous alerts require acknowledgement and this information may have been used to modify the user awareness, but the information contained in the individual alerts is not. The operator action is to cancel multiple alerts as one, chunked, automatic operation. The user is aware only of cancelling multiple alerts and their awareness therefore does not direct them to sample the situation for the cause of the alerts that could be critical in some contexts. The net result is the user has incomplete awareness of a situation despite the fact that the interface displayed the relevant information. Analysing this interaction in terms of the SA model indicates that a breakdown occurs between sampling the situation and modifying the operator awareness.

The model encapsulates a particular view of SA as the fit between a subjective interpretation (awareness) of a situation and the actual situation built through an individual's interaction with their environment (Flach 1996). This perspective of SA suggests that a strong correspondence between the awareness and the situation indicates high SA, while weak correspondence means low SA.

The potential of the model lies in analysing difficulties that affect the user-system coupling, such as interaction breakdowns. The division of the model into areas of activity on the individual's part (sample–modify–direct) provides a structure for researchers to analyse and categorise SA problems. For example, the model could be used to question where the problems in particular situations might have arisen: what information did the individual sample from their environment?; how did this lead them to modify their awareness (what was available through the interface)?; and how, subsequently, did this direct their actions? The structure of the model partitions different areas of interest to allow researchers to concentrate on each as a distinct dimension contributing to awareness that can bring its own set of potential problems. It also allows us to consider the boundaries between these partitions, which is where we believe that many SA difficulties might arise. As individuals integrate sampled information, for example, the modification of their awareness may loosen the coupling between subjective interpretation and the objective situation leading to a reduction in SA.

5 Hazard Analysis

We suggest that the dynamic model of SA proposed in the previous section can be used as a framework for the identification and analysis of hazards relating to operator awareness in the context of system use. Specifically, there are two ways in which the model can contribute to the design of safer systems: identifying interaction breakdowns and identifying automatic interactions, both of which are key to SA. The two areas can be related to research in cognition, specifically the concepts of conscious and automatic cognition, also referred to as reflective and experiential cognition respectively (Norman 1993). Differentiating these two modes of cognition enables us to highlight and compare different aspects of human action which will be of use to our discussion of SA, interaction breakdowns and automatic interaction, and to the improved design of safety-related systems.

Experiential cognition involves the skill of an expert responding automatically to events – without conscious reflection or awareness; in contrast, reflective cognition requires different mental processes based on a higher level of consciousness (Norman 1993). Both modes of cognition are needed and neither is superior to the other – they simply differ in requirements and functions. Rasmussen (1983) also provides a similar view through his 'skill-rule-knowledge' based framework of human behaviour which suggests that human behaviour occurs as a result of different levels of cognition and, implicitly, different levels of consciousness. For example, human behaviour at the skill level, such as an experienced driver changing gears in a car, occurs automatically and without conscious effort (i.e., by experiential cognition).

These issues raises considerations of whether particular interactions undertaken by safety-related system operators should be designed to 'require' automatic or conscious cognition and also how designers might ensure the required cognition through their design. These considerations are important since they have extreme safety implications through their impact on SA.

System interactions should also support the users in achieving their tasks and the design of the interface can have a tremendous affect on the safety of the system (Rajan 1997). Interaction breakdowns can occur when human-computer communication is interrupted - in a safety-related system this could have potentially lethal consequences. Interaction breakdowns occur when a system behaves differently than was anticipated by the user (Winograd and Flores 1986) – when automatic cognition becomes conscious. Interaction breakdowns can trigger an inappropriate action (an act of commission) or it may not trigger any action at all (an act of omission).

An interaction breakdown causes an operator to apply a proportion of their finite cognitive resource to the interaction and not to the system objective. Therefore, interaction breakdowns could be disastrous in a safety-related system such as an aircraft or an air traffic control system if the operator must stop flying or controlling in order to interact with the system. Based on this understanding, it may be argued that the aim of system design should be to eliminate any potential interaction breakdowns, to develop a transparent interface that requires minimal conscious cognition. This sentiment is prevalent within the HCI literature which often equates interface transparency with usability of the system. For example, Norman (1993) argues that interruptions are especially common in the interactions with computer systems and he suggests that to achieve 'optimal flow' (automatic interaction) it is necessary to minimise these interruptions, making the system as usable as possible.

However, it can also be argued that the greatest hazard in a system is associated with the operator 'experiencing' when he should be 'reflecting' – in other words performing automatic processing when conscious thought is required. With experience, automatic human cognition can become the norm; information is perceived, interpreted and acted upon with little or no attention to it. For example, many skilled functions of an air traffic controller possess this characteristic and, for some controllers, it is intrinsic to skill acquisition. Conscious cognition bears a complex relationship to SA, yet it seems inherently unsafe to perform tasks while remaining unaware of them even if they are performed well (Hopkin 1995). The implication is that operator awareness of a situation may not be updated and may therefore be inaccurate. This raises a tension between moves to remove interaction breakdowns by making interactions transparent, and interfaces usable, and the problems caused by the emphasis this places on automatic cognition. There may, we would contend, be times when usability and safety are mutually exclusive since automatic cognition is to be avoided in favour of conscious cognition, with the implication that usability of the system is decreased if the operator is consciously engaged.

The model of SA proposed in this paper may be used as a framework for research studies that aim to identify SA problems associated with interaction breakdowns and automatic cognition by looking for related reductions in integrated SA. These reductions in SA may arise where a mismatch arises between the subjective interpretation and the objective situation. Undertaking research that helps us understand and explain these mismatches should provide input to the interaction and interface design process. They can be used as input to the next generation of the system, which can aim to mitigate against the hazards that they create in current systems.

6 Situation Awareness and Usability

Identifying potential or actual interaction problem areas and addressing them is crucial in safety-related systems and anything that can support this will be a useful addition to the field of safety engineering. Norman (1988) initially suggested that safety-related systems pose a special problem in design and he implied that system safety and usability requirements could be incompatible; although he did not identify when this may be the case.

We have suggested that modelling SA is useful in identifying areas of interface design where safety and usability are mutually exclusive. Specifically, this can occur when the user fails to assimilate critical information resulting from automated interactions as discussed in the previous section. A model of SA could also contribute to the development of system safety cases as safety-related system operators must convince regulatory authorities that their systems are safe to operate and must therefore identify the unique safety requirements relating to their interactive systems (Storey 1996).

It will also help determine the extent to which making the system more usable would actually reduce hazards and increase safety. If it can be shown that making systems more usable in certain situations encourages users to have inappropriate SA, then designers will have to take this into account in designing interfaces and interactions rather than aiming for blanket usability in their systems. This will highlight further complexity in the design of safety-related systems and, through improved understanding of this complexity, help inform interface and interaction design.

There is a general trend to make use of usability in the requirements specification for interactive systems, with usability generally taken to involve not only ease of use but also effectiveness in terms of measures of human performance (Shackel 1991). From this view of usability, safety-related system developers may be tempted to infer that a usable system is, by implication, a safe system. However, as this paper has already suggested, usability and safety can be mutually exclusive properties. So, making use of usability evidence, such as the speed at which tasks may be completed using a given interface, to support claims that aspects of the system are safe may be misleading.

Instead, since safety-related systems are primarily concerned with hazardous failures, safety arguments should focus on these failures and the evidence directly related to them. The model proposed in Figure 2 can be helpful here, in supporting the substantiation of a safety claim as highlighted in the following example:

Hazardous Failure: Controller acts inappropriately due to lack of SA.

Claim: Interface design enables adequate level of SA to be acquired and maintained.

Argument: All safety-significant interactions modify operator awareness.

Evidence A: No automatic safety-significant interactions.

Evidence B: Safety-significant interactions conform to dynamic SA model with no discontinuities, e.g., the sample/modify/direct cycle is followed throughout the user's interaction with the system.

Safety and hazard analysis involve the identification and analysis of risk in order to achieve and maintain a tolerably safe state of system operation. However, as this example shows, it is possible that making an interactive system safe will entail many trade-offs with usability – in this case safety-significant interactions could not be allowed to become automatic or be by-passed in any way. This might be in direct contrast to advice based on usability where, for example, HCI prototyping may reveal a usability requirement for particular complex keying sequences to be replaced with a macro facility allowing a function to be invoked with a single switch action. However, this usability requirement may inadvertently increase the risk of human error if a hazard is associated with the keying sequences. Furthermore, the severity of the hazard associated with the keying sequences may increase during emergency or abnormal situations of a system in use. It seems that it is not enough to simply concentrate on the usability of an interactive system to assure safe operation.

Any design trade-off between usability and safety may also affect the reliability of the cognitive processes involved with acquiring and maintaining SA. If a well-intentioned system developer attempts to eliminate interaction breakdowns in the name of usability, this may have an adverse effect on the SA of the operator; something which is likely to lead to problems in the use of the system. This suggests that SA may be thought of as a critical criterion for safety-related systems and that we should balance the requirements of both SA and usability in the design of interfaces and interaction. In order to advance the field, research needs to concentrate on quantitative measures of SA which may be used to derive safety metrics for evaluating interactive systems. These safety metrics can then, in turn, be used as evidence to support arguments for specific safety claims.

7 Conclusions

This paper has identified operator situation awareness (SA) as an important phenomenon which can be used to examine human cognition in context in order to add value to system safety. The paper reviewed different theoretical views of SA and synthesised key issues from these views into a dynamic model of SA, based upon Neisser's Perception-Action Model (1976). It is suggested that the SA model can be used in suitable studies as a framework for the analysis and identification of hazards relating to operator awareness in the context of system use, and that this might be especially useful in considering safety-related systems. In addition, the results of such studies may be useful in identifying areas of interface design where hazards arise through the development of incomplete SA and where safety and usability are mutually exclusive. Finally, the paper presented a simple example of the use of the SA model to illustrate this position and to show how the SA model can be used in generating evidence to support system safety claims.

The SA model is currently in use in studies of the use of safety-related systems to identify interaction hazards and to make subsequent design recommendations. Only through using the model in complex, real-world settings can an improved appreciation of the model's usefulness be developed as well as the criticality of SA as a phenomenon for the analysis of user-system interaction.

8 References

Adams M J, Tenney Y J and Pew R W (1995), Situation Awareness and the Cognitive Management of Complex Systems, Human Factors, 37(1), 85-104,March 1995.

Billings C E (1995), Situation Awareness Measurement and Analysis: A Commentary, in Garland D J and Endsley M R (Eds.), Experimental Analysis and Measurement of Situation Awareness, Proc. of an Int Conf, FL:USA, November 1995.

Boehm B W (1988), A Spiral Model of Software Development and Enhancement, IEEE Computer, 61-72.

CAA (1998a), Aircraft Proximity Reports: Airprox (C) – Controller Reported, August 1997 - December 1997, Vol. 13, Civil Aviation Authority, London, March 1998.

CAA (1998b), Analysis of Airprox (P) in the UK: Join Airprox Working Group Report No. 3/97, September 1997 - December 1997, Civil Aviation Authority, London, August 1998.

Card S K, Moran T P and Newell A (1983), The Psychology of Human Computer Interaction, Lawrence Erlbaum Associates, Hillsdale, NewJersey.

Charness N (1995), Expert Performance and Situation Awareness, in Garland D J and Endsley M R (Eds.), Experimental Analysis and Measurement of Situation Awareness, Proc. of an Int Conf, FL:USA, November 1995.

Dominguez C (1994), Can SA be defined?, in Vidulich M, Dominguez C, Vogel E and McMillan G (Eds.), Situation Awareness: Papers and Annotated Bibliography, 5-15, Report AL/CF-TR-1994-0085, Wright-Patterson AFB, Ohio.

Endsley M R (1995a), Theoretical Underpinnings of Situation Awareness: A Critical Review, in Garland D J and Endsley M R (Eds.), Experimental Analysis and Measurement of Situation Awareness, Proc. of an Int Conf, FL:USA, November 1995.

Endsley M R (1995b), Towards a Theory of Situation Awareness in Dynamic Systems, Human Factors, 37(1), 32-64, March 1995.

Flach J M (1995a), Situation Awareness: Proceed with Caution, Human Factors, 37(1), 149-157, March 1995.

Flach J M (1995b), Maintaining Situation Awareness when Stalking Cognition in the Wild, in Garland D J and Endsley M R (Eds.), Experimental Analysis and Measurement of Situation Awareness, Proc. of an Int Conf, FL:USA, November 1995.

Flach J M (1996), Situation Awareness: In Search of Meaning, CSERIAC Gateway, 6(6), 1-4, 1996.

Garland D J and Endsley M R (Eds.) (1995), Experimental Analysis and Measurement of Situation Awareness, Proc. of an Int Conf, FL:USA, November1995.

Harris D (Ed.) (1997), Engineering Psychology and Cognitive Ergonomics Volume 1: Transportation Systems, Proc. 1st Int. Conf EP&CE, Stratford-upon-Avon, 23-25 October 1996, Ashgate Publishing.

Hopkin V D (1995), Human Factors in Air Traffic Control, Taylor and Francis.

Hutchins E (1995), Cognition in the Wild, Bradford, MIT Press.

Isaac A R (1997), Situation Awareness in Air Traffic Control: Human Cognition and Advanced Technology, in Harris D (Ed.), Engineering Psychology and Cognitive Ergonomics Volume 1: Transportation Systems, Ashgate Publishing.

Kirwan B, Donohoe L, Atkinson T, MacKendrick H, Lamoureux T and Phillips A(1998), Getting the Picture: Investigating the Mental Picture of the Air Traffic Controller, Proc. Conf. Ergonomics Society, 405-408.

Landauer T K (1987), Relations Between Cognitive Psychology and Computer Systems Design, in Carroll J M (Ed.), Interfacing Thought: Cognitive Aspects of Human-Computer Interaction, MIT Press, Cambridge: MA.

Mace W M (1977), Ask Not What's Inside Your Head But What Your Head's Inside Of, in Shaw R E and Brandsford J (Eds.), Perceiving, Acting and Knowing, Hillsdale NJ, Erlbaum.

Nardi B A (Ed.) (1996), Context and Consciousness: Activity Theory and Human-Computer Interaction, London, MIT Press.

Neisser U (1976), Cognition and Reality: Principles and Implications of Cognitive Psychology, San Francisco, W H Freeman.

Norman D A (1988), The Psychology of Everyday Things, New York, Basic Books.

Norman D A (1993), Things that Make us Smart: Defending Human Attributes in the Age of the Machine, Addison-Wesley.

Rajan J (1997), Interface Design for Safety-Critical Systems, in Redmill F and Rajan J (Eds.) (1997), Human Factors in Safety-Critical Systems, Butterworth-Heinemann.

Rasmussen J (1983), Skills, Rules, Knowledge: Signals, Signs and Symbols and other Distinctions in Human Performance Models, IEEE Transactions: Man & Cybernetics, SMC-13,257-267.

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.

Sandom. C. (2011): Safety Assurance: Fact or Fiction? Conferences in Research and Practice in Information Technology (CRPIT), Vol. 133. T. Cant, (Ed.), *Australian System Safety Conference*, Melbourne.

Sarter N B and Woods D D (1991), Situation Awareness: A Critical but Ill-Defined Phenomenon, Int J of Aviation Psychology, 1, 45-57.

Shackel B (1991), Usability - Context, Framework, Definition and Evaluation, in Shackel B and Richardson S (Eds.) (1991), Human Factors in Informatics Usability, Cambridge University Press.

Smith K and Hancock P A (1995), Situation Awareness is Adaptive, Externally Directed Consciousness, Human Factors, 37(1), 137-148, March 1995.

Storey N (1996), Safety-Critical Computer Systems, London, Addison-Wesley.

Suchman L (1987), Plans and Situated Actions, Cambridge, Cambridge University Press.

Whitfield and Jackson (1982), The Air Traffic Controller's Picture as an Example of a Mental Model, in Johannsen G and Rijnsdorp J E (Eds.), Proc. of IFAC Conf. on Analysis, Design and Evaluation of Man-Machine Systems, 45-52, London, Pergamon.

Winograd T and Flores F (1986), Understanding Computers and Cognition: A New Foundation for Design, Norwood, Ablex.

Author Index

Acfield, Anthony P. and Weaver, Robert A., 5

Becht, Holger, 49
Benard, Vincent, 15
Braband, Jens, 23
Broy, Manfred, 29

Cant, Tony, iii, 37
Caulier, Patrice, 15
Connelly, Simon, 49

Edwards, C.B.H., 57

Hardy, Terry L., 71
Hawkins, Richard, 85
Hussey, Andrew, 49

Liu, Zhuojun, 79

Mahony, Brendan, 37
McDermid, J. A., 93
Mu, Huina, 79

Rae, Andrew, 85
Reinhardt, D. W., 93
Richard, Philippe, 15

Sandom, Carl, 109

Vanderhaegen, Frédéric, 15

Westcott, M., 57

Yu, Peng, 79

Zhang, Yongguang, 79

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, 1720 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, Qeensland 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.