SAFETY CASES

The Development of Safety Cases for Complex Safety Critical Systems
Module Learning Objectives

Learning Objectives

– The student will understand what a safety case is
– The student will be able to structure and prepare the content of a safety case
– The student will understand the strengths and weaknesses of the technique
– The student will be able to critically review a safety case report prepared by others
Module Outline

- Overview
- What is a safety case?
- Advantages and limitations
- Elements of the safety case
- Goal structuring methodology
- Challenges in developing a safety case
- Maintaining the safety case
- Reviewing a safety case
- Key Points and issues

Annexes
- A: Safety case patterns
- B: Safety case modules
- C: Fallacies of safety arguments
Overview
History

Safety cases

- Originated in the British chemical industry CIMAH regulations
- Applied to oil industry (OISC Regs) after the Piper Alpha oil rig fire
- Applied to UK Rail after Clapham junction accident
- Have become part of the EU safety culture
- Embedded in various safety standards
  - DEF-STAN 00-56
  - DEF (AUST) 5679
  - Australian DMO SAMS Framework
  - CMMI SAFE+
  - IEC 61508
Why do we need a Safety Case?

There are various reasons

- You may need a tool for managing the safety of a plant or system
  - Identifying and managing the safety impacts of change
  - Setting safety targets
  - Confidence in meeting safety targets
- To address and reduce legal liability:
  - Statute (COMCARE)
  - Common law (Negligence)
- To make a reasoned argument that a system is (or will be) safe
- It may be a direct or implied regulatory requirement
- To organise and structure a project or plants safety data, information and logical argument
- To identify constraints and where tradeoffs of safety against mission effectiveness have been made
- To identify aspects of operational risk management
What a Safety Case is Intended to Provide

A safety case can provide a consistent and complete:
- Assurance viewpoint
- Focus and rationale for safety activities
- Auditable or review-able approach
- Linkage between hazards and countermeasures
- Demonstration of the interaction of design solutions & standards

A safety case focuses on the product system NOT the development process
- A documented body of evidence that provides a compelling case that the system satisfies certain critical properties for specific contexts
  - DO-178B, Common Criteria, IEC-61508,…
  - “safety case”, “certification evidence”, “security case”,…
How is a Safety Case Different to MIL-STD-882?

■ A MIL-STD-882 System Safety Program
  – Is acquisition focused
  – Looks at the hardware, software, people and procedural hazards
  – Often runs dead after delivery and site activation
  – More likely to address proximal (system) causes of accidents
  – The Safety Assessment Report (882C Task 301) is a close analogue to a safety case report

■ A Safety Case
  – Addresses both acquisition and operations (more operational)
  – Oriented to convince a regulator the plant is safe to operate
  – Strong emphasis on “methods of work” or procedures
  – Requires the definition of the safety management system
  – More likely to address root (organisational) causes of accidents

■ Differences driven by different needs
So What is a Safety Case?

“A safety case is a documented demonstration by an organisation of the way in which hazards at a facility (read weapon system) are managed”

IEAUST, 2002

“A documented body of evidence that provides a convincing and valid argument that a system is adequately safe for a given application in a given environment”

Adelard

Safety Case ≠ Document
What is a Safety Case?

■ Requires at a minimum [5]:
  – Supporting evidence
    • Results of analysis, test, demonstration etc that provide the fundamental evidence upon which the argument for safety is based
    • Because argument without evidence is unfounded
  – A High level argument
    • A number of separate arguments regarding safety aspects
    • A convergent conclusion: “That the system meets its safety criteria and therefore is acceptably safe to be allowed to operate in accordance with its defined objectives and procedures…”
    • Because evidence without argument is unexplained

■ A safety case
  – Is the totality of the safety evidence NOT just a safety case report
  – What you have achieved, NOT how hard you tried…
Advantages & Limitations
A Safety Case Will Not Affect Organisational Culture

- The *original* use was as an argument to a regulator
- Independent review and certification is an essential part of the methodology
- A safety case will not *in itself* make an organisation safer
  - If an organisation exhibits:
    - Over-confidence & complacency
    - A low priority assigned to safety
    - Flawed resolution of competing goals
  - It is *likely* that any safety case developed is flawed as well
- It can be used as a tool by a regulatory authority to change the culture of an organisation
Case Study: Clapham Junction Accident - British Rail

- A mature, regulated, safe transport system
- In 1988 35 people were killed in a collision between two trains at Clapham junction resulting from a signaling failure.
  - The signal failure caused by a wiring fault introduced in maintenance.
  - A wire was improperly terminated and by-passed crucial interlocks
- It was felt in the ensuing enquiry that the accident had been symptomatic of the whole culture
- British Rail could not demonstrate why their system was safe
  - No whole-system safety case
  - No ability to effectively evaluate the impact of change upon safety
- Institutional + Technical change ⇒ Accidents
- UK Regulator now requires comprehensive safety cases that demonstrates sufficient consideration of management of all credible hazards
Addressing the Aspects of Negligence

- As a **legal** concept, negligence comprises proving four key issues
  - Did we cause it?
  - Could we have foreseen it?
  - Was it preventable?
  - Would it have been reasonable for us to have prevented it?

- A safety case can reduce the likelihood of an accident occurring which was:
  - Caused by us…
  - Foreseeable…
  - Preventable…
  - Reasonable for us to have prevented…

- Provides documented evidence to the above
Addressing Statutory Requirements

- As an example
  - For Commonwealth facilities and systems the *Occupational Health and Safety (Commonwealth Employment) Act 1991* is applicable.

- A safety case can address the following regulatory elements:
  - 1.05 (1) Identification of hazards
  - 1.05 (2) Risk assessment of the Identified hazard
  - 1.05 (3) Apply an acceptable method of risk assessment
  - 1.06 (1) Introduce controls to eliminate (a) or minimise (b) risk
  - 1.06 (2) Apply a hierarchy of hazard control measures
  - 4.13 Provide designers with a source of safety risk data
  - 4.18 (1) Identify hazard control procedures and their minimum competency standards
  - 4.19 Provide a record of relevant safety data
Elements of a Safety Case
Elements of a Safety Case

Safety Cases normally contain:

- High Level Argument & Narrative
  - Goals & strategies
  - Constraints & assumptions
- System Description
  - Requirements, design & operational aspects
- Formal Safety Assessments (Of the developed system)
  - Safety analyses (focusing on hardware, software and personnel)
  - Safety verification results (Independent or otherwise)
- Safety Management System
  - How the system is/will be maintained and operated safely through life
- Safety of Development Program
- Supporting Material
  - Detail analysis data, reference material etc…

It should also provide traceability amongst safety data

- For example from case to the system safety requirements
The System Description

Describe the system

- **SYSTEM = A composite, at any level of complexity, of personnel, procedures, materials, tools, equipment, facilities, and software (MIL-STD-280).**
- DEFINE THE BOUNDARY CAREFULLY IT AFFECTS SCOPE
- Mission goals
- Mission functions needed to achieve the objectives.
- **Mission Profiles or a “Design Reference Mission”**
  - What Performs them (Human, Hardware, Software, Documents)
  - System Components (Hardware, Personnel)
  - System Interfaces (Internal and External)
  - System Operating Procedures (How Documented)
  - Facilities – Repair and Training

The operational context – Important for software
The Formal Safety Assessment

- Identify safety risk acceptance criteria
  - Who does it (or did it for the development)
  - What level of risk is acceptable (bounds the ALARP region)
  - Rules for prioritising the risks

- Identify the bounds of the analysis
  - System boundary, mission phases, lifecycle etc
  - Hazard types addressed, severity levels etc
  - Qualitative versus Quantitative Analytical Techniques

- Identify the accidents (Top level event(s)) you are concerned about

- Does mission timeline affect safety targets?
The Formal Safety Assessment

- Identify Hazards and Targets
- Link the Hazards to the Accident (The Sequence)
- Assess Risks (Causal Factors, Probability, Severity)
- Identify Controls
  - Include Performance Standards for Procedural Controls
  - Identify Costs of Controls
- Document in a Hazard Log (PHA form can be used)
- Review and evaluate overall residual risk
- Prepare the capstone High Level Argument
  - Concise
  - Argue from the facts
  - identify assumptions and limitations
Developing the High Level Argument (HLA)

- Supports the primary purpose of the SCR
  - It forms part of the formal safety assessment
  - It should communicate a clear, comprehensive and defensible argument that a system is acceptably safe to operate in a particular context [4]
  - HLA should form the ‘spine’ of the safety case showing how these elements are related and combined to provide assurance of safety

- Within the limits defined [Scope], the system [System Description] is SAFE because all identified hazards [System Hazards] and requirements [Safety Requirements] have been addressed. Hazards have been sufficiently controlled and mitigated [Hazard Control / Risk Reduction Measures] according to the safety risk posed [Risk Assessment]. Evidence [Safety Analysis / Test] is provided that demonstrates the effectiveness and sufficiency of these measures. Appropriate roles, responsibilities and methods were defined throughout the development of this system [Development Process Justification] [Safety Management System] and defined future operation [5]
Developing the HLA

- If a HLA is not clear to its audience the SCR is a failure!

- It should:
  - Record and justify the safety goals & strategies
    - Eliminated all Class I hazards
    - Reduced Class II hazards to acceptable risk levels
    - Applied DEF-STAN 00-56 Safety strategies 3.1 etc
  - Record constraints placed on safety by the development e.g.
    - COTS/NDI
    - Software Language
    - Legacy system issues
  - Organise the usual multitude of project documents into a concise and integrated statement
Developing the HLA

- The HLA should be a natural outcome of the safety management process
  - Planning Aspects
    - Defines goals of the safety program
    - Defines the constraints to work within
    - Establishes verification requirements
  - Control Aspects
    - Do subordinate goals actually achieve the parent goals?
    - Are all justifications/assumptions valid?
    - Is the supporting context correct and consistent?

- For complex cases consider the use of formal techniques such as goal structuring notation to organise the HLA
The Safety Management System Summary

- Structured description of safety management strategies
  - Identifies Procedural Control Methods per element
    - Safety management policy (Technical and Operational)
    - Design safety procedures
    - Operating safety procedures
    - Maintenance safety procedures
  - Risk acceptance/carriage responsibilities

- Usually has supporting appendices
  - Organisational Control Hierarchy
  - Accident Procedural Control Measures
  - List of Principal Manuals & Documents that support the safety management system
  - Identify each safety management element and who is the owner.
  - Top “10” hazards as ranked and their controls
What the Safety Management System Does

- Organisational and supervisory functional failures are significant causal factors in human error
- Organisational change can lead to loss of responsibility and accountability for safety
- A safety management system is intended to
  - Provide a set of competent technical activities
    - Change management
    - Risk control
    - Collection, storage, analysis and distribution of safety data
  - Reduce the likelihood of day to day human error in technical activities
  - Minimise the effects of organisational change
  - Explicitly identify the ‘cost of safety’
The Safety Management System Summary

DEFENCE AGAINST ADVERSE ORGANISATIONAL CHANGE:
- a. Diffusion of Responsibility
- b. Lack of independence/status
- c. Poor Communication Channels

DEFENCE AGAINST ADVERSE TECHNICAL ACTIVITIES:
- a. Superficial Safety Effort
- b. Ineffective Risk Control
- c. Failure to Evaluate Changes
- d. Information Deficiencies

DEFENCE AGAINST COST CUTTING

IMPORTANT BUT OFTEN NEGLECTED

- Organisation Interfaces
- Organisation Structure Diagram
- Responsibility Matrix

- Identify Management Policy
- Identify Risk Acceptance Responsibilities
- Identify Safety Procedures
- Identify Performance standards
- Identify Cost of Control

- Safety Management Policy Statement
- Operating Safety Procedures
- Design Safety Procedures
- Maintenance Safety Procedures

- Justify System Safety
- Develop Safety Case
- Establish Safety Management System
- Identify Controls for Element “X”
The Safety Case Report

- The Safety Case is normally documented in a Safety Case Report
- Presented to a Regulator.
- Usually issued in a Conceptual, Preliminary & Final version
- Answers the Question:

**What assurance can you provide that this system is of such integrity?**
Goal Structuring Methodology
Goal Structures and Notation

■ Purpose of a goal structure
  – To show how safety goals are
    • Broken down into sub-goals
    • Eventually supported by evidence (solutions)
  – To making clear the
    • Strategies adopted
    • Rationale for the approach (assumptions + justifications)
    • Context in which goals are stated

■ Goal structuring notation (GSN)
  – Simple graphical notation (syntax) of symbols and relations
Goal Structuring Notation

- Advantages:
  - Simple
  - Structured hierarchical breakdown
  - Expressive (captures the elements most important to the HLA)
  - Capable
  - Can be used at various stages of argument development
  - Method guidance exists (e.g. concerning syntax)
  - Semantics well defined and understood
  - Increasing industry adoption (especially for software)
  - Can use it as the ‘organising principle’ for large textual reports

- Disadvantages:
  - Learning curve (Easy to read, harder to develop)
  - Doesn’t stop you writing bad arguments!
  - Viewed as an end in itself, it’s a communication tool
Goal Structuring Methodology

- Establish top level goals (customer/statutory)
  - Record the stakeholders for the goals

- Define derived requirements (standards, codes etc)
  - Establish as goals and link to top level goals
  - Some requirements may be constraints

- Break down the top level goals into sub-goals

- Show how design and analysis decisions meet requirements (goals) by use of strategies
  - When the safety case is produced in step with the development then the safety case can be used to record the decisions as they are made

- Justify strategies
  - Especially useful for qualitative arguments regarding human error
Goal Structuring Methodology

G1. Prevent Mach. Induced Human Error

G7 Eliminate Inconsistent Behaviour (3)

S12 Provide consistent behaviour for all modes of {Interface Y}

S1 Strategy Eliminate Accidental complexity

C1 Safety Related Interface of System X (n = # functions)

{Interface Y} Provides {Interface Y}

S1 Eliminate Accidental complexity

G2 Eliminate Unintended side effects errors

n

G3 Eliminate Operator Authority Limits errors

n

G4 Eliminate Indirect Mode Change

n

G5 Eliminate Interface Interpretation

n

G6 Eliminate Inadequate Feedback

n

S3 Annunciate operator limits changes

S4 Map limits one-to-one to hazards

S5 Minimise Indirect Mode Changes

S6 Annunciated mode consistent with internal

S7 Conditioning of I/f not mode dependant

S8 Annunciate operator limits changes

S9 Annunciate 'interpretation' mode changes

S10 Provide system status to operator

S11 Provide independent feedback

S13 Provide consistent behaviour for all modes of {Interface Y}

G7 Leads to supervisory. error of omission if operator scanning for expected result

© Jacobs Sverdrup Australia Pty Ltd

REV 5.0 11 APR 2005 Page 33
Goal Structuring Methodology

- State any assumptions made
  - Requirements (goals), environmental conditions, usage restrictions, independence assumptions etc

- Document alternative design choices and decision criteria

- Make explicit which parts of the system a goal relates to by linking goals to the ‘system model elements’
  - Sub-goals inherit the models of its parent unless otherwise specified

- Plan the verification effort so its contribution to higher goals is clear

- Conduct the verification program to verify requirements

- Link this detailed analysis (supporting evidence) to the HLA
Extensions to GSN

GSN has been extended in recent years to include:

- Safety case patterns
  - Standardised templates to encourage re-use of successful arguments
- Safety case modules
  - Allow the partitioning of cases into more easily managed modules

See Annex A & B for further discussion
Case Study: Weapon System Safety Case

- Integration of a new missile to an aircraft
- Multiple domains and stakeholders
  - Explosive ordinance safety
    - Insensitive munitions
    - Safety and arming functions
    - Weapon context
  - Missile level safety
    - Post launch flight reliability
    - Active separation flight control
    - Safe carriage, launch and jettison envelopes
    - Aircraft carriage and launch context
  - Aircraft weapon system level safety
    - Mechanical and electrical compatibility
    - Safe separation launch and jettison
    - Flutter effects
    - Active oscillation control
    - Weapon system roles, states and modes context
- Multiple modes ground vs air vs training or tactical
- Layered and partitioned safety case
Case Study: Weapon System Safety Case

**Vertical**

- **F/A-18 WEAPON SYSTEM GROUND OPS ARE SAFE**
  - ARGUMENT

- **EXPLOSIVE ORD. INHERENT FLIGHT SAFETY**
  - ARGUMENT

  *EO vendor*

  **Assumed context** that All explosives hazards are either internally controlled or identified to the weapon design process for control

**Horizontal**

- **F/A-18 WEAPON SYSTEM FLIGHT OPS ARE SAFE**
  - ARGUMENT

- **MISSILE INHERENT SAFETY**
  - ARGUMENT

  *Missile vendor*

- **AIRCRAFT INHERENT FLIGHT SAFETY**
  - ARGUMENT

  *System integrator*
  - Functional safety
  - Stores clearance

**Assumed context** that All missile hazards are either internally controlled or identified to the aircraft design process for control.
Case Study: RVSM Pre-Implementation Safety Case [6]

- EU Air Traffic Management 2000+ ‘One European Sky’
  - Major institutional changes
  - Major technical changes

Diagram:

- Whole-airspace services
  - Area-based services, federated systems
  - Integrated systems

To

Area-based services

Whole-airspace services

Integrated systems
**Case Study: RVSM Pre-Implementation Safety Case**

**Overall argument**

- **G001** Risk under RVSM is tolerable
  - Cr001
    1. Meet TLS for vertical risk
    2. Overall risk not to increase

- **A001** Pre-RVSM risk is tolerable

- **G002** Safety Reqs for vertical separation (RVSM) are complete and correct

- **G003** Safety Reqs are fully realised in Concept

- **G004** Safety Reqs will be fully realised in implementation
  - ArgReqmtsRealised

- **G005** ‘Switch-over’ will not endanger ongoing ATS

- **G006** All arguments and evidence are based on known, consistent system configuration
Case Study: RVSM Pre-Implementation Safety Case

**Implementation**

**ArgReqmtsRealised**

**C1 States involved (n = # states)**

**ST003** Argue that there is sufficient evidence that implementation will meet SRs

- **G022** Impl. Guidelines specified for States
- **G023** Each State has produced detailed Safety Plans to satisfy Guidelines
- **G024** Each State will certify compliance with its Safety Plan
- **G0025** Height monitoring will assess achieved aircraft height-keeping performance against MASPS

**ST004** Argue that evidence of implement. is trustworthy

- **G011** Database will record MASPS approval of aircraft using data from independent sources
- **G0025** Height monitoring will assess achieved aircraft height-keeping performance against MASPS
Challenges in Developing the Safety Case
Challenges

- **Challenge**
  - **Cost**
    - Complexity of argument
    - Volume of data (Refer Case Study)
    - Multi year effort (for major system)
    - Iterative approach
  - **Lack of structure/clarity**
    - Large amounts of supporting material
    - Structure of argument poorly recorded
    - System of systems safety cases
  - **Inconsistent/Incomplete**
    - Inconsistent analyses
    - Incomplete assessment
    - Constructed post development

- **How to Address**
  - **Cost**
    - Rigorous control of scope
    - Integrate with development
    - Incremental objectives
    - Maturity requirements
  - **Lack of structure/clarity**
    - A single data model to integrate analyses
    - **Structured High Level Argument**
  - **Inconsistent/Incomplete**
    - Consider tool support
    - Cross check data as part of analysis
    - **Use of Goal/Structuring notation**
Legacy System Safety Cases

- Legacy system safety cases
  - Usually required after a major accident or incident
    - Development sometimes complicated by internal politics & tensions
    - Actual accident data can be of great utility
    - Important to maintain balance in the safety case
  - Usually have a lot of data about actual hazards
    - Actual statistical data from operations can aid quantitative assessment
    - In a large organisation data may be dispersed and difficult to obtain
    - Can use this existing data to bootstrap the analysis
    - Can structure the case effort around extant hazards
    - Far greater reliance and involvement of operational staff
    - Usually need an initial “brainstorm” to collate hazards/controls
  - Safety case should reflect a warts and all view
    - ‘What we achieved not how hard we tried’
Maintaining the Safety Case

“… any amendments to the deployment of the system should be examined against the assumptions and objectives contained in the safety case.”

DEF STAN 00-56

“Regulation 6(1) requires a safety case to be revised whenever appropriate, that is whenever any of its contents would otherwise become inaccurate or incomplete.”

HSE Railways (Safety Case) Regulations 1994, quoted in [4]
System Change Analysis and the Safety Case

- The HLA can be used to evaluate the impact of changes upon the overall safety case
  - Safety Case needs to be in place before changes
  - If the system is ‘brittle’ the safety case may also be brittle

- Changes can be evaluated for adverse effects upon:
  - Safety goals
  - Solutions
  - Safety Strategies
  - Consistency with system and environmental models
  - Impact is propagated through the HLA until all issues identified and resolved
    - Have the claims changed?
    - Are arguments invalidated or new ones needed?
    - Is evidence still relevant, new evidence needed?
Maintaining and Reusing the Safety Case Report

Once developed it is generally accepted that the safety case be a living document

– Judging when appropriate to do so can be a challenge
– A 20 year life of the system can also be a challenge

  • **Case study:** The Long Term Safety Review of the U.K’s Magnox reactors, quoted in [4] found that:

    – Lack of maintenance to the original safety case had caused it to become inconsistent with current plant design and operations
    – Adding to and re-evaluating a safety case that has become ‘out of date’ with respect to current safety standards was problematic

– In large organisations it may be difficult to establish the ‘owner’

Ideally we could reuse the patterns of successful safety cases

– In practice problems may arises due to:

  • Inappropriate reuse of arguments
  • Lack of traceability
Reviewing a Safety Case
Reviewing a Safety Case

Safety cases are at their heart logical reasoning

- Based on a mix of inductive and deductive reasoning...

  - Inductive argument
    - ‘Reliability trials give a MTBF of “X” hrs, therefore in service we can expect a MTBF of “X”’

  - Deductive argument
    - ‘If fault X and human error Y then hazard state Z will occur’
    - ‘If interlock A active in the presence of fault state X then Z will not occur’

How do we evaluate the validity of these claims?

- Basic principles of inductive and deductive reasoning
- Evaluate for fallacies or errors in reasoning (acceptable, relevant, sufficient)
- Principles of falsification and strong proofs
- Quality of evidence
Inductive and Deductive Arguments

- An argument contains one or more claims (premises) and a conclusion
  - Premises must support the claim

- Deductive argument
  - Premises must be true
  - Argument must be valid e.g. logically entail the conclusion
  - To assert the premises but deny the conclusion should throw up a logical contradiction
  - Examples are fault tree node arguments

- Inductive argument
  - Premises support but do not logically entail their conclusion
  - Assumption of uniformity, as Hume points out, this may be false
    - ie extending bench test reliability data to assert a field reliability figure
## Fallacies of Safety Arguments

<table>
<thead>
<tr>
<th>Relevance fallacies</th>
<th>Acceptability fallacies</th>
<th>Sufficiency fallacies</th>
</tr>
</thead>
<tbody>
<tr>
<td>Appeal to Improper authority</td>
<td>Fallacious (ambiguous) use of language</td>
<td>Hasty inductive generalisations</td>
</tr>
<tr>
<td>Red herrings</td>
<td>Arguing in a circle</td>
<td>Arguing from ignorance</td>
</tr>
<tr>
<td>Drawing the wrong conclusion</td>
<td>Fallacy of composition</td>
<td>Omission of key evidence</td>
</tr>
<tr>
<td>Using the wrong reasons</td>
<td>Fallacy of division</td>
<td>Ignoring counter-evidence</td>
</tr>
<tr>
<td>Post hoc fallacy</td>
<td>False dichotomy</td>
<td>Confusion of Necessary &amp; Sufficient Conditions</td>
</tr>
<tr>
<td>Slippery slope</td>
<td>Faulty analogy</td>
<td>Gambler’s fallacy</td>
</tr>
<tr>
<td></td>
<td>Distinction without a difference</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Pseudo-precision</td>
<td></td>
</tr>
</tbody>
</table>

See Annex C For a full discussion
Quality of Evidence

- Usually a mix of analysis, test and field history
  - Check for validity as a support for a premise
  - Watch for too much reliance on too small a pool of data
  - Sensitivity analyses of statistical analyses should be provided
  - In small sample sets look for discarded outliers and consider why they were rejected
  - Watch for magic bullet approaches that seek to address all issues with a single analysis technique or method

- Generate metrics
  - Common hazard countermeasure types and their frequency of use
  - Countermeasures per hazard
    - What countermeasures are vulnerable to single point of failures
  - What countermeasures are procedural only
  - What are preventive vs protective and what is their hazard coverage
Principle of Falsification

- Good arguments are those that are subject to falsification
  - Arguments should be uniformly applied and also specific
  - This allows us to test the argument against the evidence provided
  - Prevents side stepping of issues by the ‘not what I meant’ response
  - Should not rely on ‘ad hoc’ moves to explain apparent inconsistencies of the safety case argument with real data

- Safety should be able to be strongly confirmed
  - Challenge (test) the system and it still maintains safe behaviour
  - System responses to unexpected (un-designed, un-tested for) challenges are still safe
  - Classic role of IV&V is to provide this ‘strong confirmation’
  - A robust system’s safety can be strongly confirmed, does not rest upon narrow assumptions as to safe inputs
Review Process

- The types of errors found by different reviewers will vary widely
  - Ensure that a safety case is exposed to broad peer and independent review
  - Ensure that reviewers are familiar with possible argument error types
Key Points and Issues
Key Points and Issues

- Safety Cases are an effective means to draw together the threads of a large safety program
  - Many of the analysis and assessment products of a MIL-STD-882 safety program can be used to support the safety case
  - It has a greater focus upon operational safety than a MIL-STD-882 safety program

- A major program safety case requires significant effort, but such effort is still small in comparison to overall lifecycle costs

- Properly structured and maintained it can support:
  - Safety analysis of modifications to the system
  - Changes in the operational context

- It should reflect what you achieved **not** how hard you tried
References


Annex A

Safety Case Patterns
High level Argument Patterns [5]

- Capture successful argument approaches that can be used within the safety case
  - ‘Best practice’ arguments, capturing:
    - Corporate expertise and standards captured ‘lessons learned’
    - Previous project’s successful approaches
    - ‘Tools of the trade’ e.g. design patterns such as Non-homogenous redundancy, safety kernel etc
  - Semantics (how we structure the argument) rather than the syntax (how it is expressed) of the safety case
  - Introduces the concept of patterns

- Patterns can emerge at different levels in the argument:
  - Architectural (Top down): e.g. Hazard → countermeasure directed breakdown
  - Mechanistic: e.g. Design margin, robustness, redundancy
  - Component/Object (Bottom up): e.g. Fault tree evidence
High level Argument Patterns

- Re-use:
  - Horizontal (across domains), for example from software integrity argument to an ALARP argument
  - Vertical (within a specific domain) for example against standardised regulatory principles:
    - Example: US DOD Nuclear safety assessment criteria identifies mandatory hazards to be addressed
      - An accident, incident or jettison involving a weapon
      - Unauthorised pre-arming, arming, launching, or release of a weapon
      - Inadvertent pre-arming, arming, launch or release of a weapon
      - A breach of security involving a weapon
Structural and Entity Patterns Abstraction [5]

- Need abstraction syntax to support patterns
  - Structural abstraction
  - Entity abstraction
  - Allows us to show:
    - Instantiation (argue from general case to specific)
    - Subsequent development (Placeholders for specific arguments)
    - Choosing between alternate paths (For this case choose from menu of options defined by higher GSN structure)
  - Re-use is a critical issue for product lines or families

- A pattern by definition needs to contain
  - The ‘common’ safety argument including it’s context
  - A generalised (abstract) approach to the solution
  - Consequences of using the safety case (e.g. constraints)
    - For example: Applying a SPOF safety strategy leaves you vulnerable to common cause and multiple failure scenario’s
C1 Safety Related Interface of System X (n = # critical functions)

Provides {Interface Y}

Instantiate

Multiplicity

G2 {Function Y} IS SAFE

G1. SYSTEM IS SAFE

S1 Argument that All functions safe

n

Selection from option

G3 {functional Safety} Exclusive goal

G4 {functional Safety} Exclusive goal

Develop further

Instantiate for specific function and develop further
Annex B

Modular Safety Cases
Modular Safety Cases [5]

- System development paradigms
  - Modular
  - Compositional
  - Low coupled interfaces, abstraction, data hiding and partitioning

- Should (in theory) be able to ‘modularise’ safety cases
  - Key Assumption: That your system is tractable e.g. modularised
  - Introduces the useful concept of safety case ‘interfaces’

- Possible uses
  - Modular avionics and software (Application vs. infrastructure)
  - ‘System of Systems’
  - System ⇔ Software ⇔ Hardware safety case interrelation
  - Integrating multiple contractors safety cases
  - Making certification arguments when you are not the developer
Modular Safety Cases – Horizontal & Vertical Partitions

Requirement: Argument is ‘top down’
- Can partition into horizontal and vertical modules

- Vertical
  - Claim at one level is objective of lower
  - E.G. Partitioning a system safety case into a system level and software safety case

- Horizontal
  - One argument provides assumed context of another
  - E.g ‘All hazards controlled’ claim relies on assumed context that ‘all hazards have been identified’
Modular Safety Cases – Horizontal & Vertical Partitions

ARGUMENT

MEETS GOAL

ARGUMENT

Development OR Contractual Interfaces

Disconnect from goal and 'Unroll' to full argument structure

Vertical

Horizontal

STRATEGY

MODULE

'Distant' goal
Modular Safety Cases – Horizontal & Vertical Partitions

**Advantages**
- It may be possible to partition the safety case into assurance layers:
  - Valuable for distributed system design where ‘liveness’ dominates safety properties
  - If the calls can ‘jump’ service layers will this show as ‘short-circuits’ in the safety case?
- Modularising can potentially enhance maintainability
- Can support separate development and subsequent integration
- Can be used to clearly define reliance on ‘arguments of others’
  - A major issue in design acceptance activities, especially COTS

**Disadvantages**
- Interface needs to be explicitly defined and agreed
- Interface specification or contract required
Modular Safety Cases - Interfaces

- GSN has been augmented with interface notation and contracts (specifications)
Modular Safety Cases – Interface Specifications

For each interface element specify the ‘contract’

- For example Module A requires Goal ‘A-G1’ to be satisfied, module B contracts to do so, the contract must define which goal in module B does this e.g. B-G3 satisfies A-G1

- Should (at a minimum) identify
  - Participating modules
  - Goals matched between modules (see above example)
  - Collective context and evidence that is deemed to be consistent across modules
  - Resolved distant goal, context and solution references amongst modules

- Can use interface analysis (N2 diagrams etc) and other systems engineering techniques to expose critical areas of argument
## Modular Safety Cases – Interface Specifications

<table>
<thead>
<tr>
<th>ATTRIBUTE</th>
<th>COMMENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>MODULE ATTRIBUTES</td>
<td>Defines module contents</td>
</tr>
<tr>
<td>Goals addressed</td>
<td>Purpose of module</td>
</tr>
<tr>
<td>Evidence module contains</td>
<td>Compare for global consistency</td>
</tr>
<tr>
<td>Internal context module refers to</td>
<td>Compare for global consistency</td>
</tr>
<tr>
<td>Arguments requiring external support</td>
<td>External dependencies</td>
</tr>
<tr>
<td>MODULE INTERFACES</td>
<td>Defines interdependencies</td>
</tr>
<tr>
<td>Reliance on external goals</td>
<td>Reference and status re. goal</td>
</tr>
<tr>
<td>Reliance on external evidence</td>
<td>Reference and status re. goal</td>
</tr>
<tr>
<td>Reliance on external context</td>
<td>Reference and status re. goal</td>
</tr>
</tbody>
</table>
Annex C

Fallacies of Safety Arguments
Fallacies of Safety Arguments

The following are definitions of less common fallacies:

- **Post hoc fallacy.** One event follows another therefore there must be a causal relationship

- **Argument from authority.** Basing argument on authority not logic
  - For example, ‘the US Weapon Safety Explosives Review Board (WSERB) has issued an EO rating for this weapon, therefore it must be safe’
  - ‘Historically this high degree of mission success …’

- **Slippery slope.** Arguing that one follows automatically from another
  - ‘If component X fails then component Y will fail and then a catastrophic hazard will always occur…’ BUT not necessarily, this error often occurs in risk assessment

- **Ignoring counter evidence.** Argument is based only on confirmatory evidence and not negative (attempts to disprove it)
  - Reliance on an accident free period as evidence of safety rather than trying to force a system into hazardous states and evaluating its response
Fallacies of Safety Arguments

– **The gamblers fallacy.** Basing estimates of probability upon small samples and prescribing short terms trends to a change in the underlying probability of events
  - Use of ‘accident clusters’ to falsely judge safety
  - Use of a small number of test events to define reliability
  - Use of a small numbers of operating hours to confirm the effectiveness of system safety improvements

– **Circular justification.** Justifications run in a circle also known as ‘begging the question’
  - ‘Since the shuttle is a manned vehicle, the probability of mission success is necessarily very close to 1.0’

– **Red herrings.** The use of irrelevant arguments or data to support a safety claim

– **Arguing from ignorance.** An argument from ignorance asserts that a claim regarding a system or process is true solely because there is no evidence to suggest otherwise
Fallacies of Safety Arguments

– **Pseudo-precision.** This fallacy may occur in two ways:
  • An argument asserts a quantitative claim about a system, component, or process using greater precision than that which may be ascertained from the premises to the claim
  • An argument asserts a quantitative claim that is supported only in qualitative premises to the claim

– **Confusion of necessary & sufficient conditions.** An argument asserts without evidence that a condition that is necessary for a claim to be true is also sufficient for it to be true. For example
  • A claim is made that ‘hazards have been mitigated’ but the evidence demonstrates only that hazard identification has been completed adequately

– **Fallacious use of language.** Language can be used fallaciously within a safety case argument in at least three ways
  • Use of a key term in multiple senses, without distinguishing them
  • Use of a term or reference that has multiple senses consistently, but without indicating which particular sense is intended
  • Defining terms circularly, or contain premises that are nothing more than restatements of claims they are intended to support