Structuring and potentially formalising (Assurance) Case Arguments

Tim Kelly
tim.kelly@york.ac.uk
Overview

• Safety Cases and Safety Arguments
• Structured (but Informal) Arguments
• Considerations in Formalisation
• Structured Assurance Case Metamodel (SACM)
Safety Cases

• The purpose of a safety case can be defined in the following terms:

  A safety case should communicate a clear, comprehensive and defensible argument (supported by evidence) that a system is acceptably safe to operate in a particular context

• Communication is an important aspect
Synthesis of Evidence

- (Dynamic) Test Results
- Analysis
- In-Service Fault Data
- CVs
- Procedures
- Human Reviews
- Failure Modes and Effects Analysis
- Timing Analysis
- Static Code Analysis
- Hardware – software testing
- Simulation results …
Three types of argument

- (Causal) Behavioural arguments of risk management, i.e. how the causes of hazards are eliminated or mitigated, or how the consequences of hazards are mitigated.

- Confidence arguments – arguments that provide confidence in the adequacy of the details of the risk management argument, e.g. justifying the adequacy of hazard identification techniques, or the sufficiency of verification results presented.

- Arguments of conformance / compliance with safety standards, regulations, and legislation – where compliance is not straightforward it is necessary to justify how a project, system design and operation have addressed legal and regulatory obligations.
Arguments

- Historically, narrative text commonly used
  - Shared understanding?
- Structured Argumentation Approaches
  - GSN - Goal Structuring Notation, CAE etc.
  - GSN clearly disambiguates the structure and elements of the argument, it cannot ensure that the argument itself is ‘good’ or sufficient for its purpose
GSN Example

Control System is Safe

Hazard Identified from FHA (Ref Y)

Tolerability targets (Ref Z)

1 x 10^-6 p.a. limit for Catastrophic Hazards

H1 has been eliminated

Probability of H2 occurring < 1 x 10^-6 per annum

Probability of H3 occurring < 1 x 10^-7 per annum

Primary Protection System developed to I.L. 4

Secondary Protection System developed to I.L. 2

Formal Verification

Fault Tree Analysis

Process Evidence of I.L. 4

Process Evidence of I.L. 2

Software developed to I.L. appropriate to hazards involved

I.L. Process Guidelines defined by Ref X.
Supporting Informal Arguments

- **Deductive** arguments (Formal Logic)
  - if the premises are true, then the conclusion must also be true

- **Inductive** arguments (Informal Logic)
  - the conclusion follows from the premises not with necessity, but only with ‘probability’
Formalising the Informal

- Growing interest in how these informal safety arguments may be modelled in formal logic
- The informality of the underlying reasoning present in safety assurance cannot be eliminated
  - e.g. justification of the domain experience of personnel involved in hazard analysis
- However, the informal arguments can be represented by formal logic
Inductive -> Deductive?

- formalisation can involve axiomatising (informal) aspects of the argument at the 'edge' of our argument
  - e.g. ‘all hazards identified’ argument
  - Of course, could structure this further
    - Kicking the can down the road?
    - Further set of axioms covering the informal aspects of the formalised argument
Are all types of safety case argument equally amenable to formalisation?

- valuable service has been performed by 'annexing' the informal arguments to an easily identified location (a form of reductionism)?

- **concern**: illusion of formality created through hiding problematic informal and subjective arguments behind an abstraction

- formalised ‘core’ with informality pushed to the periphery of the formalisation is advantageous or dangerous for evaluation and review?

- formalisation will not reduce perhaps the most significant aspect of the review burden – namely individual review and acceptance of subjective (informal) assertion
Does the subject matter of a safety case argument affect the value of formalisation?

- deductive arguments can form part of a safety case
- when subject matter domain is itself logical
- asserted inferences can become provable inferences
- When safety case arguments (or at least portions of them can become provable) are they perhaps not better represented as evidence (i.e. proof), rather than as informal logic?

- value of a safety case is to represent the informal logical ‘glue’ that pulls together different forms of the evidence (including deductive results – proof being one such example)
Supporting Model Based Safety Cases

- Systems Assurance Task Force within the OMG (Object Management Group) has been developing a standard for the interchange ‘model’ of assurance cases for 10+ years
  - First ARM (Argumentation Metamodel) + SAEM Software Assurance Evidence Metamodel
  - Then SACM 1.0 in 2012
  - Them SACM 2.0 in 2018
SACM 2.0
Supporting Dialectic Arguments

- Neglected aspect of assurance cases
- Dialectic argumentation is healthy
Supporting Confidence

Arguments

- Example: Argumentation about the adequacy of the claims, inferences, evidential links
Supporting Modularity / Packaging

- Modular assurance case management: Managing the division of assurance case arguments and evidence into modules / packages
  - E.g. aligned with architecture, or with supply chain
Supporting Patterns

- Patterns are abstract argument structures with appropriate constraints
  - E.g. long history in GSN (1997)
  - Useful to capture reusable, ‘typical’ argument structures
  - Patterns in SACM generalised beyond simply argumentation (also Artefact and Terminology)
Example: GSN Patterns

A 1-to-n relationship

n = no. of safety related functions

Goal 1
{System X} is acceptably safe

Element requires instantiation

Strategy 1
Argument over all safety related functions implemented by system

Context 1
The safety related functions of {system X} are {functions}

A choice

Goal 2
{Function Y} is acceptably safe

Element requires instantiation and development

Goal 3
Interactions between system functions are non-hazardous

Element requires further development (support)

Goal 4
All system functions are independent (no interactions)
Example: Artefact Patterns
Example: Expression Patterns

NoAcceptCriteria
No explicit acceptability criteria for \{System X\} exist

SysAccSafe
\{System X\} is acceptably safe

SysDesc
\{System X\} Definition

ExistSysSafetyRec
\{Existing System\} Safety Record

AtLeastAsSafeArg
At least as safe as existing system argument

ExistSysDesc
\{Existing System Definition\}

ExistSysAccSafe
\{Existing System\} was acceptably safe

SimilarSys
\{System X\} sufficiently similar to \{Existing System\} to merit comparison

SuffSimilarDefn
\{Definition of 'sufficiently' similar\}

ExistTargets
Safety targets implied by \{Existing System\} safety record

SysMeetsExistTargets
\{System X\} meets or exceeds safety target requirements implied by \{Existing System\} safety record

SimilarOpn
\{System X\} operating behaviour sufficiently similar to \{Existing System\}

SimilarEnv
\{System X\} is operating in a sufficiently similar operating context to \{Existing System\}
Supporting Machine Processing

SACM models are machine processable
- Standardised format for model interchange
- But reasoning is limited by ability to process expressions
Support beyond Natural Language

- MultiLangString could support several ‘dialects’
- Formal expressions
- OCL (e.g. for *ImplementationConstraints*)
- Languages that could support machine evaluation
- Powerful combination with abstract argumentation, and evidence, structures (and appropriate *ImplementationConstraints*)
SACM Concrete Syntax

- **ID**
  - Statement
  - (Asserted) Claim

- **ID**
  - Statement
  - Assumed Claim

- **ID**
  - Statement
  - Axiomatic Claim

- **ID**
  - Statement
  - NeedsSupport Claim

- **ID**
  - Statement
  - Defeated Claim

- **ID**
  - Cited Package [Cited ID]
  - Statement
  - AsCited Claim

- **ID**
  - Statement
  - Abstract Claim
SACM Diagrams
Summary

- Safety case arguments are often informal
- growing interest in formalisation
- Some discussion points:
  - value gained over merely ‘structured’ (model-driven) approaches
  - tradeoffs between precision and accessibility
  - whether all forms of argument are equally amenable to formalisation
- SACM 2 Designed to support all of current (e.g. GSN) practice but not limited to it (e.g. dialectic, better packaging, more support for patterns)
  - Attempting to pave the way towards machine readable and processable arguments