Assurance Case Research: Themes and Directions

Tim Kelly
University of York, UK

E-mail: tim.kelly@cs.york.ac.uk

Overview

- Chronology of Past Research
- Recent Past
- Current Topics
- Research Themes

© Copyright Tim Kelly, 2011. Not to be reproduced without permission of author.
Past Research

- High Integrity Systems Engineering Group has been working on topic of Safety Cases for 20 years
- Early (1990s) focus on improved method and tool support
  - Support for structured arguments
    - Initially Toulmin (Micro Arguments)
    - Then, Goal Hierarchies (initially inspired by RE ideas)
  - Support for safety evidence
    - Data Model for Combining Safety Models (e.g. FTA, HAZOP etc.)
  - ASAM (A Safety Argument Manager), ASAM2, SAM2000 tools
- Late 1990s focus on maturing the argumentation work
  - Refining / Simplifying the Goal Structuring Notation (GSN)
  - GSN Method - 1997
  - Safety Case Patterns – 1997

GSN Pattern Description

```
Goal 1
(System X) is acceptably safe

Strategy 1
Argument over all safety related functions implemented by system

Context 1
The safety related functions of (System X) are (functions)

Goal 2
(Function Y) is acceptably safe

Goal 3
Interactions between system functions are non-hazardous

Goal 4
All system functions are independent (no interactions)
```

A 1-to-n relationship

\( n = \text{no. of safety related functions} \)

Element requires instantiation

Element requires further development (support)

Element requires instantiation and development

A choice
**Past Research**

- **Modular and Compositional Safety Cases - 2001**
  - Initially focused on Integrated Modular Avionics (IMA) examples
  - Modular GSN
  - Safety Case Interfaces and Contracts
  - Utilised and Extended in Later Work:
    - Dependability Cases (Despotou 2004) & Survivability Cases (2009)
    - Product Line Safety Cases (Habli 2010)

- **Systematic Review Processes**
  - Initial done with military and civil aerospace organisations
  - Increased our interest in the fallibilities of argument (Informal Logic)

(Please note that the diagram on the page is not transcribed into text.)
GSN Module Extensions

Extensions:
- Ability to mark a goal as 'public'
- Ability to refer to goals defined in other modules
- Ability to refer to modules
- Ability to place one argument in the context of another

Module Reference

Public Goal

SysAccSafe
(System X) is acceptably safe

ArgOverFunctions
Argument over all identified safety related functions of (System X)

ArgOver Goals

‘Away’ Goal

To be resolved by contract

Module Reference

SRFunctions
Safety Related functions of (System X)

FnASafe
Function A operation is acceptably safe

FnBSafe
Function B operation is acceptably safe

FnCSafe
Function C operation is acceptably safe

FunctionInd
All functions are independent

IndependenceArg

GSN Based Safety Case Interface

Goals 'Provided' / Addressed

Safety Case Module

‘Away’ Context

Context Defined

Evidence Presented

Goals Required

‘Away’ Solution

‘Away’ Goal
Safety Case Contracts

- Safety Case Modules can be composed if:
  - Goals Match (both ways)
  - Context is compatible
- Results can be recorded in a safety case contract
- Establishes a defined record of the inter safety case agreement
- Supports management of change

Example: Safety Case Architecture for IMA

- From FOAS Study

© Copyright Tim Kelly, 2011 Not to be reproduced without permission of author
**Recent Past**

- **Work on Software Safety Cases**
  - Long-term focus on moving people away from simply process assurance, instead hazard focused
  - Weaver – Software Safety Case Patterns based on principles of:
    - Argument according to software failure mode type (Omission, Commission, Value, Early, Late)
    - Primary, Secondary and Control Failure – Similarities with Software Fault Tree Deduction
  - Revision of approach in Hawkins and Kelly – Software Safety Case Patterns – 2009
    - Focus on Derivation of Software Safety Requirements through tiers of development – coupled with ‘backing arguments’ concerning requirements decomposition / refinement / allocation & coverage of hazardous behaviour emergent from software design commitments
    - Coupled with work on Software Safety Evidence Selection
Weaver Omission Argument

DefnOPS

ContextCSF

Identified failure mechanisms describe all possible paths through CSF that include a unique output statement.

ArgFailureMech

Hazardous Software Failure Mode (HSFM) of type Omission abscent in contributary software functionality (CSF)

Software Contribution Pattern

Software functionality which contributes to software hazardous failure mode (SHFM)

SelfResCSF

Safety Requirements of contributory software functionality

HSFM

Hazardous Software Failure Mode

DefnOPS

Definition of output statement

AsOnPrimary

All feasible control paths through CSF include a unique output statement

AsOnSecondary

Failures of other components which could lead to CSF Omission Hazardous Failure Mode are acceptable

AsOnControl

CSF is scheduled and allowed to run once

© Copyright Tim Kelly, 2011 Not to be reproduced without permission of author

Assurance Case Research - 13

THE UNIVERSITY OF YORK

Page 7
Recent Past

- Work on Confidence
  - Deductive-Inductive Argument distinction
    - Closely related to notions of Defeasible reasoning
  - Weaver work on ‘Safety Assurance Levels’ (SALs) alongside consideration of ‘type’ of argument structure (Govier) – 2005
    - Introduced notions of Trustworthiness of Evidence and Relevance
    - Introduced notion of meta-arguments
  - (Also work by BAE Systems on Safety Evidence Assurance Levels)

- As Confidence As Reasonably Practicable – 2009
  - Term first coined by Brian Randell
  - ALARP (As Low As Reasonably Practicable) Risk Management manages hazards
  - ACARP manages assurance deficits – shortfalls from 100% confident, deductive arguments – known unknowns

Assurance Deficits

- Recognised assurance deficits = Something we don’t know (haven’t addressed in the case)
  - A known unknown
  - Potential source of counter evidence
- Increase assurance by addressing deficits
Extending the 6 Step Method

- Explicitly identify potential assurance deficits at each step in argument construction
- Performed deviation-style analysis of each step
  - Based the analysis on HAZOP technique
    - HAZOP guidewords
      - No or none, more, less, as well as, part of, other than, reverse
- Apply and interpret for each step
  - Identify what assurance deficits may arise

Identifying Assurance Deficits During Argument Construction

<table>
<thead>
<tr>
<th>Step</th>
<th>Purpose</th>
<th>Assurance impact</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Identify goals to be supported</td>
<td>To identify and unambiguously state the goals to be supported.</td>
<td>Move, Less, Other than. The scope of the claim is too broadly defined. The effect of this is to be less or none. The result of the claim is that although a certain level of assurance may be obtained, the scope of the claim is too broadly defined. It may be more difficult to determine the level of confidence associated with the claim. Another way of putting it is that confidence is not achieved.</td>
</tr>
<tr>
<td>2. Define basis on which goals are stated</td>
<td>To clarify the scope of the claim, to provide definitions of terms used, to support argument construction.</td>
<td>None. Any claim is too vague or false. The scope of the claim is unclear, due to lack of context. As a result, it is more difficult to understand the scope of the claim.</td>
</tr>
<tr>
<td>3. Identify strategy to support goals</td>
<td>To identify how a goal can be supported.</td>
<td>Move, Less, Other than. The scope of the claim is too broadly defined. The effect of this is to be less or none. The result of the claim is that although a certain level of assurance may be obtained, the scope of the claim is too broadly defined. It may be more difficult to determine the level of confidence associated with the claim. Another way of putting it is that confidence is not achieved.</td>
</tr>
<tr>
<td>4. Define basis on which strategy is stated</td>
<td>To identify and explain assumptions upon which the strategy depends.</td>
<td>All assumptions are, by definition, unexplored. The argument holds only on the basis that the assumptions are true. If there is a lack of confidence in the strategy, then it also has a lack of confidence in the truth of the claim.</td>
</tr>
</tbody>
</table>

© Copyright Tim Kelly, 2011. Not to be reproduced without permission of author.
### Recent Past

- **Work on Confidence (continued)**
  - ACARP work brought out clear distinction between:
    - Epistemic and aleatoric uncertainty
    - Knowledge uncertainty needs to be managed as well as risk

- **Parallel work on Argumentation Metamodel (ARM) and Structured Assurance Case Metamodel (SACM)**
  - OMG Software Assurance Working Group -> OMG System Assurance Task Force
  - Helped document semantics of informal arguments (GSN & Claims-Argument-Evidence)
    - E.g. Inferences as Assertions, Implicit Assertions of Context in GSN
    - Reassuring work!

---

**Identifying Assurance Deficits During Argument Construction**

<table>
<thead>
<tr>
<th>assured</th>
<th>Assurance Deficit</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>To provide justification for why a particular strategy is being used.</td>
<td>Assurance is not provided so any assumptions are not justified. The assumption that there is a lack of assumptions in the argument. The argument is not sound.</td>
<td>No, Less - No justification is provided so any assumptions are not justified.</td>
</tr>
<tr>
<td>More - Although the lack of direct evidence of assurance is important to note, it is only obvious if the assumptions are not met.</td>
<td>More - Although the lack of direct evidence of assurance is important to note, it is only obvious if the assumptions are not met.</td>
<td></td>
</tr>
</tbody>
</table>

### 5. Elaborate Strategy
- Specify the goals that emerge from the chosen strategy.
- Assured: As well as, Part Of

### 6. Identify Basic Solution
- Identify the solution which provides adequate support to the claim.
- Assured: The solution provides some confidence in the goal being supported (is required). Assurance is not at this step, it is unclear why the evidence provides confidence (infer goal being supported). It may be unclear because:
  - There may be an inductive gap between the claim and the evidence. The evidence does not provide a compelling reason to believe the claim is true.
  - There is uncertainty about the truthfulness of the evidence itself. Note that evidence which is untruthful will undermine assurance even in the situation where there may be a deductive relationship between the claim and the evidence.

- Other than: Counter-evidence in any evidence which undermines the confidence in the claim being made. The presence of counter-evidence does not necessarily mean that the argument is inadmissible. It simply means that the confidence in the claim may be lower than it was before the counter-evidence was provided. It is not necessary to determine the impact of the counter-evidence on the claims confidence. In many cases it may still be possible to make a sufficiently compelling argument despite the identification of counter-evidence. Particularly where there are indications which lead the uncertainty caused by the counter-evidence.

- Counter-evidence is not correctly identified then the potential effect on the assurance of the argument cannot be determined. It is not necessary to determine the impact of the counter-evidence on the claims confidence. In many cases it may still be possible to make a sufficiently compelling argument despite the identification of counter-evidence. Particularly where there are indications which lead the uncertainty caused by the counter-evidence.
Recent Past

- Nimrod Review
  - Shone a spotlight on safety case practice
  - Picked up on a number of concerns (as highlighted in 'Safety Cases – Are They Working?' Kelly, 2008)
  - Why were people creating bad safety cases?
    - Many cultural, commercial, and organisational issues
    - Some concerns re: scale and lack of concision
- Joint York-Virginia work on ‘New Approach Creating Clear Safety Arguments’ (Hawkins, Graydon, Kelly Knight)
  - Principle – Safety Cases should have TWO arguments: Safety Argument and Confidence Argument
  - Assurance Claim Points (ACP) provide link between two
  - Qualitative Treatment of Confidence, but not prohibitive of quantitative approaches

Safety vs. Confidence Arguments

- The safety argument documents the asserted arguments and evidence of risk reduction
  - RULES:
    - Everything cited in the safety argument should have a direct role as part of the causal chain to the hazard;
    - All claims in the safety argument must be claims about the system or parts, properties, or properties of parts thereof
    - Artefacts from system development (e.g. test reports and, by extension, their contents) may be referenced only as evidence or context
- The confidence argument documents the reasons for having confidence in the safety argument
  - RULES:
    - confidence argument claims must address (only) the structure of the safety argument (i.e. it's not a free-for-all!)
Assurance Claim Points

- These assertions could, and should, be debated
  - This is the role of the CONFIDENCE argument

- These ACPs correspond to three different types of assertion:
  - Asserted inference (ACP1)
  - Asserted context (ACP2)
  - Asserted solution (ACP3)

Current Activities

- Continue to work on Safety-Confidence Work
  - Additional perspective – Compliance Arguments
  - Showing the effects of refactoring existing safety cases

- Assured Argument Driven Development (A²D²)
  - Joint Virginia-York work
  - Building upon principles of Phased Safety Case Construction (e.g. Bate et al, 1997), Assurance Based Development (Greenwell and Knight, 2010) and Boehm

- Application of Assurance Arguments to domains other than Safety
  - Current UK Government work on Security Cases
  - OMG work is largely security driven
  - DHS Assurance Case Work
Current Activities

- ‘Codification’ of previous work on Software Safety Evidence Selection and Confidence
  - Working closely with a number of companies and helping them develop in-house software safety assurance processes
  - Guidance on how to ‘turn the dial up and down’ on confidence, rather than retrospective evaluation

- Looking at the formal-informal argument interface
  - 2011 Habli (ESEM11) work with NASA Ames
    - Exploring Bayesian models of confidence
  - Example – recent work on autonomous software safety arguments, model checking in context
  - Work on Software Safety Argument Patterns proving to be useful framework

Current Activities

- Integration of Arguments and Evidence (Sun, 2011)
  - Dealing with inconsistency in evidence
  - Meta-model and model-mapping based approach
  - Integrating Arguments and Evidence
    - Interesting split in OMG work: Software Assurance Evidence Metamodel (SAEM) and Argumentation Metamodel (ARM) – leads to examination of the interface
    - Evidence Assertions
      - Results Assertions
      - Descriptive Assertions
    - Relates to Internal and External Validity, and the concepts of Trustworthiness (Integrity of Results) and Relevance (w.r.t. Descriptive Assertions)
Current Activities

- Assurance Cases for ‘Systems of Systems’
  - Utilising Modular Case Concepts (Contracts, Rely-Guarantee)
  - Micro-scale: e.g. Work on Safety Case Architecture to correspond with Autonomous Decision Making Architecture (for UAVs)
  - Mid-scale: e.g. Work on Logistics Applications integrated on Defence Information Infrastructure
  - Large-scale: e.g. for Battlefield Systems of Systems
  - Work on defining and agreeing safety case interfaces

Summary

- Fortunate to have been working with a ‘maturing’ industry, e.g.
  - Those already using GSN, were keen to explore Modular GSN
  - Those practicing safety cases, keen to look at ways to improve practice (such as safety-confidence work)
- Themes
  - Improving structure and rigour, e.g.
    - GSN -> SACM
    - Software Safety Case Patterns
    - Safety-Confidence split
  - Improving explicit management of the (ever-present) informality and confidence issue in assurance argument
  - Improving assurance case processes and their degree of integration with ‘core’ development activities