AUTOMATICALLY DETECTING FALLACIES IN SYSTEM SAFETY ARGUMENTS

Tangming Yuan, Suresh Manandhar, Tim Kelly, and Simon Wells
Universities of {York | Edinburgh Napier}

CMNA 15 @ PRIMA 2015 (Bertinoro, Italy)

- Safety Cases (Tim)

- NLP (Suresh)

- Argumentation & Dialogue (Tommy)

SAFETY CASES

- Long established in many industries as a key element of the safety assurance process
- Critical role in development of safety-critical systems... *where failure could result in loss of life, &/or significant environmental or property damage*
    - Defence, Aerospace, Energy (e.g. Nuclear), Transport (e.g. railways)
- Represents a shift in responsibility onto developers & operators of (potentially unsafe) systems to construct & present well-reasoned arguments so that it is reasonable to conclude that the system can achieve acceptable levels of safety
    - "Communicate a clear and defensible argument that a system is acceptably safe to operate in a given context" Kelly & Weaver (2004)
- Safety Case: Arguments (increasingly GSN)+supporting evidence (Regulatory + management information + review)

ARGUMENTS IN SAFETY CASES

- A Key Components of a safety case: **Safety Arguments**

- These are structured arguments designed to argue that the system is acceptably safe

- Can be free text, e.g.

- But this has drawbacks...

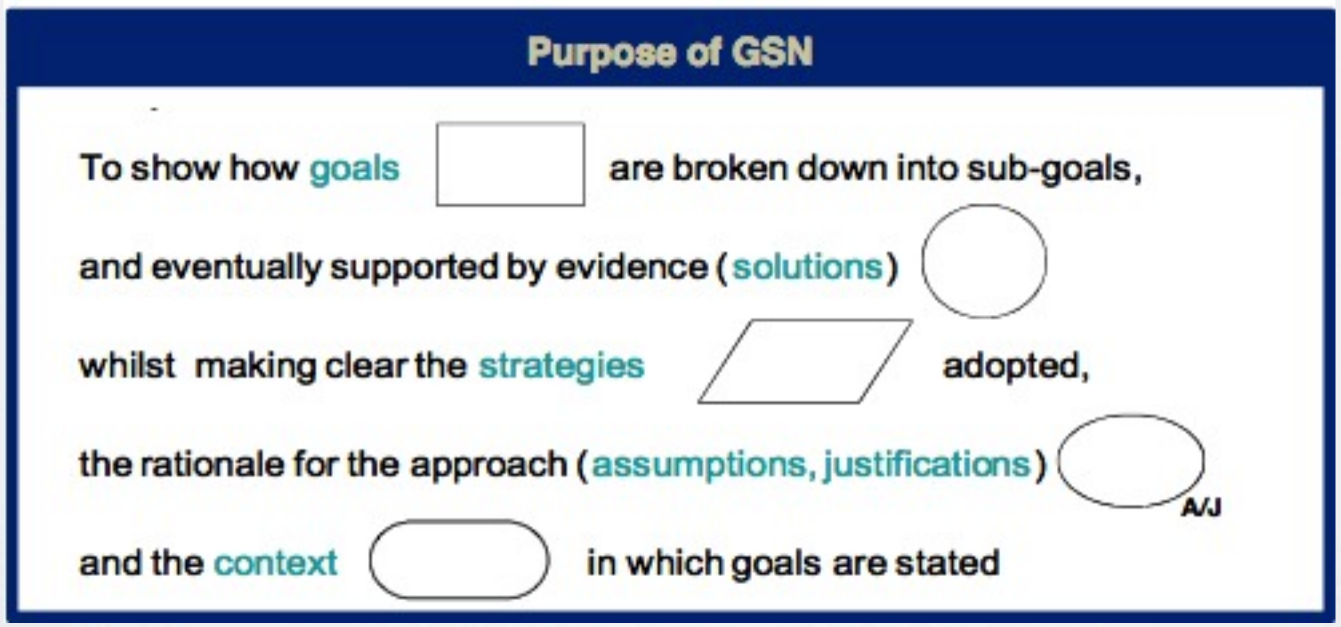
The Defence in Depth principle (P65) has been addressed in this system through the provision of the following:

- Multiple physical barriers between hazard source and the environment (see Section X)
- A protection system to prevent breach of these barriers and to mitigate the effects of a barrier being breached (see Section Y)

For hazards associated with warnings, the assumptions of [7] Section 3.4 associated with the requirement to present a warning when no equipment failure has occurred are carried forward. In particular, with respect to hazard 17 in section 5.7 [4] that for test operation, operating limits will need to be introduced to protect against the hazard, whilst further data is gathered to determine the extent of the problem

GSN

- Goal Structuring Notation
  - Graphical Argument Notation
    - Represent individual elements of a safety argument, e.g.
      - Elements of argument:
        - Requirements, claims, evidence, context
      - Relationships between elements
- Aims to clearly, explicitly, & unambiguously document the safety case



Purpose of GSN

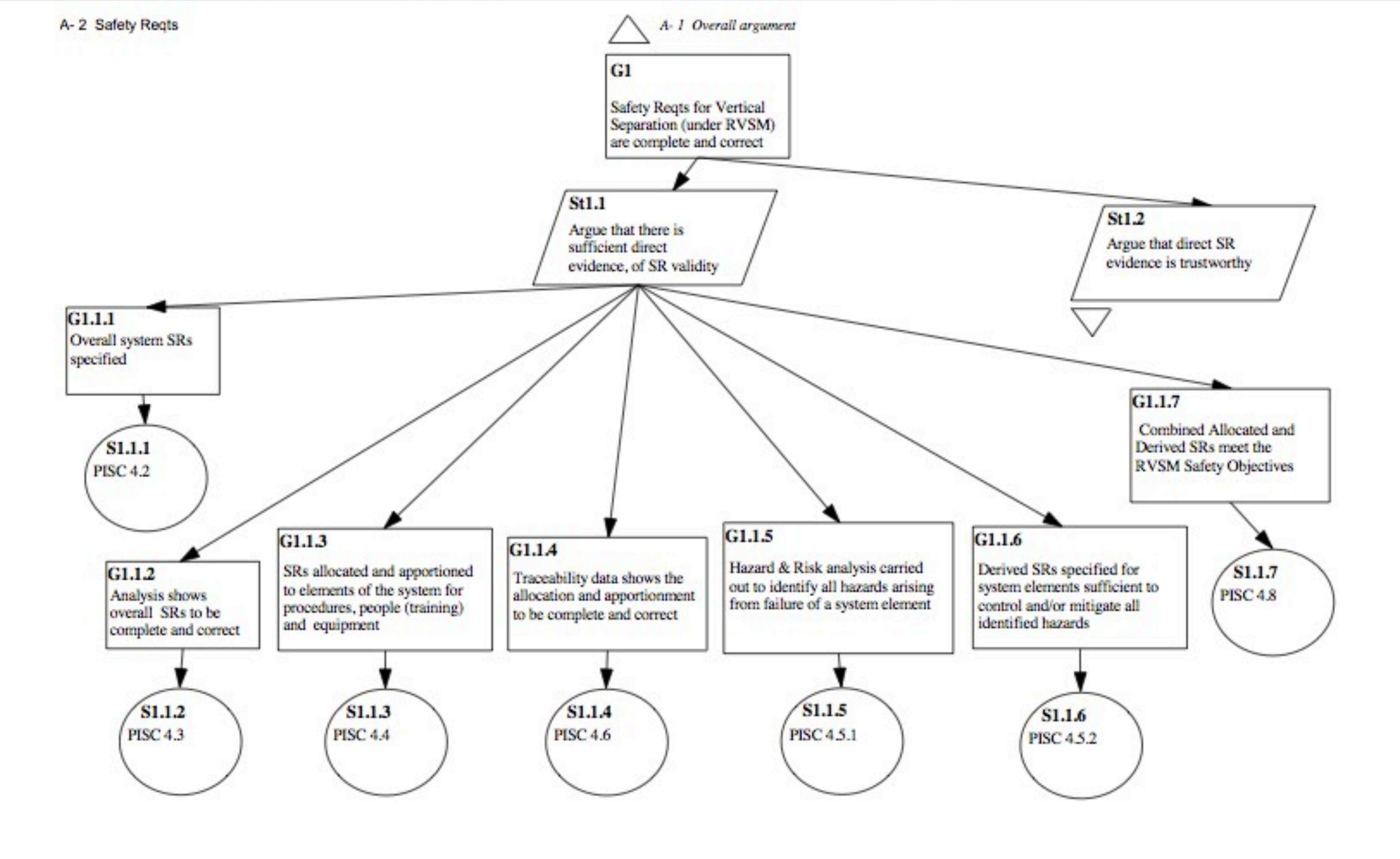
To show how goals □ are broken down into sub-goals,

and eventually supported by evidence (solutions) ○

whilst making clear the strategies ▱ adopted,

the rationale for the approach (assumptions, justifications) ⬭

and the context ⬭ in which goals are stated



A- 2 Safety Reqts

A- 1 Overall argument

G1

Safety Reqts for Vertical Separation (under RVSM) are complete and correct

St1.1

Argue that there is sufficient direct evidence, of SR validity

St1.2

Argue that direct SR evidence is trustworthy

G1.1.1

Overall system SRs specified

S1.1.1

PISC 4.2

G1.1.2

Analysis shows overall SRs to be complete and correct

S1.1.2

PISC 4.3

G1.1.3

SRs allocated and apportioned to elements of the system for procedures, people (training) and equipment

S1.1.3

PISC 4.4

G1.1.4

Traceability data shows the allocation and apportionment to be complete and correct

S1.1.4

PISC 4.6

G1.1.5

Hazard & Risk analysis carried out to identify all hazards arising from failure of a system element

S1.1.5

PISC 4.5.1

G1.1.6

Derived SRs specified for system elements sufficient to control and/or mitigate all identified hazards

S1.1.6

PISC 4.5.2

G1.1.7

Combined Allocated and Derived SRs meet the RVSM Safety Objectives

S1.1.7

PISC 4.8

EVALUATING SAFETY ARGUMENTS

- Reviews are used to increase the soundness of arguments
- Generally 2-person, e.g.
    - proposer who asserts and defends the safety case
    - Assessor who scrutinises & attacks the arguments to discover vulnerabilities
    - Objective: To form a mutual acceptance of the subjective positions

PROBLEM & MOTIVATION

- Whilst GSN tackles many of the overt drawbacks of plain-text approaches
- & reviews uncover many issues
  - No guarantee of the quality of the post-review arguments
    - Very much dependent upon the experience, expertise & strategic wisdom of the proposer & assessor
- A complimentary approach is to use software to support the construction and evaluation of the safety arguments, e.g.
  - An agent that can assist by detecting (where possible) flaws in the arguments

SOLUTION OVERVIEW

- GSN nodes are black-boxes - they still contain free text
- But if node contents were formalised, or at least written in a restricted language, then GSN nodes could be machine processable
    - *Predicate-logic based approach as the GSN node content language*
    - *Attempt to capture abstract errors as expressions of the language*
    - AIM: To (eventually) automatically identify as many errors as possible (& ideally apply any findings to natural language arguments in other domains)

BUILDING AN ONTOLOGY

- Preliminary study of the *Europe Air Traffic Management (ATM) System* Safety Case used to build ontology of constant, function, and predicate symbols
    - Used to form an initial vocabulary of GSN node expressions
        - *Wan (2015) "Auto-detecting fallacies in system safety arguments" MSc Thesis, University of York*

FALLACIES IN SAFETY ARGUMENTS

- Working within the following framework:
    - Fallacy as a mistake in argument that violates 1or more of following criteria 1. well-formed structure, 2. relevance of premises to conclusion, 3. acceptability of premises, 4. sufficiency of grounds to support conclusions, 5. provision of effective rebuttals to anticipated criticisms
- Within safety domain, Greenwell *et al.* (2005) studied a range of safety cases, and identified 3 categories: relevance, acceptability, & sufficiency
    - We looked at a subset:
        - appeal to improper authority, fallacious use of language, faulty analogy, circular argument, fallacy of composition, confusion of necessary & sufficient conditions
    - Important: Safety cases can be big & complex hence they can lead to errors. (fallacies). Aim is to support human actors in reducing issues (raising quality of cases) by detecting most egregious cases so that the proposer can rectify the situation

APPEAL TO IMPROPER AUTHORITY

- Fallacy that violates the relevance criterion
- Employment of premises that appeal to authorities who aren't relevant, e.g. not an authority in the field
- Transfers one authorities competence in one field into another field in which its competence is not valid

- Use constant symbols standard() & authority()
- For the domain, build a database of authorities and fields of expertise.
  - e.g. Within safety arguments authorities validly cited tend to be individuals, committees, standards documents, best practices
- Determine when a safety argument cites a standard or authority where the ascribed competence is incorrect

USE OF LANGUAGE

- Lack of clarity & consistency
  - e.g. inconsistent use of words referring to desirable system properties, e.g. expressions that describe safety, reliability, or dependability
- In safety context, police use of common words & phrases, e.g.
  - isSafe(), isReliable(), is Dependent() have distinct meanings in context so their use should be tracked
  - *NB. Not a solution to all problems of this nature, but a start...*

FAULTY ANALOGY

- Assumption that if two things are alike in one or more respects then they are necessarily alike in some other respect
    - In safety cases this assumption can be disastrous
        - e.g. Ariane 5 explosion due to faulty analogy between rocket safety cases
- e.g. if isAlike(system(x), system(y)) & isSafeSystem(y) then isSafe(isSafeSystem(x))
- This assertion, without justification, could be bad

CIRCULAR ARGUMENT

- Essentially, using the concluding portion of the argument to support itself, e.g. using (part of) the conclusion as evidence to support itself
- *Not withstanding earlier work (particularly, Mackenzie(1979), Woods & Walton (1978))*
- Tracing the use & reuse of predicate clauses through longer arguments, e.g.

isSafe(system(x)) => meetStandard(processOf(System(x)),Standard(y))

meetStandard(processOf(System(x)), Standard(y)) => isSafe(System(x))

FALLACY OF COMPOSITION

- Assumption that if every part is true then the whole is true
- If supporting sub-systems A, B, & C are safe the the system is safe,
  - but what about interactions between those sub-systems?
- Can detect patterns of argument associated with system decomposition (previous work by Yuan & Kelly) which should prompt a check for component interaction arguments, if any of those arguments are missing then the author should address that, e.g. provide a component-interaction case or modify the composition argument

CONFUSION OF NECESSARY & SUFFICIENT CONDITIONS

- A necessary condition must be present for an event to occur
- Sufficient conditions will trigger the event
- In a safety case, the argument "hazards have been mitigated" is common with evidence given to show that this is the case
    - However for this to be sufficient *all* hazards must be identified
    - We can identify arguments of type
        - isSafe(system(x), condition1) => isSafe(system(x))
        - & flag this to the author as condition1 might be insufficient to conclude that the entire system is safe

FUTURE WORK

- Enrich the ontology vocabulary
- Apply to different domains (within the assurance context, e.g. security/privacy assurance)
- Incorporation within existing safety-argument capture tools
- Wider application

CONCLUSIONS

- Safety Arguments
- GSN
- Approaches to minimising errors (using review processes)
- Automatic fallacy detection in safety arguments

THANKS FOR YOUR ATTENTION

QUESTIONS?