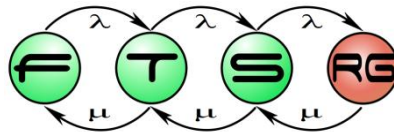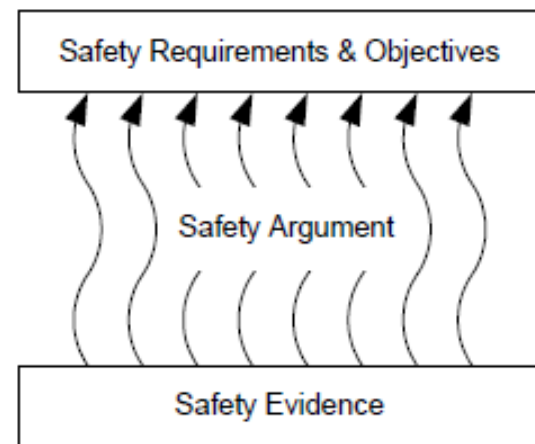# The Safety Case

## Structure of Safety Cases
Safety Argument Notation

The safety case

- Definition (core): The documented demonstration that the product complies with the safety requirements
- Role:
  - A safety case should communicate a comprehensive and defensible argument that a system is acceptably safe to operate in a particular context
  - Condition for safety acceptance and approval
- To be prepared by: Developers and/or operators
- To be accepted by: Safety authority and/or customer
- Principal elements:
  - Safety requirements (goals, objectives)
  - Arguments (relations)
  - Evidences
    - Analysis results (e.g., FTA, FMEA)
    - Formal verification
    - Test results
    - …

Safety Requirements & Objectives

Safety Argument

Safety Evidence

Standard structure of a safety case

- Conditions for safety acceptance
  - Evidence of quality management
  - Evidence of safety management
  - Evidence of technical safety
- Structured presentation of evidence and arguments
- Example: EN50129 (railway)
  - Part 1: Definition of the system
  - Part 2: Quality management report
  - Part 3: Safety management report
  - Part 4: Technical safety report
 - Part 5: Related safety cases
  - Part 6: Conclusion

Part 6: Conclusion
Part 5: Related Safety Cases
Part 4: Technical Safety Report
Part 3: Safety Management Report
Part 2: Quality Management Report
Part 1: Definition of System
SAFETY CASE
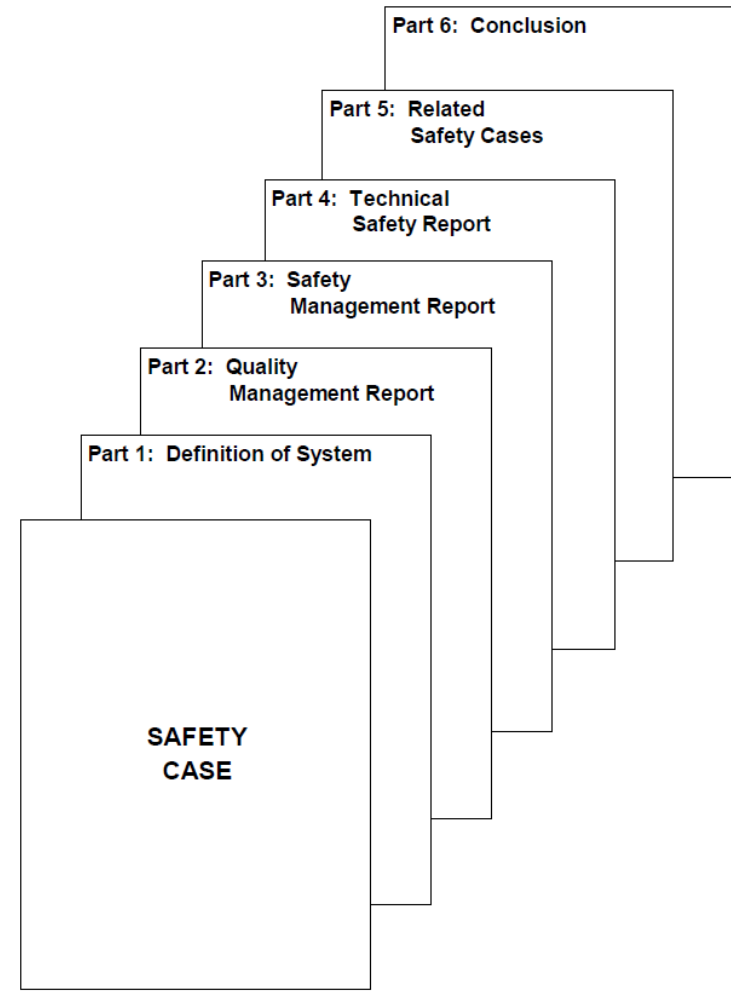
Quality related parts of the safety case

Part 2: Quality management report

- o Minimize the incidence of human errors at each stages in the lifecycle: Reduce the risk of systematic faults

Part 3: Safety management report

1. Safety lifecycle: From requirements to validation
2. Safety organization: Roles and competence
3. Safety plan: Activities and approval milestones + review
4. Hazard log: Hazards + risks + risk control
5. Safety requirements
6. System design
7. Safety reviews
8. Safety verification and validation
9. Safety justification
10. System handover (to authority)
11. Operation and maintenance
12. Decommission and disposal
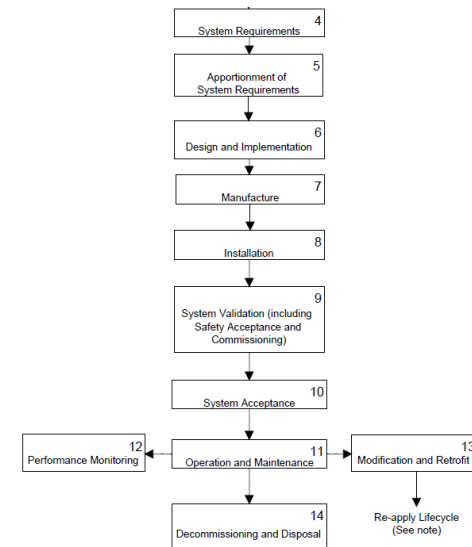
Technical parts of the safety case

Part 4: Technical safety report

1. Introduction:
   o Summary of technical principles and standards

2. Assurance of correct functional operation
   o Architecture, interfaces, fulfillment of requirements, assurance of correct hardware and software behavior

3. Effects of faults
   o Random hardware faults: Quantified safety target
      o Detection, actions after detection, effects, independence, multiple faults
   o Systematic faults: Risk reduction

4. Operation with external influences
   o Demonstration of operability and safety

5. Safety-related application conditions
   o Rules, conditions, constraints

6. Safety qualification tests
   o Evidence to demonstrate completion

Safety argumentation

Communicating safety arguments

- Typical: Free text
  - Structured form (items, enumerations, references)
  - Complex arguments are difficult to describe
    - Review, management, tracking, coordination is difficult

> 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.

?

- Graphical notation: Goal Structuring Notation
  - Elements of safety arguments
  - Relationships between the elements

Elements

- **Goal**: Objective, claim about the system
  - Compliance with requirements
  - Sufficient mitigation / avoidance of hazards
  - Without evidence it is unfounded!
- **Strategy**: Decomposition method
  - Derivation of sub-goals
- **Evidence** (solution)
  - Results of observation, analysis, test, simulation, …
  - Fundamental information from which safety can be inferred
- **Context**
  - Context of demonstrating safety
- **Assumption** or **Justification**
  - Limits, conditions etc.
- **Undeveloped goal**
  - Further development is necessary

System can tolerate single component failures

Argument by elimination of all hazards

Fault Tree for Hazard H1

All Identified System Hazards
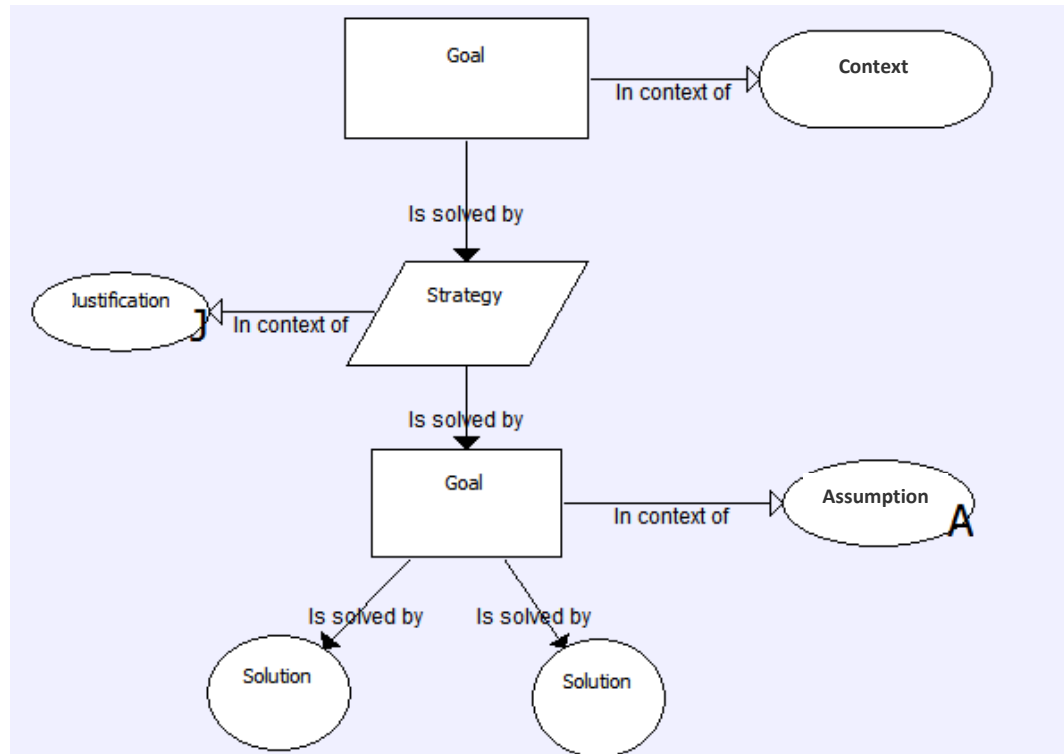
A1
Sub-systems are independent
A

Relations

- **"Is solved by"**
  - Applied between goals, strategies, evidences
- **"In context of"**
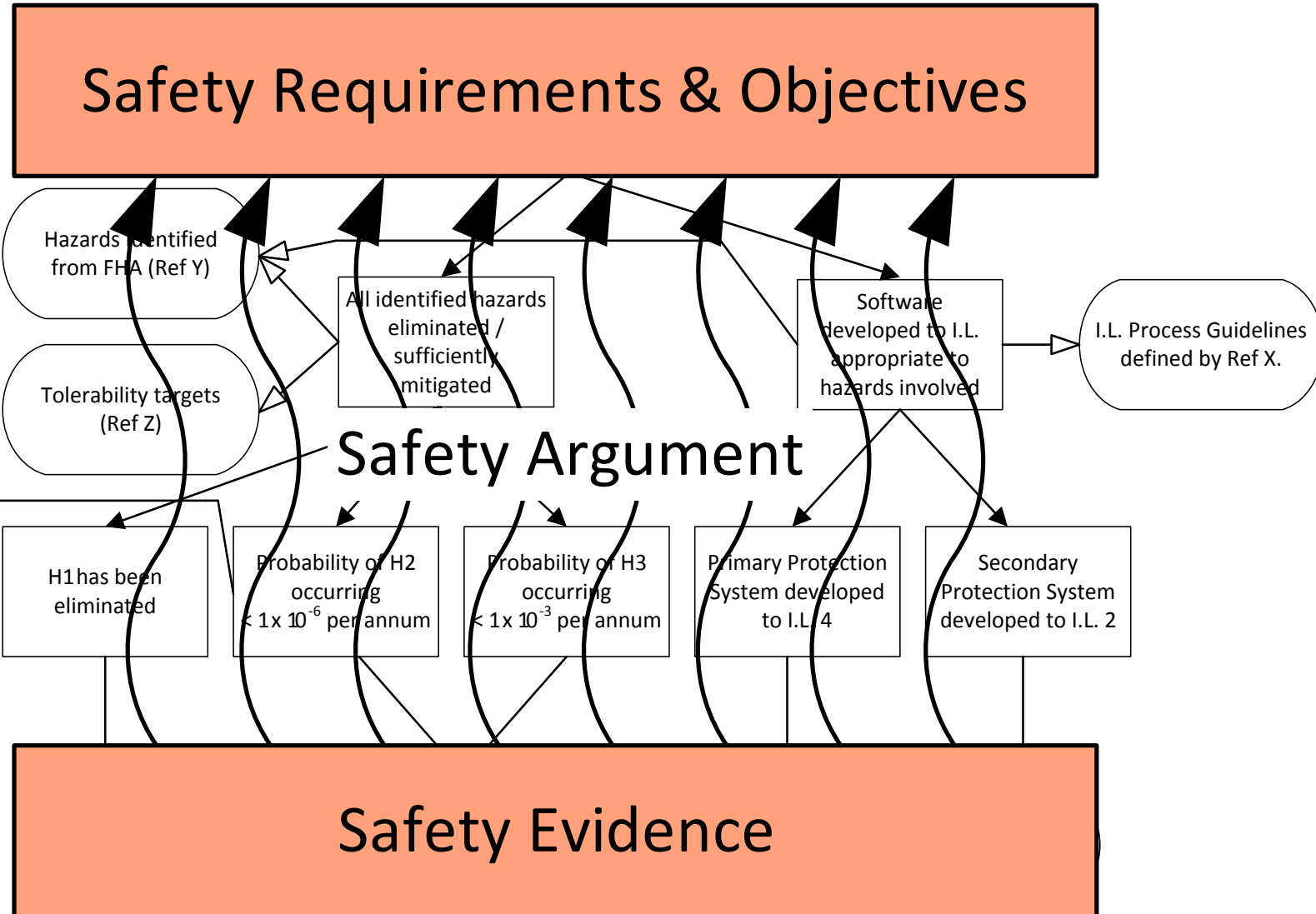 - Applied between contexts / assumptions / justifications and other elements
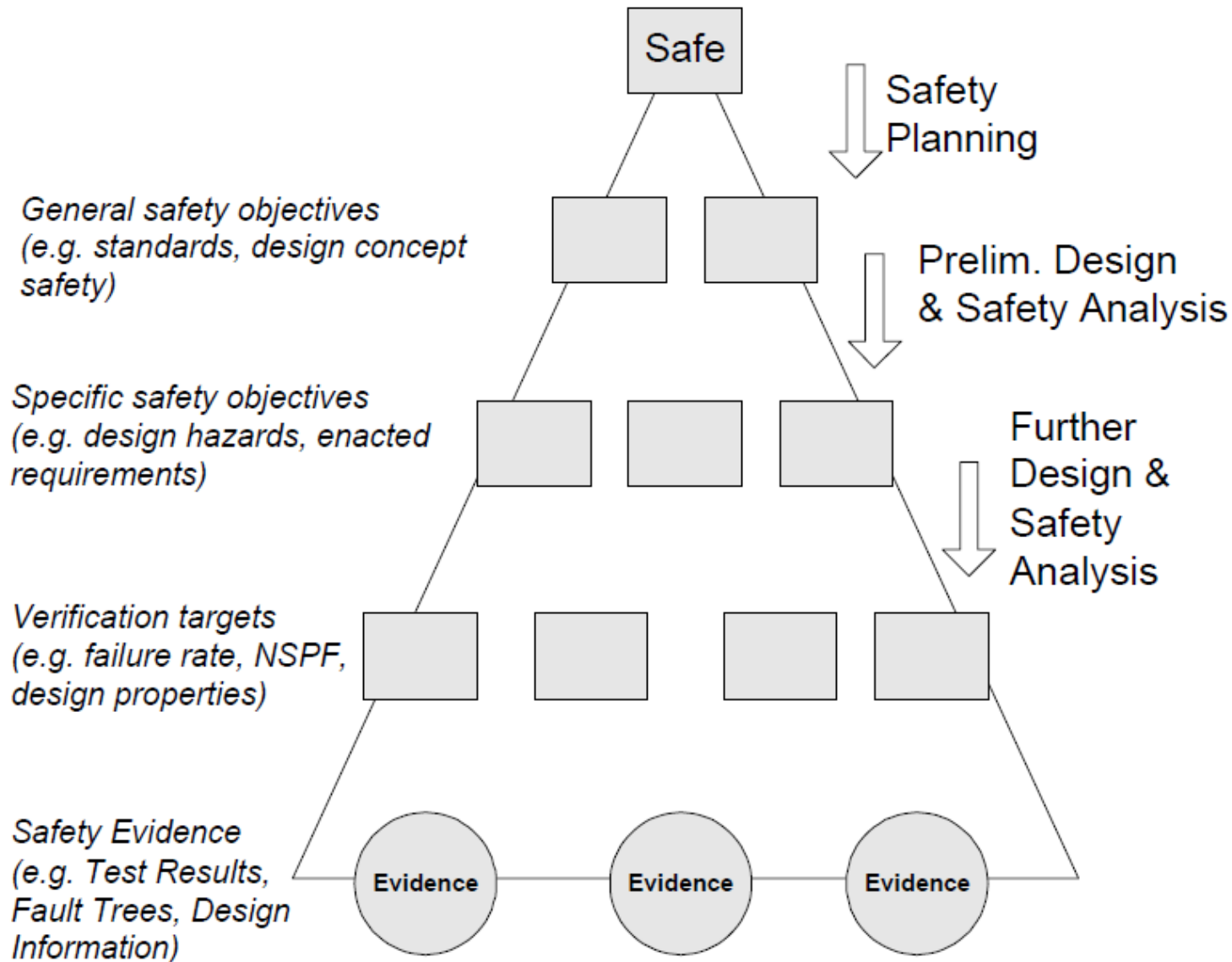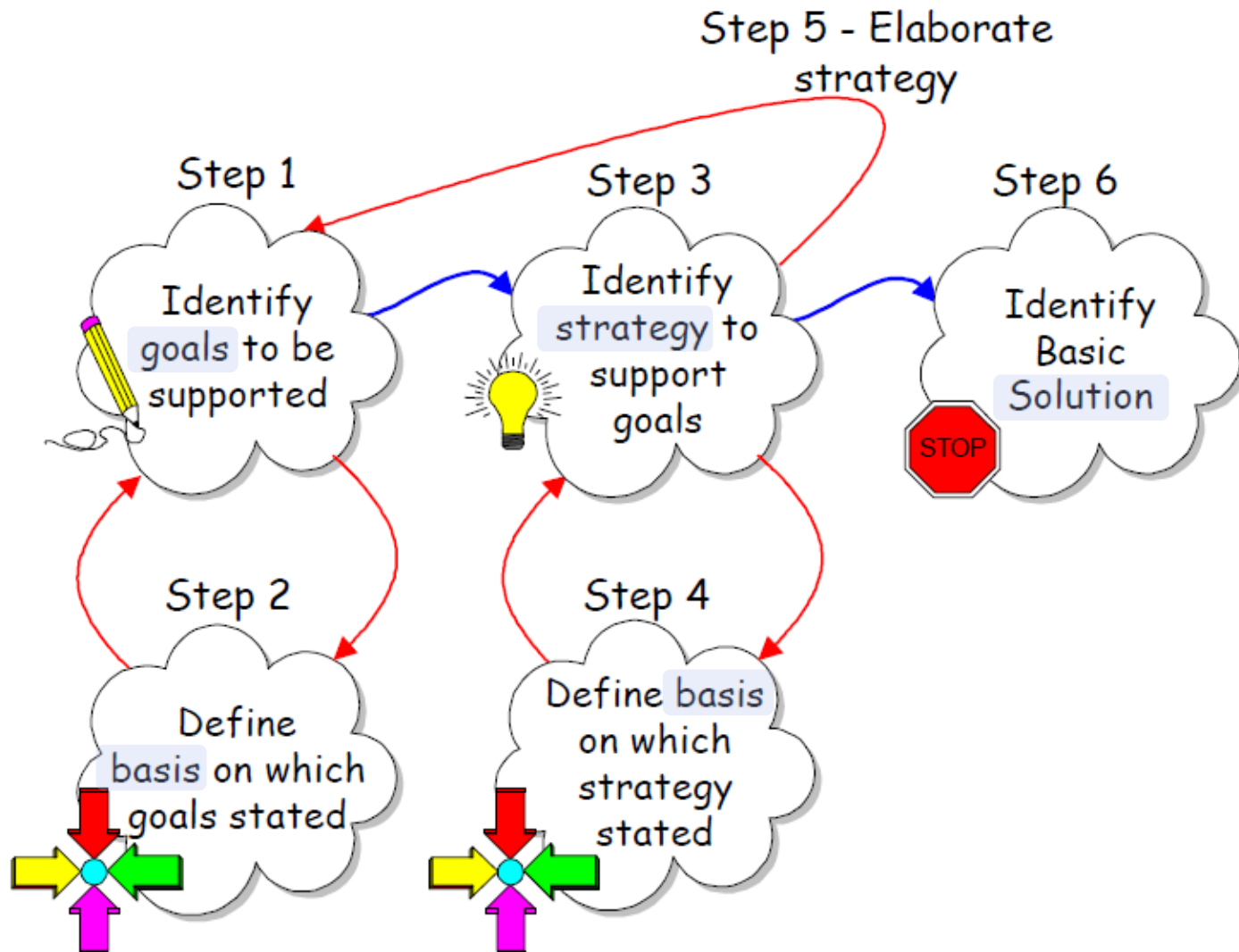
SolvedBy

InContextOf

Safety Requirements & Objectives

Hazards identified from FHA (Ref Y)

All identified hazards eliminated / sufficiently mitigated

Software developed to I.L. appropriate to hazards involved

I.L. Process Guidelines defined by Ref X.

Tolerability targets (Ref Z)

Safety Argument

1×10^{-6} p.a. limit for Catastrophic Hazards

H1 has been eliminated

Probability of H2 occurring $< 1 \times 10^{-6}$ per annum

Probability of H3 occurring $< 1 \times 10^{-3}$ per annum

Primary Protection System developed to I.L. 4

Secondary Protection System developed to I.L. 2

Safety Evidence

Source: T. Kelly

General safety objectives
(e.g. standards, design concept
safety)

Specific safety objectives
(e.g. design hazards, enacted
requirements)

Verification targets
(e.g. failure rate, NSPF,
design properties)

Safety Evidence
(e.g. Test Results,
Fault Trees, Design
Information)

Safe

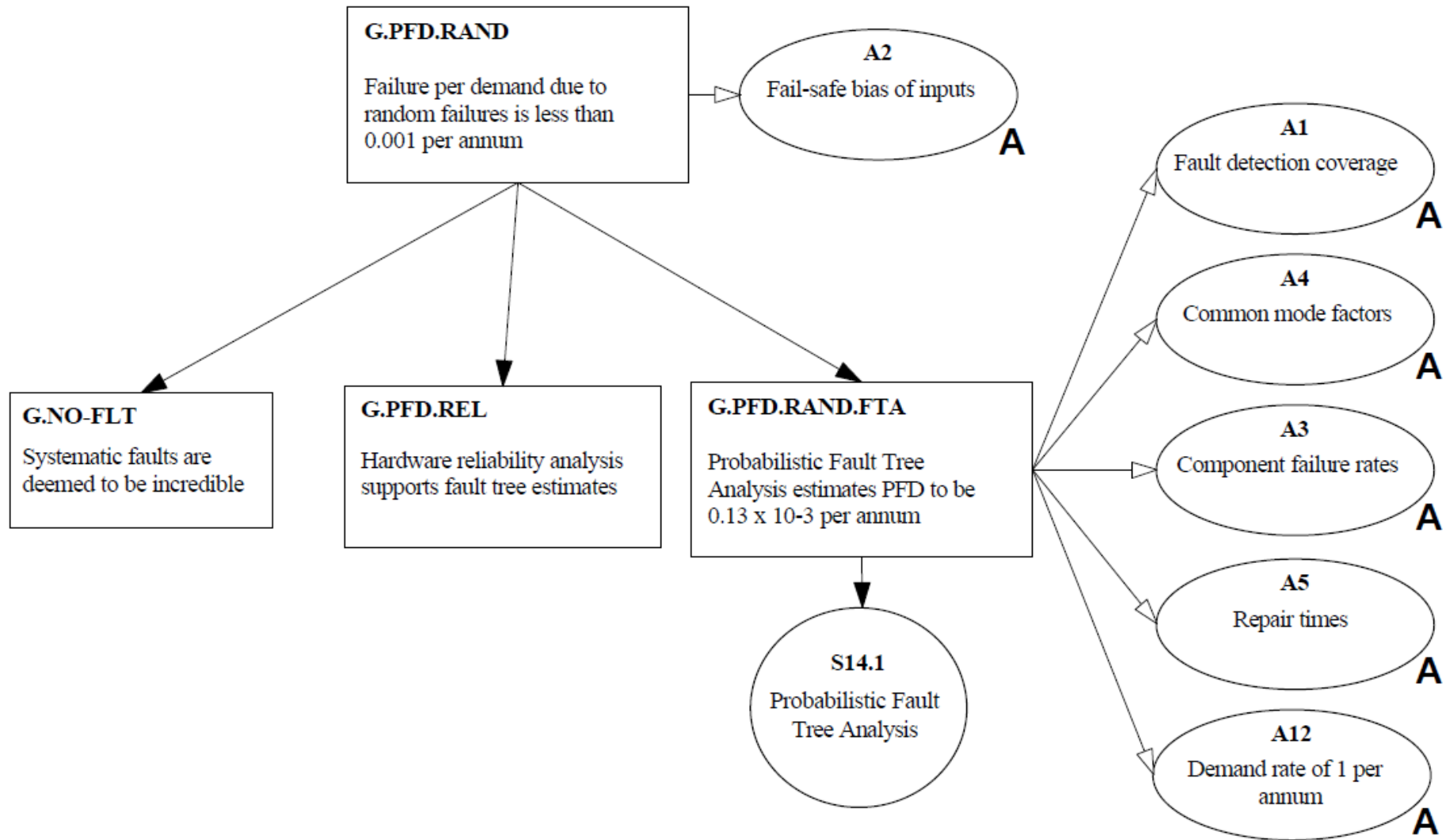Safety
Planning

Prelim. Design
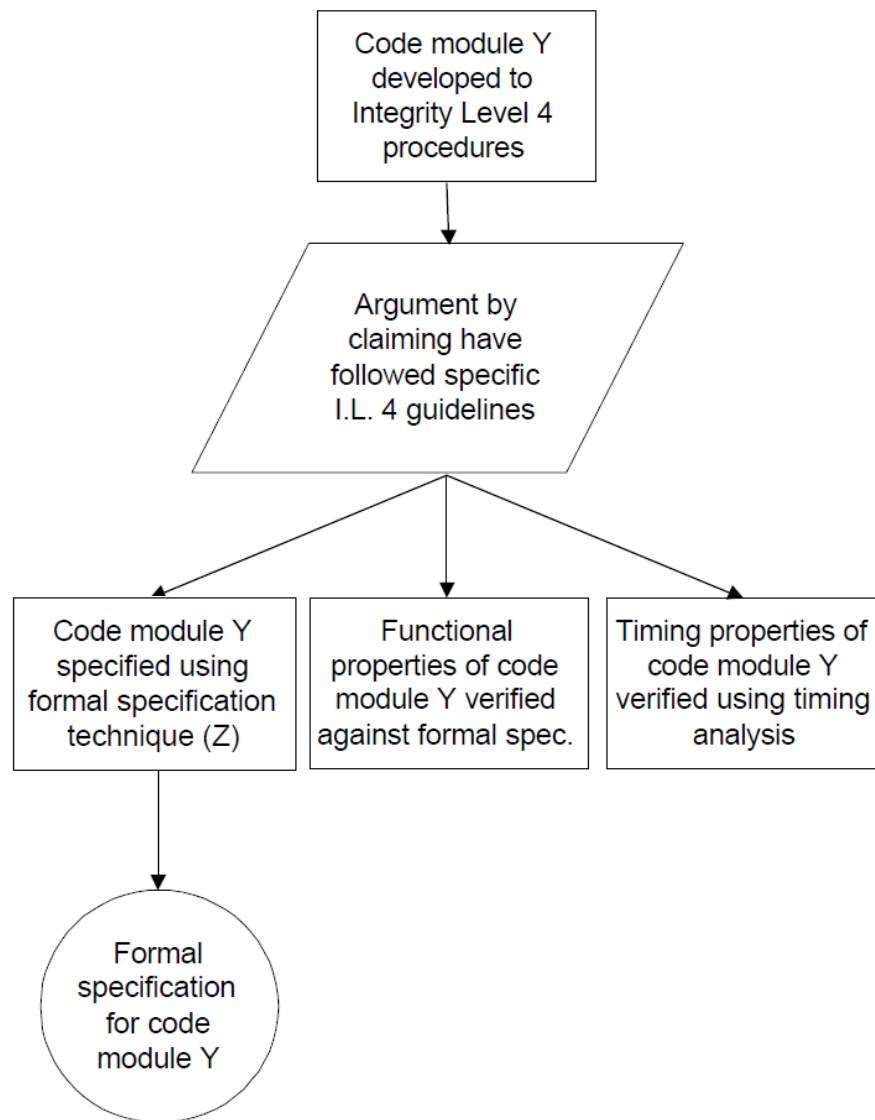& Safety Analysis

Further
Design &
Safety
Analysis
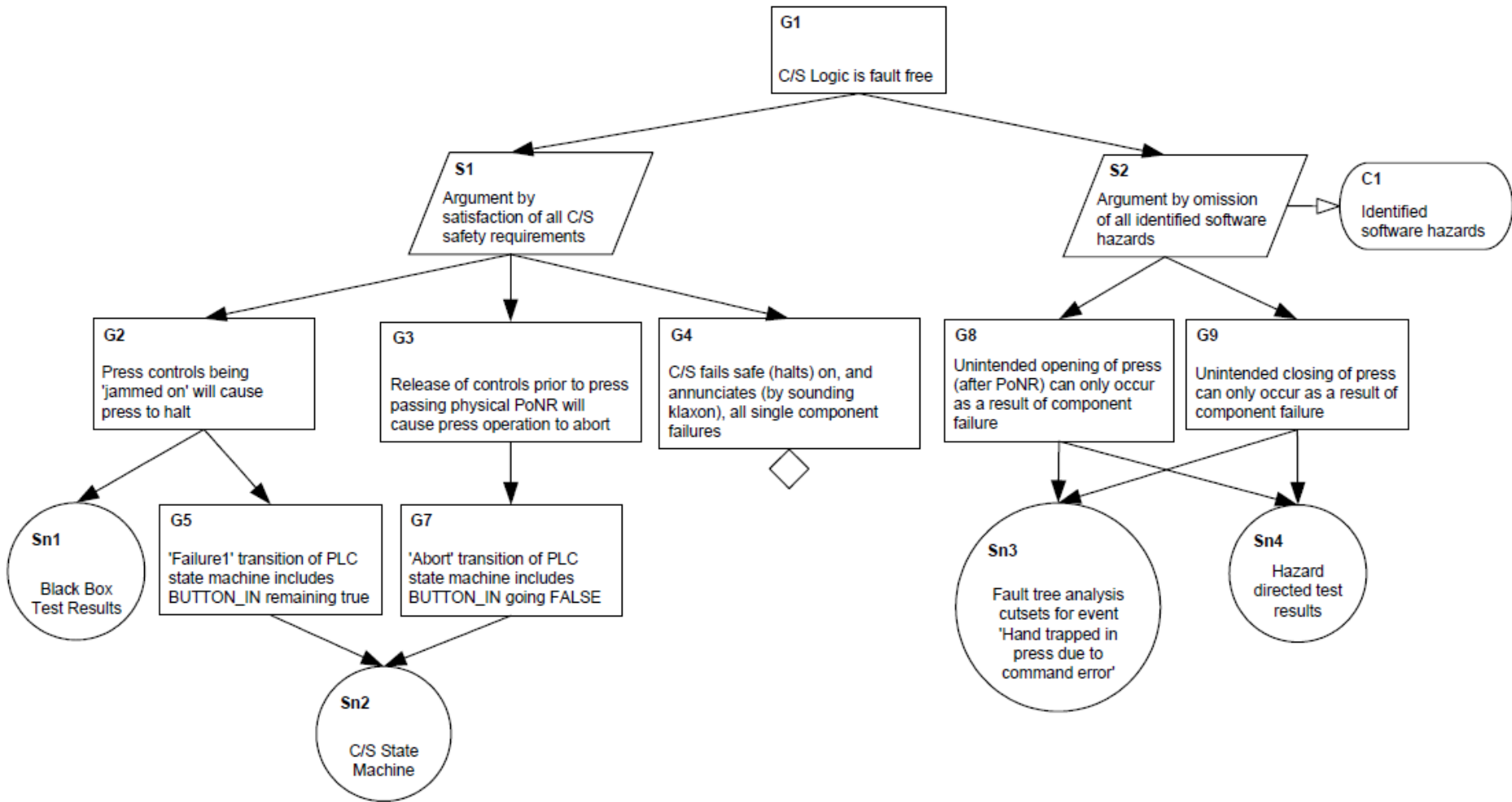
Steps of safety case construction

Source: T. Kelly

Safety arguments for software

- Software SIL:
  Required techniques and measures
  form arguments and evidences

- Example: Guidelines followed for SIL4
  - Formal specification
  - Formal verification of functionality
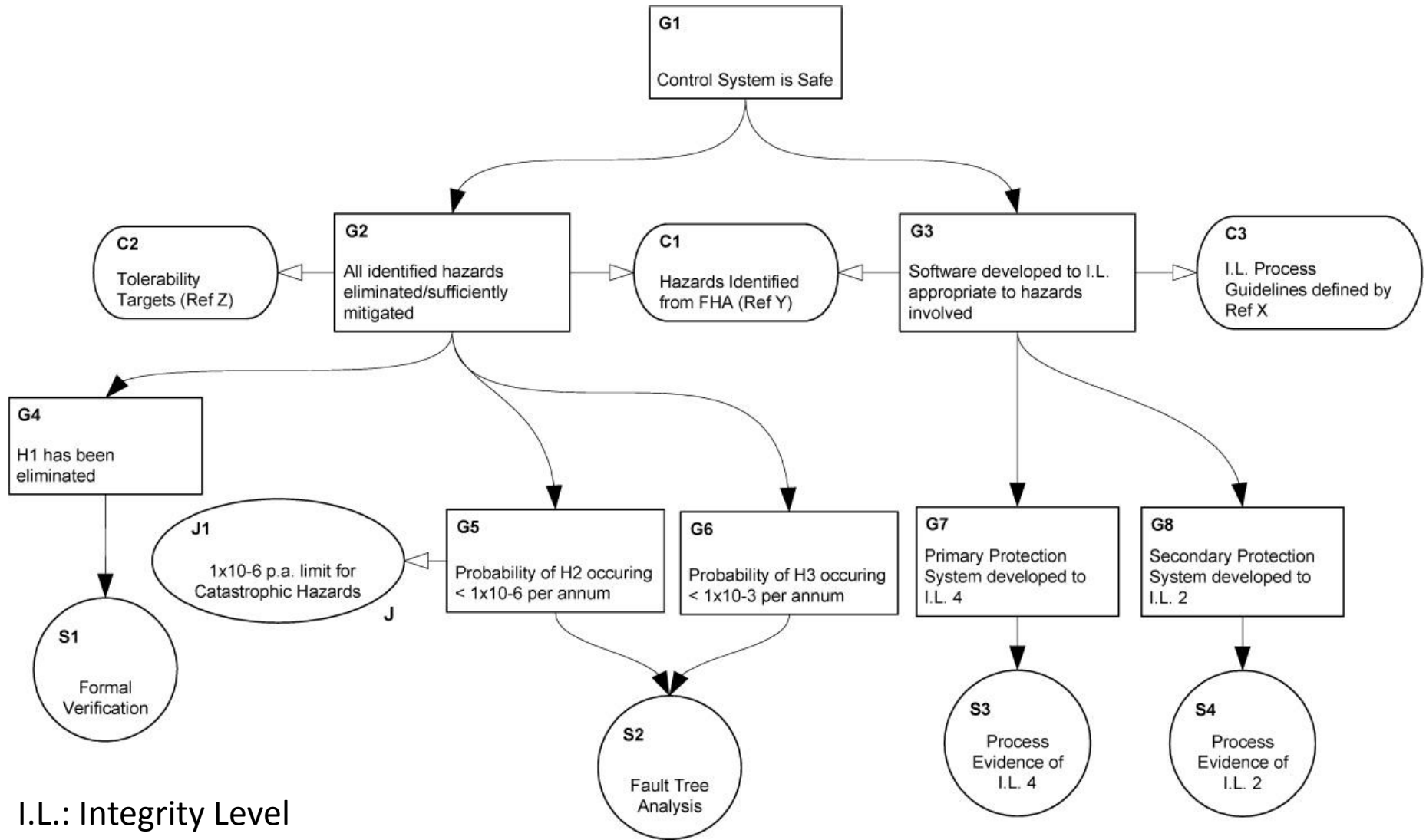 - Formal verification of timing

An example goal structure

Evidences: Test results, state machine analysis, fault tree analysis, directed testing

Generic goal structure

I.L.: Integrity Level
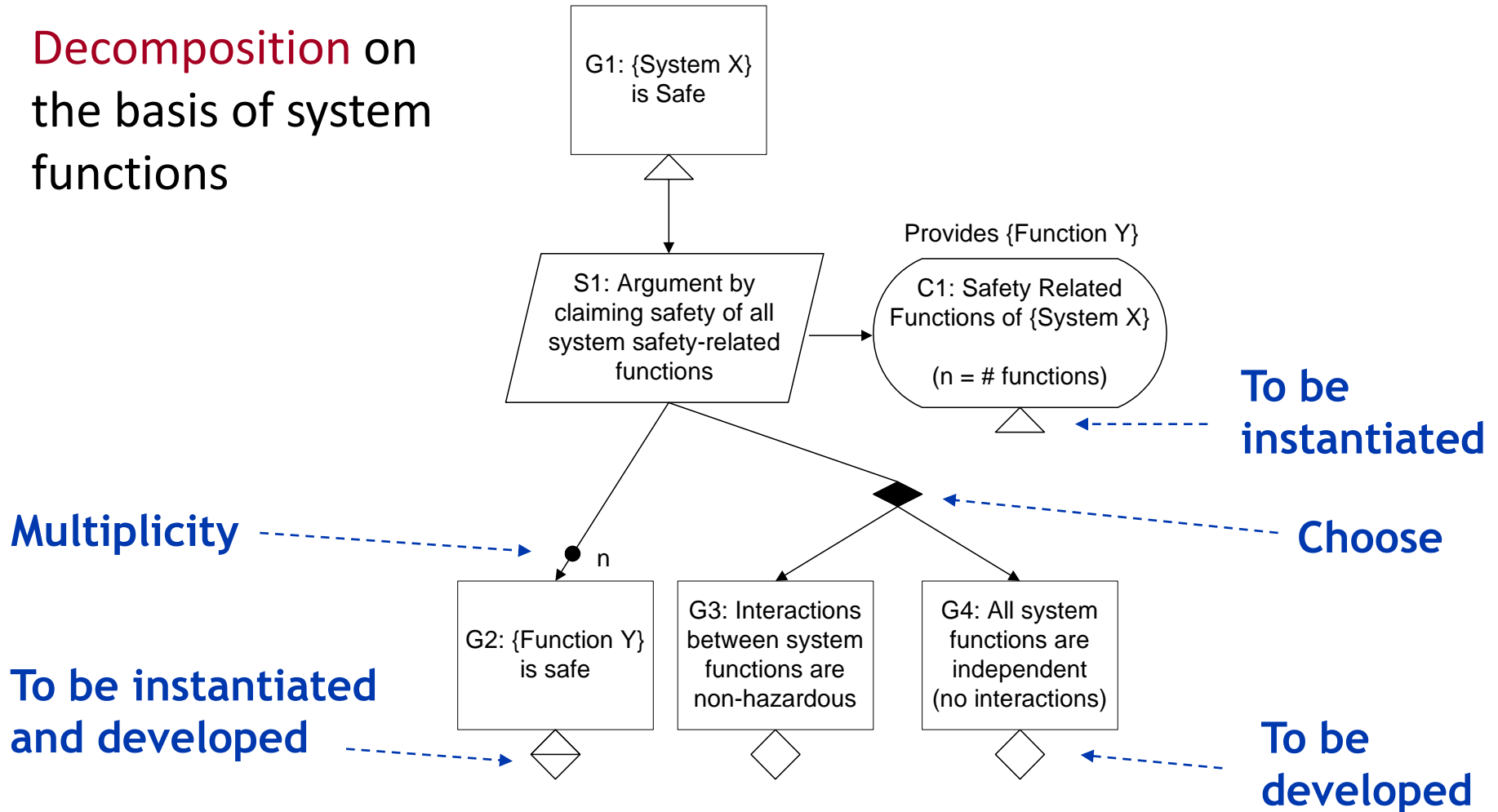FHA: Functional Hazard Assessment

Safety case patterns

- Combines argumentation and patterning
  - Supports the re-use of successful argument approaches (best practice)
  - Focus on semantics rather than the syntax of the safety case
- GSN extensions to support capturing patterns
  - Multiplicity
  - Instantiation
  - Develop
  - Instantiation and develop
  - Choice

Decomposition on the basis of system functions

G1: {System X} is Safe

S1: Argument by claiming safety of all system safety-related functions

Provides {Function Y}

C1: Safety Related Functions of {System X}

(n = # functions)

To be instantiated

Multiplicity

Choose

n

G2: {Function Y} is safe

G3: Interactions between system functions are non-hazardous

G4: All system functions are independent (no interactions)

To be instantiated and developed

To be developed

How a fault tree analysis can be used as evidence

Fault Tree Evidence

G2 {Probability of X occuring is ...} (Quantitative)

G1 {Causes of X are ...} (Qualitative)

G3 No single point of failure can lead to {X}

A1 Basic Failure Events used in Sn1 are independent **A**

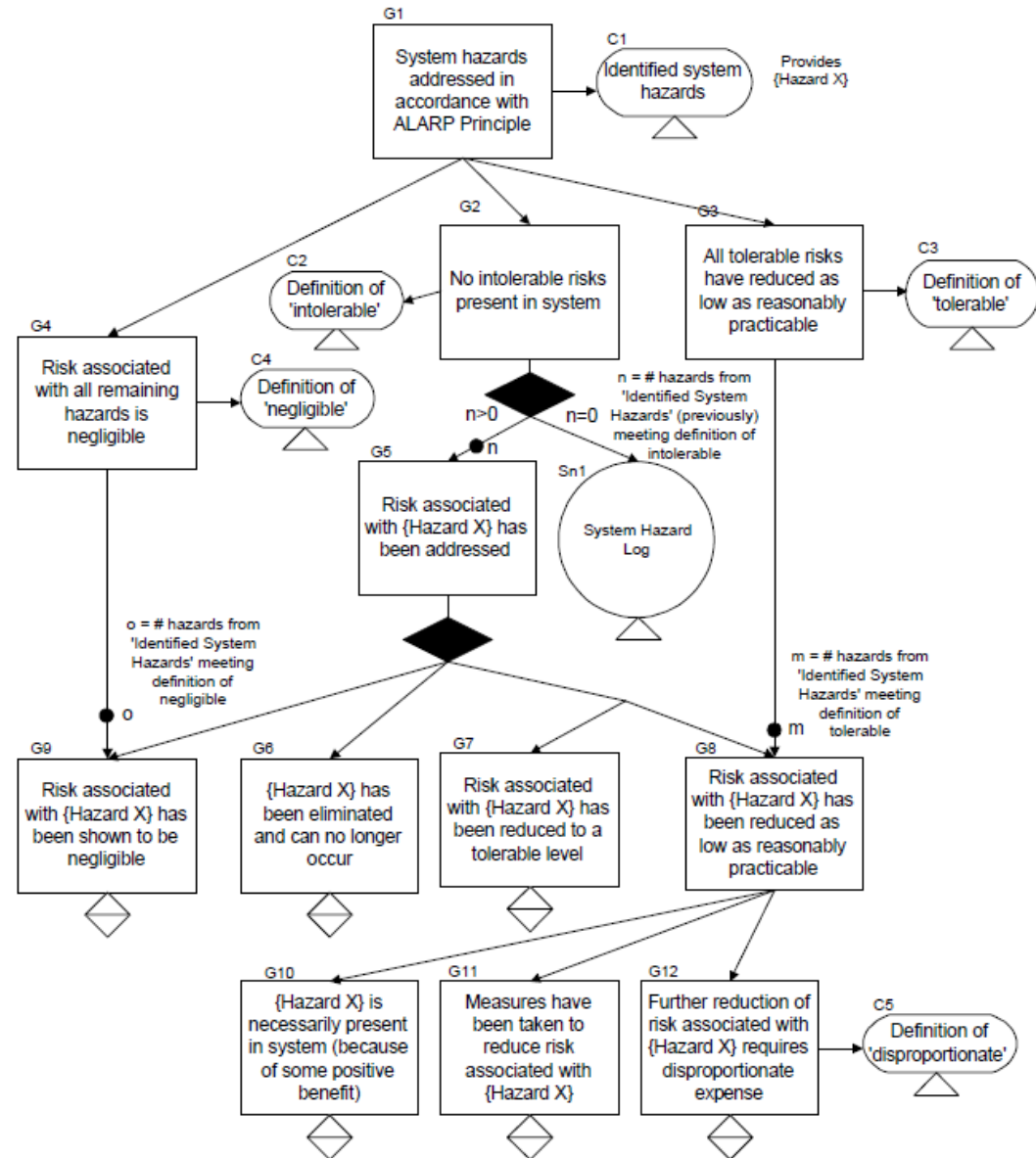**Sn1** Fault Tree for Condition {X}

J1 Sn1 is an accurate representation of the causes of X **J**
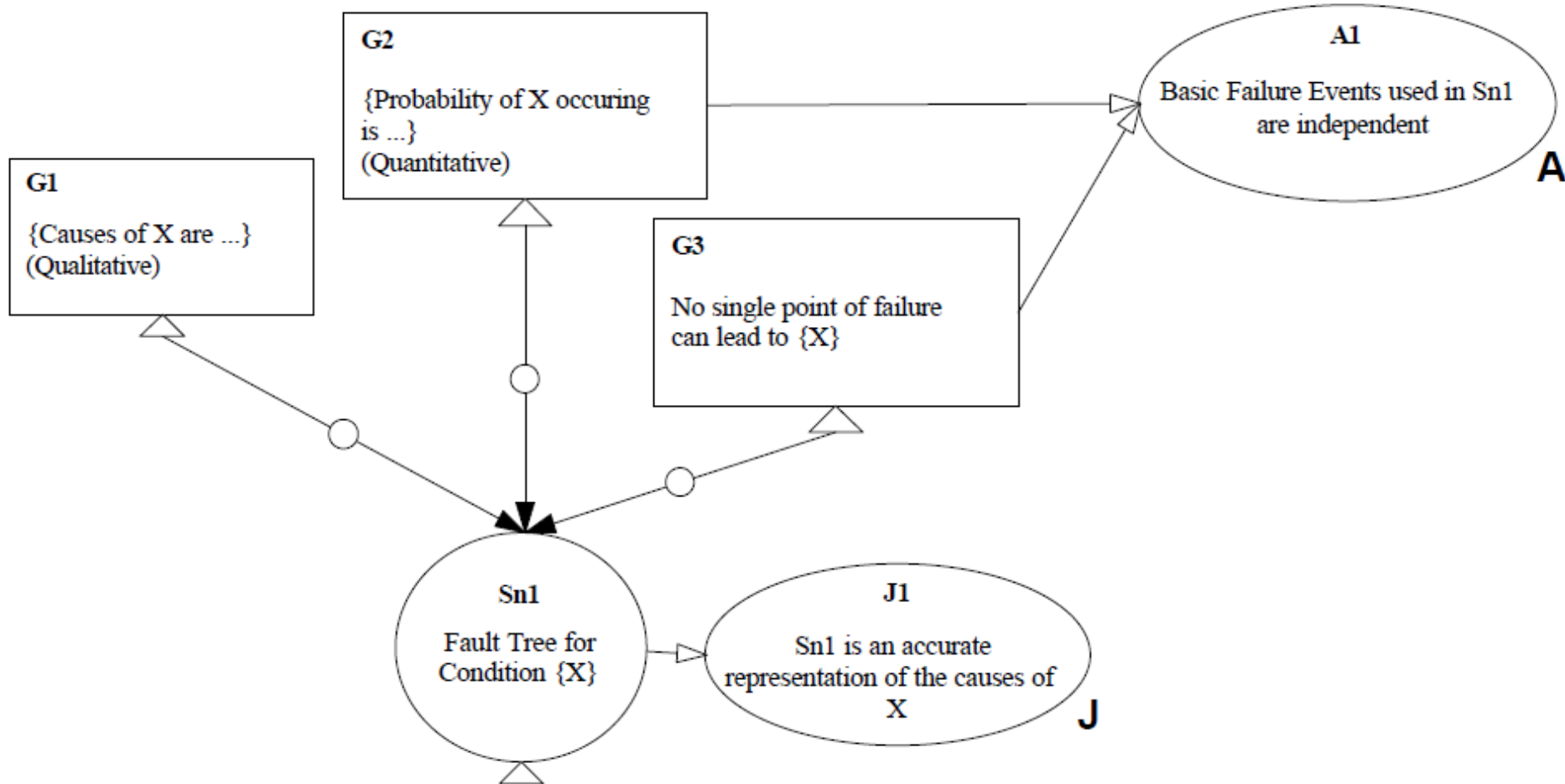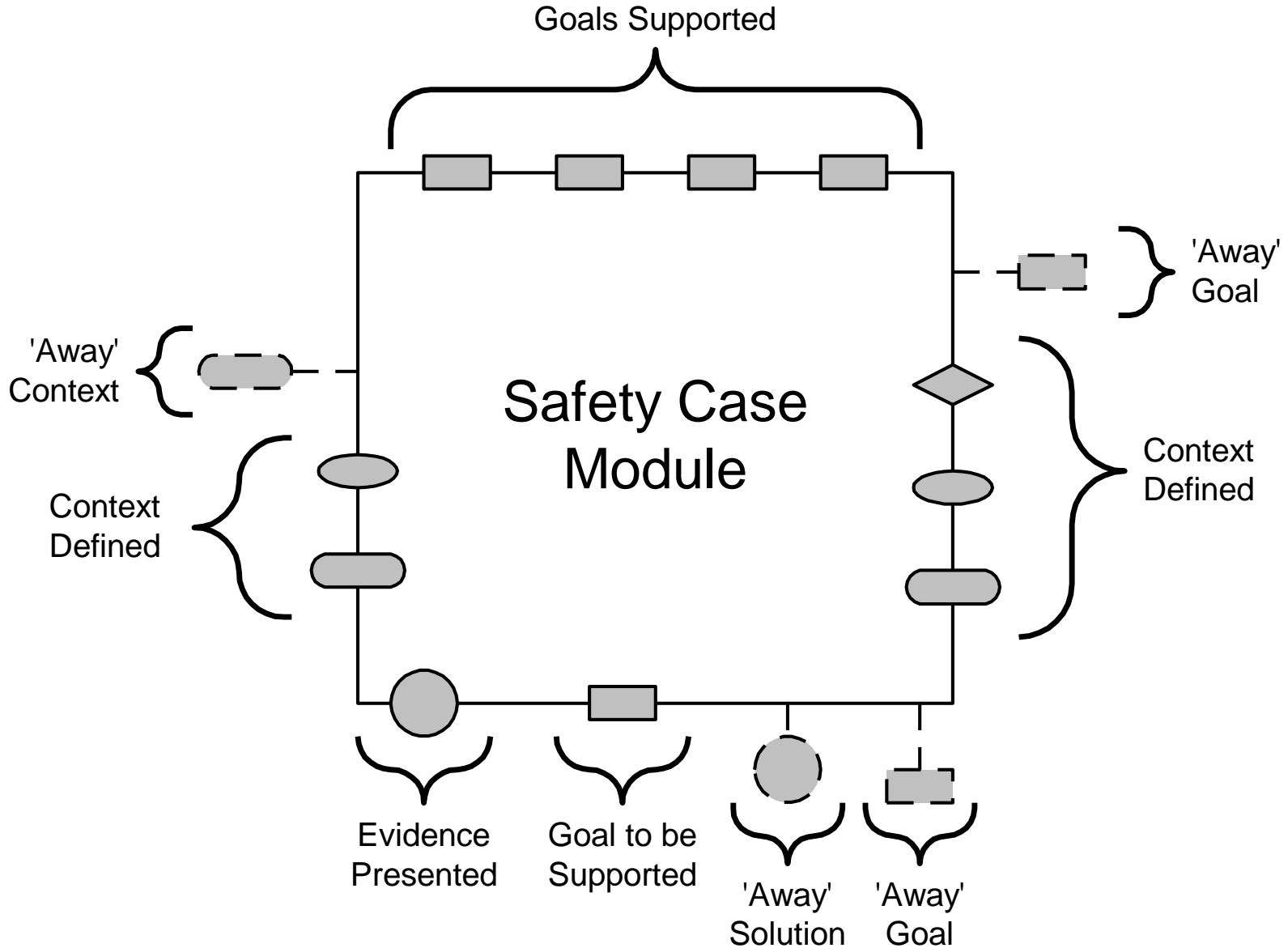
ALARP: As Low As Reasonably Practicable

- No intolerable risk
- All tolerable risks have reduced as low as reasonably practicable
- All remaining hazards have negligible risks

Modular safety cases

- Goal: Modular, compositional construction of safety cases (corresponding to system structure)

- Partitioning of modules
  - Vertical (hierarchical) partitioning
    - Claims of one argument are objectives of another
    - E.g., case split of system and software safety case
  - Horizontal partitioning
    - One argument providing the assumed context of another
    - E.g., "All system hazards have been identified" provides assumed context of an argument that "All identified system hazards have been sufficiently mitigated"

- Module interfaces
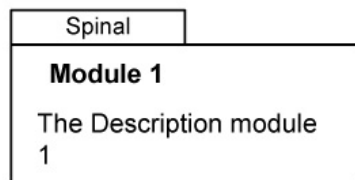 - Dependency of objectives, evidence, context of other modules
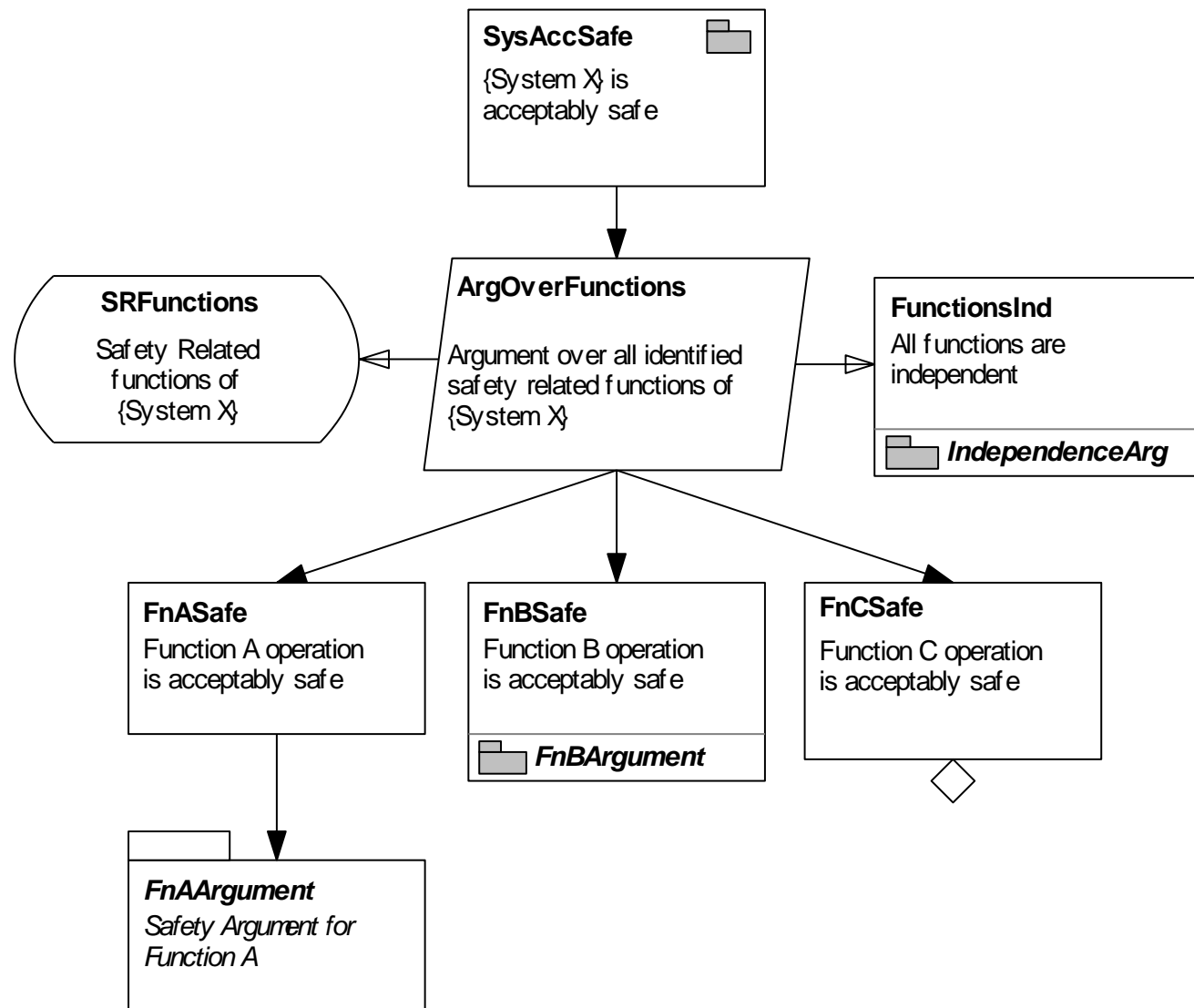
Example of a modular safety case

Elements:
- **Safety case modules**

Spinal

Module 1

The Description module 1

- **"Away" goals**

G0_Module 1

Software Condition X cannot occur

Module 1

SysAccSafe

{System X} is acceptably safe

SRFunctions

Safety Related functions of {System X}

ArgOverFunctions

Argument over all identified safety related functions of {System X}

FunctionsInd

All functions are independent

IndependenceArg

FnASafe

Function A operation is acceptably safe

FnBSafe

Function B operation is acceptably safe

FnBArgument

FnCSafe

Function C operation is acceptably safe

FnAArgument

Safety Argument for Function A

Example: Change of the
context of hazard logs
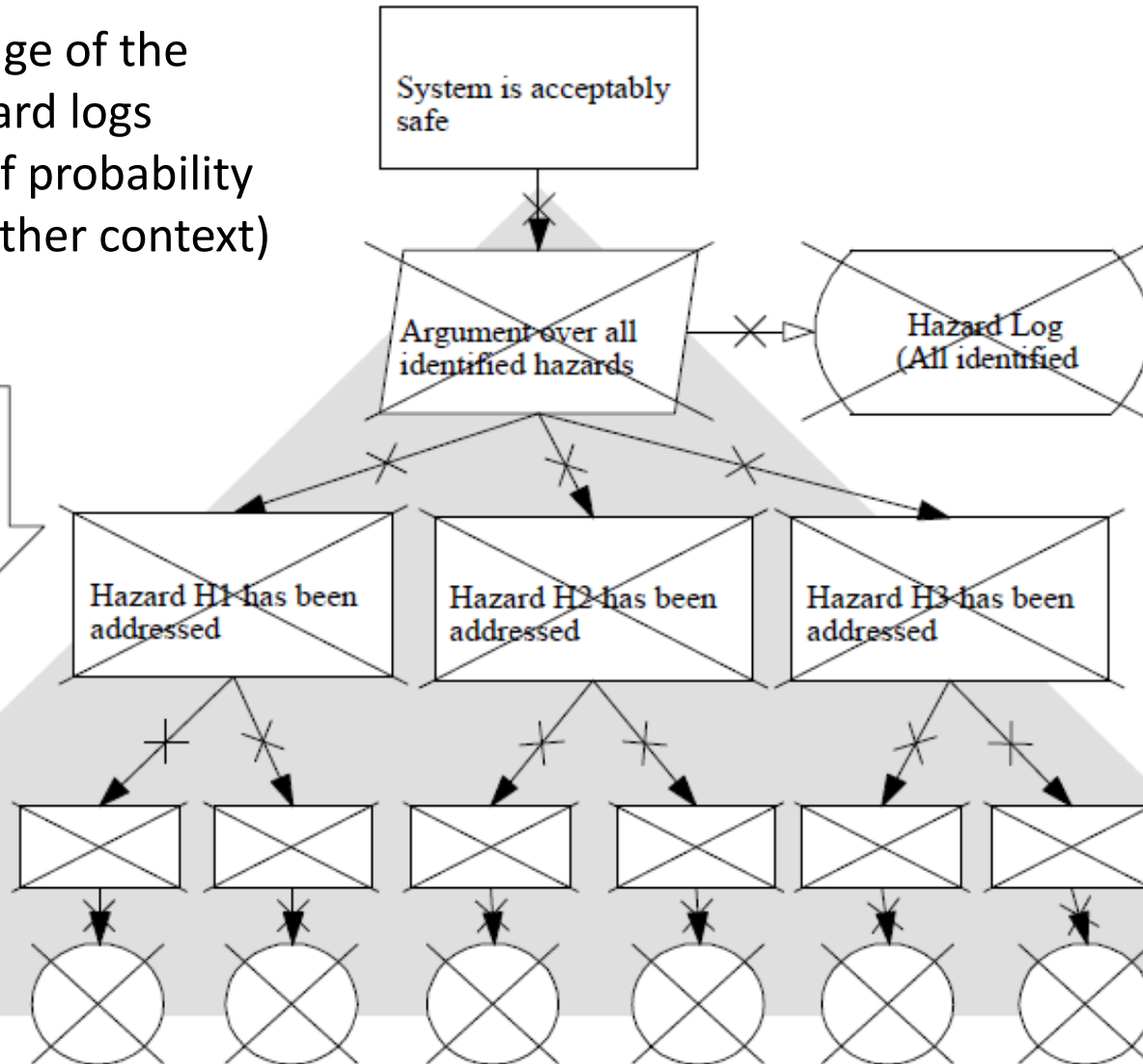(e.g., change of probability
of hazards in other context)

Advantages and disadvantages of GSN

- Advantages:
  - Simple elements
    - Captures the elements most important to safety arguments
  - Structured hierarchical breakdown
    - Method guidance exists
  - Semantics well defined and understood (first order logic)
  - Can be used at various stages of argument development
  - Increasingly being adopted by companies

- Disadvantages:
  - Easy to read, harder to write ☺
  - Doesn't stop you writing bad arguments ☹

- **ASCAD: Adelard Safety Claims Arguments Data**
  - ○ Claim:
    Assertion
    to be proven
  - ○ Argument:
    How evidence
    supports
    claim
  - ○ Evidence:
    Required
    observation,
    analysis, test,
    …
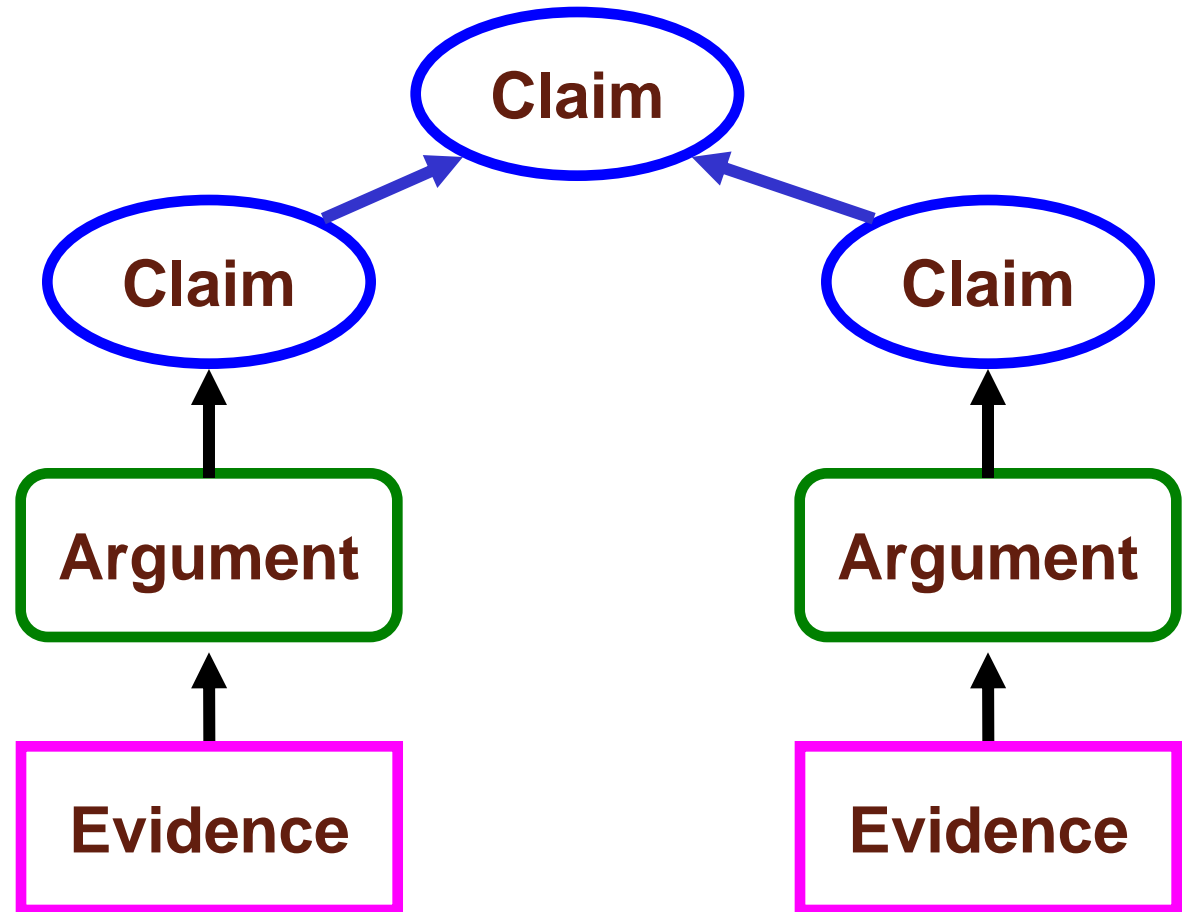
Generalization

- **Assurance cases**
  - Safety cases
  - Security cases
  - Dependability cases

- Definition
  - A documented body of evidence that provides a convincing and valid argument that a specified set of critical claims regarding a system's properties are adequately justified for a given application in a given environment

- Examples of using assurance cases
  - Security-critical applications: Based on Common Criteria
  - Medical devices: Based on ISO 14971

Supporting tools

- **Adelard Safety Case Editor (ASCE)**
  - Adelard, www.adelard.co.uk
  - Supports both GSN and ASCAD
- **E-Safety Case**
  - Praxis HIS, www.esafetycase.com
- **GSN CaseMaker**
  - ERA Technology, www.era.co.uk
- **ISCADE (Integrated Safety Case Development Environment)**
  - RCM2, www.iscade.co.uk
- **ISIS**
  - High Integrity Solutions, www.highintegritysolutions.com
- **Freeware Visio Add-on**
  - University of York, http://www.cs.york.ac.uk/~tpk/gsn/gsnaddoninstaller.zip

Summary

- **Structure of safety cases**
  - Evidence of quality management
  - Evidence of safety management
  - Evidence of technical safety

- **Safety argumentation –**
  presented using the Goal Structuring Notation
  - Elements: Evidence, Strategy, Goal, Context
  - Patterns
  - Modular safety arguments
  - Maintenance of safety arguments

- **Generalization: Assurance cases**