What does it mean to have a dynamic safety case?

Tim Kelly

University of York

'Living' Safety Case

- A safety case is the argument and evidence supporting the claims about the safety of the system in operation in a defined context
- You can ask "What is the safety case"? at any time
- Safety Case *Reports* are simply the 'snapshots' of the status of the safety case at a given point in time
  - Current status of the arguments
  - Current status of the evidence

Maintenance

- Typically safety case reports prepared for an acceptance milestone before operation permitted
  - Necessarily *prediction* (modelling, estimation), therefore challenge should be expected
- Many elements can be challenged during operation:
  - System - e.g. configurations
  - Evidence - e.g. failure rates not as predicted
  - Assumptions - e.g. operator behaviour
  - Requirements - e.g. tougher regulations brought in
- Need:
  - a) to monitor such things
  - b) to assess the continuing 'truth' of the safety case in the light of these challenges

Safety Case 'with variables'

- Some of the challenges are predictable
- Can leave placeholders (and criteria) within the safety case and check at run-time
  - e.g. "Calculated failure rate is *X*"
- Presents opportunities for dynamic evidence generation, assurance case generation (really instantiation of a well-known - patterned - structure) and checking
  - fits well with our existing work on Model-Based Assurance Cases

Can Modular Safety Cases help?

- Modular safety cases allow the packaging of a monolithic safety case into modules of argument and evidence with well-defined interfaces
  - Safety Case architecture can correspond with System architecture
- Originally intended to cope with relatively 'slow-time' change - e.g. to system configuration
  - During system lifetime, but off-line checking of satisfaction of necessary dependencies and guarantees
- No reason why couldn't be run-time checked
  - However, biggest problem is in the validation of the necessary properties for system safety (e.g. over all allowed configs.)

Back to Patterns

- My original notion of safety case patterns has both:
  - Entity abstraction (placeholder, types etc.)
  - Structural abstraction (e.g. "n arguments of the form are required" or "an argument of the form X *or* Y is required)
- Safety Case pattern (with choices and multiplicity) can be considered to be a little more like a program

Summary

- *All* safety cases should be dynamic!
- Various interpretations:
  - Maintenance - challenge & response
  - Planned abstraction and run-time criteria
  - 'Plug and Play' Modular Safety Cases
  - Safety Case Pattern as more complicated run-time logic to be checked
- To some extent all have problems of prediction and validation
- What are we ready for? Which form best suits given application domains