Transferable Safety

Motivation, Challenges and Potential Pitfalls

Jane Fenn – December 2013

Motivation – Interpreting the Question

NB. Comments predominantly refer to defence software-intensive systems

- Cost and development timescales for software systems have grown significantly with complexity
- Typically ‘accepted’ that use of open systems and/or Commercial Off-The-Shelf systems and components are viable strategies for reducing cost and timescales
  - *however*
- Reality is that addressing safety for these cases can out-weigh the cost/time savings for system development
  - Notable defence examples, eg. use of American-certified aircraft in UK – direct ‘transfer’ of an existing safety certification not accepted
- Intend to interpret the question as transferring existing ‘assurance’, and primarily ‘assurance evidence’, to facilitate the evaluation of product safety in a new environment

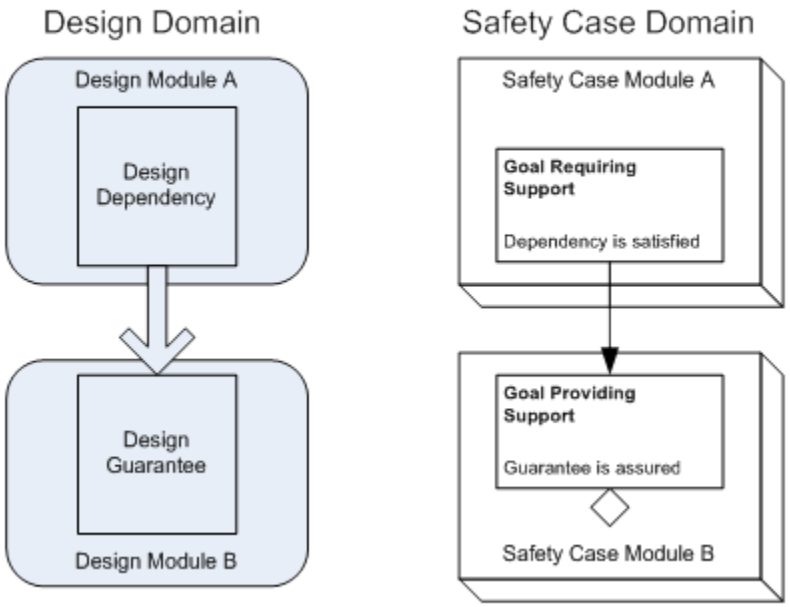
Motivation – Why am I interested?

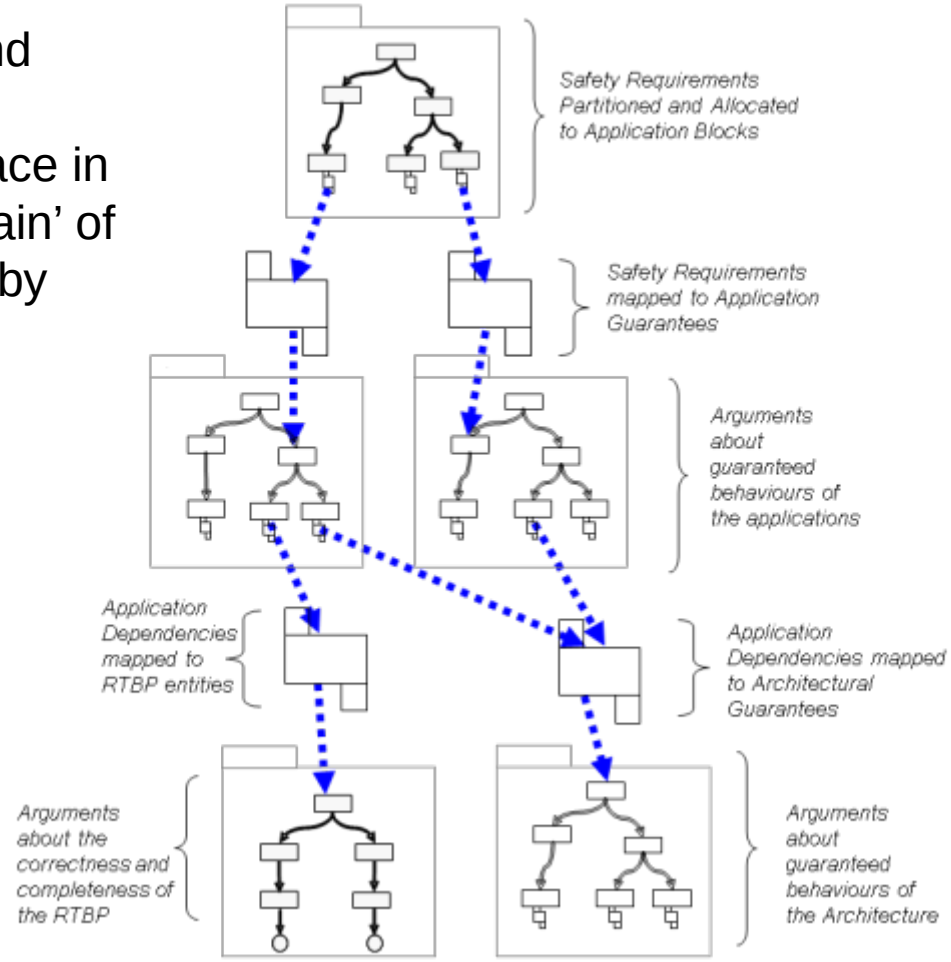
- Current practice
  - BAE Systems buys-in significant amount of software for use on its project
  - Where the software is pre-existing, assurance evidence may be offered by the software's supplier
  - Particularly where equipment is common with civil aircraft, likely that it has been developed to different standards, mainly DO-178
    - e.g. Radios, navigation equipment, TCAS
  - Desirable to re-use this evidence in the military regulatory environment
- Future practice
  - Currently working on a strategy for designing software for reusability
  - Will facilitate exchange of defence software components between aircraft, manufacturers and nations, and open up the marketplace for defence software
  - Need a strategy that addresses software by documenting the reusable software and providing assurance evidence which will proactively support software reuse
    - Particularly where different development standards might be used

Motivation – Isn't that what Modular Safety Cases do?

Brief summary:

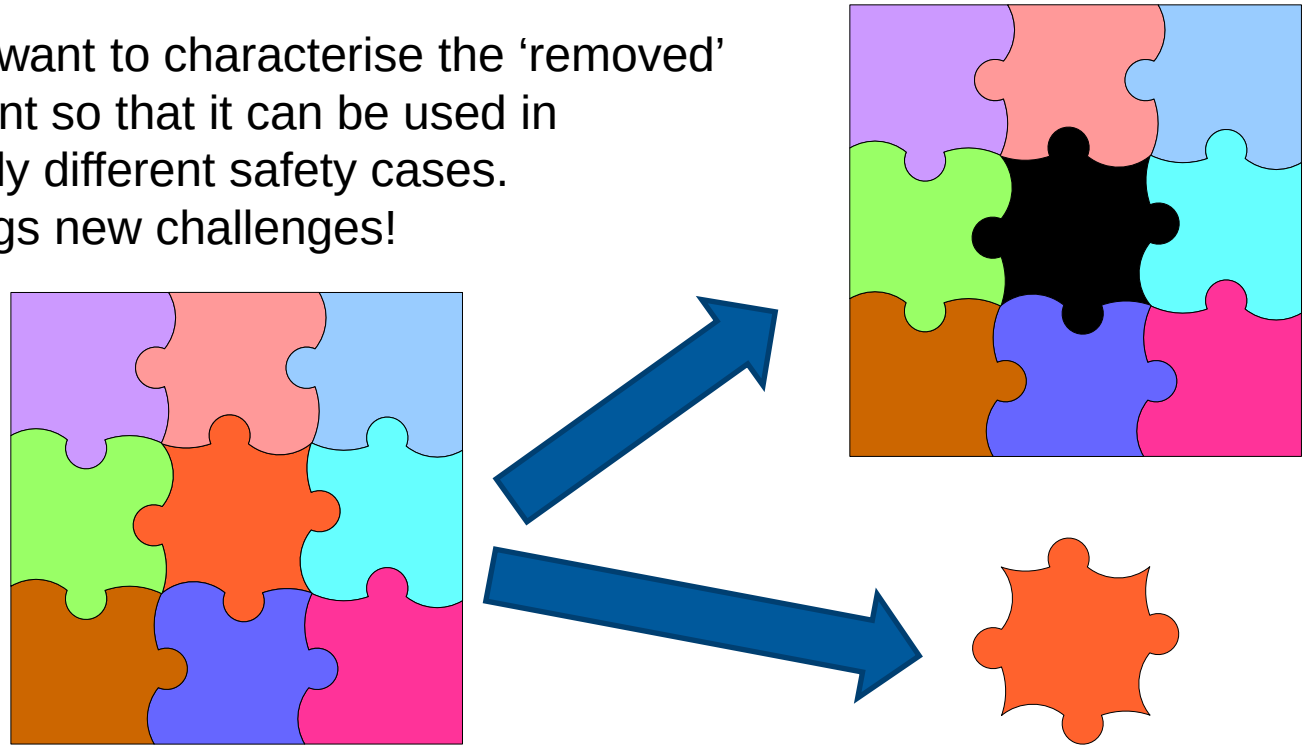
- If we can identify safety guarantees and associated dependencies at design interfaces, we can replicate that interface in the safety case and create a 'daisy chain' of dependencies and guarantees, linked by safety case contracts





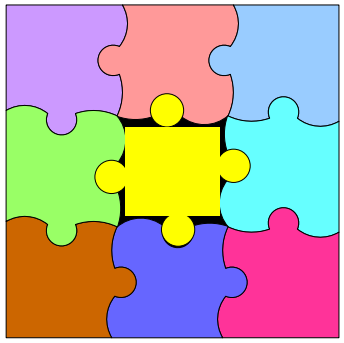
Motivation – Isn't that what Modular Safety Cases do? (2)

- Essentially, the IAWG version of modular safety cases were trying to characterise the 'gap' left when the component was removed from a system, to support change impact analysis.
- Now, we want to characterise the 'removed' component so that it can be used in completely different safety cases.
- That brings new challenges!



Challenges

- ***'Context'*** is the heart of the challenge, and at various levels of abstraction
- 'Context compatibility' was also what we needed to address to justify the validity of the safety case contracts in modular safety cases!
  - Low Level Context
    - E.g. Mars Rover had mis-match in units of measurement at an interface
    - E.g. Assumptions about bandwidth of internet connection available on network
  - How to unambiguously record sufficient data about the interface
    - Lots of existing work on techniques for making designs and their interfaces more rigorous, but need something that is 'portable'
  - Essentially:
    - How to know what is sufficient?
    - How to know what is relevant?
    - How to know what is important?
  - But not JUST at low level…..



Challenges

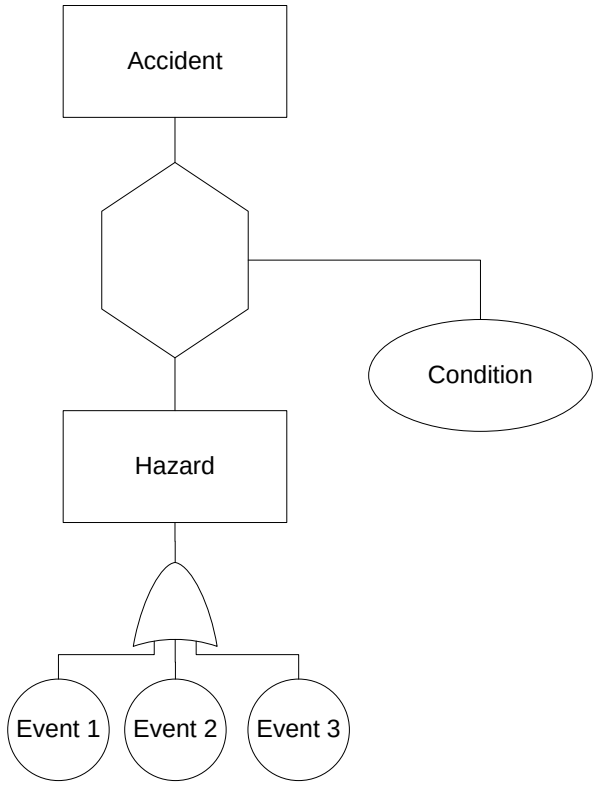
- High Level Context
  - Different regulatory environments which mandate different standards, and in some cases, even design choices
    - Even where domains are very similar
    - E.g. military and civil aerospace

Item	Process present in military standard	Process present in civil standard
Identify ways in which unexpected system behaviour can cause harm	√	√
Allocate a severity to the potential accident	√	√
Identify mechanisms for reducing the likelihood of the unexpected behaviour happening	√	√
Identify assurance requirement for those mechanisms	√	√

Challenges – Example – Military Aircraft

BUT Military standard (00-56) requires

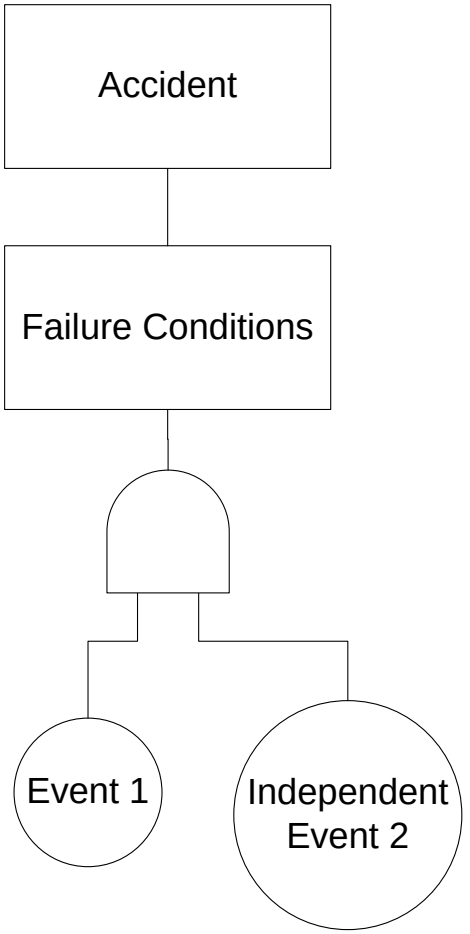
- All *accident* sequences to be assessed
- Highest severity allocation is 'CATASTROPHIC'
- Considers the **risk** of the *accident* occurring
- i.e. makes provision for mitigation to be through either
  - Reducing the probability of the causal events occurring
  - Reducing of the probability of the hazard propagating to an accident
    - By reducing the probability of the conditions occurring or introducing new 'conditions'
- Assurance allocation to causal events may take into account the hazard-to-accident mitigation
  - Assurance allocated to mitigating functions-only



Challenges – Example – Civil Aircraft

BUT Civil standard (ARP 4754A + circulars) requires

- 'Expected' *accident* to be considered in assessing severity
- Highest severity allocation is 'CATASTROPHIC'
- Specific requirements apply if 'catastrophic' severity allocated
- Assurance allocation to causal events may only be reduced by taking into account independent systems
  - Assurance allocated to whole equipment or module which generates causal events
- *Huge challenge to write a process so as to be compatible with both standards!*



BAE SYSTEMS
INSPIRED WORK

Potential Pitfalls

- Already considered some of the potential pitfalls, essentially insufficient or undeclared context
- Lists exist of properties to check, e.g. units, precision, endianism, etc
- But, often the problem is context that the original designer didn't consider 'important'
  - Seemed too 'obvious', based on 'custom and practice' on the original project
  - However, might too much contextual information be as bad as too little?
- Context issues that are specific to reusable software:
  - Consider case where software component is developed under DO-178, tested using target hardware and aircraft certification achieved
  - What if a variant aircraft has exactly the same functional/behavioural requirements but:
    - Uses a different compiler
    - Uses a faster processor
    - Has a different scheduling policy
      - What assurance evidence might be reusable in each case?

BAE SYSTEMS
INSPIRED WORK

Wish List?

- ‘Portable’ mechanism for rigorously defining interface properties that may be relevant to safety
  - OPENCOSS?
    - ***Short-term workaround:***
      - *use abstract language – XML*
      - *Enforce a template for the safety-relevant information required when identifying any safety-related component as reusable*
- ‘Portable’ mechanism for describing required or achieved assurance
  - ???
    - ***Short-term workaround:*** *Developers to declare assurance information available about any reusable safety-related component*
- Assurance Evidence characterisation that includes contextual dependencies
  - ??? – propose update to OMG and/or Assurance Evidence meta-model OPENCOSS?
    - ***Short-term workaround:*** *Use checklist of anticipatable context*

Thank you