McMaster University

McSCert

The Recent Trend to Assurance Cases – Pros and Cons

Tom Maibaum

Alan Wassyng, Mark Lawford, Hans Bherer

McMaster University

SQRL

CASCON 2010 SCC

Software Certification

- There are increasing problems related to software use in critical systems
- There are increasing problems for critical systems regulators
- Software is intrinsically different in the way it goes wrong – and we must cope with that
- But does this give us licence to approach regulation in a way which is different than used by classical engineers?

Certification: Product vs Process

- Process based or product based
  - Process essential to company for developing good software
  - Should be irrelevant in certifying software applications
- We check process because we can – examining evidence related to the product is difficult – in fact, an open research topic
- A good process is not a guarantee of a quality product; at best it offers an increased probability of quality

Certification:
Product vs Process

Bloomfield & Bishop:

"… what has been achieved, *not how hard you have tried*" (our emphasis)

Certification: Product vs Process

- We propose a product based approach:
  - Model entity in terms of measurable attributes characterising it
  - Measure value associated with each attribute
  - Aggregate measured values
  - Make engineering decision based on aggregated value (perhaps to issue a licence)
- If only life were that simple! ☺
- How do we identify the relevant attributes?
- How do we “measure” safety?

What Engineers Do and SEs (mostly) Don't Do

- Should what SEs do be that different form what classical engineers do?
  - Classical engineers have highly prescriptive, highly domain specific, highly product focused standards for certification
  - Properties to be determined, and sometimes the exact analysis method to be used, are defined in detail (possibly by reference to standards)
- In contrast, software related standards are very generic, focus on process elements and say almost nothing about the products manipulated by the process and their properties

Safety Cases

- Significant product focus
- Structured approach in which we
  - Make safety claims
  - Present arguments
  - Use evidence related to or derived from the product
- Mandated in the UK

  Defence standards, Air traffic management
- Recommended in an influential (US) NAS report

  Jackson et al: Software for Dependable Systems

Why Safety Cases?

- Reasons pro Safety Cases
  - “Demonstrates” that safety properties are satisfied & risks mitigated (?)
  - Mechanism for efficient review & involves all stakeholders
  - Provides a focus & rationale for safety activities
  - Demonstrates discharge of duty to public & shareholders
  - Allows for application of different standards at different times
  - Supports innovation (radical design) (?)

Bloomfield & Bishop again

Why Safety-Cases?

- Problems with prescriptive regulation
  - Safety may be seen as the regulator’s responsibility
  - Built on past experience – may not be current enough
  - Encodes current best practice that may eventually stifle progress
  - If overly restrictive may be barrier to open markets
    - I think I heard this one recently … ☺
  - Can adversely affect cost & quality
    - You can see the headline: SAFETY COSTS TOO MUCH!

Bloomfield & Bishop (2010), citing Robens (1972), Cullen (1990))

Why Safety Cases?

- The original motivation for safety cases produced a framework/approach for the structural organization of the safety argument
- It was designed to be high level and to be applied in many domains. It was certainly not software specific
- It is safety oriented – still true to a large extent for the *assurance case* approach currently being explored, for example, by John Knight

(Software) Engineering

McSCert

- Engineers have a duty to society to build effective (and cost-effective) artifacts that do not jeopardize public safety
- They use a variety of methods, heuristics and techniques to do this, and often use mathematical analysis to model and predict behaviour
- They are often extremely prescriptive in their regulations and in accepted professional practice

Software Engineering

- Is this really engineering?
- It should be!
- Much of the time we seem to believe in the rigour, and methods, and mathematics
- We fall down badly in a few areas:
  - Empirical basis for standards
  - Empirical confirmation of efficacy
  - Measurement
  - Prescription

Empirical Software Engineering

- Most of our software standards are anecdotal – this is a (poor) substitute for being based on empirical evidence
- We cannot really talk about the efficacy of our processes with any sort of authority – most of our processes are judged again on anecdotal evidence – sometimes very biased
- Well-founded software experiments are amazingly few in our literature

Measurement

- There are a number of excellent works on measurement and metrics in software engineering – and in spite of these, we have very few accepted measures related to the quality of a software product
- No wonder we rely on checking adherence to process for software certification – we do not yet know how to judge (measure) the products

Prescription

McSCert

- One of the major points of this talk!
- Most engineering regulation is prescriptive
- Much in engineering practice is prescriptive
- Engineers do this because:
  - They can (they have empirical and theoretical evidence)
  - It is safe – conservative maybe, but safe definitely
  - It takes into account the varying capabilities of practicing engineers

Civil Engineering Example

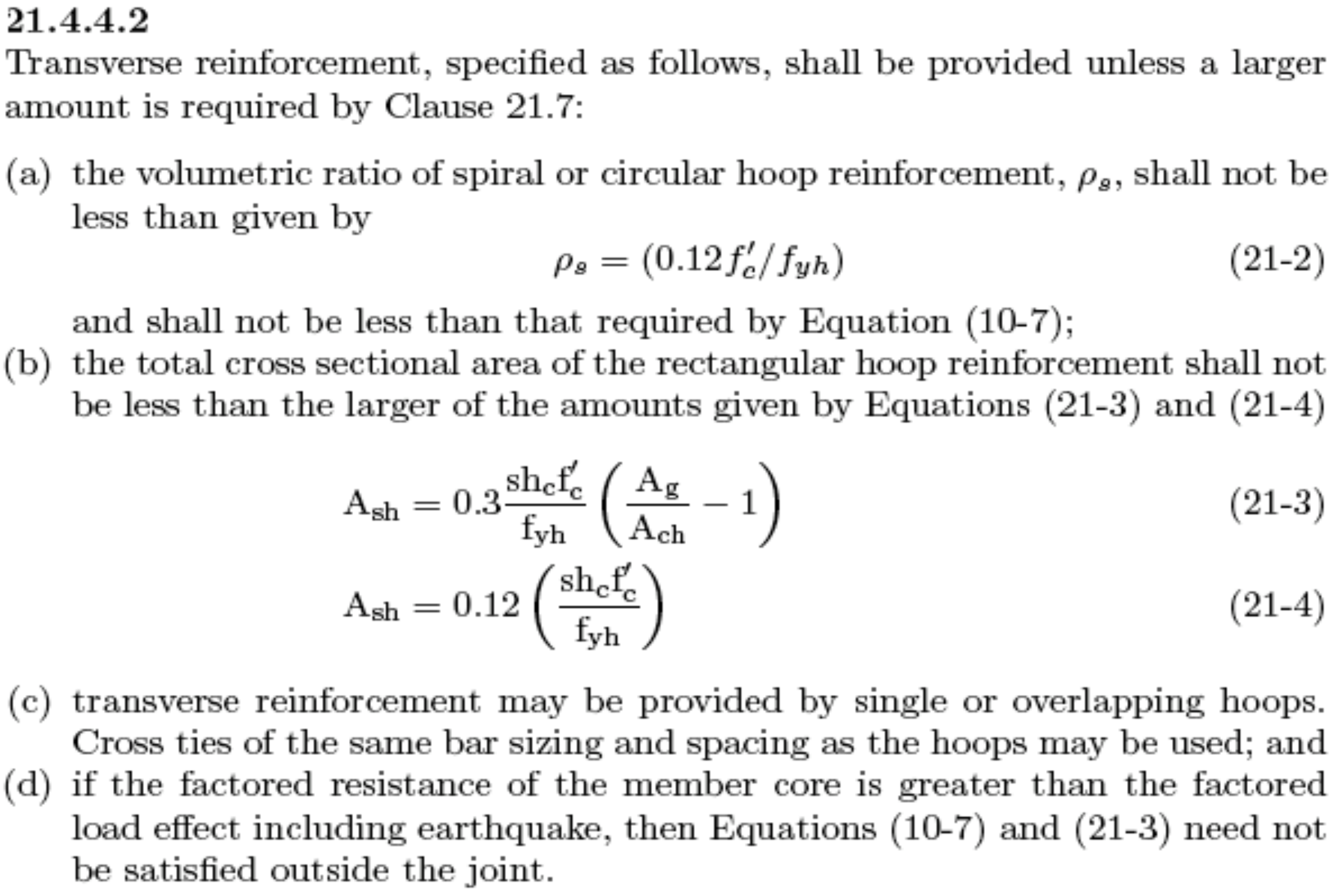
- Civil Engineers use Engineering Codes
- For example, the *CSA Standard CAN3-A23.3, Design of Concrete Structures for Buildings*
  - 15.4.1 The external moment on any section of a footing shall be determined by passing a vertical plane through the footing and computing the moment of the forces acting over the entire area of the footing on one side of that vertical plane
  - Prescriptive and conservative

Civil Engineering Example

- 19.2.1 Elastic behaviour shall be an accepted basis for determining internal forces and displacements of thin shells. This behaviour may be established by computations based on an analysis of the uncracked concrete structure in which the material is assumed linearly elastic, homogeneous, and isotropic. Poisson's ratio of concrete may be assumed to be equal to zero
- 19.3.1 The specified compressive strength of concrete, t'_c, at 28 days shall be not less than 20MPa
- Conservative, specifies acceptable assumptions and includes prescriptive requirements on materials

Civil Engineering Example

Even complex seismic design



21.4.4.2
Transverse reinforcement, specified as follows, shall be provided unless a larger amount is required by Clause 21.7:

(a) the volumetric ratio of spiral or circular hoop reinforcement, ρ_s, shall not be less than given by

$$\rho_s = (0.12 f'_c / f_{yh}) \qquad (21\text{-}2)$$

and shall not be less than that required by Equation (10-7);

(b) the total cross sectional area of the rectangular hoop reinforcement shall not be less than the larger of the amounts given by Equations (21-3) and (21-4)

$$\mathrm{A_{sh}} = 0.3 \frac{\mathrm{s h_c f'_c}}{\mathrm{f_{yh}}} \left(\frac{\mathrm{A_g}}{\mathrm{A_{ch}}} - 1 \right) \qquad (21\text{-}3)$$

$$\mathrm{A_{sh}} = 0.12 \left(\frac{\mathrm{s h_c f'_c}}{\mathrm{f_{yh}}} \right) \qquad (21\text{-}4)$$

(c) transverse reinforcement may be provided by single or overlapping hoops. Cross ties of the same bar sizing and spacing as the hoops may be used; and

(d) if the factored resistance of the member core is greater than the factored load effect including earthquake, then Equations (10-7) and (21-3) need not be satisfied outside the joint.

- Prescriptive and conservative and can be checked for compliance during and after

Lessons from Being Civil

- In the balance between *safety* and *creativity/ efficacy*, safety always wins
- Accepted as a way of life in the profession
- Prescriptive regulation is updated frequently – but not in a chaotic way
- Smart prescriptive regulation can be incredibly powerful
    - Canadian nuclear regulations – separate control and safety
- “Code” applies to the complete domain (concrete in our example)

Lessons from Being Civil

McSCert

- The standard imposes constraints and requirements on the product
  - Compliance can be determined objectively since it is defined in the context of the standard scientific measurement framework
- The standard is unashamedly prescriptive on analysis as well
- The standard is based on empirical confirmation of theory

SQRL

Downside of Safety-Cases

- Engineers classically rely on established and recognized methods for designing artifacts - Vincenti calls this *normal design*
- These assurances are backed up by standard analyses and measurement procedures.
- In contrast, *radical design* is where some element of a normal design method is absent, say because untried technology is used

Downside of Safety Cases: a Case of NSV?

- Software engineers have avoided developing a normal design culture
- Safety cases seem to be promoting the software industry’s avoidance of normal design
- This makes the regulators’ task difficult, and their processes become unpredictable & unreliable
- Regulators need *normal evaluation* methods – they will not be able to cope with hundreds of *one off* safety cases

Downside of Safety Cases

- It is not good enough that the producer of the product supplies the evidence and the supporting arguments in the safety-case
- What matters is that the certifying agent then cannot expect the same type of evidence and argument throughout the agent’s case load – thus agents have little chance of building essential expertise (safety-case templates may help – but probably not enough)

Downside of Safety Cases

- Safety cases quite clearly have been designed to present evidence of safety. In some domains, efficacy is also extremely important
- Medical devices in the US have to be proven to be both effective and safe
- There is almost always some tension between efficacy and safety, and safety cases were not created to deal with this complication

Are Safety Cases Safe?

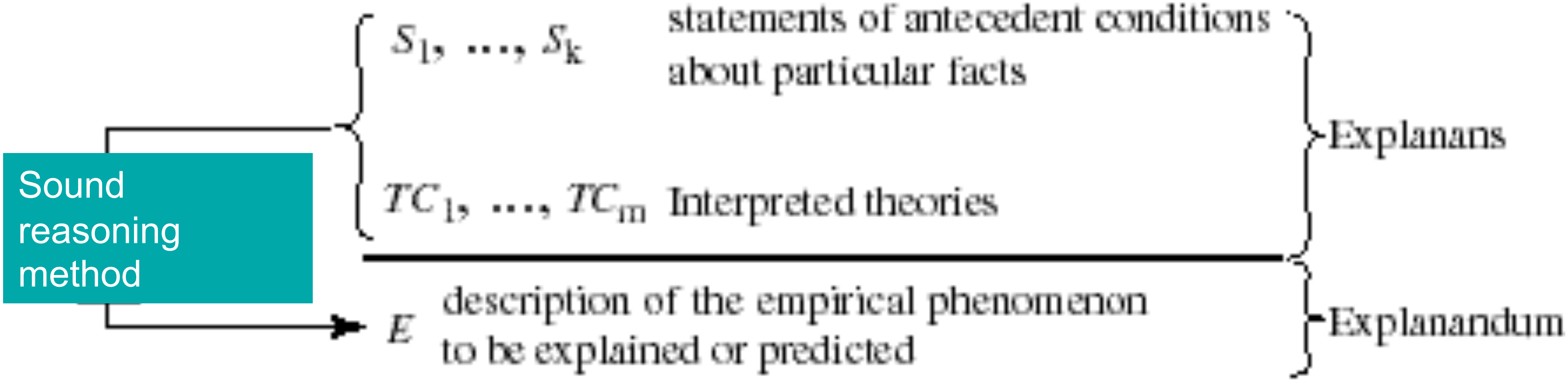
- Finally, how do we “measure” a safety/ assurance case for safety and efficacy?
- Given a safety/assurance case, how should a regulator decide to accept it or not?
- Is the argument presented in the safety case sound? How do we judge?
- All the work on safety cases has given us very few tools for making such judgments

A Conjecture

- There is a "tried and true" method that may be applicable: scientific explanation
- Scientists use rigorous reasoning based on theories of science and contingent facts (observations) to
  - Explain some observed phenomenon, or
  - Make an observable prediction about the consequences of a theory, given contingent facts
- A safety case may be seen as a prediction (of safety properties) based on underlying theory and contingent facts related to the system/ software

Scientific Explanation

- If a safety case is not an example of scientific explanation, I don’t know what it is!
- Argumentation (the standard way of structuring safety cases) invites judgment: what is the basis of this?



$S_1, \ldots, S_k$ statements of antecedent conditions about particular facts

$TC_1, \ldots, TC_m$ Interpreted theories

Explanans

E description of the empirical phenomenon to be explained or predicted

Explanandum

Sound reasoning method

Scientific Explanation

- So, a grand challenge, just because they are in fashion and I am a dedicated follower of fashion!

  Systematise the reasoning behind scientific explanation so that you can automate it!

Conclusions

- Safety cases are proposed as THE way of certifying systems – and that software specific certification processes can be used within the context of safety cases
- We like safety cases as a way of structuring safety arguments, but the non prescriptive approach exemplified by safety cases will trickle down into the software specific process – ruining our chance of developing effective, predictable, certification methods
- And there is the small problem of assessing safety cases on a repeatable, objective basis

Conclusions

- The arguments against prescription given by safety case proponents seem thoroughly unconvincing
- Prescriptive regulation does not need to be overly static – it usually is not
- The point on responsibility has some merit – but is true in all engineering jurisdictions and does not seem to have been a real problem
- These arguments tend to favour creativity and progress over safety – strange for safety case proponents!