FDA Assurance Cases, 21, 22 Feb 2008
Based on Open Group Paris 23 April 2007, slight revisions of
Open Group San Diego 31 January 2007, major rewrite of
HCSS Aviation Safety Workshop, Alexandria, Oct 5,6 2006
Based on University of Illinois ITI Distinguished Lecture
Wednesday 5 April 2006
based on ITCES invited talk, Tuesday 4 April 2006

Assurance Cases and Ultra-High Confidence

John Rushby

Computer Science Laboratory
SRI International
Menlo Park CA USA

Justifiable Confidence

- Let's look at an area where ultra-high confidence is justified
- Software in modern civil aircraft
- Enough flight experience to substantiate failure rates close to 10^{-9} per hour in critical software
- Largely standards-based
  - DO-178B (software)
  - DO-254 (complex hardware)
  - DO-297 (integrated modular avionics)
- Can we learn from these?

Maybe Not: They Are Fallible

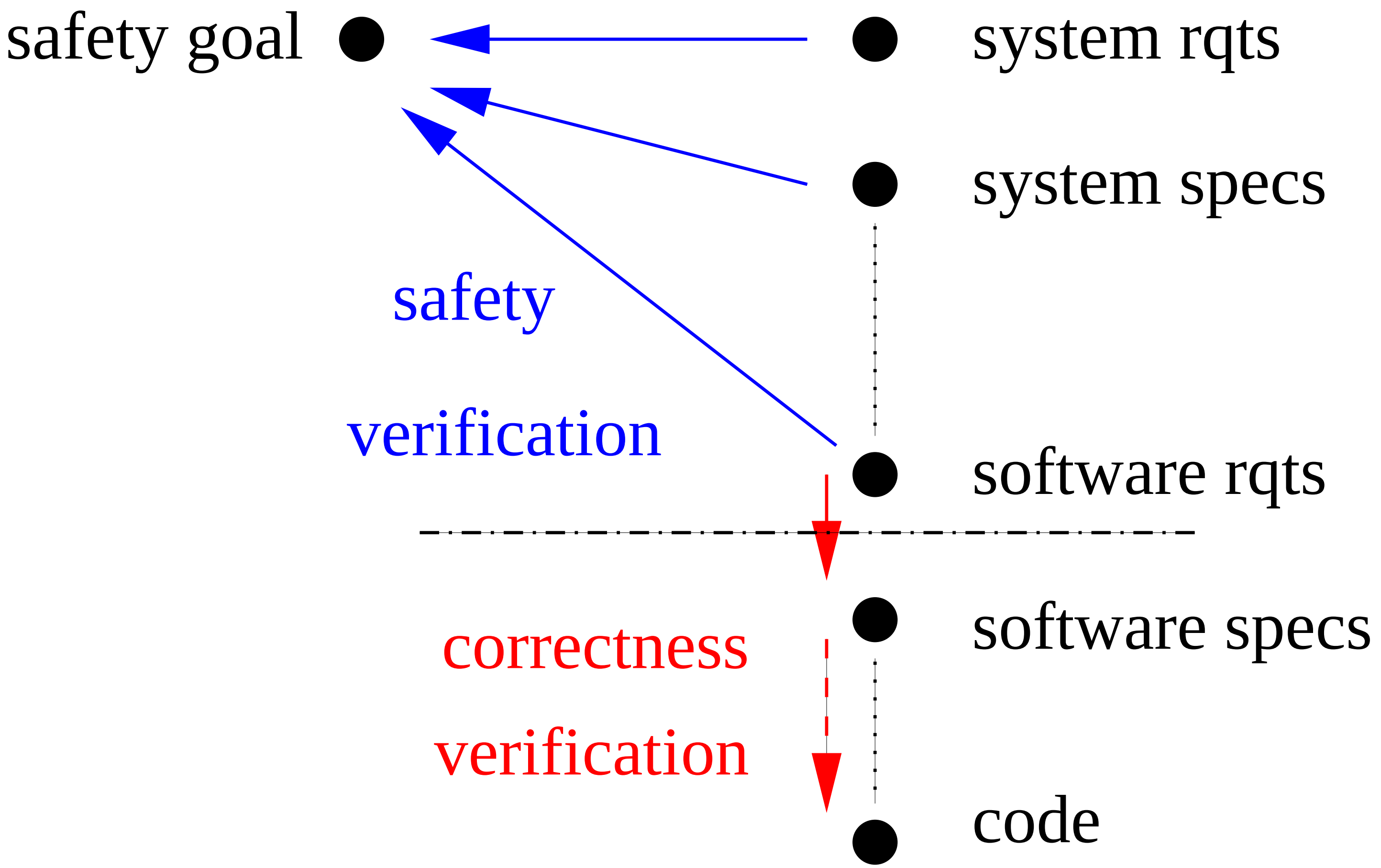
- Fuel emergency on Airbus A340-642, G-VATL, on 8 February 2005 (AAIB SPECIAL Bulletin S1/2005)
- Toward the end of a flight from Hong Kong to London: two engines shut down, crew discovered they were critically low on fuel, declared an emergency, landed at Amsterdam
- Two Fuel Control Monitoring Computers (FCMCs) on this type of airplane; they cross-compare and the "healthiest" one drives the outputs to the data bus
- Both FCMCs had fault indications, and one of them was unable to drive the data bus
- Unfortunately, this one was judged the healthiest and was given control of the bus even though it could not exercise it
- Further backup systems were not invoked because the FCMCs indicated they were not both failed

Safety Culture

- See also incident report for Boeing 777, 9M-MRG
  (Malaysian Airlines, near Perth Australia)
- We don't know what in DO-178B makes it work
  - Most of the time

  But sometimes fail
- Maybe current development and certification practices may be insufficient in the absence of safety culture
- Current business models are leading to a loss of safety culture
  - Outsourcing, COTS
- Safety culture is implicit knowledge
- Surely, a certification regime should be effective on the basis of its explicit requirements

Maybe Not:
They Focus On Correctness More Than Safety



The (premature?) focus on correctness is hugely expensive

Maybe Not:
We Don't Know Why Some Things Are Required

E.g., MC/DC testing

- Structural test coverage criterion required for Level A software
- Tests are generated from requirements
- Coverage is measured on the code
- Can only get high coverage if the requirements are highly detailed

Is it evidence for good testing or good requirements?

Maybe Not:
We Don't Know Why We Have To Do More

- Level A software requires MC/DC test coverage
- Level B does not
- Static analysis finds significant anomalies in avionics code
  - A worrying discovery on its own
- No discernible difference in anomaly rates between Levels A and B
- So what did the extra work buy us?
- Actually, there is lots of work on the efficacy of testing
- But does this remain valid when tests are auto-generated?
  - Auto-generated tests are often minimal

Maybe Not: Why Are Multiple Forms of Evidence Required?

- More evidence is required at higher Levels/EALs/SILs
- What's the argument that these deliver increased assurance?
- Generally an implicit appeal to diversity
  - And belief that diverse methods fail independently
  - Not true in n-version software, should be viewed with suspicion here too

Critique of Standards-Based Approaches

- Goals, evidence, argument are surely present
  - What other basis for certification is there?

  But they are mostly implicit
- Explicitly they define only the evidence to be produced
- The goals and arguments are implicit
- Hence, hard to tell whether given evidence meets the intent
- On the other hand: can work well in fields that are stable or change slowly
  - Can institutionalize lessons learned, best practice
    - ⋆ e.g. evolution of DO-178 from A to B to C
- What can we adopt for assurance cases for medical devices?
  - Airplanes are much simpler than physiology

Rational Safety Cases

- Currently, we apply safety analysis methods (HA, FTA, FMEA etc.) to an informal system description
  - Little automation, but in principle
  - These are abstracted ways to examine all reachable states
- Then, to be sure the implementation does not introduce new hazards, require it exactly matches the analyzed description
  - Hence, DO-178B is about correctness, not safety
- Instead, use a formal system description
  - Then have automated forms of reachability analysis
  - Closer to the implementation, smaller gap to bridge
- Analyze the implementation for preservation of safety, not correctness
  - Favor methods that deliver unconditional claims

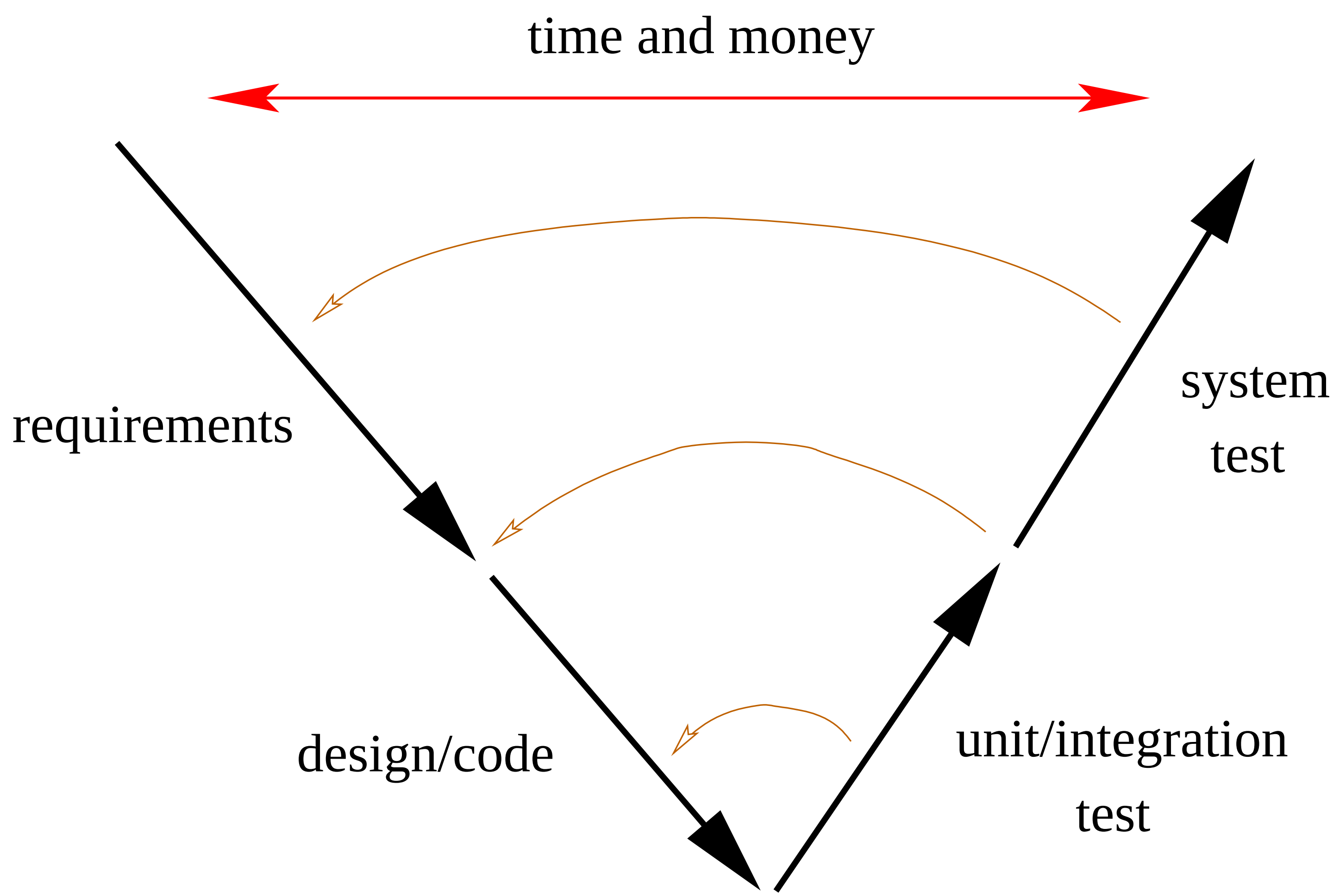
Formal Methods (aside)

- Formal methods are not about priestly ways to complicate life
- They are about automated analyses that consider all possible executions
- To make them tractable, may need to approximate
  - Crude: downscaling
  - Principled: predicate abstraction, abstract interpretation, etc
- Most of the action is in improved automation, and automated abstraction

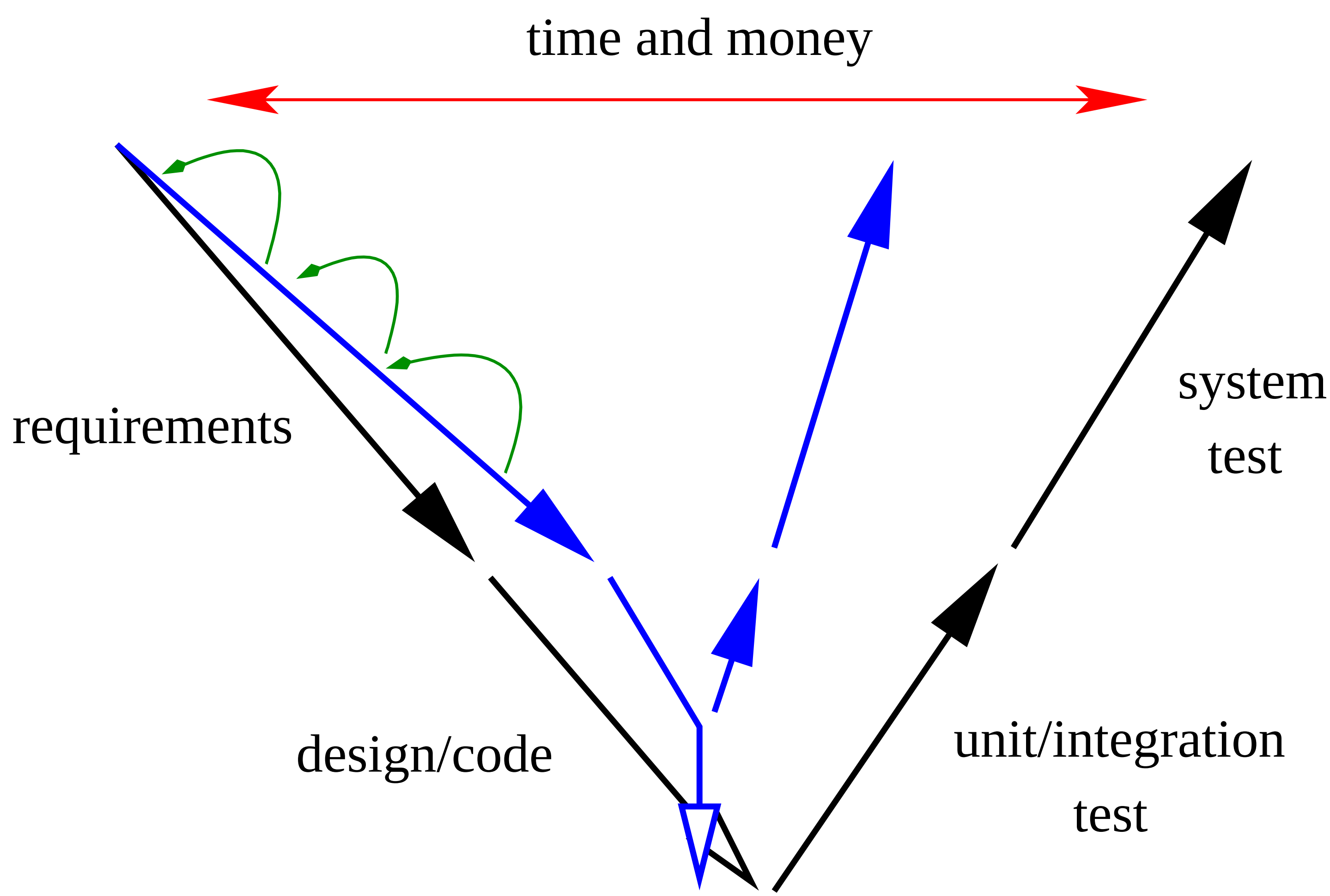
Formal Methods

- The move to model based development presents a (once in a lifetime) opportunity to move analytic methods into the early lifecycle, mostly based on formal methods
- Modern automated formal methods can deliver unconditional claims about small properties very economically
  - Static analysis, model checking, infinite bounded model checking and k-induction using SMT solvers, hybrid abstraction (which uses theorem proving over reals)
- Larger properties will require combined methods
  (cf. the Evidential Tool Bus)
- The applications of formal methods extend beyond verification and refutation (bug finding): test generation, fault tree analysis, human factors,. . .
- Tool diversity may be an alternative to tool qualification

Traditional Vee Diagram (Much Simplified)



Vee Diagram Tightened with Formal Methods



Example: Rockwell-Collins

Multi-Legged Arguments

- Need to know the arguments supported by each item of evidence, and how they compose
- Want to distinguish rational multi-legged cases from nervous demands for more and more and . . .

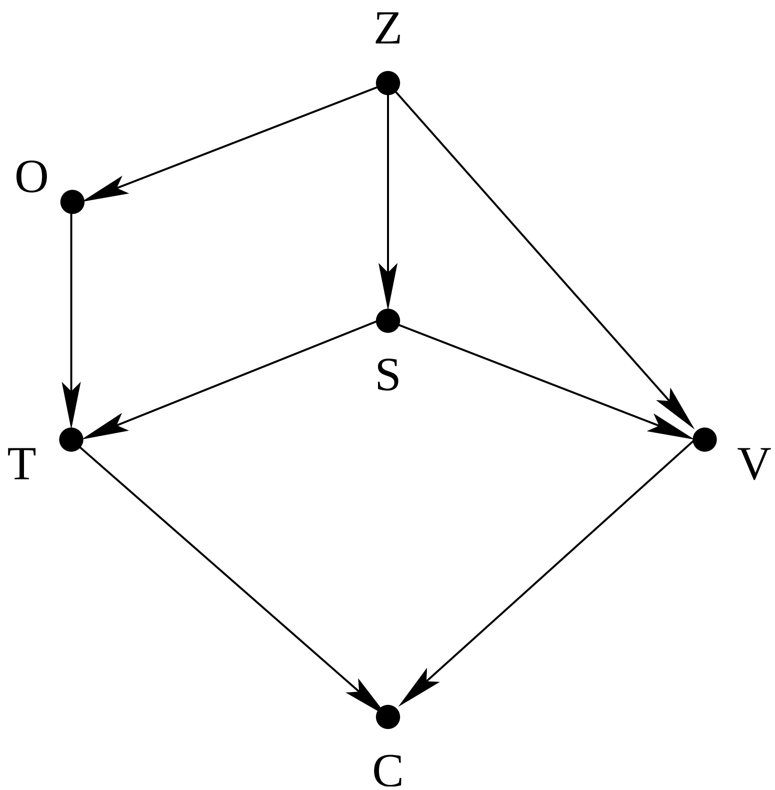
Two Kinds of Uncertainty In Certification

- One kind concerns failure of a claim, usually stated probabilistically (frequentist interpretation)
  - E.g., 10^{-9} probability of failure per hour, or 10^{-3} probability of failure on demand
- The other kind concerns failure of the assurance process
  - Seldom made explicit
  - But can be stated in terms of subjective probability
    - E.g., 95% confident this system achieves 10^{-3} probability of failure on demand
    - Note: this does not concern sampling theory and is not a confidence interval
- Demands for multiple sources of evidence are generally aimed at the second of these

Bayesian Belief Nets

- Bayes Theorem is the principal tool for analyzing subjective probabilities
- Allows a prior assessment of probability to be updated by new evidence to yield a rational posterior probability
  - E.g., P(C) vs. P(C | E)
- Math gets difficult when the models are complex
  - i.e., when we have many conditional probabilities of the form p(A | B and C or D)
- BBNs provide a graphical means to represent these, and tools to automate the calculations
- Can allow principled construction of multi-legged arguments

A BBN Example



Z: System Specification

O: Test Oracle

S: System's true quality

T: Test results

V: Verification outcome

C: Conclusion

Absolute Claims in Multi-Legged Arguments

- Can get surprising results (Littlewood and Wright)
  - Under some combinations of prior belief, increasing the number of failure-free tests may decrease our confidence in the test oracle rather than increase our confidence in the system reliability
- The anomalies disappear and calculations are simplified if one of the legs in a two-legged case is absolute
  - E.g., 95% confident that this claim holds. . . period
  - Formal methods deliver this kind of claim
  - E.g., Spark Ada (with the Examiner): guaranteed absence of run time exceptions
- Extends to multiple unconditional claims

Flies in the Ointment

- These results assume the verification leg considers the same system description and requirements as the other leg
- But this is seldom the case
  - Verification of weak properties: static analysis etc.
  - Verification of specific critical properties (subclaims)
  - Verification of abstractions of the real system
- It's a research challenge to develop the theory to cover these issues
- Aside: philosophers studying confirmation theory (part of Bayesian Epistemology) formulate measures of support differently than computer scientists
  - e.g., $c(C, E) = P(E \mid C) - P(E \mid \text{not } C)$

From Software To System Certification

- The things we care about are system properties
- So certification focuses on systems
  - E.g., the FAA certifies airplanes, engines and propellers
- But modern engineering and business practices use massive subcontracting and component-based development that provide little visibility into subsystem designs
- Strong case for "qualification" of components

  Business case: Component vendors want it (cf. IMA)

  Certification case: system integrators and certifiers do not have visibility into designs and processes
- But then system certification is based on the certification data delivered with the components
  - Must certify systems without looking inside subsystems

Application to Medical Devices

- We certify individual medical devices
- Then assemble them into larger systems (PnP)
  - The patient
  - The operating room
  - Hospitals
- Often without conscious system-level design
- We need compositional assurance cases to support PnP

Compositional Design and Development

- Compositional certification will be impossible unless there is a deliberate (and successful!) attempt to control subsystem interactions during design and development
- It's also what's needed for safety: cf. Perrow's tight coupling and high interactive complexity
  - Would be manifested through excessively complex mutual assumptions and guarantees
- The alternative is massive testing at every stage (cf. NASA), and you still have no guarantee of success
- Aside: Boeing 787 has 20,000 LOC in its IMA, a few million LOC of safety-relevant software, and 750,000 lines of XML in configuration data

A Science of Certification

- Certification is ultimately a judgment that a system is adequately safe/secure/whatever for a given application in a given environment
- But the judgment should be based on as much explicit and credible evidence as possible
- A Science of Certification would be about ways to develop that evidence

Making Certification "More Scientific"

- Favor explicit over implicit approaches
  - i.e., goal-based over standards-based
  - At the very least, expose and examine the claims, arguments and assumptions implicit in standards-based approaches
- Be wary of demands for multiple forms of evidence, with implicit appeal to diversity and independence
  - Instead favor explicit multi-legged cases
  - Use BBNs to combine legs
  - Favor methods that deliver unconditional claims
- Use formal ("machinable") design descriptions
  - Automate safety analysis methods
  - Analyze implementation for preservation of safety

Compositional Certification

- This is the big research challenge
- It demands clarification of the difference between verification and certification (because we know how to do the former compositionally, but not the latter)
- And explication of what constitutes an interface to a certified component
    - The certification data is in terms of the interface only
    - You cannot look inside
- Compositional certification should extend to incremental certification, reuse, and modification
- It's also the big challenge for regulatory agencies
    - A completely different way of doing business

A Research Agenda

- The Science of Certification
  - Or a science for certification
- Specification and verification of integration frameworks
  - Partitioning, separation, buses, kernels
- High-performance automated verification for strong properties of model-based designs
  - Mostly infinite state and hybrid systems

  And automation of related processes (test generation, FTA)
- Compositional certification
  - Composition of hybrid systems
- Tool qualification
  - Evidence management
- Just-in-time certification and runtime synthesis