



CITY UNIVERSITY
LONDON

Introducing probabilistic reasoning into Assurance Cases

presented by **Lorenzo Strigini**
Centre For Software Reliability
City University London, U.K.

Outline

Why probabilistic reasoning?

Where? How?

Directions of CSR's contributions and ongoing research

Some experience: advantages, difficulties

Conclusions

Why probabilities for assurance cases?

We build structured assurance cases to organise reasoning

- showing how complex, disparate evidence actually supports a claim, e.g., “this system is safe enough”
 - via connected sub-arguments:
given this specific evidence and in view of this general knowledge, we can make this specific claim
 - so that arguments can be *communicated* clearly and *examined* systematically, avoiding wrong deductions
- formal, mathematical representation is the standard way of improving quality of reasoning
- when dealing with *uncertainty*, the mathematics of *probabilities* provides an appropriate language
- indeed, reasoning with probabilities is inevitable if
 - the claim to be supported is probabilistic
 - the evidence is statistical

Directions of work. 1: applying probabilistic reasoning to more aspects of argument

... where not yet standard practice

- software reliability
 - successful quantitative, probabilistic methods
- human aspects; security
 - see other presentations, posters
- one often needs to refute generic objections against applicability of probabilities
- while qualifying the extent of the gains expected
 - may be numerical assessment, or just insight about factors affecting system dependability

Example: software reliability

- old objection: “probabilities make sense for *random* events, but software is *deterministic*”
- our counter-argument:
 - probabilities make sense for describing *uncertainty*
 - “software is deterministic” means that the response to a given input value (and state) will always be the same
 - + for some values, a *wrong* response (a failure)
 - uncertainties:
 - + we do not know which input values these are (the undiscovered bugs)
 - + we do not know when, or how often, they will arise
- results: a probabilistic approach allows us to describe the effect of software on system dependability; apply rigorous methods for reliability assessment; reason about the effectiveness of fault tolerance; ...

Directions of work.

2: linking the probabilistic and informal parts of a case

A case often contains probabilistic arguments

- e.g., calculated reliability of a subsystem
 - using an accepted mathematical model and assumed values of reliability of its components
- trust in this reliability claim depends on trust that e.g.
 - this model describes the real system with sufficient fidelity
 - the component reliability values are accurate enough
- that is, it is affected by “epistemic” uncertainty
 - including “unknown unknowns”
 - usually *not* described mathematically, or even explicitly
- some directions of work:
 - describing confidence in parts of arguments (evidence, models,...) and its effect on claims
 - e.g. possible trade-offs between confidence and claim levels
 - clarifying notions of “sufficient” confidence

Directions of work.

3: drawing conclusions from complex evidence

- cases include disparate evidence and arguments
 - e.g. evidence concerning software: production process, development organization, test results, static verification ...
- usually combined into final claim through “judgement”: an *intuitive, unverifiable* process
- but we’d wish to check more in depth:
 - e.g., how much confidence does each item of evidence add to a claim?
- can we express all this mathematically?
 - allow communication and checking
 - make hidden premises explicit
 - detect inconsistencies
 - ...
- ...making the assessment more scientific than otherwise feasible

Experience with probabilistic formalisms for cases

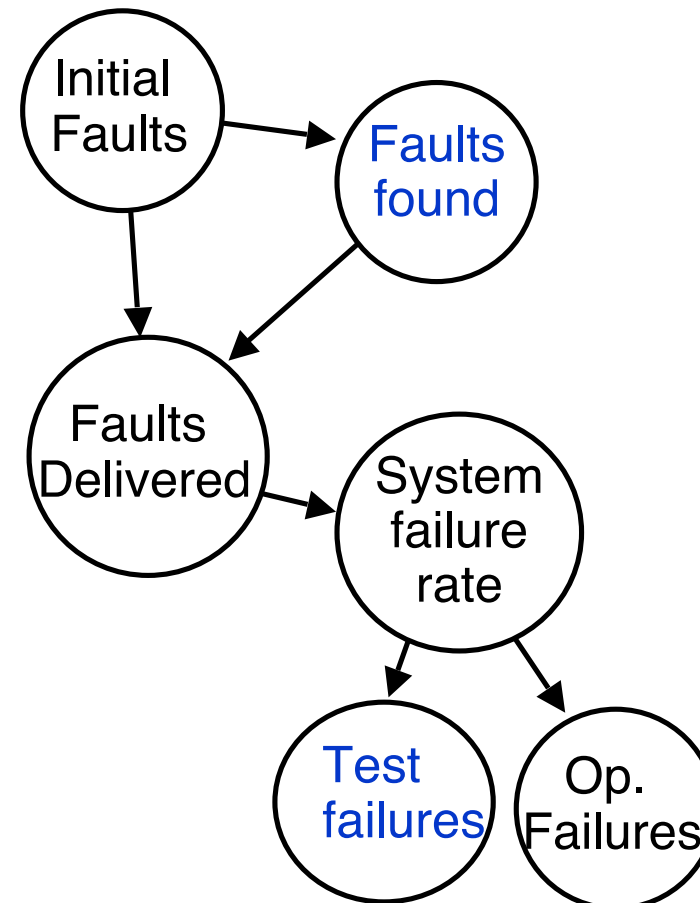
We present a few examples represented via Bayesian networks

- a language for describing complex probabilistic models
- with intuitive visual representation: a graph with associated probability tables
- supported by software tools

An example about software reliability

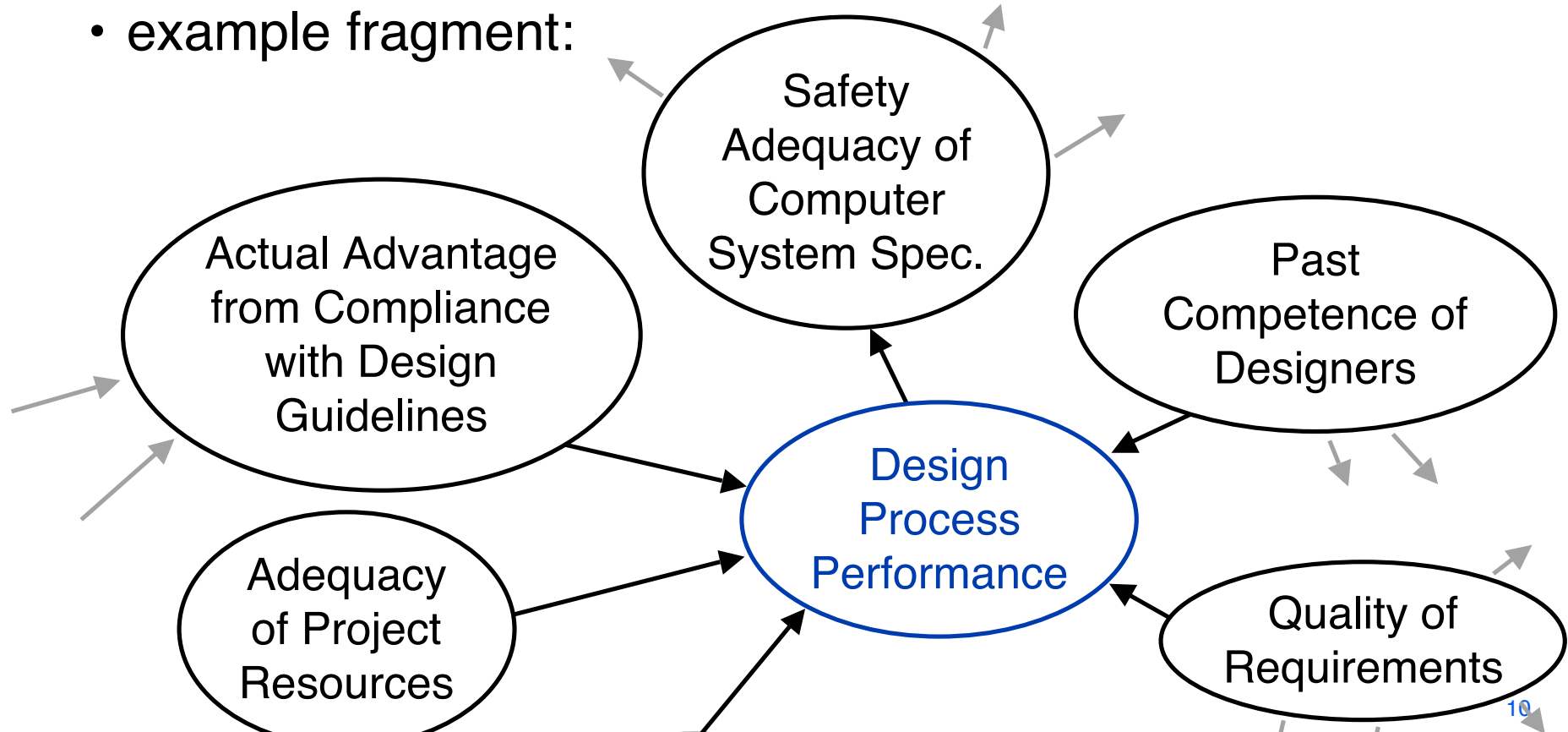
e.g., this Bayesian network clarifies what we can or cannot claim about failures in future operation, based on development data, given

- assumptions specified with the network
- evidence (in blue)



A more complex case study

- part of safety argument for a class of software-based systems used in nuclear plants
- addressing the early part of the lifecycle
- describing an expert safety assessor's judgement
- example fragment:



Experience with this case study

- the formalism helps the expert to examine his own judgement process
- less likely to support discussion towards scientifically based consensus
 - it described informal judgement without reference to agreed basis of knowledge, theory
- practical difficulties, and methods for alleviating them
 - eliciting the necessary probability parameters
 - feedback of the model's implications

“2-legged argument” Bayesian network

- claim about a bound on software probability of failure
- based on testing plus static verification

S: system's true unknown pfd, $0 \leq S \leq 1$

Z: system specification, {correct, incorrect}

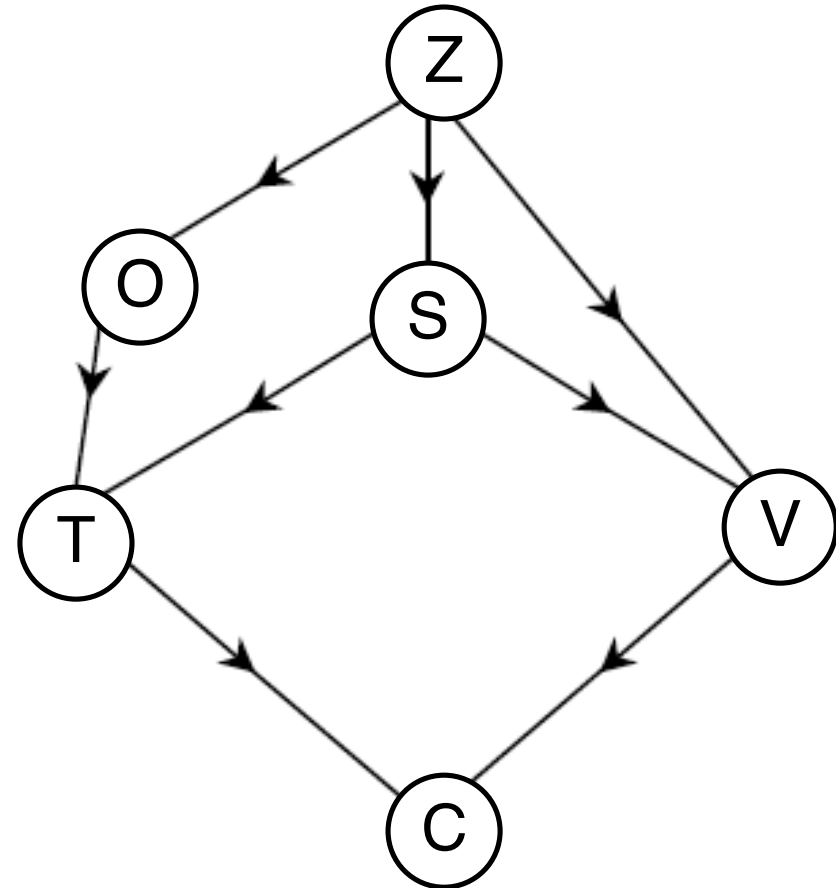
O: testing oracle, {correct, not correct}

V: verification outcome, {verified, not verified}

T: test result, {no failures, failures}

C: final claim, {accepted, not accepted}

(V,T) represents evidence



Interesting findings, e.g.: when is it that *positive* evidence should *decrease* your confidence?

- e.g. testing results “too good to be true”

Summary

- probabilistic reasoning is a natural way of making reasoning more explicit
 - clarity for understanding, communicating, verifying arguments
 - different uses and extent of results depending on context
 - risks to manage: complexity, potential for unwarranted extensions, spurious authority
- we have identified three directions for extending the application of probabilistic reasoning
 - arguments for different sub-systems, or subclaims
 - clarifying some informal reasoning, “epistemic uncertainty”
 - drawing conclusions from complex, disparate evidence
- and reported some examples and experience

Questions, comments?