CHAPTER 11

Littlewood and Rushby's Approach Through Specialist Architecture

11.1 Evaluating a System Through Running It

Testing involves giving a system some judiciously chosen input values and seeing what behavior is produced. But testing is not the only way a system is evaluated by running it. Systems are also used, and if a system has been used for long enough, maybe one can say enough about the real behavior of a component when running to use that component somewhere else. A technical term for this in the IEC 61508 standard is «proven in use». An assessor will very likely be presented with data about a system or system component as it has been run, and must somehow judge the system or component as fit for purpose. This is a problematic area, in which scientific results are often at variance with the reality of attempted evaluation.

Some guidance for «proven in use» is given in the standard, in Part 7 Annex D, and of course it is logically the same guidance which would be given for drawing conclusions from tests. But it is not clear how useful the guidance really is. (Actually, some of us believe it is clear: it's not very useful!)

The guidance is based on classical frequentist reasoning. For example, one may be able to conclude by performing nearly 500,000 operational hours without seeing a failure that one has attained a 99% confidence level that the software will run for 100,000 hours without a failure. But 100,000 hours without failure is the absolutely lowest level at which the standard recognises a reliability condition for critical hardware (SIL 1 HW is to run between 100,000 hours and 1 million hours without failure). And 500,000 operational hours are hard to come by, because one must be sure that the software is exactly the same throughout (no updated versions!). Furthermore, the operational conditions (environment and distribution of inputs) must be exactly the same as for the intended future use, and the oracle for detecting failure must have been perfect! Subtle differences in operational use can invalidate any hoped-for conclusion. And one failure in this time invalidates the positive conclusion one might hope to draw, so one needs to be sure any failure has been detected: a perfect oracle. These conditions on versions and operational profile are strenuous enough to make the guidance all but inapplicable in most cases for software [1, 5].

Since that is so, can we relax the conditions somewhat, and relax our confidence comparatively? The answer is simple but firm: no! The general reason, as we all should know, is that discrete systems controlled by software have sharp choice points, at which behavior is highly discontinuous – and lots of them. You cannot just alter an ops profile «a little bit» and hope the resulting behavior will only alter «a little bit»: the statistical criterion is «no failures», not «one or two».

11.2 So what can we do that is more practical?

There are considerable sources of uncertainty in evaluating software. There are versions: does a new, modified «update» share relevant properties of the original version, such as freedom from behavior which would or could cause a dangerous failure? Say, we have seen software from such-and-such a developer before, and they have a detailed track record for quality. How confident may we be that their new software under evaluation attains the level of quality we have come to expect from them?

I think that we might well have to approach this issue step by careful step. Not in general, as attempted by Part 7 Annex D, but through many particular instances. Some recent work of Littlewood and Rushby have produced some guidance on how a statistical evaluation explicitly incorporating uncertainty might work, for a very specific architecture [4]. I include a brief sketch here with permission.

11.3 Developing Confidence in Certain Architectures Through Practical Statistical Reasoning: Littlewood and Rushby

In [4] Littlewood and Rushby consider certain types of so-called 1oo2 divergent architectures. 1oo2 divergent architectures are common, although not ubiquitous, in critical control systems and protection systems in diverse industries such as nuclear power, aviation, rail and medical devices.

The calculations concern two-channel architectures in which (i) the system as a whole has at least one guaranteed "safe state", (ii) one channel, Channel A, provides the required functionality in normal operations, and is as complex as necessary, and (iii) the second channel, Channel B, is very simple, provides limited functionality, but in particular the function of bringing the system into a safe state if Channel A fails dangerously (here the system needs to include a perfect oracle for Channel A's "health"). Both Channels are "hot". They are assumed to be SW-based, so SW reliability models are appropriate. I consider here demand-based functionality only.

Assumption: the system is such that it does not fail dangerously if either Channel A does not fail or Channel B does not fail.

A key property of the architecture is that Channel B is arguably perfect. The functionality implemented in Channel A is assumed to be complex enough that one can construct an argument for its reliability (in terms of probability of dangerous failure on demand, pdfd), but its perfection is implausible.

There are two kinds of probability calculation involved, technically called aleatory and epistemic, the terms used by Littlewood and Rushby.

Aleatory probabilities are those taken to be inherent in the situation itself, given a sample space. For example, in a sequence of rolls of a fair die, it is taken that each face appears according to a uniform distribution: each of the six faces has an aleatory probability of 1/6 of appearing on any given roll. The aleatory probability of, say, a dangerous failure of the system, will in general not be known.

The epistemic probability reflects what an assessor can know about the system behavior, given the evidence. The assessment of the system is based on the epistemic probability as calculated by the assessor.

The crucial practical point is as follows. Although the aleatory probability can likely not be known, its form is determined by the architecture, and this form leads to a particularly simple formula for the epistemic probability, which is easily calculated by any engineer (it is pure arithmetic). Furthermore, the level of evidence required to attain high confidence in appropriately high levels of freedom from dangerous behavior appears to be practical.

Let the aleatory probability of dangerous failure on demand of Channel A on a randomly-selected input be denoted by $pdfd_A$, and the aleatory probability that Channel B is not perfect by pnp_B. The probability $pdfd_A$ is the usual kind of well-understood probability random variable. The probability of non-perfection of B, pnp_B , can also be given a frequentist interpretation, as the probability, say, that a SW-subsystem of this size with this kind of functionality, developed using such-and-such methods, is not perfect. Let the (unknown) values of these probabilities be p_A , respectively p_B.

Littlewood and Rushby show that the probability that the system fails, that is, that Channel A fails dangerously, and that Channel B also fails at this point, is bounded above by $(p_A \times p_B)$.

This means that the probability of dangerous failure of Channel A is conditionally independent of the probability of imperfection of Channel B in the architecture presented. (This calculation is performed on the conservative assumption not only that Channel B is imperfect, given by p_B , but that it actually does fail, given a dangerous failure on demand of Channel A.)

The question is, now, what epistemic probability an assessor should assign to a system failure on demand. Most general is that an assessor has a joint probability distribution $F(p_A, p_B)$. The probability that the system fails on a random demand is then $\int (p_A \times p_B) dF(p_A, p_B)$. If the assessors' beliefs are independent, so that $F(p_A, p_B) = F(p_A) \times F(p_B)$, then

$$\int (p_A \times p_B) dF(p_A, p_B) = \int p_A dF(p_A) \times \int p_B dF(p_B) = P_A \times P_B$$

The question is whether it is reasonable for an assessor to believe that flaws in Channel A and imperfections in Channel B are independent, and the answer is: likely no. For example, misinterpretations of the engineering requirements may be a common cause of Channel A failing and Channel B not being perfect. Similarly, shared mechanisms and higher-level mechanisms could fail, such as the communications of the coordination between Channels A and B to the encompassing system. Such a failure would also constitute a common-cause failure.

However, Littlewood and Rushby point out that common-cause failures are the only plausible mechanism by which the joint distribution $F(p_A, p_B)$ may not be factored as above. They proceed as follows.

Let C be the assessor's a priori estimate of the probability of common-cause failure of both channels. Then C is placed at point (1,1) in the probability space (meaning: both channels fail with certainty during a common-cause failure). It is further assumed that the probability of common-cause failure is independent of the probability of individual happenstance failure of each channel simultaneously.

The probability that the system fails on a random demand is then given by

$$C + (1 - C) \times (PA)^* \times (PB)^*$$

where $(.)^*$ denotes the mean value of the posterior distribution.

(Strictly speaking, $(P_A)^*$ is the conditional distribution given that A is not certain to fail, and $(P_B)^*$ the conditional distribution given that B is not certain to be imperfect. This technical subtlety does not affect the use to which we put this estimate below.)

This is a particularly simple formula, and it is applicable, as follows.

An Illustration

There are claim limits established in specific safety-relevant or -critical engineering domains; for example, in the atomic power industry in Great Britain there is a claim limit of 10^{-5} for the probability of common-cause faults per operating hour (ophour). This means that one may not claim a lower probability than this for common-cause faults in a fit-for-purpose certification exercise. This gives us a lowest value for C of 10^{-5} in the above formula.

To estimate $(PA)^*$, observe that it is possible to estimate it to $O(10^{-4})$ per op-hour through using statistically-valid random testing ("statistically-valid" means here that test case selection probabilities are exactly those that are (to be) encountered during operations, and that the oracle, the device which tells whether a test case has passed or failed, is perfect) [5]. However, it may be easier to obtain an estimate more like $O(10^{-3})$ in many cases, so let us suppose that we have obtained an estimate for $(P_A)^*$ of 10^{-3}.

To estimate $(P_B)^*$ is a little more unusual. The posterior probability that Channel B is imperfect is affected by the assessment criteria used for Channel B. For example,

(I) The theorem prover, model checker, or whatever other tools used to perform rigorous checks in II may be unsound. (II) The requirements may be misunderstood. However, this event is included in the estimate of C and thus plays no further role here. (III) The requirements, use assumptions, and design may be formulated incompletely or incorrectly. This is further divided into three cases: (a) the specification is inconsistent (then no system can be built to this spec!); (b) elements of the specification may be wrong (although the spec is consistent); (c) the formal specification and verification has discontinuities in it, or is otherwise incomplete.

One can argue that an assessor can form a posteriori judgements about these probabilities which would be adequate to estimate $(PB)^*$ to values in the region of 10^{-3}. (To reach this figure from an estimate of the likelihood of II, III, recall we are assuming conservatively that, if Channel B is imperfect, it will fail when Channel A fails.)

Suppose, then, that we have achieved a posteriori estimates such as these for $(P_A)^*$ and $(P_B)^*$. It follows that the probability of dangerous failure of the system is given by

$$C + (1 - C) \times (PA)^* \times (PB)^*$$

that is

$$10^{-5} + (1 - 10^{-5}) \times 10^{-3} \times 10^{-3} \sim 1.1 \times 10 - 5$$

We have thereby constructed a high degree of confidence in the dangerous-failure-freeness of a 1oo2 system by combining two lower degrees of confidence that attach to each channel separately.

Bibliography

[1] R. Butler and G. Finelli, *The Infeasibility of Quantifying the Reliability of Life-Critical Real-Time Software*, IEEE Trans. Soft. Eng., vol. 19(1), January 1993.

[2] International Electrotechnical Commission, IEC 61508, *Functional safety of electrical/electronic/programmable electronic safety-related systems*, 7 parts, 2010.

[3] Peter Bernard Ladkin, *An Overview of IEC 61508 on E/E/PE Functional Safety*. Available at http://www.causalis.com/IEC61508FunctionalSafety.pdf , Causalis Limited, 2008.

[4] Bev Littlewood and John Rushby, *Reasoning about the Reliability of Diverse Two-Channel Systems in which One Channel is "Possibly Perfect"*, IEEE Transactions on Software Engineering, 38(5), pp. 1178–1194, 2011.

[5] Bev Littlewood and Lorenzo Strigini, *Validation of Ultra-High Dependability for Software-based Systems*, Communications of the ACM, vol. 36(11), pp. 69-80, November 1993.