City Research Online

City St George's, University of London

Citation: Littlewood, B. (2000). The use of proofs in diversity arguments. IEEE Transactions on Software Engineering, 26(10), pp. 1022-1023. doi: 10.1109/32.879822

This is the unspecified version of the paper.

This version of the publication may differ from the final published version. To cite this item please consult the publisher's version.

Permanent repository link: https://openaccess.city.ac.uk/id/eprint/1616/

Link to published version: https://doi.org/10.1109/32.879822

Copyright and Reuse: Copyright and Moral Rights remain with the author(s) and/or copyright holders. Copies of full items can be used for personal research or study, educational, or not-for-profit purposes without prior permission or charge, unless otherwise indicated, provided that the authors, title and full bibliographic details are credited, a hyperlink and/or URL is given for the original metadata page and the content is not changed in any way. For full details of reuse please refer to City Research Online policy.

City Research Online: http://openaccess.city.ac.uk/ publications@citystgeorges.ac.uk

The use of proof in diversity arguments

Bev Littlewood

Centre for Software Reliability, City University, London

Abstract

The limits to the reliability that can be claimed for a design-diverse fault-tolerant system are mainly determined by the dependence that must be expected in the failure behaviours of the different versions: claims for *independence* between version failure processes are not believable. In this note we examine a different approach, in which a simple secondary system is used as a back-up to a more complex primary. The secondary system is sufficiently simple that claims for its perfection (with respect to design faults) are possible, but there is not complete certainty about such perfection. It is shown that assessment of the reliability of the overall fault-tolerant system in this case may take advantage of claims for independence that are more plausible than those involved in design diversity.

Index Terms: Software fault tolerance, reliability, safety, probability, verification, proof

Background

It is common in several industries to have a form of diversity or redundancy in which the different components or versions have different levels of trust placed in them. In some cases a highly functional primary system is backed up by a simple secondary procedure. For example, the present UK air traffic control system can revert to manual operation, involving paper records of aircraft movements, in the event of certain types of computer failure. Some computerised fly-by-wire aircraft flight control systems have a series of successively more degraded modes of operation, providing less and less functionality [Rouquet and Traverse 1986]. In the case of the UK Sizewell B nuclear reactor, the protection system has two elements: a computerised Primary Protection System (PPS), and a hard-wired Secondary Protection System (SPS) [Hunns and Wainwright 1991].

All these examples have in common one or more processes providing high functionality at the price of complexity, backed up by simpler (but less functionally capable) processes. The design principle behind systems like these seems sensible, but its efficacy will clearly vary from one application to another. It seems to require that some procedures that are inherently simpler and/or less extensive than those provided for normal operation will be sufficient to provide safe operation, possibly for a short time only, in the event of failure of a primary system.

The extensive extra functionality of the primary systems in these examples is present, of course, for good reason. The increasing use of software-based systems, in particular, seems to provide opportunities for novel functionality which, in addition to

improving efficiency, sometimes increases safety and reliability in new ways. Thus the software-based Sizewell PPS is more complex than earlier non-computerised systems partly because it provides novel safety advantages (e.g. via the provision of extensive built-in hardware self-testing capabilities).

As systems increase in complexity, however, it becomes harder to ensure that they do not contain residual design faults that can cause failure during operation. Indeed, in all the cases mentioned above the systems are of such complexity that it must be assumed that they will indeed contain design faults. Assessment of reliability therefore forms an important part of the safety cases of the wider systems of which they are part. Such assessment is difficult in a design-diverse fault-tolerant architecture because claims for statistical independence between the version failure processes do not seem sustainable [Eckhardt and Lee 1985; Knight and Leveson 1986; Littlewood 1996; Littlewood and Miller 1989].

In the case of the Sizewell protection system, this assessment problem seems to have been addressed via a 'conservative' argument based upon claim limits. The required reliability for the overall system was 10^{-7} probability of failure on demand (*pfd*). A claim of 10^{-3} *pfd* could be justified for the PPS alone. Whilst it could not be assumed that the PPS and SPS would exhibit independent failure behaviour, the relative simplicity of the SPS meant that a claim for it of 10^{-4} *pfd* was regarded as very conservative. Thus the claim of 10^{-7} *pfd* for the overall protection system was really a trade-off between excessive conservatism in a claim for a version reliability (the SPS) and dependence between version failures.

The difficulty with an argument of this kind is that the trade-off is taking place between two very different things - version reliability and dependence. It assumes that the *degree* of conservatism in the former is sufficient to compensate for the 'amount' of dependence. This in turn seems to imply that the latter is known (or could be estimated conservatively).

In the next section a different kind of argument for this situation is examined - one in which the simplicity of the SPS allows probabilistic claims for its perfection.

The possibility of perfection

If we allow the possibility that a system is completely free of design faults - 'perfect' - we are accepting that it may never fail in its entire lifetime[1]. The evidence for such a claim, of course, is different in nature from the evidence of a claim for high reliability. It is this differing nature of evidence that gives rise to such apparently paradoxical assertions as: 'I might believe you if you claim that this system has a zero failure rate, but I would never believe a claim that it has a 10^{-9} probability of failure per hour'. The evidence for the first claim is logical in nature: it is a claim based upon simplicity, and involving proof. In the second case, on the other hand, the presence of design faults is acknowledged, and impracticable quantities of evidence would be required for a claim that the impact of these faults is infinitesimal.

In this section we consider whether it is possible to take advantage of logical evidence in our probabilistic claims for design-diverse, fault-tolerant systems. Consider as an example a safety system, like the Sizewell protection system, comprising an

[1] Due to design faults - there may still be failures for other reasons, e.g. because of failed hardware components.

extensively functional PPS and a simple SPS. The latter is assumed to be sufficiently simple that a claim for perfection is feasible: specifically we are prepared to make a claim for the *probability of perfection* of the SPS. Such a claim might be supported, for example, by a claim that the formal specification really does accurately reflect the engineering requirements (because of the simplicity of these), together with a formal verification that the implementation is a correct translation of this specification.

In such a case it would be conservative to say that, in the event that the SPS is *not* perfect, it will always fail precisely in those circumstances where the PPS fails - in other words, there is then 'complete' dependence. We have, therefore, the following:

$$\begin{aligned} & P(\text{safety system fails}) \\ &= P(\text{safety system fails}|\text{SPS perfect})\,P(\text{SPS perfect}) \\ &+ P(\text{safety system fails}|\text{SPS not perfect})P(\text{SPS not perfect}) \\ &= P(\text{PPS fails}|\text{SPS not perfect})P(\text{SPS not perfect}) \end{aligned} \quad (1)$$

If we were prepared to assume that imperfection of the SPS and failure of the PPS were statistically independent, we have

$$P(\text{PPS fails}|\text{SPS not perfect}) = P(\text{PPS fails}) \quad (2)$$

and so

$$P(\text{safety system fails}) = P(\text{PPS fails})\,P(\text{SPS not perfect}) \quad (3)$$

Thus, if we were able to claim a reliability for the PPS of 10^{-3} *pfd*, we could obtain our required system reliability of 10^{-7} *pfd* by claiming a probability of 10^{-4} that the SPS is not perfect.

Superficially, the crucial independence argument here is similar to the naïve argument used to claim that the *pfd* of a 1-out-of-2 system is the product of the individual *pfd*s. In fact, though, there are two important differences. In the first place, the assumption of independence that is made in (2) seems more plausible than the independence of version *failure processes.* In the latter case, the argument for the inevitability of positive dependence in [Eckhardt and Lee 1985] seems compelling: there will be such dependence so long as there is variation of 'difficulty' over the input space. Here, on the other hand, the relationship between the PPS *failure process* and the fallibility of the SPS *proof process* seems much more tenuous. Even if independence cannot be claimed, it may be possible to make a conservative claim for the conditional probability of failure, $P(\text{PPS fails}|\text{SPS not perfect})$.

The second difference between the two arguments lies in requiring here an estimate of a probability of perfection, or correctness. It is certainly true that we have no direct means of measuring such a probability, in contrast to the SPS *pfd*, which is measurable in principle from operational testing. On the other hand, such direct measurement is only rarely carried out, even for safety-critical systems; instead an assessment is based upon expert judgement involving, for example, evidence from static analysis and quality of the production process. It may be possible here, similarly, for an expert to make a conservative claim for probability of perfection.

Discussion

The idea of backing up a primary system with a much simpler secondary system is an old one. It seems a plausible means of *achieving* a high overall system reliability. Here we have looked at the contribution that such an architecture makes to reliability *evaluation.*

The approach looks promising as a means of making believable quantitative claims for systems, as part of plant safety cases. It is possible to build into the argument some conservatism - e.g. that in the event that the SPS is *not perfect* its failure always coincides with PPS failures. It may also be possible to be conservative in estimating the key conditional probability, *P*(PPS fails|SPS not perfect), rather than assuming independence as was done above. On the other hand, the estimation of the probability that the SPS is not perfect poses some problems.

However, at the very least, the problems associated with this kind of argument seem less profound than those associated with conventional reliability analysis of a fault-tolerant system. There the key difficulty is in estimating the *dependence* between the version failure processes, since independence here is *never* believable: this estimation problem seems very difficult.

Notice that an implication from (3) is that a perfect SPS implies a perfect safety system - failures are impossible because the SPS always works correctly. In these extreme circumstances, we can cease to treat the PPS as a critical system. Its function, however, is not unimportant: in fact the expectation is that demands will almost always be met by the PPS because of its superior functionality, and only rarely will the simpler 'inferior' SPS be called upon.

Although an *assured* perfection of the SPS is unlikely to be a practical proposition in real applications, the observation nevertheless reveals how the architecture described here differs from a more conventional 1-out-of-2 system, where there is often a near symmetry of treatment of the two versions. Reactor protection systems seem ideal candidates for the approach outlined here, since the protection function is intrinsically fairly simple. Thus the SPS can be made simple, with the hope that it can be proven correct, and the PPS can have extensive functionality, since its reliability requirement is sufficiently modest to be demonstrated, e.g. by direct evaluation from operational testing.

In nuclear protection, as elsewhere, there is a tendency towards building systems with the extensive functionality and other benefits that only software can provide. This trend is coupled with one of backing up the primary (software-based) system with a software-based secondary [Temelin 1993]. In such cases it may be better to use the very asymmetric architecture - and its associated safety and reliability argument - discussed above, than the more common solution of two fully functional diverse subsystems.

Acknowledgements

This work was supported partially by Scottish Nuclear under the QUARC and DISPO projects, and by EPSRC under the DISCS project.

References

[Eckhardt and Lee 1985] D.E. Eckhardt and L.D. Lee, "A Theoretical Basis of Multiversion Software Subject to Coincident Errors," *IEEE Trans. on Software Engineering*, vol. 11, pp.1511-1517, 1985.

[Hunns and Wainwright 1991] D.M. Hunns and N. Wainwright, "Software-based protection for Sizewell B: the regulator's perspective," *Nuclear Engineering International*, vol. September, pp.38-40, 1991.

[Knight and Leveson 1986] J.C. Knight and N.G. Leveson, "Experimental evaluation of the assumption of independence in multiversion software," *IEEE Trans Software Engineering*, vol. 12, no. 1, pp.96-109, 1986.

[Littlewood 1996] B. Littlewood, "The impact of diversity upon common mode failures," *Reliability Engineering and System Safety*, vol. 51, no. 1, pp.101-113, 1996.

[Littlewood and Miller 1989] B. Littlewood and D.R. Miller, "Conceptual Modelling of Coincident Failures in Multi-Version Software," *IEEE Trans on Software Engineering*, vol. 15, no. 12, pp.1596-1614, 1989.

[Rouquet and Traverse 1986] J.C. Rouquet and P.J. Traverse. "Safe and reliable computing on board the Airbus and ATR aircraft," in *Safecomp: 5th IFAC Workshop on Safety of Computer Control Systems*, Pergamon Press, 1986.

[Temelin 1993] Temelin, "Modernising the Temelin VVER power plant," *Modern Power Systems*, pp.34-37, 1993.