Littlewood, B., Popov, P. T. & Strigini, L. (1999). A note on reliability estimation of functionally diverse systems. Reliability Engineering & System Safety, 66(1), 93 - 95. doi: 10.1016/S0951-8320(99)00014-9 <http://dx.doi.org/10.1016/S0951-8320(99)00014-9>

A note on reliability estimation of functionally diverse systems

Bev Littlewood, Peter Popov, Lorenzo Strigini

Centre for Software Reliability, City University, Northampton Square, London EC1V 0HB, UK

Abstract

It has been argued that *functional diversity* might be a plausible means of claiming independence of failures between two versions of a system. We present a model of functional diversity, in the spirit of earlier models of diversity such as those of Eckhardt and Lee, and Hughes. In terms of the model, we show that claims for independence between functionally diverse systems seem rather unrealistic. Instead, it seems likely that functionally diverse systems will exhibit positively correlated failures, and thus will be less reliable than an assumption of independence would suggest. The result does not, of course, suggest that functional diversity is not worthwhile; instead, it places upon the evaluator of such a system the onus to estimate the degree of dependence so as to evaluate the reliability of the system.

Key words: Functional diversity; design diversity; system reliability

Background

There is now a considerable literature on probabilistic conceptual models for design diversity and its impact upon the dependence of system failures [Eckhardt & Lee 1985, Hughes 1987, Littlewood & Miller 1989, Littlewood 1996].

Whilst the earliest of these, by Eckhardt and Lee, dealt with diversity in multi-version software, an essentially identical model for hardware systems was developed independently by Hughes. The key idea in both cases is that the demands placed upon systems by their environment (i.e. a wider system) can vary in 'difficulty', and that this variation induces dependence upon the failure processes of different 'diverse' versions. Very informally, we say that observing version *A* to fail causes us to believe that the demand was probably a 'difficult' one, *and thus increases the chance that* B *will fail*. It is shown that the greater the variation in 'difficulty' - which is given a formal definition within the model - the greater the dependence in their failure behaviour. Indeed, they will fail independently *only* in the case where there is no variation of difficulty at all: a situation which might reasonably be claimed never to occur in practice.

In this model, using the software terminology, we can think of the versions as having been developed 'independently' in the sense that there was no communication between the development teams, nor any overriding design authority (apart, of course, from a common specification). Thus all the differences between the developed versions will have arisen 'willy-nilly'. The later models extended this basic idea to incorporate a notion of 'forced diversity', where a higher authority imposes upon the teams specific

differences in the way that they work: for example, the use of different programming languages, different testing regimes, etc. The intuitive reasoning here is that forced differences in the design process might induce the faults in the versions to be different in kind. In the ideal case, the kinds of faults that would enter version *A* would not be present in version *B*, and *vice-versa*. In the terminology of 'difficulty', those inputs that were difficult for *A* would tend not to be difficult for *B*, and *vice-versa*.

Functional diversity carries the idea of forced diversity a stage further. Whilst forced diversity makes the designs of the versions different, each version still receives the same inputs. In functional diversity, on the other hand, there is a deliberate decision to make the inputs different. Thus, for example, a protection system might comprise two versions, one of which makes its trip decision based on temperature inputs, and the other on pressure inputs.[1]

The practical intuition here is straightforward and appealing. Forcing the design teams to use functionally diverse inputs is an attractive way of forcing them to be 'intellectually diverse' in their solutions to the design problem. If the designs are 'very different' in some meaningful way, there is a good chance that they will differ usefully in the faults they contain, and thus tend not to fail together.

Our intention in this note is not to criticise the overall approach of functional diversity: it looks a sensible approach to *achieve* reliability in a diverse system. Rather, we wish to examine *how much* benefit it is reasonable to claim because of the use of functional diversity; in particular, whether it is reasonable to claim that the versions will fail *independently*.

The model

For ease of exposition we shall continue to use the simple example of a protection system using temperature inputs, *t*, and another using pressure, *p*. We shall also continue to use the language of software, but the arguments deployed here apply equally well to more general systems.

The figure below shows a grossly simplified version in which there is only one temperature dimension, and one pressure dimension. A particular demand is then a point (p,t). This generalises without difficulty to the more realistic case where each of *p* and *t* are vectors.

In the spirit of the earlier models, based upon that of Eckhardt and Lee, we have a set, Π, of programs that could be built to take *p* inputs, and a set of programs, $\Tau$ that could be built to take *t* inputs. A particular development of a 'pressure' program can be thought of as the random selection of a program, π, from Π; a development of a 'temperature' program is the random selection of a program, τ, from $\Tau$. These selections are determined by probability measures on the sets Π and $\Tau$.

In operation, demands are selected randomly from the space of all demands with probabilities that are determined by the operational profile: the probability of selecting (p,t) we shall denote by $f(p,t)$. For such a demand, a 'temperature' program will use *t* as input, a 'pressure' program will use *p*.

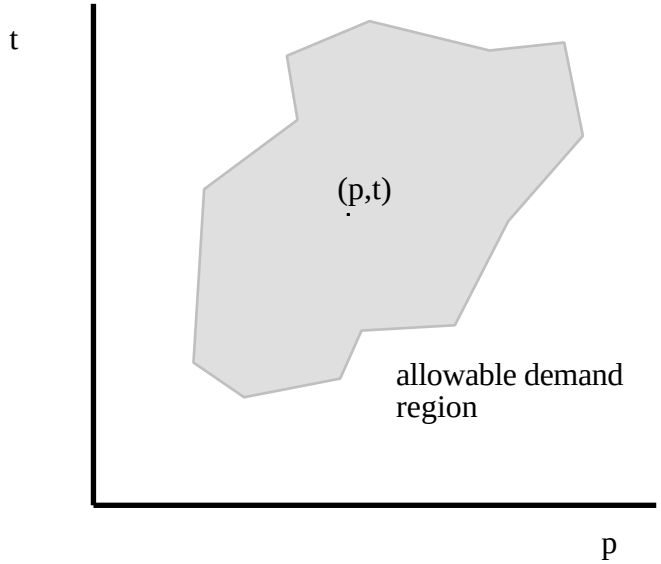
---

[1] We use this simple example only for clarity of exposition, and do not mean to suggest that real systems would have such simple diversity.

We can now define 'difficulty' functions for the *p* inputs, and for the *t* inputs:

$\theta_\pi(p)$ = *P*(randomly selected π fails on *p*)

and similarly for $\theta_\tau(t)$.



Consider now a 1-out-of-2 system built from a randomly chosen 'pressure' program and an independently randomly chosen 'temperature' program. We shall define the *demand* difficulty to be:

$$\theta_{\pi,\tau}(p,t) = P(\text{randomly selected } (\pi,\tau) \text{ fail together on } (p,t)) \quad (1)$$

An interpretation of 'functional diversity' within this model is that there is independence here, i.e. that for every (p,t) the two programs fail independently:

$$\theta_{\pi,\tau}(p,t) = \theta_\pi(p)\theta_\tau(t) \quad (2)$$

This *conditional* failure independence is similar to the conditional independence in the Eckhardt and Lee model, and in the forced diversity model of Littlewood and Miller [Eckhardt & Lee 1985, Littlewood & Miller 1989].

Unfortunately, this independence does not imply that there is *unconditional* independence between version failures. This can be seen as follows.

$$P(\text{randomly selected } (\pi,\tau) \text{ fail together on } \textit{randomly selected } (p,t)) = \mathbf{E}_{p,t}[\theta_{\pi,\tau}(p,t)]$$

$$= \mathbf{E}_{p,t}[\theta_\pi(p)\theta_\tau(t)] = \sum_{p,t} \theta_\pi(p)\theta_\tau(t) f_{p,t}(p,t) \quad (4)$$

and in general this is not equal to $\mathbf{E}_p[\theta_\pi(p)]\mathbf{E}_t[\theta_\tau(t)]$, the product of the individual version probabilities of failure.

There *will* be equality here if

$$f_{p,t}(p,t) = f_p(p) f_t(t) \quad (5)$$

i.e. if in the selection of a demand (p,t) the values of 'pressure' and 'temperature' are independent of one another. However, this seems unlikely ever to be so.

In fact, the reliability of the 1-out-of-2 system will be less than what could be inferred by an assumption of independence of failures between the two versions, so long as

$$\mathbf{E}_{p,t}[\theta_\pi(p)\theta_\tau(t)] > \mathbf{E}_p[\theta_\pi(p)]\mathbf{E}_t[\theta_\tau(t)]$$

i.e.

$$\mathbf{Cov}_{p,t}[\theta_\pi(p),\theta_\tau(t)] > 0\,. \qquad (6)$$

In other words, so long as the difficulty functions are positively correlated, the system will be less reliable than if the versions exhibited failure independence. Conversely, if it can be argued that this correlation is *negative*, the system will be *more* reliable than under version failure independence. Claims for such negative correlation might be hard to justify, however.

Discussion

One interpretation of the result here is that functional diversity is just a kind of forced diversity, similar to that already seen for diverse software versions that execute *the same* inputs as one another: the result here is very similar to that reported in [Littlewood & Miller 1989]. Once again, the result stems from variation of 'difficulty' within the subspaces P and T, and in particular how these difficulty functions are correlated over the allowable set of demands in $P \quad T$.

It seems reasonable to represent the informal claim that 'functional diversity gives independent failures' by conditional independence, (2). This seems to capture the intuitive notion that for each demand, (p,t), the versions are doing such different things that the failure of one does not tell us anything about the chance of failure of the other.[2] Unfortunately, this conditional independence does not imply unconditional independence: it is unconditional independence that is needed to be able to claim that the probability of failure of the system is just the product of the version probabilities of failure for a future randomly selected demand.

Clearly, the process of selecting demands will never result in statistical independence of the version inputs, p and t, (5). Whilst this observation does not rule out the possibility of zero correlation between the version difficulty functions - (5) is sufficient for this but is not necessary - there will clearly be a strong possibility of positive correlation, (6).

It would seem unreasonable simply to *assert* that there is failure independence between versions: the result above shows that this would be a strong assertion about the nature of the different difficulty functions. Similarly, it would be even harder to justify a claim that functional diversity induced negative correlation.

Of course, this reasoning does not suggest that the use of functional diversity is not good practice: on the contrary, it seems likely that it is indeed a good means of *achieving* high reliability. What is not possible, however, is to claim that the use of

---

[2] Notice that if the claim for conditional independence here cannot be supported, then reasoning about the *degree* of dependence will be difficult and most probably will need to take account of our understanding of the process by which faults are generated.

functional diversity is sufficient in itself to justify an assumption of independence in the version failures. It leaves the system assessor with the task of evaluating precisely *how* dependent the versions are before he/she can evaluate the reliability of the system. This is not easy.

In fact this task seems as difficult as evaluating the reliability of the fault-tolerant system from scratch as if it were a black box, since it requires sufficient empirical evidence of coincident failures to enable statistical estimation of the probability of coincident failure. This contrasts with the much easier task we would face if we could assume independence, since in that case only the (comparatively) modest individual version reliabilities need to be estimated.

Acknowledgement

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

References

[Eckhardt & Lee 1985] D. E. Eckhardt and L. D. Lee, "A Theoretical Basis of Multiversion Software Subject to Coincident Errors", *IEEE Trans. on Software Engineering*, 11, pp.1511-7, 1985.

[Hughes 1987] R. P. Hughes, "A new approach to common cause failure", *Reliability Engineering*, 17, pp.211-36, 1987.

[Littlewood 1996] B. Littlewood, "The impact of diversity upon common mode failures", *Reliability Engineering and System Safety*, 51 (1), pp.101-13, 1996.

[Littlewood & Miller 1989] B. Littlewood and D. R. Miller, "Conceptual Modelling of Coincident Failures in Multi-Version Software", *IEEE Trans on Software Engineering*, 15 (12), pp.1596-614, 1989.