Langley Gran[illegible]
IN-61-CR
104541
17P.

SOFTWARE RELIABILITY MODELS FOR FAULT-TOLERANT AVIONICS COMPUTERS AND RELATED TOPICS

(NASA-CR-181440) SOFTWARE RELIABILITY MODELS FOR FAULT-TOLERANT AVIONICS COMPUTERS AND RELATED TOPICS Final Report, 1 Jul. 1981 - 30 Jun. 1987 (George Washington Univ.) 17 p Avail: NTIS HC A03/MF A01 CSCL 09B G3/61
N88-11382
Unclas
0104541

Final Report

National Aeronautics and Space Administration

Grant NAG 1-179

1 July 1981 through 30 June 1987

Douglas R. Miller

Principal Investigator

Department of Operations Research

School of Engineering and Applied Science

The George Washington University

Washington, D.C. 20052

Table of Contents

Abstract

Software reliability research supported by National Aeronautics and Space Administration Grant NAG 1-179 is briefly described. General research topics are reliability growth models, quality of software reliability prediction, the complete monotonicity property of reliability growth, conceptual modelling of software failure behavior, assurance of ultrahigh reliability, and analysis techniques for fault-tolerant systems.

1. INTRODUCTION

Research performed under National Aeronautics and Space Administration Grant NAG 1-179 was primarily concerned with system design flaws. Design flaws may cause a system to behave in unpredicted and undesirable ways. In systems that control aircraft or nuclear reactors, design flaws may cause system failures which are catastrophic. Extreme effort should be undertaken to prevent such events from occurring, through either flawless or fault-tolerant design. Better methods must be developed for approaching this goal and for assessing whether or not particular systems are acceptable.

Much of the research described here is applicable to most systems designed by man; however, our focus is on software in digital computer systems. (An important example is the advanced flight-critical digital avionics which is being introduced into commercial aircraft.) By its nature, software seems especially prone to flaws in design; furthermore, apparently innocuous design flaws can have overwhelming consequences. Under the current state of the art in software design development, flaws can never be ruled out. Instances of perfect software may occur, but they cannot be identified *a priori* with certainty; there is always some degree of *uncertainty* about whether a piece of software is acceptable. How should this uncertainty be handled?

One method of dealing with uncertainty is to use probability models and statistical analysis. Some of the more philosophical members of the statistics community argue that the only logically coherent way to quantify one's notion of uncertainty is with probability. There may be alternatives: Some people seem to be accepting a nonquantitative approach.* Others argue that any uncertainty regarding the perfection of software can be removed using such methods as correctness-proving. We believe that uncertainty exists, and that the scientific way to deal with it must be quantitative, which leads to a statistical approach. We do have doubts about the ability of statistical methods to deal with situations that require ultrahigh reliability. With these thoughts in mind, we have pursued a research program of developing probability models of software failure behavior and statistical methods for software quality assurance.

2. SUMMARY OF RESEARCH

The work of NAG 1-179 mainly focused on statistical methods and models for software reliability. The work is applicable for moderate levels of reliability. It is not clear whether any of this work contributes in a positive way to the assurance of ultrahigh levels of reliability required by digital avionics. Our work relevant to ultrareliability tends to show difficulties and limitations in the assurance of these levels of reliability.

---

*"Software considerations in airborne systems and equipment certification," DO-178A, Special Committee 152, Radio Technical Commission for Aeronautics, Washington, D.C., March 1985.

The effort resulted in 20 research papers, which are listed in Section 4. The research falls into the six general areas briefly described in the following subsections.

2.1 Reliability Growth Models

A system contains design flaws, each of which eventually manifests itself at some time, whereupon the system is redesigned in order to remove the design flaw. If failure times are indexed chronologically they can be represented as

$$0 \le s_1 \le s_2 \le s_3 \le \dots$$

Because there is uncertainty about when failures will occur, these times can be modelled as the realization of a stochastic process:

$$0 \le S_1 \le S_2 \le S_3 \le \dots$$

The process $\{S_n, n = 1,2,3,\dots\}$ is a "reliability growth process." This process can also be represented as a counting process $\{N(t), 0 \le t\}$, where

$$N(t) = \max\{n: S_n(t) \le t\},$$

or as an interfailure time process $\{X_i, i = 1,2,3,\dots\}$, where

$$X_i = S_i - S_{i-1}, \quad i=1,2,3,\dots$$

Numerous stochastic process models have been proposed as "reliability growth" processes.

Modifications to two models, the Jelinsky-Moranda model and the Duane model, have been developed. Littlewood [10] modified the Duane model so that it has a finite intensity at $t = 0$; this makes it more plausible as a failure model. Littlewood and Sofer [14] present a Bayesian version of the Jelinsky-Moranda model; this improves the statistical inference related to the model but it must still be used with great care as a reliability growth model.

A major innovation in reliability growth modelling is the adaptive approach of Keiller and Littlewood [5]. Their idea is to fit any particular reliability growth model to data, make predictions about future data, and then, as future data come in, compare predictions with reality. If past prediction errors show a consistent bias, it is possible to modify future predictions to lessen this bias. These "adaptive" predictors tend to behave at least as well as the original predictors, and often perform significantly better.

A very general class of reliability growth models called exponential order statistic models is presented by Miller [16]. In this case, $\{S_n, n=1,2,3,...\}$ are order statistics of independent, nonidentically distributed exponential random variables. The exponential random variables have rates $\{\lambda_i, i=1,2,3,...\}$. This very general class of models includes many of the standard software relibility growth models as special cases.

The exponential order statistic modelling paradigm yields insight into the modelling of reliability growth. For example, there are virtually no a priori restrictions on the parameter set $\{\lambda_i, i=1,2,3,...\}$; however, for example, the Duane model is equivalent to an EOS model with $\lambda_i \cong bi^{-a}$, $i = 1,2,3,\ldots$. Thus it is difficult to justify a restricted class of models like the Duane model. Similar reasoning applies to the Musa-Okumoto model; see [16]. Some EOS models are useful examples for the study of worst case reliability growth scenarios; see [17].

2.2 Quality of Prediction

Software reliability growth models can be used to estimate and predict various properties of a piece of software that is being redesigned as bugs are discovered. The distribution of the time until the next failure can be estimated. The failure rate of the software can be estimated (assuming no further corrections will be required). The expected number of failures occurring over some finite horizon can be estimated. In all these cases it is important to have some knowledge of the quality of the predictions.

Littlewood has described a "prediction system" as consisting of three components:

(1) Probability models that completely specify the distribution of failure times $\{S_1, S_2, \ldots\}$;

(2) An inference procedure for picking a specific single model for particular observed data $\{s_1, s_2, \ldots\}$;

(3) A prediction mechanism that combines (1) and (2) to give probability statements about future failure times $\{S_{N(t)+1}, S_{N(t)+2}, \ldots\}$.

Predictive quality addresses the performance of the entire prediction system. This is in contrast to the usual "goodness-of-fit" approach, which focuses on models and inference alone. Models that satisfy goodness-of-fit criteria like chi-squared or Kolmogorov-Smirnov tests may still give unacceptable predictions. The focus should be on the quality of the prediction.

Several tools for assessing the quality of prediction have been developed for the case of the distribution of the time until next failure. Keiller, Littlewood, Miller and Sofer [6,7] and Keiller, Littlewood and Sofer [8] describe "u-plots" and "y-plots". Roughly speaking, the u-plot identifies bias in the prediction of time until next failure. The y-plot examines how well the trend (of reliability growth) is captured in the predictive distribution of time until next failure. Abdel-Ghaly, Chan, and Littlewood [1,2] describe the prequential probability ratio (PLR), a statistic which can distinguish between the amounts of noise in different prediction systems. All these methods are used to identify which models have highest predictive quality on any given set of failure data. This use is illustrated using real data.

Keiller and Littlewood [5] use the u-plot quality of prediction tool to modify the predictive distribution of time until next failure. These adapted predictors usually have higher predictive quality than the original predictors.

Some preliminary work assessing absolute quality of estimates of current program failure rate was done by Miller and Sofer [18,19] in a Monte Carlo study; however, this does not give a real-time indication of how well the prediction system is working, which u-plots, y-plots, and PLR do.

2.3 Complete Monotonicity Property

Miller [16] shows that a reasonable characteristic of reliability growth models is a complete monotonicity property for the cumulative mean function. Furthermore, any additional constraints or restrictions on the cumulative mean function may not be justified. In particular, let N(t) equal the number of failure events in [0,t] and let M(t) = EN(t) equal the expected number of events in [0,t]; then

$$(-1)^{n+1} \frac{d^n M(t)}{dt^n} \geq 0;$$

this is equivalent to M'(t) being completely monotone.

Miller and Sofer [18,19,20] use the complete monotonicity property as the basis of model fitting procedures that are generalizations of the method of isotonic regression. Completely monotonic sequences of failure rates are fit to raw empirical failure rates using the criterion of least-squares. The last value in the sequence of completely monotone rates is used as an estimate of the current program failure rate.

The completely monotone characterization has certain ramifications. It becomes difficult to justify the use of single parametric families of reliability growth models, and unreasonable to expect accurate prediction of reliability growth very far into the future.

2.4 Conceptual Modelling of Software Failure Behavior

Mathematical models may serve various purposes. One purpose might be to organize thinking about a system. Another is to obtain qualitative results, such as relative comparisons or inequalities. There are areas of design and behavior of software for which, while quantitative analysis is desirable, a more realistic first step is a descriptive, conceptual modelling of phenomena. In the area of multiversion software, such models were developed by Eckhardt and Lee (IEEE Transactions on Software Engineering, SE-11, (1985): 1511-1517) and then extended by Littlewood and Miller [12,13]. These models describe how programs created independently will exhibit dependent failure patterns: Eckhardt and Lee assumed a common development methodology and obtained positive correlation between failure patterns. Littlewood and Miller show that by using diverse development methods it is possible to get a negative correlation between failure patterns in different versions. In [13] they show that increased diversity between development methods decreases the correlation between failure patterns in different versions. This has ramifications in the n-version approach to software fault-tolerance.

In another instance of conceptual modelling, Littlewood [11] looks at the DFR-mixture closure theorem. He presents an intuitively appealing subjectivist interpretation. It provides some motivation for using DFR interfailure time distributions in reliability growth models.

2.5 Assuring Ultrahigh Reliability

The uncertainty always present regarding the reliability of an item of software leads us inevitably to a statistical analysis of the phenomenon. Indeed, there has been some success using statistical methods to deal with the uncertainty of software quality; however, there appear to be limitations to statistical analysis, especially when ultrahigh reliability is sought. It is difficult to prove that such a negative position is correct, but the lack of progress in finding assurance methods for ultrahigh reliability seems to support this position. Miller [17] discussed the problem of statistical assurance of ultrareliability in a paper presented to the American Statistical Association.

2.6 Analysis Techniques for Fault-Tolerant Systems

The work performed under NAG 1-179 included development of some methods useful in analyzing fault-tolerant systems. These methods involve efficient estimation and calculation procedures for performance measures of fault-tolerant systems.

Arsham and Miller [3] developed an extension of the Kolmogorov-Smirnov confidence interval procedure for estimating a distribution function. The confidence intervals are narrower in one tail of the distribution than in the other. Such confidence intervals are useful for estimating the distribution of fault-coverage times in fault-tolerant systems.

Gross and Miller [4] present the randomization numerical technique for calculating state probabilities of transient Markov chains. Miller [15] specializes the randomization technique for efficient calculation of system degradation and failure probabilities from Markov models of certain fault-tolerant systems.

Kioussis and Miller [9] present an efficient Monte Carlo simulation method for estimation of reliability for fault-tolerant systems. It is a variance reduction technique based on the importance sampling idea of path-splitting. It allows more efficient estimation, with confidence, of small system failure probabilities.

3. CONCLUSIONS AND FUTURE WORK

There are two aspects to the problem of reliable software: achievement and verification. Our research deals with the second aspect. Because of uncertainties in the production and usage of software, statistical models are appropriate. The statistical methods mentioned in this report can play a useful role in assuring software quality. Methods developed thus far are especially useful in situations requiring moderate to high levels of reliability; however, we know of no way to assure ultrahigh levels of reliability (e.g., failure rates of 10^{-9}/hour).

To assure ultrahigh levels of reliability the existing statistical methods are inadequate. Unreasonably large test samples would be necessary; but in most cases even huge samples would not be sufficient because previously negligible factors now become crucial. The exact usage distribution and exact knowledge of interfaces and interactions with the whole system are required. This problem is present for design flaws in any radically new system. There is a limit to the level of reliability in which we can have sufficient confidence.

There is still much work to be done in developing statistical methods for software quality assurance: In the realm of quality of prediction (Section 2.2) absolute measures and confidence interval capabilities are needed. Realistic models of software behavior in real-time control systems are needed; most current models are oriented toward software operated in a batch mode. To understand software failure behavior and software development processes, extensive controlled experimentation with real software projects is required; sophisticated statistical methods will make this more efficient and economical. Methods must be developed to integrate verification information from diverse sources: test data for the product, success and failure data of the development process on related products, expert opinion about quality, etc.

4. RESEARCH PAPERS

[1] A. A. Abdel-Ghaly, P. Y. Chan, and B. Littlewood. "Evaluation of competing software reliability predictions," IEEE Transactions on Software Engineering, SE-12 (1986): 950-967.

[2] A. A. Abdel-Ghaly, P. Y. Chan, and B. Littlewood. "Tools for the analysis of the accuracy of software reliability predictions." In Software System Design Methods, NATO ASI Series, F22 (J. K. Skwirzynski, Ed.), Springer-Verlag, Heidelberg, 1986: 299-335.

[3] H. Arsham and D. R. Miller. "A Poisson process approximation for generalized K-S confidence regions," Statistica Neerlandica, 39 (1985): 291-302.

[4] D. Gross and D. R. Miller. "The randomization technique as a modelling tool and solution procedure for transient Markov processes," Operations Research, 32 (1984): 343-361.

[5] P. A. Keiller and B. Littlewood. "Adaptive software reliability modelling," Proceedings of the Fourteenth International Conference on Fault-Tolerant Computing, IEEE Computer Society, Silver Spring, MD (1984): 108-113.

[6] P. A. Keiller, B. Littlewood, D. R. Miller and A. Sofer. "On the quality of software reliability prediction." In Electronic Systems Effectiveness and Life Cycle Costing, NATO ASI Series, F3 (J. K. Skwirzynski, Ed.), Springer-Verlag, Heidelberg, 1983: 441-460.

[7] P. A. Keiller, B. Littlewood, D. R. Miller and A. Sofer. "Comparison of software reliability predictions," Proceedings of the Thirteenth International Symposium on Fault-Tolerant Computing, IEEE Computer Society, Silver Spring, MD (1983): 128-134.

[8] P. A. Keiller, B. Littlewood and A. Sofer. "Predictive quality of software reliability models," Proceedings of the Fourth Jerusalem Conference on Information Technology, IEEE Computer Society, Silver Spring, MD (1984): 216-225.

[9] L. C. Kioussis and D. R. Miller. "An importance sampling scheme for simulating the degradation and failure of complex

systems during finite missions," Proceedings of the 1983 Winter Simulation Conference. IEEE, New York (1983): 631-639.

[10] B. Littlewood. "Rationale for a modified Duane model," IEEE Transactions on Reliability, R-33 (1984): 157-159.

[11] B. Littlewood. "Subjective probability and the DFR-mixture closure theorem," Communications in Statistics, Theory and Methods, 13 (1984): 859-863.

[12] B. Littlewood and D. R. Miller. "A conceptual model of multiversion software," Proceedings of the Seventeenth International Symposium on Fault-Tolerant Computing, IEEE Computer Society, Washington, DC (1987): 150-155.

[13] B. Littlewood and D. R. Miller. "A conceptual model of the effect of diverse methodologies on coincident failures in multi-version software," Proceedings of Third International Conference on Fault-Tolerant Computing Systems, Bremerhaven, September 1987.

[14] B. Littlewood and A. Sofer. "A Bayesian modification to the Jelinski-Moranda software reliability growth model," Software Engineering Journal, 2:2 (March 1987): 30-41.

[15] D. R. Miller. "Reliability calculation using randomization for Markovian fault-tolerant computing systems," Proceedings of Thirteenth International Symposium on Fault-Tolerant Computing, IEEE Computer Society, Silver Spring, MD (1983): 284-289.

[16] D. R. Miller. "Exponential order statistic models of software reliability growth," CR-3909, National Aeronautics and Space Administration (July 1985). Abridged version: IEEE

Transactions on Software Engineering, SE-12 (1986): 12-24.

[17] D. R. Miller. "Making statistical inferences about software reliability," Appendix of Final Report for NASA Grant NAG 1-179.

[18] D. R. Miller and A. Sofer. "Completely monotone regression estimates of software failure rates," Proceedings of Eighth International Conference on Software Engineering, IEEE Computer Society, Washington, DC (1985): 343-348.

[19] D. R. Miller and A. Sofer. "A nonparametric approach to software reliability using complete monotonicity." In Software Reliability: State of the Art Report 14:2 (A. Bendell and P. Mellor, Eds.), Pergamon Infotech, London, 1986: 183-195.

[20] D. R. Miller and A. Sofer. "Least-squares regression under convexity and higher-order difference constraints with application to software reliability." In Advances in Order Restricted Statistical Inference (R. L. Dykstra, T. Robertson, and F. T. Wright, Eds.), Springer-Verlag, New York, 1986: 91-124.

5. APPENDIX: "Making Statistical Inferences About Software Reliability"

This paper was an invited talk at the Joint Statistical Meetings (American Statistical Association, Institute of Mathematical Statistics, and the Biometric Society), Chicago, August 1986. It was also printed in Software Reliability and Metrics Newsletter, Centre for Software Reliability, London, December 1986.