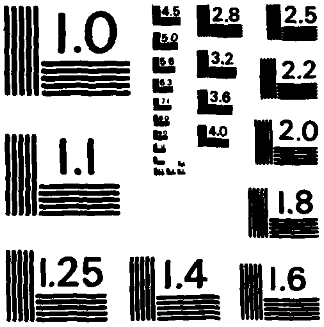
AD-A160 234 RELIABILITY EVALUATION OF FAULT-TOLERANT MULTIPROCESSOR 1/1
SYSTEMS(U) DUKE UNIV DURHAM NC DEPT OF COMPUTER SCIENCE
K S TRIVEDI 14 MAY 85 AFOSR-TR-85-0783 AFOSR-84-0132

UNCLASSIFIED F/G 9/2 NL

END
FILMED
DTIC



MICROCOPY RESOLUTION TEST CHART

NATIONAL BUREAU OF STANDARDS-1963-A

2

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION
Unclassified

1b. RESTRICTIVE MARKINGS

2a. SECURITY CLASSIFICATION AUTHORITY

2b. DECLASSIFICATION/DOWNGRADING SCHEDULE
N/A

3. DISTRIBUTION/AVAILABILITY OF REPORT
Approved for public release; distribution unlimited

4. PERFORMING ORGANIZATION REPORT NUMBER(S)

5. MONITORING ORGANIZATION REPORT NUMBER(S)
AFOSR-TR- 85-0783

6a. NAME OF PERFORMING ORGANIZATION
Duke University

6b. OFFICE SYMBOL *(If applicable)*

7a. NAME OF MONITORING ORGANIZATION
AFOSR

:. ADDRESS *(City, State and ZIP Code)*
Department of Computer Science
Durham, NC 27706

7b. ADDRESS *(City, State and ZIP Code)*
Bldg. 410
Bolling AFB, D.C. 20332-6448

. NAME OF FUNDING/SPONSORING ORGANIZATION
AFOSR

8b. OFFICE SYMBOL *(If applicable)*
NM

9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
AFOSR-84-0132

ADDRESS *(City, State and ZIP Code)*
Bldg. 410
Bolling AFB, D.C. 20332-6448

10. SOURCE OF FUNDING NOS.

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNIT NO.
61102F	2304	A5	

TITLE *(Include Security Classification)*
Reliability Evalu of Fault-Tolerant Multiprocessor Systems

PERSONAL AUTHOR(S)
Kishor S. Trivedi

TYPE OF REPORT
Interim

13b. TIME COVERED
FROM ________ TO ________

14. DATE OF REPORT *(Yr., Mo., Day)*
14 May 1985

15. PAGE COUNT
6

16. SUPPLEMENTARY NOTATION

17. COSATI CODES

FIELD	GROUP	SUB. GR.
XX	XXXXXXXXX	XXXX

18. SUBJECT TERMS *(Continue on reverse if necessary and identify by block number)*
Model construction, model reduction, fault trees, PMS notation, Extended Stochastic Petri Nets

19. ABSTRACT *(Continue on reverse if necessary and identify by block number)*

The major issues involved in modeling modern computer systems can be broadly classified into those arising form themodel construction, model reduction and solution, and in the interpretation of the model solution. Modeling languages such as fault trees, the PMS notation, and Extended Stochastic Petri Nets can be valuable in simplifyning the task of model construction. The goal of the languages is to provide well difined constructs to the user and let the modeling package automatically generate the details of the underlying stochastic model. The language constructs should correspond closely to the system constructs and yet should produce a concise representation.
Specifying the relevent deatils of the system being modeled can require a tremendous number of states to be considered (in excess of 100,000). Techniques must be developed to reduce the model to one that is computationally tractable, and then to solve the reduced model in a computationally efficient manner. Once the solution is obtained, it must be interpreted carefully. The errors introduced by the model reduction step and in the (cont)

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT
UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT. ☒ DTIC USERS □

21. ABSTRACT SECURITY CLASSIFICATION
Unclassified

22a. NAME OF RESPONSIBLE INDIVIDUAL
Brian W. Woodruff, Maj, USAF

22b. TELEPHONE NUMBER *(Include Area Code)*
(202) 767-5027

22c. OFFICE SYMBOL
NM

Block #19 cont.

solution must be bounded, and sensitivity of the solution with respect to input parameters should be estimated.
We have made considerable progress under th e auspices of this grant in both model construction techniques and model reduction and solution techniques.

AFOSR-TR- . 783

Interim Technical Report
AFOSR-84-132
RELIABILITY EVALUATION OF FAULT-TOLERANT MULTIPROCESSOR SYSTEMS

Kishor S. Trivedi

Principal Investigator
Department of Computer Science
Duke University

AFOSR-84-0132

1. Introduction

The major issues involved in modeling modern computer systems can be broadly classified into those arising from the model construction, model reduction and solution, and in the interpretation of the model solution. Modeling languages such as fault trees,[1] the PMS notation,[2] and Extended Stochastic Petri Nets[3] can be valuable in simplifying the task of model construction. The goal of the languages is to provide well defined constructs to the user and let the modeling package automatically generate the details of the underlying stochastic model. The language constructs should correspond closely to the system constructs, and yet should produce a concise representation.[4]

Specifying the relevent details of the system being modeled can require a tremendous number of states to be considered (in excess of 100,000). Techniques must be developed to reduce the model to one that is computationally tractable, and then to solve the reduced model in a computationally efficient manner. Once the solution is obtained, it must be interpreted carefully. The errors introduced by the model reduction step and in the solution must be bounded, and sensitivity of the solution with respect to input parameters should be estimated.

We have made considerable progress under the auspices of this grant in both model construction techniques and model reduction and solution techniques. This progress will be outlined in the next two sections.

2. Model Construction

Three sets of inputs are necessary to construct a reliability model including the system structure and fault-occurrence behavior, and the fault and error handling behavior. The description of the system structure (the set of resources, their interconnections and the conditions under which the system is operational) and the fault-occurrence behavior determine the structure of the dependability model.

Fault trees are often used to specify the conditions under which a system fails, and by implication, the set of resources and their interconnections. A fault tree is a logical diagram that describes the various combinations of events that lead to the undesirable top event, system failure. The top event is divided into its consitituent events (subsystem failures), which are then similarly subdivided. The lower level events are connected to the higher level events by the means of Boolean logic gates. The lowest level events are called basic events, and usually correspond to the failure of components. Reliability block diagrams are similar to fault trees in that they are simple to understand and construct, but where a fault tree is a 'failure' diagram, the reliability block diagram is a 'success' diagram.[2] Each component or subsystem is represented by a block; the logical dependencies are represented by connections between the blocks. Each path between the ends represents a configuration that leaves the system operational. One major drawback to both fault trees and reliability block diagrams is that they are 'static' diagrams; they are not designed to model dynamically reconfigurable systems for example.

More general system structure characteristics can be modeled with state transition diagrams. In this framework, every possible state of the system must be enumerated and classified, as well as the transitions between the states. If the transition rates are constant with time, then the resulting state transition diagram is a Markov chain.[5] The constant transition rates imply that the time spent in each state is exponentially distributed. If the transitions between the states depend on the time spent in the individual state, then the resulting chain may be semi-Markovian.[6] Semi-Markov processes allow the time spent in each state to be generally distributed; this generality makes the solution of all but the smallest models difficult. If the distributions of the holding time in each state are limited to exponential polynomials (a very minor restriction), it can be solved much more easily.[7, 8]

A PMS (processor-menory-switch) diagram is a higher level description of the structure of the system; it shows more explicitly the components and their physical interconnections. A PMS diagram is often accompanied by a set of 'assertions,' a listing of the requirements that must be fulfilled for the system to be operational. The PMS diagram must be 'translated' into another form before the system can be analyzed.[2]

Performance analysts may prefer to represent the system in terms of a queueing network[9] with two service centers, one corresponding to the failure process and the other corresponding to the repair process. The major advantage of this approach is that the performance analyst can form the model in a familiar language. Also, a great deal of study has been performed on queueing networks.

Yet another powerful tool for describing the system structure is the extended stochastic Petri net (ESPN).[3] The ESPN is especially useful for modeling systems that exhibit asynchronous concurrent activities, and is more general than the other languages mentioned.[10] The major drawback to using ESPN's is that it may be difficult for the analyst unfamiliar with the intricasies of ESPN's to develop a correct representation of the system. However, its generality allows us to develop model reduction and solution methods for the ESPN, with the knowledge that the techniques are applicable to the other model types. Further, it permits us to study the relationships between the different model types. An ESPN model can serve as an 'intermediate' language for comparison of the versatility and ease of specification of the other model types. A comparison of the different modeling languages is necessary to objectively determine the pros and cons of each one, and to investigate their ranges of applicability. This may allow us to define new constructs for the other languages, to increase their modeling power and applicability. Then each analyst may continue to operate within a familiar and comfortable environment, without sacrificing versatility or speed.

The last set of inputs required for a dependability model pertain to the behavior of the system upon the occurrence of a fault. This part of the model, called the fault/error-handling model (also called a coverage model) may include such behavior as fault and error detection, transient recovery and automatic reconfiguration. Several different models have been developed to represent this behavior, most of which are included in the *HARP* reliability prediction package.[11, 12] A major difference between modeling system structure and fault occurrence and modeling fault/error handling is that specific *languages* have been developed for the former, while specific *models* have been developed for the latter. The concept of modeling coverage[13, 14] is a fairly recent one whose importance is just beginning to be appreciated.

Under the current Air Force grant, we have made important advances in defining modeling languages, such as the Extended Stochastic Petri Net (ESPN)[15] as discussed in this section. A new system is under development that allows the hierarchical definition of models. Each subsystem can be specified and combined by using fault trees, reliability block diagrams, Markov chains, semi-Markov processes and/or stochastic precedence graphs.

3. Model Reduction and Solution

Once the complete description of the system being modeled is generated, and the fault/error handling behavior is described, the resulting model is often too large and complex to solve. Another problem that will frequently arise is stiffness: competing events whose time constants differ by many orders of magnitude. Stiffness causes difficulties in both numerical and simulative solutions. We can often redude the model to one that is more tractable, by exploiting the characteristic that makes the model stiff. Informally, we can decompose the model into two submodels, one that represents the 'fast' behavior and the other the 'slow.' These two models may be solved separately, and their solutions be aggregated into the overall model solution.

One such model reduction technique that is often used divides the model into distinct fault-occurrence and fault/error-handling models. This technique, termed ***behavioral decomposition*** has been utilized in CARE II,[16] CARE III[17] and HARP.[18, 19] The fault/error-handling model is solved in (semi-) isolation for coverage factors, which are then combined with the system structure and fault arrival information for solution of the overall model.

Another technique for the reduction of the overall model has been presented by Bobbio and Trivedi.[20] They present an approximation algorithm for systematically converting a stiff markov chain into a non-stiff chain with a smaller state space. This method works on the matrix representation of the Markov chain, rather than interpreting the underlying behavior of the system being modeled. Obviously, the problem of model reduction needs to be studied and extended to other model types, and applicability of the various techniques must be investigated. We are continuing a serious study of decomposition/aggregation methods applicable to large, stiff Markov reliability and availability models.

When the model is reduced to an acceptable size, the most appropriate solution technique must be chosen. An analytic solution is desirable since it is often the fastest and the most efficient. A combinatorial solution is often used when the input is specified in terms of a fault tree or reliability block disgram. To predict the reliability of a system at some time t, this solution method considers the combinations of events that cause the system to fail (or remain operational) and assign a probability to each combination. A more general combinatorial method has been implemented in SPADE[7] in which t remains symbolic. The system does not need to be re-solved for each value of t for which the solution is desired. Also, the times of interest may be generally distributed (exponential polynomials). The SPADE solution method is applicable to fault trees and reliability block diagrams. In fact, it is applicable to any system that can be specified as a directed acyclic graph.

Recently, we have developed a general model that allows subsystems to be specified as fault trees, reliability block diagrams, stochastic precedence graphs and/or semi-Markov processes. The solution method developed earlier for SPADE extends to such a hybrid model and combines the efficiency of combinatorial approaches and the versatility of a Markovian approach.

A markov chain produces a set of ordinary differential equations

$$P'(t) = P(t)A(t) \qquad\qquad P(0) = P_I$$

where $P(t)$ is the probability vector for operational states and $A(t)$ is the associated matrix of (possibly) time dependent transition rates. This analytic model is then solved numerically for the state probabilities $P_i(t)$. The reliability or availability of the system is then given by the sum of state probabilities for operational states. In a reliability model the failure states are absorbing, while for an availability model, repair can cause a transition from a failure state to an operational state. We have generally used a Runge-Kutta Fehlberg type quadrature routine to solve the set of equations associated with a Markov chain, but have recently begun serious study of numerical methods more suitable for the specific kinds of matrices associated with stochastic systems.

In order to analytically combine the study of performance and reliability/availability, a Markov reward process is often used. In these models, a reward (relating the performance of the system to the structure) is associated with each state. Kulkarni, Nicola and Trivedi have proposed a unified model that relates performance and reliability measures for the analysis of fault-tolerant systems.[21] The solution of the reward process is given in terms of double transforms (one for the time variable, and the second for the reward variable). Since analytical inversion is not tractable, they resort to a hybrid analytical-numerical approach for the inversion of the double transforms. The computational procedure involves the numerical evaluation of the roots of a polynomial followed by an analytic inversion with respect to the Laplace transform variable. The Laplace-Stieltjes inversion is then carried out numerically.[22]

Often a simulative solution is preferable to an analytic one, especially if the model includes concurrency (and non-exponential distributions). If the model is phrased in terms of an ESPN, it can be simulated using DEEP (the Duke ESPN Evaluation Package).[10, 15] DEEP provides either a transient (for reliability analysis) or steady state (for availability or performance analysis) solution. The major advantage to simulating a system for solution is the flexibility that is possible. The major disadvantage to simulation arises when trying to solve a stiff system. Many simulation trials are needed if the model includes very rare events.

We are investgating techniques for more efficient simulation of stiff systems, in which the occurrence (or non-occurrence) of rare events is recognized. The rare events can then be forced to occur in the simulation. The statistical analysis of the simulation runs would then "weigh" the results accordingly. Developing techniques for the ESPN model assures us that the techniques are applicable to the other model types, since they are all special cases of an ESPN model.

In many cases neither a simulation model nor a analytic model are sufficient to include all the system aspects in one model. In this case a hybrid model, a judicious combination of simulation and analytic models may be used. HARP is such a hybrid model, since the fault handling model might be simulated, but the aggregated Markov model is solved analytically (numerically). The interface between the hybrid parts of such a model must be designed carefully.

Thus we have made major advances in solving complex perfromability models[23] and in deriving a hybrid combinatorial-Markov model for solving complex realistic models.

4. Interpretation of the Solution

There are errors involved in any type of modeling for system evaluation. It is necessary to identify all assumptions and sources of error in model prediction, define proof procedures to verify or experiments to validate the assumptions. In case the assumptions are not supported by these procedures, either the model needs to be modified or errors in model predictions need to be bounded.

Often the values of the transition rates are known to lie within a certain range of values, with a very high probability. Also there may be a positive (although very small) probability that the initial state of the system does not correspond to the initial state of the model. In these cases we are interested then in the *range* of values between which the reliability lies, rather than a point estimate. Smotherman has devised a technique[24] for converting a complex reliability model to a much simpler model. This simple model can then be used to bound the final result with respect to the parametric sensitivity.

In addition to point estimates of such metrics as availability and mean time to failure, SAVE[9] produces an estimate of the *sensitivity* of the estimate to various input parameters. The user can then have an idea as to which system parameters are most crucial to the operation of the system. Parametric sensitivity measures can also be useful when optimizing a system with respect to reliability, performance, cost, etc.

We are continuing a study of error bounding techniques and sensitivity analysis for complex reliability and availability models.

References

1. R. E. Barlow and H. E. Lambert, "Introduction to Fault Tree Analysis," in *Society for Industrial and Applied Mathematics*, ed. J. B. Fussell and N. D. Singpurwalla, pp. 7-35, 1975.
2. Daniel P. Siewiorek and Robert S. Swarz, *The Theory and Practice of Reliable System Design,* Digital Press, 1982.
3. Joanne Bechta Dugan, Kishor S. Trivedi, Robert Geist, and Victor F. Nicola, "Extended Stochastic Petri Nets: Applications and Analysis," *Performance 84*, December 1984.
4. Kishor Trivedi, "Modeling and Analysis of Fault-Tolerant Systems," in *Proceedings of the International Conference on Modeling Techniques and Tools for Performance Analysis*, Paris, May 1984. Invited Paper.
5. Kishor S. Trivedi, *Probability & Statistics with Reliability, Queuing & Computer Science Applications,* Prentice-Hall, 1982.
6. W. Feller, "On Semi-Markov Processes," in *Proceedings National Academy of Sciences,* vol. 51, pp. 653-659, 1964.
7. Robin Sahner and Kishor Trivedi, "SPADE: A Tool for Performance and Reliability Evaluation," *International Workshop On Techniques and Tools for Performance Analysis*, Sophia Antipolis, France, June 1985.
8. Robin Sahner and Kishor S. Trivedi, "Performance and Reliability Analysis Using Directed Acyclic Graphs," Computer Science Department Technical Report, Duke University, April 1985. Submitted to IEEE Transactions on Software Engineering.
9. Ambuj Goyal, Steve Lavenberg, and Kishor Trivedi, "Probabilistic Modeling of Computer System Availability," *The Conference on Statistical and Computational Problems in Probability Modeling, Annals of Operations Research*, 1986.
10. Joanne Bechta Dugan, *Extended Stochastic Petri Nets: Applications and Analysis,* Ph.D. Dissertation, Department of Electrical Engineering, Duke University, 1984.
11. Kishor Trivedi, "Reliability Evaluation for Fault-Tolerant Systems," in *Mathematical Computer Performance and Reliability*, ed. G. Iazeolla, P. J. Courtois, A. Hordijk, Elsevier Science Publishers, North-Holland, Amsterdam, 1983. Invited Paper.

12. Joanne Bechta Dugan, Kishor S. Trivedi, Mark K. Smotherman, and Robert M. Geist, "The Hybrid Automated Reliability Predictor, Phase 1," Computer Science Department Technical Report, Duke University, April 1985. In preparation.

13. W. G. Bouricius, W. C. Carter, and P. R. Schneider, "Reliability Modeling Techniques for Self-Repairing Computer Systems," in *Proceedings 24th Annual ACM National Conference*, pp. 295-309, 1969.

14. Robert Geist, Kishor Trivedi, Joanne Bechta Dugan, and Mark Smotherman, "Modeling Imperfect Coverage in Fault-Tolerant Systems," in *Proceedings IEEE 14-th Fault Tolerant Computing Symposium*, pp. 77-82, June, 1984.

15. Joanne Bechta Dugan, Andrea Bobbio, Gianfranco Ciardo, and Kishor Trivedi, "The design of a Unified Package for the Solution of Stochastic Petri Net Models," in *Proceedings International Workshop on Timed Petri Nets*, Torino Italy, July, 1985.

16. J. J. Stiffler, "Computer Aided Reliability Estimation," in *Proceedings AIAA/NASA/IEEE/ACM Computers in Aerospace Conference*, November 1977.

17. J. J. Stiffler and L. A. Bryant, *CARE III Phase III Report - Mathematical Description*, NASA Contractor Report 3566, November 1982.

18. Robert Geist, Kishor Trivedi, Joanne Bechta Dugan, and Mark Smotherman, "Design of the Hybrid Automated Reliability Predictor," in *Proceedings IEEE/AIAA 5th Digital Avionics Systems Conference*, November, 1983.

19. Kishor Trivedi, Robert Geist, Mark Smotherman, and Joanne Bechta Dugan, "Hybrid Modeling of Fault-Tolerant Computer Systems," *International Journal of Computers and Electrical Engineering*, 1984. To appear, special issue on "Reliability and Verification of Computing Systems."

20. Andrea Bobbio and Kishor Trivedi, "An Aggregation Technique for the Transient Analysis of Stiff Markov Systems," Computer Science Technical Report Number CS-1984-25, November 1984. Submitted for publication.

21. Vidyadhar Kulkarni, Victor Nicola, and Kishor Trivedi, "On Modeling the Performance and Reliability of Multi-mode Computer Systems," in *Internatinal Workshop on Modeling and Perfomance Evaluation of Parallel Systems*, North Holland, Grenoble, 1985. Invited Paper.

22. Vidyadhar Kulkarni, Victor Nicola, Kishor Trivedi, and Roger Smith, *A Unified Model for the Analysis of Job Completion Time and Performability Measures in Fault-Tolerant Systems*. In review, JACM.

23. Vidyadhar Kulkarni, Victor Nicola, and Kishor Trivedi, "A Unified Model for Performance and Reliability of Fault-Tolerant/Multi-Mode Systems," Technical Report CS-1984-12, Department of Computer Science, Duke University. Submitted for Publication to JACM.

24. Mark Smotherman, Robert Geist, and Kishor Trivedi, "A Methodology of Simple Models," Duke University Department of Computer Science Technical Report CS-1985-2, November 1984.