MEADEP — A Dependability Evaluation Tool for Engineers

Dong Tang, Myron Hecht, Jeffrey Miller and Jady Handal

Keywords: Availability, Data analysis, Evaluation, Markov chain, Parameter estimation, Reliability, Software tool.

Summary & Conclusions

MEADEP is a user-friendly dependability evaluation tool for measurement-based analysis of critical systems. MEADEP consists of four software modules: a data preprocessor for converting data in various formats to the MEADEP format, a data analyzer for graphical data presentation and parameter estimation, a graphical modeling interface for building block diagrams (including the exponential block, Weibull block, and k-out-of-n block) and Markov reward chains, and a model solution module for availability/reliability calculations with graphical parametric analysis. Use of the tool on failure data from measurements provides quantitative evaluations of dependability for the target system, while greatly reducing requirements for specialized skills in data analysis and system modeling from the user. MEADEP has been applied to evaluate availability for two air traffic control systems based on operational failure data and the produced results have provided valuable feedback to the project management of these critical systems. MEADEP has also been used to analyze a nuclear power plant safety model, based on the Eagle 21 architecture and its early field failure data. The study identified the most sensitive parameter and its most sensitive value segment to the plant mean time between hazards.

1. Introduction

Automation of dependability evaluation has been realized by the computer engineering community for over 15 years during which many dependability modeling tools were developed [1, 6]. Some of the representative tools are SAVE [2], SHARPE [8], and UltraSAN [9]. The emergence of these tools has given impetus to the applications of advanced modeling and evaluation techniques. Many theoretical modeling and solution issues such as non-exponential failure arrival/recovery times and numerical stiffness[1] have been addressed. However, practical issues such as data analysis, parameter estimation and graphical user interface (GUI) have rarely been considered in these tools.

Given a tool in the above category and a set of field failure data (typically in a database format), reliability engineers often hesitate to make use of the tool and data to generate a quantitative evaluation of dependability. This is due to the difficulties involved in data processing, parameter estimation (especially for cases where failures are rare), model specification for the tool and appropriate mapping from data to the model, and the lack of GUI features. The need to develop tools that can reduce these difficulties has thus been apparent.

From the viewpoint of engineers, it is desirable to have software tools which integrate data processing, statistical analysis, typical dependability modeling and evaluation,[2] and a user-friendly interface to provide non-expert users with an easy-to-use environment for producing quantitative dependability evaluations for real systems. This paper introduces just such a tool — MEADEP (MEAsure DEPendability)[3]. The purpose of this development was to facilitate the use of measurement-based dependability analysis methods [4, 10] and to reduce the cost of such analyses so that they can become an integral part of engineering projects where dependability is an important consideration.

2. Overview of MEADEP

MEADEP is a failure data based dependability analysis and modeling tool. Dependability measures generated by MEADEP are either directly obtained from data, such as failure rate and event distribution, or evaluated by combined use of failure data and dependability models, such as system level reliability and availability. Thus, two basic types of

[1] In a Markov dependability model, failure rates tend to be very small and recovery rates tend to be much larger. Stiffness means the technical difficulty in model solution caused by the difference between the largest and the smallest parameters in the model.

[2] The exponential distribution is typically assumed in the model evaluation.

[3] The academic version of MEADEP is available to the public at www.sohar.com.

input to MEADEP are:

- Data — structured failure reports containing information on failure time, location, type, impact and other failure characteristics
- Models — graphical specifications of dependability models including serial and parallel reliability blocks (with the exponential or Weibull distribution), weighted blocks, k-out-of-n blocks, and Markov reward chains

The output of MEADEP consists of results obtained from data and results evaluated from models. Results from data include various graphical representations (pie charts and curves) of event distribution, Mean Time Between Events (MTBE) and its confidence interval, histograms for Time Between Events (TBE) and for Time To Recovery (TTR) distributions, with super-plotting of typical analytical functions accompanied by the results of their goodness-of-fit tests, and the mean, lower and upper bounds for failure rate, recovery rate, and coverage. Results obtained from models include: Mean Time Between Failures (MTBF) for repairable and non-repairable systems, steady-state availability, reliability for a given time point, and interval reliability for a time period (average reliability over the period).

Figure 1 is the layout of MEADEP. The *Data PreProcessor* (DPP) module, interacts with the user to convert source data to MEADEP internal data. The source data can be manually generated structured reports, typically in a database format, or computer generated event logs. The *Data Editor and Analyzer* (DEA) module is used to edit internal data and to perform statistical analysis on the data. Parameter values estimated from the data by this module can be inserted into the text modeling file generated by another module, the *Model Generator* (MG). The MG module provides a graphical user interface for the user to draw model diagrams and then to generate, from the diagrams, a text modeling file that contains model specifications suitable for solution. Model diagrams can be imported from library files containing predefined models to save development time. The *Model Evaluator* (ME) module produces results based on the model specifications and parameters in the text modeling file. All modules are integrated with the *Graphical User Interface* (GUI).

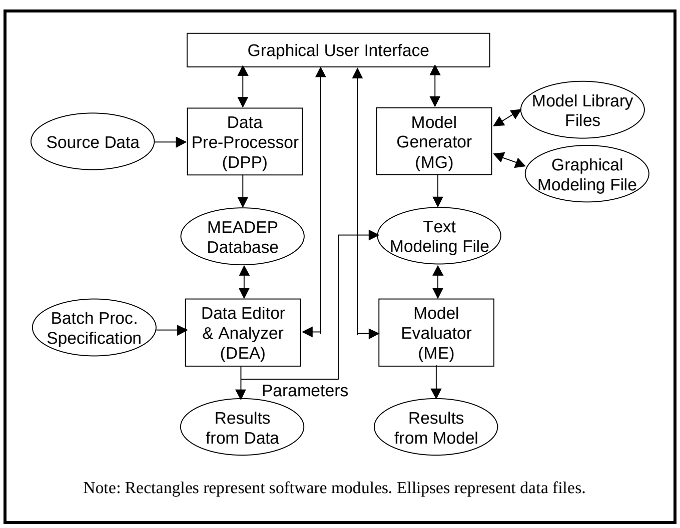

Figure 1 Layout of MEADEP

The DEA module works on data converted by the DPP module and performs statistical analysis. It has three categories of functions: data editing, graphical analysis, and parameter estimation. In addition to individual parameter estimates, it can generate multiple parameters and confidence intervals by processing a query file. The module can also plot, over a histogram, five different analytical probability distribution functions (pdf) determined by the sample mean and sample variance: exponential, gamma, Weibull, normal and lognormal. Meanwhile, the estimated parameters for these functions as well as the results of the Chi-Square and Kolmogorov-Smirnov goodness-of-fit tests [4] are provided on the screen (an example is shown in Figure 6).

The MG module is a graphical "drag and drop" interface for constructing dependability models. A model is developed hierarchically, from the top level to the bottom level, forming a tree-structure. Each node in the tree is a diagram of serial or parallel reliability blocks, a diagram of weighted blocks, or a diagram of Markov reward chain. The user can navigate from one diagram to another to build models. The weighted block diagram is hardly seen in other modeling tools. It was added into MEADEP based on our practical modeling experience [15]. The overall availability of a weighted block diagram is the weighted sum of the availability for each individual block.

The graphical model can be translated into a text modeling file which contains model specifications for directing the ME module to solve the model for results. The ME module has two major functions: editing the text modeling file (editor) and evaluating the model (evaluator). The editor allows the user to revise models and parameters and then to immediately see the effects of revisions on results. The evaluator provides regular results and parametric analysis. For the regular results, the modeling file is evaluated once and the results for all diagrams (models) listed in the specification are generated. In the parametric analysis, the user specifies a loop, and multiple sets of results are generated graphically for one or multiple diagrams.

The implementation of MEADEP integrates techniques in GUI programming, database engineering, dependability modeling, and statistical/numerical analysis. Specifically, MEADEP was developed based on the following software platforms and tools: Microsoft Windows 95/98/NT, Visual C++, Open DataBase Connectivity (ODBC) database engines, IMSL C Numerical Libraries and the Olectra Chart graphical package. Because of the incorporation of these techniques, MEADEP provides many user-friendly features and can interface with various database formats, thereby reducing difficulties in model development and data conversion. The parameter estimation methods and model solution methods used in MEADEP were developed based on algorithms discussed in [5, 7, 16]. For several test cases, including hierarchical models of multiple Markov chains, the steady-state and transient results generated by MEADEP match those generated by SHARPE [8]. More details about MEADEP can be found in [11, 14, 15].

3. Availability Evaluation for ATC Systems

In this and the next section, we discuss MEADEP applications on two categories of systems: (1) continuously operating real-time systems, and (2) on-line protection systems. The operational profiles radically differ for the two categories: continuous input (workload may fluctuate) and intermittent input (rare events). The first category requires high availability and can tolerate component-level failures by redundancy provisions. Computer operating systems, telephone switching and ATC systems all fall into this category. Most previous work on measurement-based dependability evaluation has been for systems in this category as reviewed in [4, 10]. The second category requires successful responses to emergent demands, and a failed response can result in the loss of life and property. The nuclear power safety system is a typical representative of this category.

MEADEP has been used to evaluate availability for two major air traffic control systems: the Voice Switching and Control System (VSCS) and the Air Route Traffic Control Center (ARTCC). The VSCS is a digital communication system responsible for voice switching between pilots and air traffic controllers. The modeled ARTCC components include a set of radars, radar data processing and display subsystems. Both evaluations were based on measurements from field systems and provided valuable feedback to the project management.

For the VSCS, an operational availability model, which is a hierarchy of 23 reliability block and Markov model diagrams, was developed to represent the system. The Program Trouble Report (PTR) data from the first 12 VSCS systems installed in major U.S. air traffic control centers were used to estimate model parameters. A data conversion screen which maps the VSCS PTR data fields that are useful for parameter estimation to the MEADEP data fields and a parameter estimation screen in which a failure rate is to be estimated from the converted VSCS data can be found in [14].

Figure 2 shows the top level diagram of the VSCS operational availability model which consists of four subsystems: the Air-to-Ground switch subsystem (AG), the Ground-to-Ground switch subsystem (GG), the Central Control subsystem (CC), and the Position subsystem (POS). POS is a k-out-of-n block. Each of these blocks is further decomposed to a lower level diagram. Figure 3 shows the AG subsystem diagram which models two redundant air-to-ground channels (AGC) and a pair of redundancy control units (DMC). In the diagram, the two heavy square blocks marked as AGC and DMC represent two sub-diagrams (submodels) of this diagram (the two heavy frames can be expanded to view the lower level diagrams). Parameters λ_{agc} and μ_{agc}, which are used in this model and are also placed in the AGC block, will be evaluated from the lower level diagram AGC. Similarly, parameters λ_{dmc} and μ_{dmc} will be evaluated from diagram DMC. The data used in this evaluation represented unexpected failures and outages in a cumulated period of 2,212 system operational days. The system availability evaluated from the data as of June 30, 1996 was on the level of five 9's (0.99999) and was dominated by software failures.

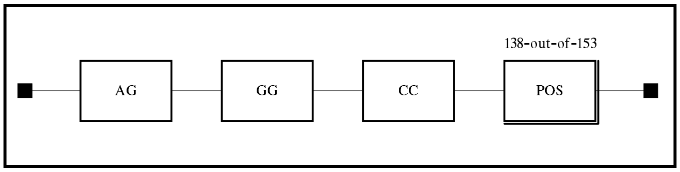

Figure 2 VSCS Top Level Model

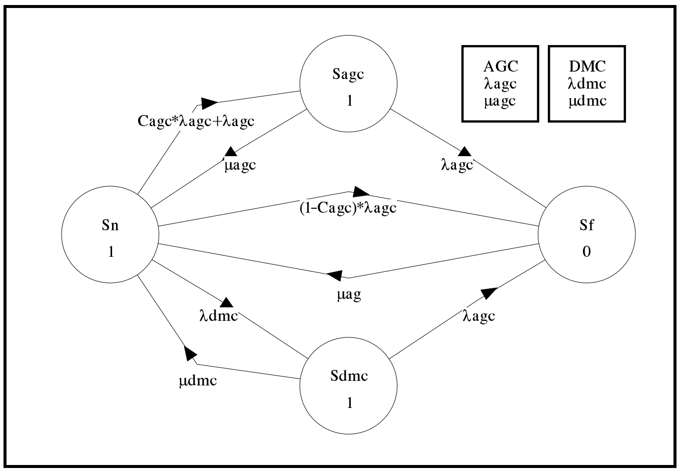

Figure 3 VSCS Air to Ground Switch Subsystem Model

Two lessons can be learned from this evaluation: First, if no major failures occur in the future, it would take 15 years of normal operation for all of the 21 sites to demonstrate an availability of the expected seven 9's at the 80% confidence level, using the upper bound based evaluation approach discussed in [12]. Therefore, in order to quantify availability for systems with such high requirements, it is necessary to introduce accelerated testing and assessment methods. Second, the only VSCS system-level failure occurring in the period was the Seattle VSCS Type I failure (a problem that precludes the primary air traffic control system mission objective of controlling aircraft) of August 11, 1995 when all of the eight air-to-ground telephony switch shelves (for simplicity, AG shelf) failed. The event was widely reported by the media. The problem diagnosis, provided by the manufacturer, identified the likely root cause as an undetected fault in a memory chip. The fault corrupted the length field (set to 0-length) of a message broadcast by an A/G shelf. All other A/G shelves (including primaries and standbys), upon receiving this invalid message, reset and cleared all application code from their processors simultaneously, due to a general protection fault caused by the 0-length. This catastrophic failure was the result of three rare conditions [3]: a memory chip fault, hardware memory error detection function disabled, and software defects in data consistency checking. This event indicates that data errors can be caused by hardware faults, and that software should prevent catastrophic failures from these errors. To target this issue, testing with random data error injection may be a solution.

For the ARTCC system, a hierarchical model consisting of 14 reliability block and Markov model diagrams was developed. One of the two major parts modeled is the Radar System. A submodel in the Radar System is the Radar Coverage Markov Reward Model shown in Figure 4. This model represents 12 radar sites which provide surveillance coverage for a certain airspace. The 12 circular areas covered by these radar sites intersect with each other. If all radar sites are operational (State S_0), there is 100% airspace surveillance coverage and the reward rate is thus 1 (shown in the circle). As the number of failed radars increases, the surveillance coverage, or reward rate, decreases. In the figure, the reward rate for a single radar site failure (S_1) is 0.992, and for two radar site failures (S_2) it is 0.98, etc. Each radar site is modeled by a lower level diagram, Rsite. The failure rate (λ_{rsite}) and recovery rate (μ_{rsite}) of the radar site is evaluated by this lower level diagram, as shown by the square block (which can be expanded to view the lower level diagram) in the figure. The expected airspace surveillance coverage is the availability evaluated from this Markov reward model.

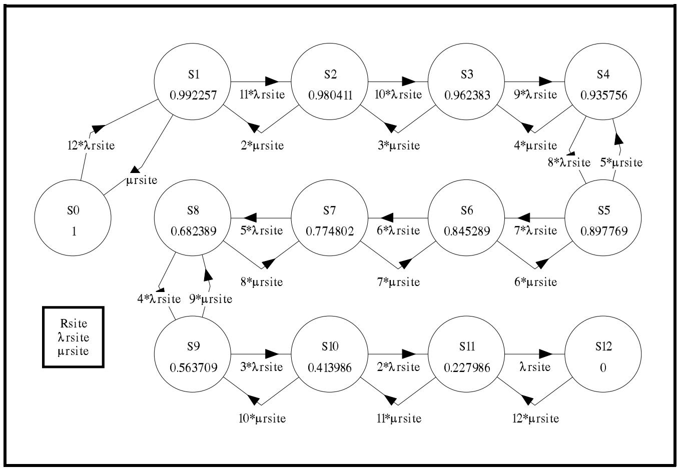

Figure 4 Radar Coverage Markov Reward Model

In addition to the 12 radar sites, the Radar System also includes the Automation Subsystem, which processes information from the radars, and the Display Channel, which displays objects detected by the radars. Figure 5 shows the relative contributions of these three subsystems to the Radar System unavailability. The radar sites are a major availability bottleneck (54%)[4]. However, all of the three subsystems have the same order of availability of three 9's, based on the data that represented both scheduled and unscheduled outages.

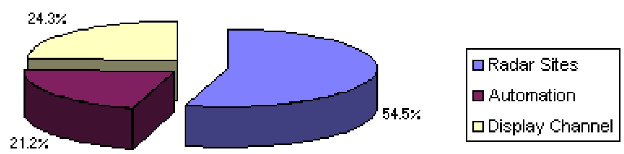

Figure 5 Radar System Unavailability Distribution

Figure 6 shows the distribution histogram of time between outage events for a particular radar site, generated by MEADEP. The curves plotted on the screen are the exponential and Weibull functions. Both functions passed both Chi-Square and Kolmogorov-Smirnov goodness-of-fit tests at the 0.1 significance level.

[4]The Radar Sites availability is performance related. It represents the average space coverage by radars for the area.

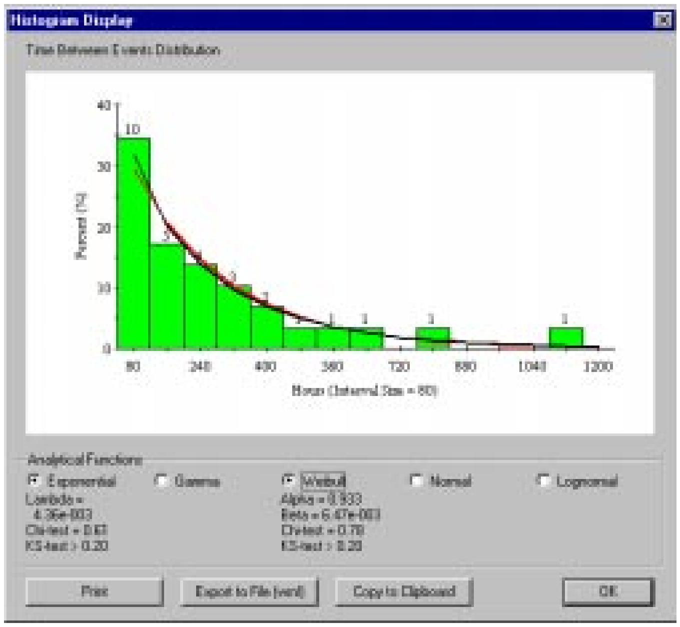

Figure 6 Distribution of Time Between Outages for a Radar Site

4. Sensitivity Analysis for a Nuclear Power Plant Safety System

Now we demonstrate how MEADEP can be used in modeling safety systems. The modeled configuration has two subsystems: a nuclear plant and a digital safety system which protects the plant by responding to and processing challenges from the plant instrumentation. A 3-level hierarchical model was developed for this configuration, where levels 2 and 3 were based on the architecture of a real digital protection system — Eagle 21 [13]. Figure 7 shows the top level plant model which reflects the intermittent operating profile (the heavy frame in this diagram means parameters λ_{ss} and μ_{ss} are evaluated from the lower level model SafSys, similar to the next diagram). Figure 8 shows the middle level model, a safety system which consists of four channels working on a basis of 2-out-of-4 votes for a reactor trip (shutdown). In this model, channel failures are assumed to be of the Byzantine type[5], because this type is the worst case failure mode and is hazardous to the protection function. The bottom level channel model is a diagram of reliability blocks representing the components of a channel.

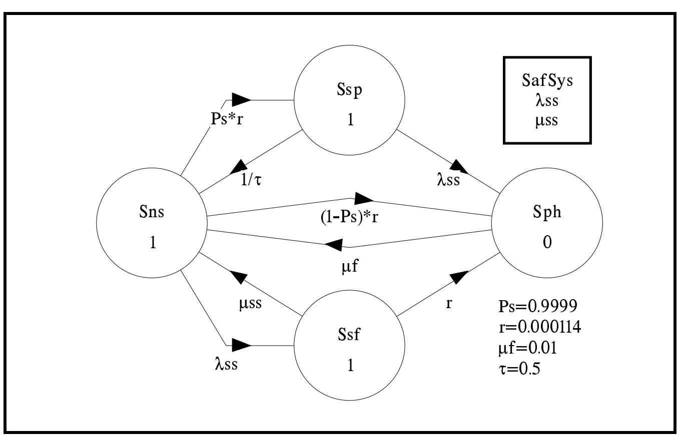

Figure 7 The Nuclear Plant Model

[5]The faulty channel continues execution and lies when asked for information [10].

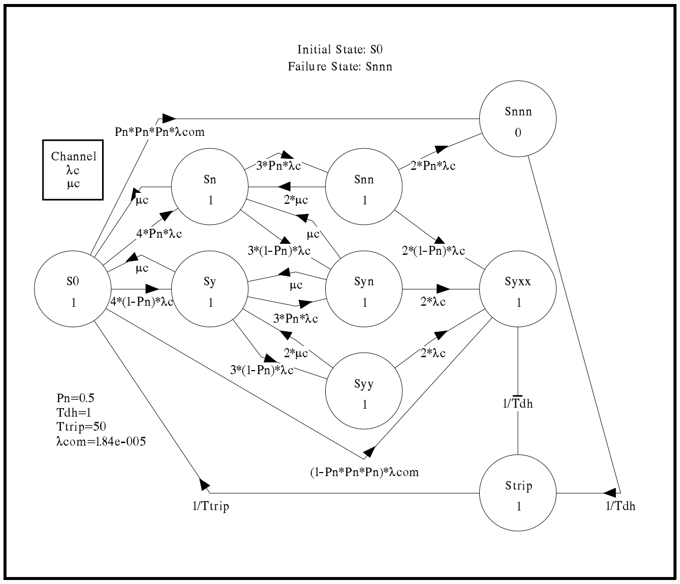

Figure 8 The Safety System Model (SafSys)

The notation used in the figures is as follows:

S_{ns} Normal/safe state in which either both plant and safety system are functioning within technical specifications or the plant is in a safe trip (reactor is shut down safely)

S_{sp} Safety processing state in which the safety system is processing a challenge

S_{sf} Safety failure state in which the safety system is not able to respond to a challenge properly while the plant is functioning within technical specifications

S_{ph} Plant hazard state which is the result of a failure of the safety system to process a challenge successfully in terms of initiating a necessary reactor trip

P_S Probability of success upon demand, i.e., the safety system will be successful in responding a challenge (initially set to 0.9999)

r Arrival rate of challenges from the plant requiring a response of the safety system (assumed to be once a year, a typical value)

τ Challenge processing time (assumed to be a half hour, a conservative assumption)

λ_{ss} Failure rate of the safety system (evaluated from the middle level model)

μ_{ss} Rate for detection and handling of a safety system failure (evaluated from the middle level model)

μ_f Recovery rate of the plant after a hazardous event

S_0 Normal state in which all four channels are functioning properly

S_n State in which one channel has failed and the output of the failed channel votes for “no trip”

S_y State in which one channel has failed and the output of the failed channel votes for “trip”

S_{nn} State in which two channels have failed and both failed channels vote for “no trip”

S_{yn} State in which two channels have failed and one failed channel votes for “trip” and another failed channel votes for “no trip”

S_{yy} State in which two channels have failed and both failed channels vote for “trip”

S_{nnn} State in which at least three channels have failed and at least three failed channels vote for “no trip”; This state is equivalent to state S_{sf} in Figure 7, because the safety system would generate a “no trip” signal should a challenge arrive.

S_{yxx} State in which three channels have failed and at least one of the failed channels votes for “trip”

S_{trip} Plant trip state (reactor is shut down)

P_n Probability that the channel output votes for “no trip”, given a channel failure (assumed to be 0.5)

λ_c Failure rate of a channel (evaluated from the bottom level model)

μ_c Recovery rate of a channel (evaluated from the bottom level model)

λ_{com} Common mode failure rate for the safety system (80% confidence upper bound based on "no common mode failures for 10 years")
T_{dh} Failure detection and handling time, given that at least three channels have failed (assumed to be one hour)
T_{trip} Plant trip duration (assumed to be 50 hours)

The data source for this analysis is the failure reports generated for the early use of Eagle 21 in Unit 1 and Unit 2 at the Sequoyah plant during a 2-year period [13]. The available data furnished the estimation of failure rate upper bounds only for the bottom level model (the channel model). Other parameters were assumed to take typical or conservative values for demonstration purposes. The dependability measure to evaluate in this analysis is the plant Mean Time Between Hazards (MTBH), i.e., the mean time to state S_{ph} (Figure 7), which represents a failure of the safety system to initiate a necessary reactor trip in response to a challenge due to its computer hardware or software faults. The MEADEP parametric analysis functionality was used to investigate the impact of the following three parameters upon the plant MTBH: (1) the safety system common mode failure rate λ_{com}, (2) the safety system failure detection and handling time T_{dh}, and (3) the probability of success upon demand P_S. As each of these parameters was selected for sensitivity study, it was varied within a reasonable range, and all other parameters were left unchanged.

The results showed that the plant MTBH is not very sensitive to λ_{com} and T_{dh}. For the selected parameter ranges, the changes of MTBH are about 8% and 2%. However, the plant MTBH is extremely sensitive to P_S: When P_S increases from 0.999 to 1, MTBH increases from 1000 years to 470,000 years, i.e., an increase of 470 times, as shown in Figure 9 which was generated by MEADEP. The largest increment segment is between 0.99999 and 0.999999 (from 82,500 years to 319,800 years), and achieving a value in this range is the most rewarding. It is clear that the most important parameter is P_S, the probability of success upon demand, and achieving a high value and estimating the achieved value for this parameter should be a key effort in the system development.

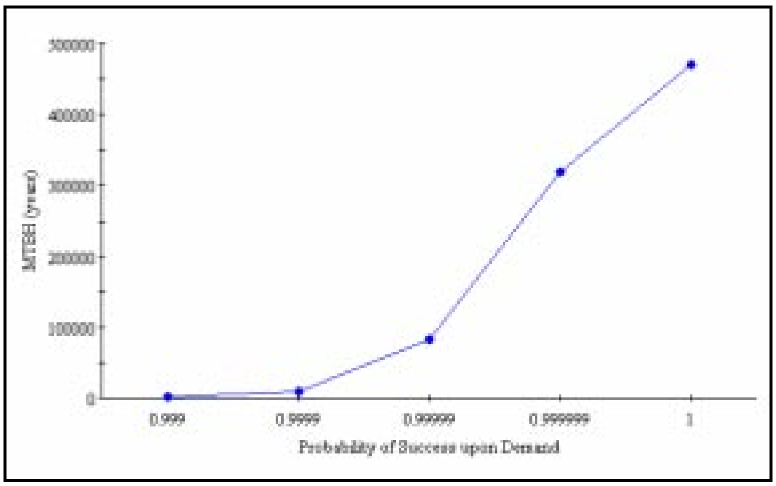

Figure 9 Sensitivity of Plant MTBH to Ps

Acknowledgments

This work was supported partially by the U.S. Nuclear Regulatory Commission under Contract NRC-04-95-081 and partially by the Federal Aviation Administration under Contract DTFA01-93-Y-0169, Subcontract 96-107.

References

[1] R. Geist and K. S. Trivedi, "Reliability Estimation of Fault-Tolerant Systems: Tools and Techniques," *IEEE Computer*, July 1990, pp. 52-61.

[2] A. Goyal, et al., "The System Availability Estimator," *Proceedings of the 16th International Symposium on Fault-Tolerant Computing*, June 1986, pp. 84-89.

[3] H. Hecht, “Rare Conditions — An Important Cause of Failures,” *Proceedings of the 8th Annual Conference on Computer Assurance*, Gaithersburg, MD, June 1993, pp. 81-85.

[4] R. K. Iyer and D. Tang, “Experimental Analysis of Computer System Dependability,” *Fault-Tolerant Computer System Design*, D. K. Pradhan (Ed.), Prentice Hall PTR, Upper Saddle River, NJ, 1996, pp. 282-392.

[5] D. Kececioglu, *Reliability and Life Testing Handbook*, Vol. 1 & 2, PTR Prentice Hall, Englewood Cliffs, NJ, 1993.

[6] J. F. Meyer, “Performability: A Retrospective and Some Points to the Future,” *Performance Evaluation*, Vol. 14, Feb. 1992, pp. 139-156.

[7] A. Reibman and K. S. Trivedi, “Numerical Transient Analysis of Markov Models,” *Computational Operations Research*, Vol. 15, No. 1, 1988, pp. 19-36.

[8] R. A. Sahner and K. S. Trivedi, "Reliability Modeling Using SHARPE", *IEEE Transactions on Reliability*, vol 36, February 1987, pp. 186-193.

[9] W. H. Sanders, W. D. Obal II, M. A. Qureshi, and F. K. Widjanarko, “The UltraSAN Modeling Environment,” *Performance Evaluation*, Vol. 24, No. 1, Oct/Nov 1995, pp. 89-115.

[10] D. P. Siewiorek and R. W. Swarz, *Reliable Computer Systems: Design and Evaluation, Digital Press*, Bedford, Mass., 1992.

[11] SoHaR Incorporated, *MEADEP User’s Manual*, Version 1.6, October 1998.

[12] D. Tang and M. Hecht, “Evaluation of Software Dependability Based on Stability Test Data,” *Proceedings of the 25th International Symposium on Fault-Tolerant Computing*, Pasadena, CA, June 1995, pp. 434-443.

[13] D. Tang, M. Hecht, H. Hecht, and R. Brill, “Measurement-Based Dependability Evaluation for Safety-Grade Digital Systems,” *Proceedings of the 1996 American Nuclear Society International Topical Meeting on Nuclear Plant Instrumentation, Control and Human-Machine Interface Technologies*, May 1996, pp. 535-542.

[14] D. Tang, M. Hecht, J. Handal and L. Czekalski, “MEADEP and Its Applications in Evaluating Dependability for Air Traffic Control Systems,” *Proceedings of the 1998 Annual Reliability and Maintainability Symposium*, Jan. 1998, pp. 195-201.

[15] D. Tang, M. Hecht, A. Rosin and J. Handal, “Experience in Using MEADEP,” *Proceedings of the 1999 Annual Reliability and Maintainability Symposium,* Washington DC, Jan. 18-21, 1999.

[16] K. S. Trivedi, *Probability & Statistics with Reliability, Queuing, and Computer Science Applications*, Prentice-Hall, Englewood Cliffs, NJ, 1982.

Author Biographies

Dong Tang, Ph.D.
SoHaR Incorporated
8421 Wilshire Blvd., Suite 201
Beverly Hills, CA 90211-3204
Phone: (323) 653-4718 ext. 104
Email: tang@sohar.com

Dr. Dong Tang received the PhD degree in computer science from University of Illinois at Urbana-Champaign (1992). He has been a Senior Research Engineer at SoHaR since 1993. He was a Visiting Research Assistant Professor at University of Illinois from 1992 to 1993 and a Research Scientist at the Institute of Computing Technology, Chinese Academy of Sciences from 1983 to 1986. His research interests include performance and dependability measurement, modeling, and evaluation, software reliability engineering, and evaluation tool development. He has been the Principal Investigator for several U.S. government sponsored research projects and has published over 25 journal and conference papers. He was the program chair of the *International Workshop on Computer-Aided Design, Test, and Evaluation for Dependability*, Beijing, July 1996. He is a member of IEEE and International Who’s Who of Information Technology.

Myron Hecht
SoHaR Incorporated
8421 Wilshire Blvd., Suite 201
Beverly Hills, CA 90211-3204
Phone: (323) 653-4717 ext. 111
Email: myron@sohar.com

Myron Hecht, co-founder and President of SoHaR, has worked in the areas of software and systems reliability. He directs SoHaR's support of Federal Aviation Administration in current and future air traffic control systems. He has also been active in nuclear reactor safety systems including development of programming guidelines for digital safety systems and the development of reliability analysis techniques. His activities in basic research and development at SoHaR have resulted in new architectures for real time distributed systems, methodologies for the development and verification of fault tolerant software, and he has also developed designs for highly reliable distributed systems for both process control and C^3I.

Jeffrey E. Miller
SoHaR Incorporated
133 Rollins Ave., Suite 5
Rockville, MD 20852
Phone: (202) 314-1355
Email: miller@sohar.com

Jeff Miller is a senior systems engineer at SoHaR Incorporated where he has been engaged in reliability modeling and analysis, fault tolerant design and development of reliability expert system tools. Prior to joining SoHaR in 1985, he had over 15 years experience in circuit and systems design for avionics and display imaging equipment while at Northrop, Xerox and RCA. He holds a BS degree in Electrical Engineering from UCLA, holds several patents, and is a member of the IEEE Reliability Society.

Jady Handal, Ph.D.
Jady Handal, Ph.D.
Federal Aviation Administration
Room 339
800 Independence Ave. SW
Washington D.C. 20591
Phone: (202) 267-3241
Email: jhandal@hq.faa.gov

Dr. Jady Handal received Ph.D. in Spectroscopy at Fordham University in 1977. He has held management and senior staff positions in Software and Systems Engineering at General Electric, Lockheed and Hughes Aircraft Company. He has worked in the areas of satellite and terrestrial communications, air traffic control and air defense systems. Dr. Handal is currently a member of the senior staff in the Federal Aviation Administration National Infrastructure Management System Program Office. He is developing a Full Context Enterprise Performance Analysis System that allows the FAA to make value-centered risk-based acquisition and operating decisions, on a facility by facility basis, for over 36,000 systems that comprise the National Airspace System (NAS). This enterprise modeling system ties Cost of Service to Quality of Service taking into account infrastructure failure rates, maintenance staffing, training and sparing. Return On Asset and Return On Investment are then used to determine Value of Service. Value of Service will drive infrastructure acquisition and operating budget decisions.