International Journal of Performability Engineering Vol. 3, No. 4, October 2007, pp. 467-480.

Printed in India

System Availability Analysis Considering Failure Severities

SWAPNA S. GOKHALE[1*], JOHN R. CRIGLER[2], WILLIAM H. FARR[2]

[1]*Dept. of Computer Science and Engineering*
University of Connecticut, Storrs, CT 06269.
[2]*Naval Surface Warfare Center Dahlgren Division (NSWCDD)*
Dahlgren, VA 22448

(Received on June 12, 2006)

Abstract: Model-based analysis is commonly used to assess the influence of different factors on system availability. Most of the availability models reported in the literature consider the impact of redundancy, fault tolerance, and system structure. However, these models treat all system failures to be equivalent or at the same level of severity. In practice, it is well-known that failures are classified into multiple severity levels according to their impact on the system's ability to deliver its services. System availability is thus influenced by only some rather than all failures. To obtain an accurate availability estimate it is then necessary to incorporate failure severities into the analysis. In this paper we present a system availability model which considers failure severities of the hardware and software components of the system in an integrated manner. Based on the model we obtain closed form expressions which relate system availability to the failure and repair parameters of the hardware and software components comprising the system. For a given choice of failure parameters, we discuss how the closed form expressions could be used to select the repair parameters to achieve specified target system availability and to establish bounds on system availability. We illustrate the potential of the model by applying it to the failure data collected during the acceptance testing of a satellite system.

Key Words: *System availability, Failure severities*

1. Introduction and motivation

The growing dependence of our society on computer systems places a heavy premium on their reliable operation. Reliability is a key metric for many life-critical systems that must operate without failure for a given period of time. Many systems, however, can tolerate some failures and can continue to operate, perhaps in a degraded mode. Also, even when a failure causes total loss of service, repair of the underlying failure can restore system operation. For such repairable systems, and for those which operate in a degraded mode, availability is a more relevant metric than reliability [1]. System availability is influenced by many factors such as the structural organization of components, level of redundancy, component failure and repair distributions and their parameters. Model-based analysis is commonly used to analyze the impact of such interdependent factors on system availability and reliability [2,3].

*Communicating Author's Email: ssg@engr.uconn.edu

Model-based availability analysis has been applied to many systems in the literature [4,5,6,7,8,9,10,11]. Most of these availability models consider degraded modes of operation arising out of redundancy and fault tolerance [10] where in the degraded modes system performance is usually lower than that of a fully operational system. The notion of "performability" has been defined to capture this interplay between performance and availability in the different modes of operation [12]. Most of the availability models ignore a system's ability to tolerate lower severity failures and still provide a smaller, essential subset of services even without redundancy. Thus, implicitly, they regard all the failures to be equivalent in terms of their consequences on the system's capabilities. Failures are usually classified into multiple severities based on their impact on the system, where the highest severity ones result in a complete loss of service and critical services can be provided despite lower severity failures. For example, a formatting failure in an account report of an online banking system does not preclude essential functions such as deposits and withdrawals. In such cases, in all the senses, the system can be considered to be available. This makes it necessary to consider failure severities to provide an accurate estimate of system availability.

It is undoubtedly important to estimate system availability and the probability of degraded mode considering failure severities for given failure and repair parameters. Prior to system deployment, however, it may be necessary to determine the repair parameters for specific failure parameters estimated at the end of testing. These values of the repair parameters must be determined based upon the target availability and probability of degraded mode. Furthermore, for known failure parameters, it may be prudent to estimate the bounds on availability and probability of degraded mode prior to entering into service level agreements with the users. Thus, the analysis methodology should not only provide an availability estimate, but also provide quantitative and systematic guidance to make informed decisions.

In this paper we present an availability model which considers failure severities of the hardware and software components of a system in an integrated manner. The model also incorporates component repairs. We derive expressions to relate availability and probability of degraded mode to the failure and the repair parameters of the components. Further, we discuss how the expressions could be used to select appropriate repair parameters to meet the specified availability target and to establish achievable bounds on availability and probability of degraded mode for given failure parameters. We then describe our experience in applying the model using failure and repair data collected during the acceptance testing of the system. Using this data we also illustrate the potential of the model to facilitate different analyses.

The paper is organized as follows: Section 2 describes system characteristics. Section 3 presents the availability model. Section 4 derives expressions for availability and probability of degraded mode and discusses the selection of repair parameters and determination of achievable availability bounds. Section 5 illustrates the model. Section 6 presents conclusions and directions for future research.

Notation:

$\lambda_{h1}(\lambda_{h2})$ – Mean failure rate of severity #1 (severity #2) hardware failures

$\lambda_{s1}(\lambda_{s2})$ – Mean failure rate of severity #1 (severity #2) software failures

$\mu_1(\mu_2)$ – Mean repair rate of severity #1 (severity #2) failures
A(D) – System availability (Probability of degraded mode of operation).
$A_{target}(D_{target})$ – Target system availability (Probability of degraded mode of operation)
$\mu_{1,t}(\mu_{2,t})$ – Target mean repair rate of severity #1 (severity #2) failures
D_b – Upper bound on probability of degraded mode of operation.
A_b – Lower bound on system availability.
$\hat{T}_1$ – Estimate of mean time to failure for severity #1 failures.
$\hat{T}_2$ – Estimate of mean time to failure for severity #2 failures.

2. System description

In this section we describe the characteristics of the system, which we inferred from the raw data consisting of times at which failures occurred. Preprocessing the raw data, each failure was classified according to the component which caused the failure (hardware/software) and its severity (#1 or #2). A severity #1 failure results in a complete loss of service, but a severity #2 failure enables a degraded mode of operation. Our objective is to build a model conforming to the data, so that at least some of the parameters can be estimated directly from the data.

We assume that the system consists of hardware and software components, both of which can fail. In general, although each component can have multiple failure modes [13,14,15], based on the data, we consider only failure two modes. Thus, the system can experience four types of failures, severity #1 and #2 hardware failures and severity #1 and #2 software failures. We assume that both the components fail independently of each other. In addition, we also assume that there is no dependence among the different failure severities of the same component. For example, a severity #2 hardware failure does not increase the likelihood of a severity #1 hardware failure. It is important to note that the hardware (software) component cannot fail when the system is down due to a severity #1 failure of the software (hardware) component. Thus, the hardware (software) component can be considered to be suspended when the system is down due to a software (hardware) failure [16]. Furthermore, a severity #2 failure of a component cannot occur once it is down due to a severity #1 failure.

A severity #1 failure (hardware or software) is catastrophic and results in a complete loss of service, whereas a severity #2 failure (hardware or software) is serious and transitions the system into a degraded mode in which a system can deliver only a small subset of critical services rather than the full suite. The system is restored back to its fully operational state from both the failed and degraded modes. Thus the system can be in three states, fully operational, degraded, and failed. Based on the contribution of component failures and repairs to system availability, the system has a series structure. A fully operational system requires both the components to be operational. When one or both the components operate in a degraded mode, the system operates in a degraded mode. Finally, system failure occurs when either of the components fail. Thus, this is a series system with three modes, a special case of a multi-state series system [17,14,15]. These system characteristics, inferred from the data, were confirmed by experts who are knowledgeable about the system.

3. Availability model

In the availability model, the state of the system is represented by a 2-tuple, where the first and the second elements represent the states of the hardware and software components respectively. For both the components, we let *U*, *2* and *1* denote fully operational, degraded, and failed states. A combination of component states result in a maximum of nine system states. Of these, state *(1,1)* cannot occur, since a severity #1 hardware failure would cause system failure and preclude a severity #1 software failure until system operation is restored. A summary of the component and the resulting system states is in Table 1.

Table 1: Component and system states

Notation	Hardware State	Software State	System State
(U,U)	Operational	Operational	Operational
(2,U)	Degraded	Operational	Degraded
(U,2)	Operational	Degraded	Degraded
(2,2)	Degraded	Degraded	Degraded
(1,U)	Failed	Operational	Failed
(U,1)	Operational	Failed	Failed
(1,2)	Failed	Degraded	Failed
(2,1)	Degraded	Failed	Failed

Consistent with the contemporary reliability and availability models [1], we assume that the time to failure for all four types is exponentially distributed. The time to restore system operation also follows an exponential distribution, with the repair rate dependent on whether the system is restored from a severity #1 or a severity #2 failure. Figure 1 shows the Markov model of the system. The evolution of system states is as follows. The system starts in state *(U,U)* when both the hardware and the software components are fully operational. In this state four events can occur, namely, severity #2 hardware and software failures and severity #1 hardware and software failures. These events transition the system to states *(2,U)*, *(U,2)*, *(1,U)* and *(U,1)* with rates $\lambda_{h2,}$ $\lambda_{s2,}$ λ_{h1} and λ_{s1} respectively. From states *(1,U)* and *(U,1)*, the system may be restored at rate μ_1 to state *(U,U)*. In states *(2,U)* and *(U,2)* multiple events are possible. We will first describe the evolution from state *(2,U)*. From this state, the system may be restored at rate μ_2 to state *(U,U)*. In addition, severity #1 hardware and software failures and a severity #1 software failure occurring at rates $\lambda_{h1,}$ λ_{s1} and λ_{s2} respectively, will transition the system to states *(1,U)*, *(2,1)* and *(2,2)*. Using similar reasoning it can be inferred that the system may transition to states *(U,1)*, *(1,2)* and *(2,2)* from state *(U,2)*. We assume that in the states where both severity #1 and severity #2 failures have occurred, namely, states *(1,2)* and *(2,1)* the restoration rate for severity #1 failures, namely μ_1, is used. We also assume that although states *(1,2)* and *(2,1)* may be reached from *(2,2)*, the system is restored to the fully operational state *(U,U)* rather than state *(2,2)*. Thus, from *(1,2)* and *(2,1)*, the system transitions to *(U,U)* with rate μ_1. Lastly, from state *(2,2)*, the system can transition to states *(1,2)* and *(2,1)* upon the occurrence of severity #1 hardware and software failures with rates λ_{h1} and λ_{s1} respectively. Also from state *(2,2)* the system is restored to state *(U,U)* at rate $\mu_{2.}$

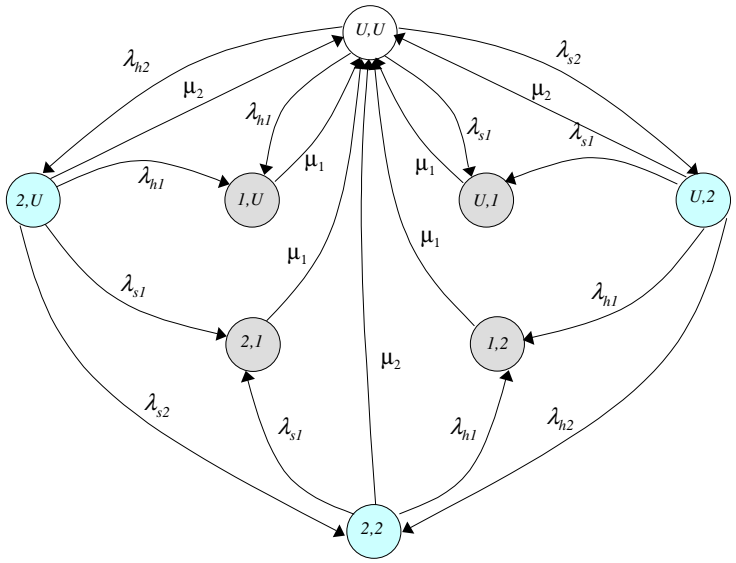

Fig. 1: System availability model with component failure severities

4. Availability analysis

We now discuss how the model in Figure 1 can be used for different types of analyses.

4.1 Availability estimation

The model shown in Figure 1 can be solved to obtain expressions for availability and probability of degraded mode. Referring to Figure 1, the system is fully operational in *(U,U)*, operates in a degraded mode in *(2,U)*, *(U,2)* and *(2,2)*, and is completely failed in *(1,U)*, *(U,1)*, *(1,2)*, *(2,1)*. The availability A and probability of degraded mode D are given by:

$$A = \frac{(\lambda_{h2} + \lambda_{s2} + \lambda_{s1} + \lambda_{h1} + \mu_2)\mu_1}{\mu_1(\lambda_{s1} + \lambda_{h1} + \mu_2) + (\lambda_{s1} + \lambda_{h1} + \mu_1)(\lambda_{s2} + \lambda_{h2}) + (\lambda_{h1} + \lambda_{s1})(\lambda_{s1} + \lambda_{h1} + \mu_2)} \quad (1)$$

$$D = \frac{(\lambda_{h2} + \lambda_{s2})\mu_1}{\mu_1(\lambda_{s1} + \lambda_{h1} + \mu_2) + (\lambda_{s1} + \lambda_{h1} + \mu_1)(\lambda_{s2} + \lambda_{h2}) + (\lambda_{h1} + \lambda_{s1})(\lambda_{s1} + \lambda_{h1} + \mu_2)} \quad (2)$$

In addition to providing an estimate of availability and probability of degraded mode, the analytical expressions in (1) and (2) can facilitate an analysis of the sensitivity of availability to failure and repair parameters. They also enable predictive or "what-if" analysis easily.

4.2 Selection of repair parameters

The users of a system may specify desired levels of availability and probability of degraded mode, which must be achieved by the system provider. One of the ways of achieving these targets may be to reduce the failure rates. However, when the system is ready to be deployed, the failure rates of the hardware and software components are usually fixed. These estimates of the software and hardware failure rates may be obtained

in several ways. For example, software failure rate may be estimated based on the failure data collected during the testing phase, whereas, the hardware failure rate may be obtained from the manufacturer's data sheets. Since the failure rates are fixed, the system provider has to achieve the availability objectives by an appropriate selection of the repair parameters. Needless to say, a very high repair rate may allow the provider to achieve the specified targets. However, the higher the repair rate, higher is the level of resources needed. A judicious selection of the repair parameters is thus necessary to achieve the desired availability target in a cost-effective manner. We now discuss how the closed form expressions can be used to select repair parameters to achieve the specified availability target.

Let A_{target} and D_{target} be the target availability and probability of degraded mode. The ratio of (1) and (2) yields:

$$\frac{A_{t\arg et}}{D_{t\arg et}} = \frac{(\lambda_{h2} + \lambda_{s2} + \lambda_{s1} + \lambda_{h1} + \mu_2)}{\lambda_{h2} + \lambda_{s2}} \quad (3)$$

Simplifying (3) and letting $F_1 = \lambda_{h1} + \lambda_{s1}$ and $F_2 = \lambda_{h2} + \lambda_{s2}$, we have:

$$\mu_{2,t} = \frac{A_{t\arg et} * F_2}{D_{t\arg et}} - F_1 - F_2 \quad (4)$$

Using (4) an estimate of the parameter μ_2, denoted $\mu_{2,t}$ may be obtained for target availability and probability of degraded mode for specified failure rates.

Substituting the value of $\mu_{2,t}$ in (2), an estimate of μ_1, denoted $\mu_{1,t}$, is given by:

$$\mu_{1,t} = \frac{D_{t\arg et}(F_1F_2 + F_1(F_1 + \mu_{2,t}))}{D_{t\arg et}(F_1 + F_2 + \mu_{2,t}) - F_1} \quad (5)$$

The above discussion assumes that the values of A_{target} and D_{target} are known. System users may, however, specify bounds on the availability and probability of degraded mode. The specific values of the metrics to satisfy these bounds then need to be determined considering their dependence. These values when substituted in (4) and (5) will provide repair parameters to satisfy these targets and hence the original bounds from which they are derived.

4.3 Availability and degraded mode bounds

In this section we discuss how the expressions could be used to establish bounds on the availability and probability of degraded mode for specific failure rates. In many cases, system users may specify a desired availability target. Since a system is available both when it is operating fully or in a degraded mode, availability and probability of degraded mode are dependent and a certain percentage of the specified availability will be due to operation in a degraded mode. Further, to satisfy a given availability target, the probability of degraded mode needs to be constrained to a certain range. The system provider is responsible for educating the user of this possibility. Also, since the degraded mode is less desirable, the provider should also provide an upper bound on the probability of degraded mode for given failure parameters. This bound can be obtained as follows.

Referring to (4) and imposing the condition $\mu_2 > 0$, provides D_b which is the bound on the probability of degraded mode of operation.

$$D_b \le \frac{A_{target} * F_2}{F_1 + F_2} \quad (6)$$

Since the probability of degraded mode of operation is always greater than 0.0, we have:

$$0.0 < D_b \le \frac{A_{target} * F_2}{F_1 + F_2} \quad (7)$$

Equation (7) thus provides an upper bound on the probability of degraded mode for a given target availability. This implies that the probability of degraded mode will not exceed D_b for repair parameters selected to meet the specified availability target.

Albeit less likely, in some cases the users may specify the desired probability of degraded mode. Once again, based on the dependence between availability and probability of degraded mode, the lower bound on availability A_b is given by:

$$\frac{(F_1 + F_2) D_{target}}{F_2} \le A_b < 1.0 \quad (8)$$

Equation (8) provides a lower availability bound, for a target probability of degraded mode. Thus, if D_{target}, is the probability of degraded mode, availability must be at least A_b.

5. Model application

In this section we discuss our experience in applying the model to the failure data collected during the acceptance testing of a satellite system. We first discuss the issue of parameter estimation. Subsequently, we demonstrate the use of the model for sensitivity analysis, selection of repair parameters, and determination of availability bounds.

5.1 Parameter estimation

The failure parameters (rates of severity #1 and #2 failures) are estimated based on the data collected during the acceptance testing of a satellite system. The data consisted of a sequence of dates on which the failures, classified according to the component and severity, were observed. Table 2 shows the sequence of failure occurrence dates for severity #1 failures. The data indicates that 11 times are available for hardware failures, whereas, only a single time is available for software failures. As a result, the data available to estimate the failure rate of severity #1 software failures is insufficient. To alleviate this, we assume that the failure rates of severity #1 hardware and software failures are identical, and use all the available data to estimate these rates. Based on the sequence of failure times, times to failure are computed and are reported in columns 3 and 6 of Table 2. The severity #1 failure rates of hardware and software components are estimated using (9).

$$\lambda_{h1} = \lambda_{s1} = \frac{1}{\hat{T}_1} \quad (9)$$

where $\hat{T}_1$ is an estimate of the mean time to failure obtained from the failure times in Table 1. The rates of severity #1 failures (hardware and software), that is λ_{h1} and λ_{s1} are 0.0037/day.

Table 2: Failure data for severity #1 failures

Date	Comp.	TTF(days)	Date	Comp.	TTF(days)
6/4/1993	HW		4/19/1994	HW	319
4/22/1994	HW	3	4/23/1994	HW	1
5/1/1994	SW	8	5/8/1994	HW	7
4/26/1996	HW	719	12/1/1996	HW	219
1/9/1997	HW	39	10/29/1997	HW	293
10/27/1998	HW	363	3/31/2002	HW	1251

Table 3: Failure data for severity #2 failures

Date	Comp.	TTF(days)	Date	Comp.	TTF(days)
4/15/1994	HW		4/16/1994	HW	1
4/28/1994	HW	12	5/2/1994	HW	4
5/16/1994	SW	14	5/18/1994	HW	2
6/17/1994	HW	30	6/20/1994	HW	3
7/16/1994	HW	26	7/18/1994	HW	2
7/24/1994	HW	6	7/29/1994	HW	5
7/30/1994	HW	1	7/30/1994	HW	1
8/18/1994	HW	19	9/15/1994	HW	28
11/1/1994	HW	47	12/1/1994	HW	30
12/14/1994	HW	13	1/18/1995	HW	35
1/31/1995	HW	13	2/21/1995	HW	21
3/24/1995	HW	31	3/25/1995	HW	1
4/1/1995	HW	7	9/22/1995	HW	174
10/30/1995	HW	38	1/11/1996	HW	73
1/29/1996	HW	18	8/25/1996	HW	209
10/1/1996	HW	37	2/21/1997	HW	143
2/25/1997	HW	4	7/14/1997	HW	139
12/16/1997	HW	155	1/31/1999	HW	411
8/9/2000	HW	556	9/2/2001	HW	389
3/1/2002	HW	180			

Table 3 shows the sequence of failure times for severity #2 failures. The data indicates that 38 failure times are available for hardware, whereas, only one time is available for software. As a result, similar to severity #1 failures, we assume that the failure rates of severity #2 hardware and software failures are identical, and use all the available failure times to estimate these rates. Using a procedure similar to severity #1 failures, the failure rates of severity #2 hardware and software failures are estimated using (10):

$$\lambda_{h2} = \lambda_{s2} = \frac{1}{\hat{T}_2} \quad (10)$$

where $\hat{T}_2$ is an estimate of the mean time to failure obtained from failure times in Table 3. The rates of severity #2 failures (hardware and software), that is, λ_{h2} and λ_{s2} are 0.0132/day.

Equation (10) assumes that severity #1 and severity #2 failures of a component are independent. However, under the assumption that a severity #2 failure cannot occur once the system is down due to a severity #1 failure, the failure rate of severity #2 failures can be estimated as follows. Let D_1, D_2, … be the failure times of severity #1 failures. Let R_1, R_2,… be the times at which the system is restored back into operation from a severity #1 failure. The relative positions of failure and restoration instants along the time axis may be as shown in Figure 2. Thus, the system is up during the intervals $(0,D_1)$, (D_1,R_2),…. , and down during the intervals (D_1,R_1), (D_2,R_2)….. During the interval when the system is up, one or more severity #2 failures may occur and thus the system may enter into a degraded mode and be restored back to the fully operational state several times. At the time of occurrence of a severity #1 failure, two possibilities with respect to severity #2 failures arise. In the first case, the system is fully operational. In the second case, the repair of a severity #2 failure is in progress and the system is operating in a degraded mode. The likelihood function for an interval in the first case may be written according to right censored (suspended) data [18], whereas for an interval where the second case holds, the likelihood function may be developed as per complete data. The joint likelihood function for all the observation intervals can then be obtained as a product of the likelihoods in the individual intervals. This joint likelihood function can then be used to estimate the failure rate of severity #2 failures of the component. The above parameter estimation procedure requires both failure and repair times for severity #1 failures. Since the repair times are not available, it cannot be applied to the data at hand.

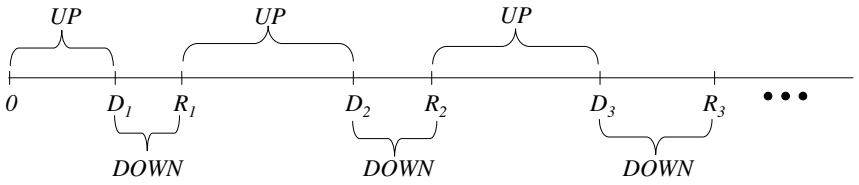

Fig. 2: Relative positions of failure and restoration times

5.2 Sensitivity analysis

The failure parameters are estimated based on the data collected during the acceptance testing of a satellite system. However, adequate data were not available to estimate the repair parameters. Such a situation arises very often in practice where only some parameters can be estimated from real data, while the remaining ones have to be "guestimated". In such cases, it is valuable to assess the sensitivity of the system availability to the parameters that could not be estimated from real data. Sensitivity analysis can be used to generate a sensitivity graph to establish a quantitative relationship between the system availability and model parameters. In this section, we illustrate through two experiments, how the expressions for availability and probability of degraded mode given by (1) and (2) can be used for sensitivity analysis.

In the first experiment, we set the repair rate of severity #1 failures, namely, μ_1 to 10, 50, 100, 500, 1000, 2500, and 5000 times their failure rate. The values of A and D were computed for each value of μ_1 using (1) and (2). The repair rate of severity #2 failures, namely, μ_2 was set to 10 times the failure rate of severity #1 failures. Figure 2 shows the availability A and the probability of degraded mode D a function of μ_1. It can be observed

that availability drops sharply when the repair rate drops from 50 to 10 times of the failure rate of severity #1 failures. Thus, to maintain the availability above 95% it is necessary to ensure that the repair rate of severity #1 failures is at least 50 times of their failure rate. The probability of degraded mode also decreases with the repair rate, however, the drop in D is not as sharp as in A when the repair rate drops from 50 to 10 times of the failure rate of severity #1 failures.

In the second experiment, we set the repair rate of severity #2 failures, namely, μ_2 to 10, 50, 100, 500, 1000, 2500, and 5000 times the rate of severity #1 failures. The values of A and D were computed for each value of μ_2 using (1) and (2) respectively. The repair rate of severity #1 failures, namely, μ_1 was set to 10 times their failure rate. Figure 3 represents the availability A and the probability of degraded mode D as a function of the repair rate of severity #2 failures. It indicates that the availability is independent of the repair rate of severity #2 failures, which is expected. The probability of degraded mode increases as the repair rate of severity #2 failures decreases. Since the availability is identical for all values of the repair rate μ_2, this implies that the probability of the system being fully operational decreases as the repair rate of severity #2 failures decreases.

The trends observed in the sensitivity graphs in Figures 3 and 4 are consistent with the expectations of the experts who are closely familiar with the system.

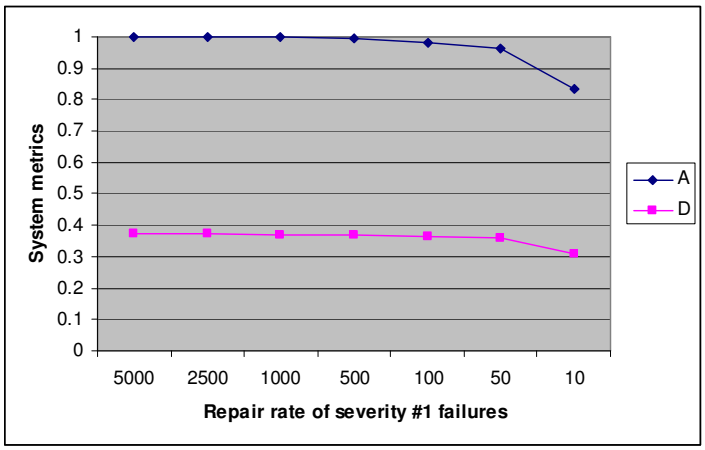

Fig. 3: A and D as a function of μ_1

5.3 Selection of repair parameters

For the failure parameters in Section 5.1, repair rates μ_1 and μ_2 were computed for different settings of A_{target} and D_{target}. The results in Table 4 indicate that the repair rate for severity #1 failures is exclusively determined by the target availability, whereas, the repair rate of severity #2 failures is determined both by the target availability and the probability of degraded mode. This is consistent with the trends in the sensitivity graph in Figure 3, where availability was unaffected by the repair rate of severity #2 failures. Furthermore, for a given target availability, as the probability of degraded mode decreases, the repair rate of severity #2 failure increases. This occurs, because a reduction in the probability of degraded mode needs to be achieved by rapidly transitioning from the degraded mode to the fully operational mode, which requires an increase in the repair rate of severity #2 failures. The results also indicate that raising the availability from 0.90 to 0.99 and from 0.99 to 0.9999, respectively, requires a 10- and a 100–fold increase in the repair rate of

severity #1 failures. Although these trends were intuitive and conformed to the experts' judgment, the expressions are valuable to obtain quantitative estimates of the repair rates for given failure parameters.

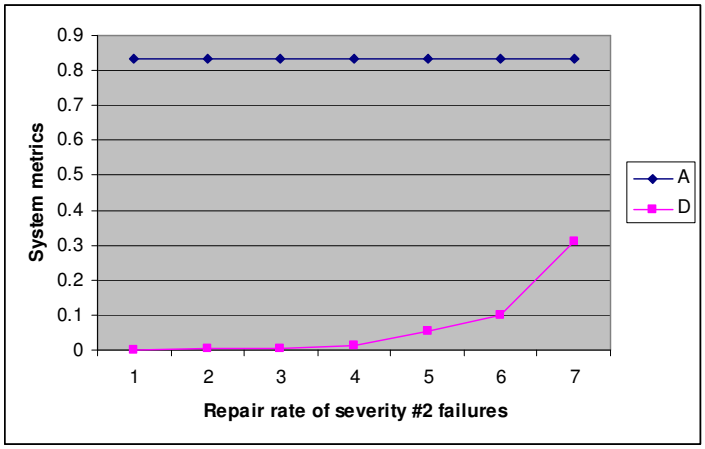

Fig. 4: A and D as a function of μ_2

Table 4: Repair rates for target availability and probability of degraded mode

A_{target}	D_{target}	μ_2	μ_1	A_{target}	D_{target}	μ_2	μ_1
0.8	0.4	0.01899	0.02964	0.85	0.4	0.02229	0.04199
0.8	0.3	0.03659	0.02964	0.85	0.3	0.04099	0.04199
0.8	0.2	0.01719	0.02964	0.85	0.2	0.07839	0.04199
0.8	0.1	0.17739	0.02964	0.85	0.1	0.19059	0.04199
0.90	0.4	0.02559	0.06669	0.95	0.4	0.02889	0.14079
0.90	0.3	0.04539	0.06669	0.95	0.3	0.04979	0.14079
0.90	0.2	0.08499	0.06669	0.95	0.2	0.09159	0.14079
0.90	0.1	0.20379	0.06669	0.95	0.1	0.21699	0.14079
0.99	0.4	0.03153	0.73359	0.9999	0.4	0.03218	74.092
0.99	0.3	0.05331	0.73359	0.9999	0.3	0.05418	74.092
0.99	0.2	0.09687	0.73359	0.9999	0.2	0.09817	74.092
0.99	0.1	0.22755	0.73559	0.9999	0.1	0.23016	74.092

5.4 Availability and degraded mode bounds

The bounds on availability (probability of degraded mode) for specified levels of probability of degraded mode (availability) for the failure parameters are in Table 5. These results indicate that as the target probability of degraded mode increases, the lower bound on availability increases. These results also indicate that the availability needs to be perfect if D_{target} exceeds a certain threshold. This threshold, which is a function of the failure parameters, is approximately 0.78 for the satellite system.

6. Conclusions

In this paper we presented an availability model which captures the impact of hardware and software failure severities in a unified manner. The model incorporates the

contribution of degraded modes arising out of tolerating lower severity failures to availability. We derived expressions to relate availability and probability of degraded mode to the failure and repair parameters of the components. We also discussed how the expressions could be used to select the repair parameters to achieve specified availability targets and also to determine bounds on availability and probability of degraded mode for given failure parameters. We described our experience in applying the model to the failure data collected during the acceptance testing of a satellite system and illustrated its use to facilitate various analyses.

Our future research involves propagating the variances in the failure and repair parameters to the variance in system availability. Enhancing the approach to quantify the service value by assigning reward rates to the different states of the model is also a topic of future research.

Table 3: Availability and degraded mode bounds

D_{target}	Availability, A_b (lower bound)	A_{target}	Degr. Mode, D_b (Upper bound)
0.2	0.2560	0.9999	0.7810
0.3	0.3840	0.999	0.7800
0.4	0.5121	0.95	0.7420
0.5	0.6401	0.90	0.7030
0.6	0.7681	0.85	0.6639
0.7	0.8962	0.80	0.6249
0.75	0.9602		
0.78	0.9986		

Acknowledgement: The research at NSWC was supported by the Software Assurance Research Program (SARP) from the Office of Safety and Mission Assurance (OSMA) of NASA, managed by the IV&V Facility at Fairmont, WV. The research at UConn was supported in part by the CT Space Grant Consortium, UConn Foundation and a CAREER award from NSF (#CNS-0643971).

References

[1] Trivedi K. S., *Probability and Statistics with Reliability, Queuing and Computer Science Applications*, John Wiley, 2001.

[2] Puliafito A., M. Telek and K. S. Trivedi, *The Evolution of Stochastic Petri Nets, Proceedings of World Congress on System Simulation*, pp. 3-15, Singapore, September 1997.

[3] Sun H., J. J. Han and H. Levendel, *Hierarchical Composition and Aggregation of State-Based Availability and Performability Models, IEEE Transactions on Reliability*, Vol. 52, No. 2, pp. 238-244, June 2003.

[4] Fota N., M. Kaaniche and K. Kanoun, *Dependability Evaluation of an Air Traffic Control Computing System, Performance Evaluation*, Vol. 35, No. 4, pp. 253-273, 1999.

[5] Kanoun K., M. Borrel, T. Morteville and Peytavin A., *Availability of CAUTRA, A Subset of the French Air Traffic Control System, IEEE Transactions on Computers*, Vol. 48, No. 5, pp. 528-535, 1999.

[6] Lyu M. R. and V. B. Mendiratta, *Software Fault Tolerance in a Clustered Architecture: Techniques and Reliability Modeling, Proceedings of IEEE Aerospace Conference*, pp. 141-150, March 1999.
[7] Sun H., J. J. Han and H. Levendel, *A generic availability model for clustered computing systems, Proceedings of IEEE International Symposium on Pacific Rim Dependable Computing*, pp. 241-248. December 2001.
[8] Chen D., S. Dharmaraja, D. Chen, L. Li, K. S. Trivedi, R. R. Some and A. Nikora, *Reliability and Availability Analysis for the JPL Remote Exploration and Experimentation System, Proceedings of International Conference on Dependable Systems and Networks*, pp. 337-342, June 2002.
[9] Yin L., R. Fricks and K. S. Trivedi, *Application of Semi Markov Process and CTMC to Evaluation of UPS System Availability, Proceedings of Annual Reliability and Maintainability Symposium*, pp. 584-591, January 2002.
[10] Smith R. M., K. S. Trivedi and A. V. Ramesh, *Performability Analysis: Measures, an Algorithm and a Case Study, IEEE Transactions on Computers*, Vol. 37, No. 4, pp. 406-417, April 1988.
[11] Kaaniche M., K. Kanoun and M. Rabaah, *A Framework for Modeling Availability of E-business Systems, Proceedings of Tenth International Conference on Computers, Communication and Networks*, pp. 40-45, October 2001.
[12] Haverkort B., R. Marie, G. Rubino and K. S. Trivedi, *Performability Modeling*, John Wiley and Sons, England, 2001.
[13] Levitin G., L. Anatoly, H. Ben-Haim and E. Elmakis, *Redundancy Optimization of Series-Parallel, Multi-State Systems*, *IEEE Transactions on Reliability*, Vol. 47, No. 2, pp. 165-172, June 1998.
[14] Hudson, J. C. and K. C. Kapur, *Reliability Theory for Multi-State Systems with Multi-State Components*, *Microelectronics and Reliability*, Vol. 22, No. 1, pp. 1-7, January 1982.
[15] Zang X., H. Sun and K. S. Trivedi, *A BDD-Based Algorithm for Analysis of Multi-State Systems with Multi-State Components, IEEE Transactions on Computers*, Vol. 52, No. 12, pp. 1608-1618, December 2003.
[16] X. Li, M. J. Zhao and R. C. M. Yam, *Reliability Analysis of a k-out-of-n System with Some Components being Suspended when the System is Down*, *Reliability Engineering and System Safety*, Vol. 91, No. 3, pp. 305-310, March 2006.
[17] Huang J, M. J. Zuo and Z. Fang, *Multi-State Consecutive k-out-of-n Systems*, *IIE Transactions*, Vol. 35, pp. 527-534, 2003.
[18] Nelson B., *Applied Life Data Analysis*, Wiley-IEEE, 2004.

Swapna S. Gokhale is currently an Assistant Professor in the Dept. of Computer Science and Engineering at the University of Connecticut. She received her B.E. (Hons.) in Electrical and Electronic Engineering and Computer Science from the Birla Institute of Technology and Science, Pilani, India in 1994, and M.S. and Ph.D. in Electrical and Computer Engineering from Duke University in 1996 and 1998 respectively. Prior to joining UConn, she spent one year as a Post Graduate Researcher at the University of California, Riverside and three years as a Research Scientist at Telcordia Technologies (Bell Communications Research), New Jersey. Her research interests lie in the areas of system and software reliability analysis, performance analysis of middleware and web-based systems and QoS issues in wireless and wireline networks. She received the

CAREER award from the National Science Foundation for her research in architecture-based software reliability assessment. She has published over 75 journal and conference papers on these topics.

John R. Crigler is a senior mathematical statistician and Complex Systems group leader in the Systems Research and Technology Department at the NSWCDD. He joined NSWCDD in 1972 and has thirty years experience in statistical consulting, analysis, and research on a variety of projects in the physical and engineering sciences. He was an Adjunct Professor in the Department of Statistics at VPI&SU from 1973 to 1995 and has authored numerous publications. His areas of interest include statistical process control, weapons accuracy assessment, software reliability and measurement, discrete event simulation, and computer performance evaluation. He is a member of the American Statistical Association, the Institute of Mathematical Statistics, and the International Council on Systems Engineering.

William H. Farr is a senior staff scientist and branch head in the Combat Systems Technology Branch of the Systems Research & Technology Department at the Naval Surface Warfare Center in Dahlgren, Virginia. He has over twenty years of experience in software system development specializing in software measurement, quality assurance, configuration management, and testing. He has extensive experience in the analysis of simulation data for complex Navy systems. In the standards arena he has been a part of a number of efforts for software reliability including the AIAA and the IEEE (Std 982.1-1988 & 982.2). He was the co-leader of the INCOSE Systems Measurement Working Group and the DoD Program for Practical Software Measurement, and past co-leader of the IEEE Systems Measurement Working Group under the Engineering of Computer Based Systems Technical Committee. He was one of four team leaders of the initiative “Engineering of Complex Systems” for ONR. He has published extensively in software metrics and reliability. He received the first "Most Significant Contribution to the Software Measurement Field" annual award by the Software Quality Institute in 1991 for his software reliability-modeling tool. He received his doctorate in Statistics/OR in 1973 from Florida State.