System Availability Analysis Considering Hardware/Software Failure Severities*

Swapna S. Gokhale
Dept. of CSE
Univ. of Connecticut
Storrs, CT 06269
ssg@engr.uconn.edu

John R. Crigler, William H. Farr
Advanced Computation Tech. Div.
NSWC, Dahlgren Division
Dahlgren, VA 22448
{william.farr,john.crigler}@navy.mil

Dolores R. Wallace
SRS Technologies
NASA GSFC
Greenbelt, MD
drwallace1@comcast.net

Abstract

Model-based analysis is a well-established approach to assess the influence of several factors on system availability within the context of system structure. Prevalent availability models in the literature consider all failures to be equivalent in terms of their consequences on system services. In other words, all the failures are assumed to be of the same level of severity. In practice, failures are typically classified into multiple severity levels, where failures belonging to the highest severity level cause a complete loss of service, while failures belonging to levels below the highest level enable the system to operate in a degraded mode. This makes it necessary to consider the influence of failure severities on system availability. In this paper we present a Markov model which considers failure severities of the components of the system in conjunction with its structure. The model also incorporates the repair of the components. Based on the model, we derive a closed form expression which relates system availability to the failure and repair parameters of the components. The failure parameters in the model are estimated based on the data collected during acceptance testing of a satellite system. However, since adequate data are not available to estimate the repair parameters, the closed form expressions are used to assess the sensitivity of the system availability to the repair parameters.

1 Introduction and Motivation

Our universe in the twenty first century has a variety of man-made systems in space. These systems have many components, usually including computer software systems either for control or communication with earth or in-space maintenance and scientific tasks. A major concern of the National Space and Aeronautics Administration and the other space agencies responsible for this space exploration is the reliable and efficient operation of these systems.

The reliability of a system may be defined as the probability of failure-free operation for a specified period of time in a specified environment [10]. Reliability is a key metric for many life-critical systems that are required to operate without failure for a given period of time. Many systems, however, are capable of tolerating some failures and continue to operate despite failures, perhaps in a degraded mode. Also, even though a failure causes total loss of service, the underlying fault may be repaired in order to restore the system back into operation. For such repairable systems as well as for systems which are capable of operating in a degraded mode, availability is a more relevant metric than reliability. The availability of a system is defined as the ability of the system to be in a state to perform a required function at a given instant of time or any instant of time within a given time interval [10]. A crucial difference between reliability and availability is that reliability refers to failure-free operation during an entire interval, while availability refers to failure-free operation at a given instant of time. This time is usually the instant when the system is first accessed to provide a required function or service.

Availability of a system is influenced by many factors such as the structural organization of the components of a system, the level of redundancy employed, and the failure and repair distributions of the components and the parameters of these distributions. These factors are inter-dependent, for example, the impact of employing a particular component in a redundant configuration will depend on the role of the component in the structure of the system. Model-based analysis has been widely used to assess the impact of these inter-dependent factors on system availability within the context of the system structure in a quantitative manner.

In the literature, model-based analysis has regarded all the failures of all the components to be equivalent. The consequence of each failure on the services provided by the

*This research was supported in part by the Software Assurance Research Program funded by the NASA Office of Safety and Mission Assurance, managed by the NASA IV&V Facility in Fairmont, WV.

Proceedings of the 2005 29th Annual IEEE/NASA Software Engineering Workshop (SEW'05)
0-7695-2306-4/05 $20.00 © 2005 IEEE

system is considered to be the same. In other words, each failure is considered to be of the same level of severity. As a result, redundancy is used to tolerate some failures and provide degraded mode of operation and repair/restoration is used to bring the system back into a completely operational state. In many real-life systems, however, all failures do not always have the same impact on system services. In fact, failures are typically classified into multiple severity levels, where failures belonging to the highest severity level cause a complete loss of service, while failures belonging to levels below the highest level enable the system to operate in a degraded mode. Thus, the system is capable of tolerating low severity failures without employing any other means such as redundancy. This makes it necessary to consider the influence of failure severities on system availability in conjunction with the system structure.

In this paper we present a Markov model which considers failure severities within the context of system structure. The model also incorporates repairs of the components. Based on the model, we derive a closed form expression which relates system availability to the failure and the repair parameters of the components. The failure parameters in the model are estimated using data collected during acceptance testing of a satellite system. The lack of appropriate data, however, precluded us from obtaining the estimates of repair parameters. As a result, the closed form availability expression derived in this paper was used to assess the sensitivity of the system availability to the repair parameters.

The rest of the paper is organized as follows: Section 2 provides an overview of the contemporary approaches to availability analysis. It then describes our approach in the light of these contemporary approaches. Section 3 describes the characteristics of the system under consideration. Section 4 presents the availability model for the system described in Section 3. Section 5 derives a closed form expression for system availability based on the model presented in Section 4. Section 6 discusses the data analysis performed to estimate the failure parameters in the model. Section 7 illustrates how the closed form availability expression could be used to assess the sensitivity of system availability to the repair parameters. Section 8 provides concluding remarks and offers directions for future research.

2 Availability analysis approaches

Contemporary availability analysis approaches can be broadly classified into two categories, namely model-based analysis and measurement-based analysis. In this section we provide a brief overview of these two approaches, and discuss the advantages and disadvantages of each. We then describe the salient features of our approach.

2.1 Model-based analysis

Model-based analysis has been used extensively for the reliability and availability analysis of several different types of systems such as a air-traffic control system [4], clustered computing systems [6, 8], remote exploration system [1], UPS system [11], and e-business systems [3]. The different model types used for analysis can be classified into two categories, namely, combinatorial models and state-space models, and a brief overview of these model types is presented here. An interested reader is referred to [10] for a comprehensive treatment.

Combinatorial models capture conditions that cause system failure in terms of the structural relationships between system components. The three model types that belong to this category include reliability block diagrams, reliability graphs and fault trees. Combinatorial models are efficient to specify and solve. However, the solution of these models assumes that the components are independent. This assumption of independence may not be satisfied by many real-life systems. State-space models such as Markov chains are used to represent complex interactions among the components. Many different types of dependencies among system components have been observed in practice and captured by Markov models. Markov models provide the ability to model systems that violate the assumptions made by the non-state-space models, but this ability is provided at the price of a state-space explosion.

The state-space explosion problem can be handled in two ways: It can be tolerated or it can be avoided. Large model tolerance must be used for specification, storage and solution of the model. If the storage and solution problems can be overcome, the specification problem can be solved by using more concise (and smaller) model specifications that can be automatically transformed into Markov models. Large models can be avoided by using hierarchical model composition [9]. Sometimes a combination of state-space and non-state-space models can be used, with the state-space models used for those parts of the system that have complex characteristics, while non-state-space models used for the more conforming parts of the system.

The primary advantage of model-based analysis is that it can be applied very early in the system lifecycle starting from the design stage. Typically, since the models can be solved relatively easily and with few computational resources, this approach can also be used to assess the sensitivity of the system availability to the different parameters. It can also be used to evaluate the impact of design changes on system availability. However, in some cases the models have to rely on simplifying assumptions in order to ensure analytical tractability, and hence may not adequately capture the characteristics of the system under consideration. Also, while trying to represent the different features of a

Proceedings of the 2005 29th Annual IEEE/NASA Software Engineering Workshop (SEW'05)
0-7695-2306-4/05 $20.00 © 2005 IEEE

system, the models may become very complex with several parameters. It is very difficult to collect data to enable the estimation of all the model parameters. As a result, model-based analysis has to rely on educated guesses of parameter values due to which only an approximate estimate of system availability can be obtained.

2.2 Measurement-based analysis

Measurement-based analysis consists of collecting data from a system and then analyzing these data to obtain an estimate of its availability. The first step in a measurement-based analysis approach consists of monitoring and recording naturally occurring errors and failures in a running system. Instrumentation techniques are necessary to obtain such measurements. The data can be of two types, namely, human-generated error reports and machine-logs [5, 7]. The benefits and obstacles in using each one of these two data types is discussed elsewhere [2]. The raw data collected during system operation usually contains a lot of redundant and irrelevant information. As a result, this data needs to be preprocessed to extract the information necessary for availability analysis. Data preprocessing is thus the second step in this approach. The preprocessed data can then be analyzed to compute mean time to failure and mean time to recovery/repair, from which an estimate of system availability can be obtained.

The primary advantage of measurement-based analysis is that it is the only way to obtain a true estimate of system availability. In addition, measurement-based analysis can be used to verify the assumptions underlying model-based analysis as well as to reveal model structure. The data collected during measurement-based analysis can be used to estimate the values of the parameters in the models. However, measurement-based analysis requires an operational system or at least a working prototype and hence it can be used very late in the system lifecycle. Also, there is no consensus or uniformity in the type of data required for measurement-based analysis and the procedures and tools used for the collection of each data. As a result, measurement-based analysis approach cannot be standardized, and needs to be customized specific to a given system. Also, it is expensive to conduct sensitivity analysis, since it entails modifying the system and repeating the data collection and analysis process.

2.3 Measurement-driven, model-based analysis

In this section we describe the essence of our availability analysis approach, which we elaborate upon in the subsequent sections. We term our approach "measurement-driven, model-based" analysis since it is a combination of both model-based analysis and measurement-based analysis approaches described above. We use the data collected from a system during acceptance testing to infer the characteristics of the system that are relevant to its availability and to reveal the structure of the model. The collected data is also used to estimate some of the parameters of the model. Thus, measurements "drive" model building as well as some parameter estimation. Since complete data were not available to obtain an estimate of the availability directly and to estimate all the parameters in the model, model-based analysis is used to assess the sensitivity of the system availability to those parameters which could not be estimated from real data.

3 System description

In this section we describe the characteristics of the system considered for availability analysis. Since the characteristics of the system were inferred from the collected data, we first describe the nature of the collected data that was used to drive the analysis. The raw data consisted of the times at which failures occurred. This raw data was preprocessed and each failure was classified along two dimensions. The first dimension attributed the failure to one of two components, namely, hardware or software. The second dimension classified each failure into two severity levels, where a failure of severity one results in a complete loss of service and a failure of severity two enables the system to operate in a degraded mode. In our measurement-driven, model-based analysis approach, the objective is to develop a model that conforms to the above properties of the collected data, so that at least some of the model parameters can be directly estimated from the data. The characteristics of the system were thus inferred directly from the data and are described below.

We assume that the system is composed of two components, namely, hardware and software. Both the hardware and the software components can fail. Each hardware and software failure can be classified into two categories, namely, severity #1 and severity #2. Thus the system can experience four types of failures, hardware failure of severity #1, hardware failure of severity #2, software failure of severity #1, and software failure of severity #2. We assume that both the components fail independently of each other. In addition, we also assume that there is no dependence among the failures of different severities of the same component. In other words, the occurrence of a failure of severity #2 from a hardware component does not increase the likelihood of the occurrence of a failure of severity #1. Similarly, a software failure of severity #1 is not more likely given that a failure of severity #2 has already occurred. It is important to note that failures of severity #2 (hardware and software) cannot occur once a failure of severity #1 (hardware or software) has already occurred.

Proceedings of the 2005 29th Annual IEEE/NASA Software Engineering Workshop (SEW'05)
0-7695-2306-4/05 $20.00 © 2005 IEEE

A failure of severity #1 (either hardware or software) is catastrophic and results in a complete loss of service, whereas a failure of severity #2 is serious (either hardware or software) and transitions the system into a degraded mode of operation. The system is restored back to its fully operational state from both failed and degraded modes. The system can thus be in three states, fully operational, degraded and failed.

The structure of a system determines whether the failure of a component causes system failure. Based on the contribution of component failure to system availability, the system can be considered to have a series structure. A fully operational system requires both the hardware and the software components to be fully operational. When either one or both the components are operating in a degraded mode, the system operates in a degraded mode. Finally, system failure occurs when either one of the components fail.

4 Availability model

In the availability model, the state of the system is represented by a 2-tuple. The first element in the tuple represents the state of the hardware component, and the second element in the tuple represents the state of the software component. Each one of the components (hardware and software) can be in three states, namely, fully operational, degraded and failed. For both the hardware and software components, we let U, 2 and 1 denote the fully operational, degraded, and failed states respectively. A combination of the states of the hardware and software components can result in a maximum of nine system states. Of these nine possible states, state $(1, 1)$ cannot occur, since a hardware failure of severity #1 would cause system failure which would preclude a software failure of severity #1 from occurring until the system is restored back into operation. Similarly, a software failure of severity #1 would result in a system failure, which would preclude a hardware failure of severity #1 from occurring until the system is restored back to operation.

We assume that the time to failure for all four types of failures is exponentially distributed. We consider two types of restoration operations, where the first type is used to remedy the cause of severity #1 failures (both hardware and software), and the second type is used to remedy the cause of hardware and software severity #2 failures. The time to restore the system back into operation also follows an exponential distribution, where the rate of the distribution depends on whether the restoration operation is being used to remedy the cause of a severity #1 or a severity #2 failure. 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 is the rate used for severity #1 failures. We also assume that although the states $(1, 2)$ and $(2, 1)$ are reached from state $(2, 2)$, the restoration operation from these states restores the system to the fully operational state (U, U) rather than the intermediate state $(2, 2)$. We use the following notation to represent the parameters of time to failure and time to restore distributions:

- λ_{h1} – Mean failure rate of hardware failures of sev. #1
- λ_{h2} – Mean failure rate of hardware failures of sev. #2
- λ_{s1} – Mean failure rate of software failures of sev. #1
- λ_{s2} – Mean failure rate of software failures of sev. #2
- μ_1 – Mean restoration rate for failures of sev. #1 (both hardware and software)
- μ_2 – Mean restoration rate for failures of sev. #2 (both hardware and software)

Figure 1 shows the Markov model of the system. The states marked in grey represent the scenario of complete loss of service. The states marked in blue represent degraded mode of operation. The single unmarked state represents a fully operational system.

5 Model solution

The steady state availability of the system denoted A is the sum of the probabilities of being in degraded states, namely, $(2, U)$, $(U, 2)$ and $(2, 2)$ and the completely operational state (U, U). Using the notation $p_{i,j}$ to denote the probability of being in state (i, j), where both i and j can take any of the three values U, 1 and 2, system availability A can be given by:

$$A = p_{U,U} + p_{2,U} + p_{U,2} + p_{2,2} \tag{1}$$

The probability of operating in a degraded mode, denoted D is the sum of the probabilities of being in states $(2, U)$, $(U, 2)$ and $(2, 2)$. D is thus given by:

$$D = p_{2,U} + p_{U,2} + p_{2,2} \tag{2}$$

In order to obtain A and D, the following system of balance equations needs to be solved.

$$\begin{aligned}(\lambda_{h1} + \lambda_{s1} + \lambda_{h2} + \lambda_{s2})p_{U,U} \quad &= \\ \mu_2(p_{2,U} + p_{U,2} + p_{2,2}) + \mu_1(p_{1,U} + p_{U,1} + p_{2,1} + p_{1,2})\end{aligned} \tag{3}$$

$$(\mu_2 + \lambda_{h1} + \lambda_{s1} + \lambda_{s2})p_{2,U} = \lambda_{h2}p_{U,U} \tag{4}$$

$$(\mu_2 + \lambda_{s1} + \lambda_{h1} + \lambda_{h2})p_{U,2} = \lambda_{s2}p_{U,U} \tag{5}$$

$$\mu_1 p_{1,U} = \lambda_{h1}(p_{U,U} + p_{2,U}) \tag{6}$$

Proceedings of the 2005 29th Annual IEEE/NASA Software Engineering Workshop (SEW'05)
0-7695-2306-4/05 $20.00 © 2005 IEEE

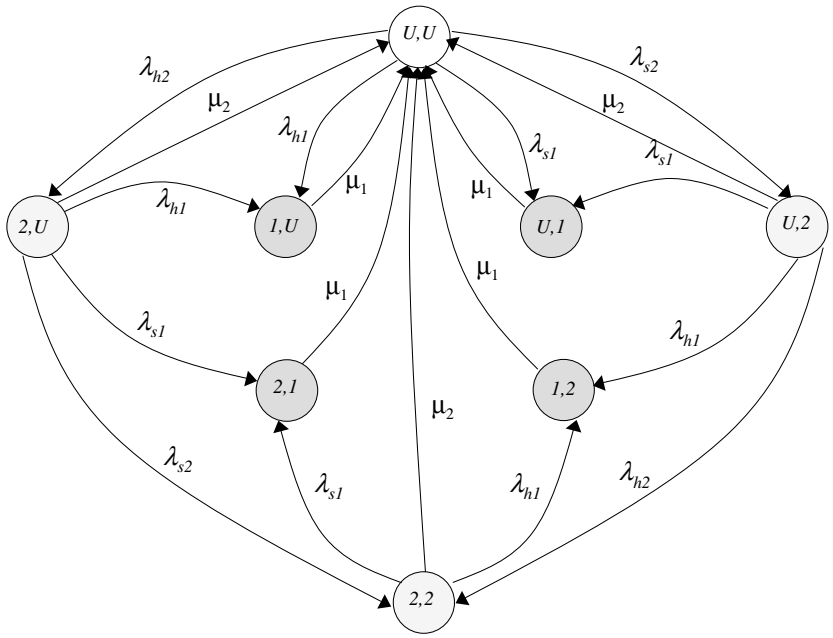

Figure 1. Availability model with failure severities

$$\mu_1 p_{U,1} = \lambda_{s1}(p_{U,U} + p_{U,2}) \tag{7}$$

$$\mu_1 p_{2,1} = \lambda_{s1}(p_{2,U} + p_{2,2}) \tag{8}$$

$$\mu_1 p_{1,2} = \lambda_{h1}(p_{U,2} + p_{2,2}) \tag{9}$$

$$\lambda_{s2} p_{2,U} + \lambda_{h2} p_{U,2} = \mu_2 p_{2,2} \tag{10}$$

$$p_{U,U} + p_{2,U} + p_{U,2} + p_{1,U} + p_{U,1} + p_{2,1} + p_{1,2} + p_{2,2} = 1 \tag{11}$$

The solution of the above set of balance equations is as follows:

$$p_{U,U} = \frac{1}{1 + \frac{(\lambda_{s1}+\lambda_{h1}+\mu_1)(\lambda_{s2}+\lambda_{h2})}{\mu_1(\lambda_{s1}+\lambda_{h1}+\mu_2)} + \frac{(\lambda_{h1}+\lambda_{s1})}{\mu_1}} \tag{12}$$

$$p_{2,U} = \frac{\lambda_{h2}}{\mu_2 + \lambda_{h1} + \lambda_{s1} + \lambda_{s2}} p_{U,U} \tag{13}$$

$$p_{U,2} = \frac{\lambda_{s2}}{\mu_2 + \lambda_{s1} + \lambda_{h1} + \lambda_{h2}} p_{U,U} \tag{14}$$

$$\begin{aligned} p_{2,1} = {} & \frac{\lambda_{s1}\lambda_{h2}}{\mu_1(\mu_2 + \lambda_{h1} + \lambda_{s1} + \lambda_{s2})} p_{U,U} + \\ & \frac{\lambda_{s1}\lambda_{s2}\lambda_{h2}}{\mu_1(\lambda_{s1} + \lambda_{h1} + \mu_2)} \frac{1}{\mu_2 + \lambda_{h1} + \lambda_{s1} + \lambda_{s2}} p_{U,U} + \\ & \frac{\lambda_{s1}\lambda_{s2}\lambda_{h2}}{\mu_1(\lambda_{s1} + \lambda_{h1} + \mu_2)} \frac{1}{\mu_2 + \lambda_{s1} + \lambda_{h1} + \lambda_{h2}} p_{U,U} \end{aligned} \tag{15}$$

$$\begin{aligned} p_{1,2} = {} & \frac{\lambda_{h1}\lambda_{s2}}{\mu_1(\mu_2 + \lambda_{s1} + \lambda_{h1} + \lambda_{h2})} p_{U,U} + \\ & \frac{\lambda_{h1}\lambda_{s2}\lambda_{h2}}{\mu_1(\lambda_{s1} + \lambda_{h1} + \mu_2)} \frac{1}{\mu_2 + \lambda_{h1} + \lambda_{s1} + \lambda_{s2}} p_{U,U} + \\ & \frac{\lambda_{h1}\lambda_{s2}\lambda_{h2}}{\mu_1(\lambda_{s1} + \lambda_{h1} + \mu_2)} \frac{1}{\mu_2 + \lambda_{s1} + \lambda_{h1} + \lambda_{h2}} p_{U,U} \end{aligned} \tag{16}$$

$$p_{1,U} = \left(\frac{\lambda_{h1}\lambda_{h2}}{\mu_1(\mu_2 + \lambda_{h1} + \lambda_{s1} + \lambda_{s2})} + \frac{\lambda_{h1}}{\mu_1}\right) p_{U,U} \tag{17}$$

$$p_{U,1} = \left(\frac{\lambda_{s1}\lambda_{s2}}{\mu_1(\mu_2 + \lambda_{s1} + \lambda_{h1} + \lambda_{h2})} + \frac{\lambda_{s1}}{\mu_1}\right) p_{U,U} \tag{18}$$

$$p_{2,2} = \frac{\lambda_{s2}\lambda_{h2}}{(\lambda_{s1} + \lambda_{h1} + \mu_2)(\mu_2 + \lambda_{h1} + \lambda_{s1} + \lambda_{s2})} p_{U,U} \tag{19}$$

Proceedings of the 2005 29th Annual IEEE/NASA Software Engineering Workshop (SEW'05)
0-7695-2306-4/05 $20.00 © 2005 IEEE

$$+\frac{\lambda_{s2}\lambda_{h2}}{(\lambda_{s1}+\lambda_{h1}+\mu_2)(\mu_2+\lambda_{s1}+\lambda_{h1}+\lambda_{h2})}p_{U,U}$$

Substituting the expressions for $p_{U,U}$, $p_{2,U}$, $p_{U,2}$ and $p_{2,2}$ into Equation (1), the steady state availability of the system can be given by Equation (20).

$$A = \frac{T_A}{T} \tag{20}$$

where T_A is given by:

$$T_A = (\lambda_{h2}+\lambda_{s2}+\lambda_{s1}+\lambda_{h1}+\mu_2)\mu_1 \tag{21}$$

and T is given by:

$$\begin{aligned} T \quad = \quad & \mu_1(\lambda_{s1}+\lambda_{h1}+\mu_2)+ \\ & (\lambda_{s1}+\lambda_{h1}+\mu_1)(\lambda_{s2}+\lambda_{h2})+ \\ & (\lambda_{h1}+\lambda_{s1})(\lambda_{s1}+\lambda_{h1}+\mu_2) \end{aligned} \tag{22}$$

The steady state probability of the system operating in a degraded mode D can be obtained by substituting $p_{2,U}$, $p_{U,2}$ and $p_{2,2}$ into Equation (23). D is given by:

$$D = \frac{T_D}{T} \tag{23}$$

where T_D is given by:

$$T_D = (\lambda_{h2}+\lambda_{s2})\mu_1 \tag{24}$$

and T is the same as in the case of system availability.

6 Parameter estimation

The parameters of the availability model described in Section 4 can be classified into two categories. The first category consists of the failure rates of severity #1 and severity #2 failures of the hardware and software components. We term the parameters belonging to the first category as "failure parameters". The second category consists of the repair rates of severity #1 and severity #2 failures for both hardware and software components, and we term these parameters as "repair parameters". In this section we discuss how the failure and the repair parameters of the model are estimated.

The failure parameters 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 were observed. The failures were classified into two categories according to their severities. They were also classified according to the component that caused the failure. Table 1 shows the sequence of failure occurrence dates for severity #1 failures. Table 2 indicates that 11 failure times are available for failures of the hardware component, whereas, a single failure time is available for failures of the software component. As a result, the data available to estimate the failure rate of severity #1 failures of the software component is insufficient. In order to alleviate this issue, we assume that the failure rate of the hardware and software failures of severity #1 are the same, and use all the available failure times to estimate these rates. Based on the sequence of failure occurrence times, time to failures are computed and are reported in column 3 of Table 1. The failure rate of hardware and software components for severity #1 failures are estimated using the following equation:

Table 1. Failure data for sev. #1 failures

Date	Component	Time to failure (Days)
6/4/1993	HW	
4/19/1994	HW	319
4/22/1994	HW	3
4/23/1994	HW	1
5/1/1994	SW	1
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

$$\lambda_{h1} = \lambda_{s1} = \frac{1}{\hat{M}} \tag{25}$$

where $\hat{M}$ is the estimate of the mean time to failure obtained from the sequence of time to failures in Table 1. The failure rate of severity #1 failures (both hardware and software), that is λ_{h1} and λ_{s1} were estimated to be 0.0037/day.

Table 2 shows the sequence of failure occurrence dates for severity #2 failures. Table 2 indicates that 38 failure times are available for hardware failures, whereas, only a single failure time is available for software failures. As a result, similar to the case of severity #1 failures, we assume that the failure rate of the hardware and software failures of severity #2 are the same, and use all the available failure times to estimate these rates. Based on the sequence of failure occurrence times, time to failures are computed and are reported in column 3 of Table 2.The failure rate of hardware and software components for severity #2 failures are estimated using the following equation:

$$\lambda_{h2} = \lambda_{s2} = \frac{1}{\hat{N}} \tag{26}$$

where $\hat{N}$ is the estimate of the mean time to failure obtained from the sequence of time to failures in Table 2. The

failure rate of severity #2 failures (both hardware and software), that is λ_{h2} and λ_{s2} were estimated to be 0.0132/day.

Due to the lack of appropriate data, the repair parameters of the model could not be estimated. Instead, the closed form expressions derived in Section 5 were used to assess the sensitivity of the system availability and probability of degraded mode of operation to the repair parameters as described in Section 7.

7 Sensitivity analysis

The failure parameters of the model shown in Figure 1 are estimated based on the data collected during the acceptance testing of a satellite system as described in Section 6. However, adequate data were not available to estimate the repair parameters of the model. Such a situation arises very often in practice where only some of the parameters can be estimated from real data, while the remaining parameters have to be "guestimated". In such a situation, instead of obtaining an estimate of system availability based on a single set of parameter values, a subset of which may be guestimated, 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 which establishes the relationship between the system availability and model parameters in a quantitative manner.

In this section we describe how the closed form expressions for steady state availability, and probability of degraded mode of operation given by Equations (20) and (23) can be used for sensitivity analysis. We designed two experiments to obtain the sensitivity graphs. The first experiment establishes the quantitative relationship between A and D, and the repair rate for severity #1 failures, namely, μ_1. The second experiment establishes the relationship between A and D and the repair rate for severity #2 failures, namely, μ_2. These experiments and their results are described below.

In the first experiment, we set the repair rate of severity #1 failures μ_1 to 10, 50, 100, 500, 1000, 2500, and 5000 times the failure rate of severity #1 failures. The values of A and D were computed for each one of the values of μ_1 using Equations (20) and (23). In all the computations, the repair rate of severity #2 failures μ_2 was set to 10 times the failure rate of severity #1 failures. Figure 2 shows the steady state system availability A as a function of μ_1. Figure 3 shows the probability of degraded mode of operation D as a function of μ_1. It can be observed from Figure 2 that the system availability drops sharply when the repair rate drops from 50 to 10 times of the failure rate of severity #1 failures. Thus, to maintain system availability above 95% it is necessary to ensure that the repair rate of severity #1 failures is at least 50 times of the failure rate of severity #1 failures. The

Table 2. Failure data for sev. #2 failures

Date	Component	Time to failure (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

Proceedings of the 2005 29th Annual IEEE/NASA Software Engineering Workshop (SEW'05)
0-7695-2306-4/05 $20.00 © 2005 IEEE

probability of degraded mode of operation also decreases as the repair rate decreases, however, the drop in D is not as sharp as the drop 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 μ_2 to 10, 50, 100, 500, 1000, 2500, and 5000 times the failure rate of severity #1 failures. The values of A and D were computed for each one of the values of μ_2 using Equations (20) and (23) respectively. In all the computations, the repair rate of severity #1 failures μ_1 was set to 10 times the failure rate of severity #1 failures. Figure 4 represents the steady state availability A of the system as a function of the repair rate of severity #2 failures μ_2. Figure 5 represents the probability of degraded mode of operation D as a function of the repair rate of severity #2 failures. Figure 4 indicates that the system availability is independent of the repair rate of severity #2 failures, which is expected. The probability of degraded mode of operation increases as the repair rate of the severity #2 failures decreases. Since the system availability is the same 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.

8 Conclusions and future research

In this paper, we present a system availability model which considers failure severities in conjunction with system structure. Based on the model, we obtain a closed form expression which relates system availability to the failure and the repair parameters of the components. We then describe availability analysis of a satellite system using the model based on the data collected during the acceptance testing of the system. We also describe the process of validating the model using simulated data. Our future research involves developing a method to propagate the variances in the estimates of failure and repair rates of the parameters to the variance in the system availability estimate. Developing techniques to estimate confidence intervals for system availability is also a topic of future research.

References

[1] D. Chen, 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". In *Proc. of Intl. Conference on Dependable Systems and Networks (DSN)*, pages 337–342, June 2002.

[2] R. K. Iyer and I. Lee. *Handbook of Software Reliability Engineering, M. R. Lyu (Ed.)*, chapter Measurement-based Analysis of Software Reliability. McGraw-Hill, 1996.

[3] M. Kaaniche, K. Kanoun, and M. Rabaah. "A framework for modeling availability of e-business systems,". In *Proc. of Tenth Intl. Conference on Computers, Communication and Networks (ICCCN)*, pages 40–45, October 2001.

[4] K. Kanoun, M. Borrel, T. Morteville, and A. Peytavin. "Availability of CAUTRA, a subset of the French air traffic control system,". *IEEE Trans. on Computers*, pages 528–535, 1999.

[5] R. Lal and G. Choi. "Error and failure analysis of a UNIX server". In *Proc. of High Assurance Systems Engineering Symposium*, pages 232–239, 1998.

[6] M. R. Lyu and V. B. Mendiratta. "Software fault tolerance in a clustered architecture: Techniques and reliability modeling". In *Proc. of Aerospace Conference*, pages 141–150, March 1990.

[7] C. Simache and M. Kaaniche. "Measurement-based availability analysis of UNIX systems in a distributed environment". In *Proc. of Intl. Symposium on Software Reliability Engineering (ISSRE)*, pages 346–355, 2001.

[8] H. Sun, J. J. Han, and H. Levendel. "A generic availability model for clustered computing systems". In *Proc. of Pacific Rim Dependable Computing*, pages 241–248, December 2001.

[9] H. Sun, J. J. Han, and H. Levendel. "Hierarchical composition and aggregation of state-based availability and performability models,". *IEEE Trans. on Reliability*, 52(2):238–244, June 2003.

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

[11] L. Yin, R. Fricks, and K. S. Trivedi. "Application of semi Markov process and CTMC to evaluation of UPS system availability". In *Proc. of Annual Reliability and Maintainability Symposium*, pages 584–591, January 2002.

Proceedings of the 2005 29th Annual IEEE/NASA Software Engineering Workshop (SEW'05)
0-7695-2306-4/05 $20.00 © 2005 IEEE

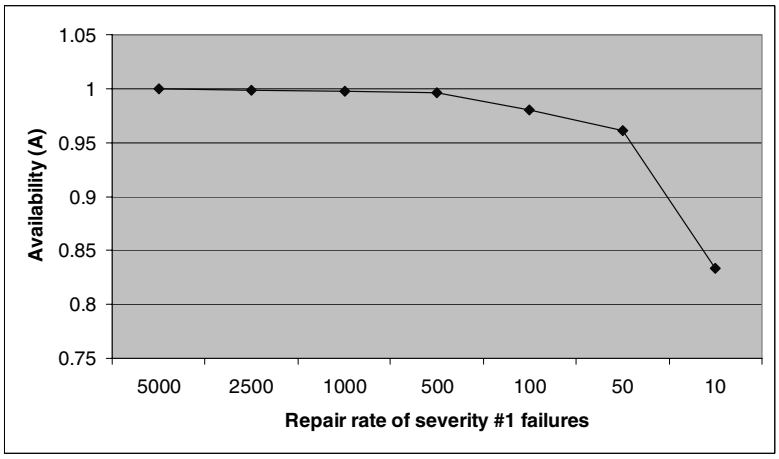

Figure 2. Availability as a function of μ_1

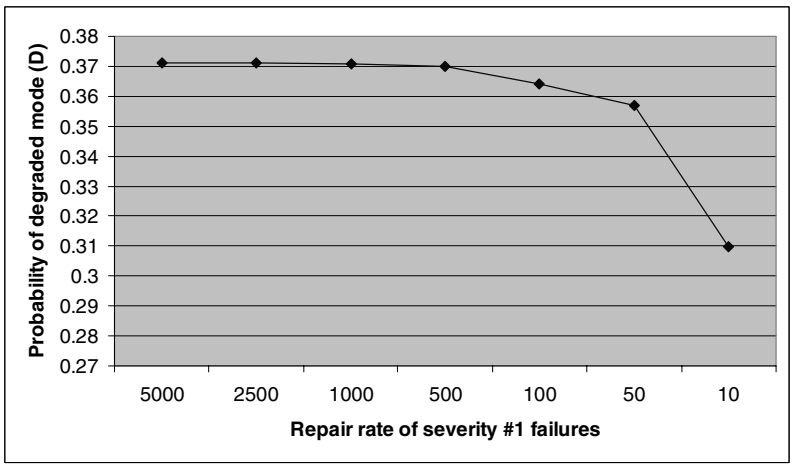

Figure 3. Probability of degraded mode of operation a function of μ_1

Proceedings of the 2005 29th Annual IEEE/NASA Software Engineering Workshop (SEW'05)
0-7695-2306-4/05 $20.00 © 2005 IEEE

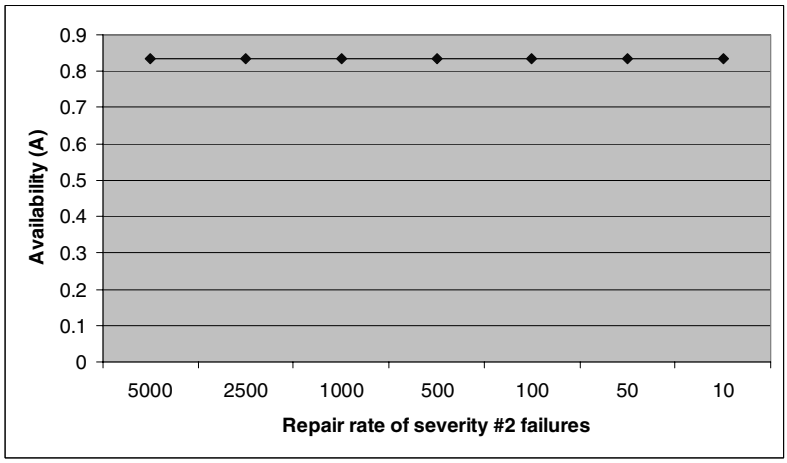

Figure 4. Availability as a function of μ_2

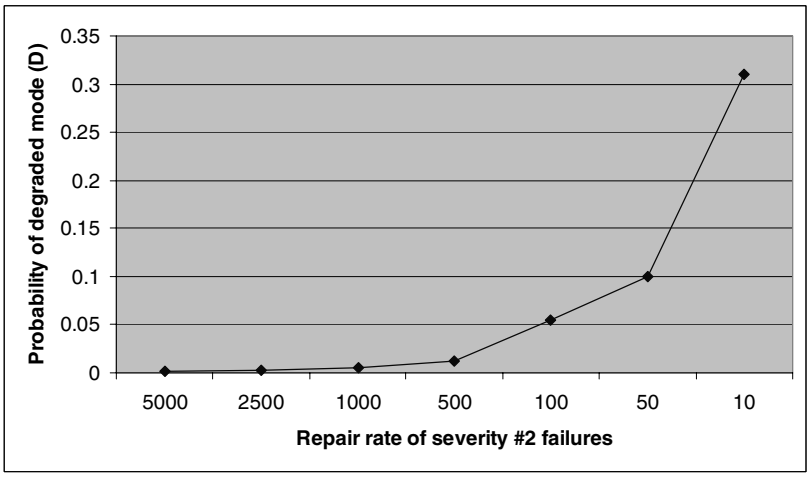

Figure 5. Probability of degraded mode of operation a function of μ_2

Proceedings of the 2005 29th Annual IEEE/NASA Software Engineering Workshop (SEW'05)
0-7695-2306-4/05 $20.00 © 2005 IEEE