SoHaR

MEADEP Application Note:

Availability Analysis of a Large Web Site

January, 2000

SoHaR Incorporated
8421 Wilshire Blvd., Suite 201
Beverly Hills, CA 90211
323-653-4717

www.sohar.com/meadep
info@sohar.com

Table of Contents

List of Acronyms

DBMS	Database management system
ISP	Internet Service Provider
MEADEP	Measurement-based Dependability software
ME	MEADEP Model Evaluator module
MG	MEADEP Model Generator Software
MTBO	Mean time between outages
MTTF	Mean time to failure
MTTR	Mean time to repair/restore
WAN	Wide Area Network

Summary

The MEADEP (Measurement-based Dependability) software tool was developed by SoHaR for the measurement, modeling, and prediction of dependability in complex systems. This paper describes how MEADEP can be used to

- *assess a large web site,*
- *identify the subsystems where the greatest improvement can be achieved, and*
- *answer business questions related to service level agreements.*

The paper shows how important, non-obvious conclusions can be reached through modeling. Two examples are to identify the subsystem that can benefit most from reliability improvement and the most important parameter to negotiate in a service level agreement in order to maximize uptime. The paper also shows how critical business decisions can be reached through such modeling.

Introduction

Availability is one of the most important attributes of online information systems – particularly those involved in e-commerce, supply chain management, medical information, and financial transactions. In the words of the Chief Information Officer (CIO) of one of the 10 largest traffic web sites, " Availability is as important as breathing in and out [is] to human beings".[1] The MEADEP tool permits CIOs and other strategic managers to make the right decisions on how to maintain and increase the dependability of their high visibility systems.

This application note describes the use of MEADEP (Measurement-based DEPendability tool) for predicting[2] the reliability and availability of an example large e-commerce web site. MEADEP's hierarchical modeling approach enables large information systems to be represented accurately and completely. Its sophisticated analytical methods provide consistent and repeatable results for high availability systems. The graphical user interface reduces this learning curve and makes MEADEP immediately usable by busy professionals. Similar tools have been developed at universities, but they are targeted at modeling specialists and researchers. MEADEP is targeted at practitioners such as the network managers, designers, and the CIO.

We first describe the system to be modeled, a large three-tiered e-commerce site. We then demonstrate how to use a modeling hierarchy to create simple, easily understood component models that are integrated to represent this complex system. We also demonstrate the power of combined Markov and reliability block diagram modeling. Finally, we discuss the results produced by the MEADEP model and how they can be interpreted from the perspective of the enterprise. The sections on modeling and initial results are technical and the intended audience is system architects, analysts, and actual and potential users.

However, the final two sections demonstrate how the results can be translated into business terms, e.g., the most important item to specify in a service level agreement or the value of capacity vs. redundancy in terms of total avoidable monthly revenue loss. These end sections are of interest to

---

[1] Kal Raman, Sr. VP and CIO of drugstore.com as quoted by Susan E. Fisher, "E-business redefines infrastructure needs", *Infoword,* January 7, 2000, available from www.infoworld.com

[2] The value of MEADEP as an analysis and prediction tool in contrast to tools for system monitoring and management that can be used for assessing existing availability and performance.

both specialists and those responsible for evaluating MEADEP from a managerial perspective. In addition, a white paper providing a broader perspective on the uses of modeling, measurement, and prediction for improving system uptime is available from SoHaR through our web site or by contacting us directly.

Example System Description

To demonstrate the capability of MEADEP to represent a large system, we will use a simplified representation of the eBay e-commerce web site shown in Figure 1 (based on a description in the *Los Angeles Times*[3 4]). Buyers and sellers enter information (through browsers) that is routed into either the Uunet or Sprint Internet backbones that are connected directly to the eBay site. Dual redundant Cisco routers are connected to a set of (Compaq) front-end web servers running on Windows NT.

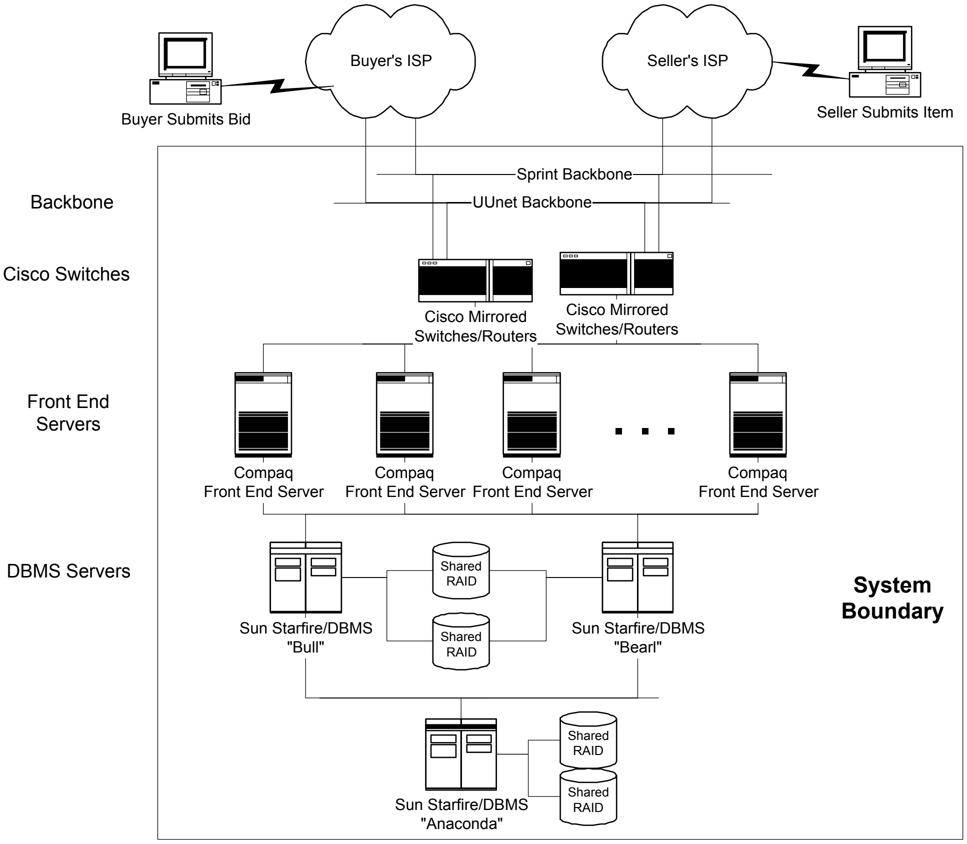


Figure 1 eBay System Configuration

The web servers in turn pass requests on auctions and/or bids to redundant Oracle Database Management System (DBMS) servers hosted on Sun Starfire servers designated as “bull” and “bear”. These DBMSs in turn retrieve and update tables representing each of the auctions. The outcomes of the auctions are recorded in the Anaconda system that then notifies the winners and losers by email.

---

[3] Joseph Menn, “Prevention of Online Crashes Is No Easy Fix”, *Los Angeles Times,* October 16, 1999, Section C, Page 1

[4] The primary reason that the eBay system was described was because of an extended 22 hour outage caused by a design deficiency and procedural problems in site restoration. Such outages are not stochastic in nature and are therefore out of the scope of a model using stochastic techniques such as MEADEP. However, although dramatic, such outages are probably less consequential over the long term than a web site which is not perceived as inherently dependable.

Model Description

MEADEP allows models to be created hierarchically and thus a complex system can be modeled as an ensemble of simpler, easily understood models which can be either system specific or adaptations of predefined models[5]. Figure 2 shows an example of a model hierarchy.

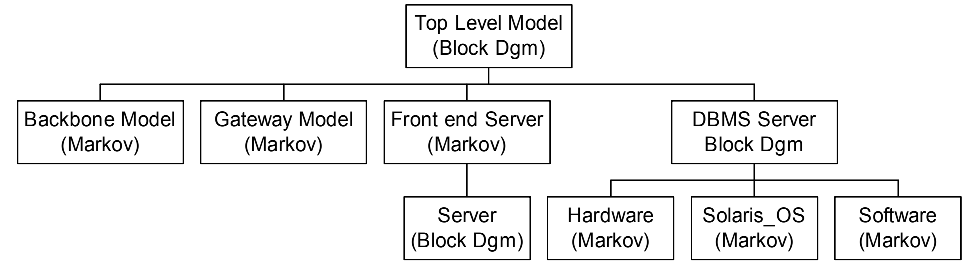


Figure 2. Organization of Hierarchical Model

In this example the top level model is a reliability block diagram that partitions the system into 4 lower level models that represent the backbone communications and tiers of the system. Three of these lower level models are Markov chains while the fourth is a block diagram.

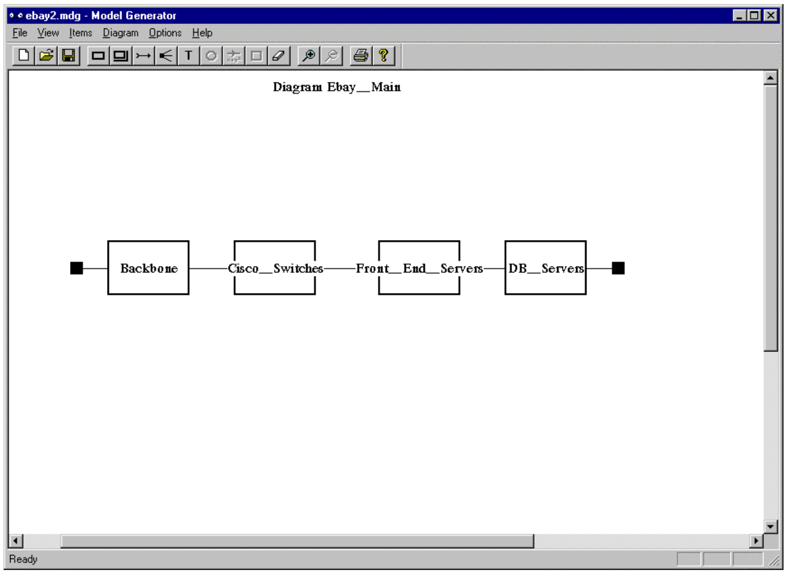


Figure 3 Top Level Reliability Block Diagram

Figure 3 is a screen dump from the MEADEP Model Generator (MG) module which shows the top level block diagram and demonstrates the graphical user interface for inputting models. There are four blocks representing the 4 top elements shown in Figure 1: the Internet backbones (UUnet and Sprint connections), the Cisco™ switches, the front end Compaq Web servers, and the bull and bear DBMS servers. The Anaconda server is not included because its failure does directly affect the on-line availability of the e-commerce site.

---

[5] As will be described below, MEADEP includes predefined libraries that can model many types of lower level blocks, and the user can add additional library functions for use in repeating elements or across multiple models.

Figure 4 shows the lower level reliability model for the two backbone networks. While reliability block diagrams use blocks to represent *structural elements,* Markov models represent the system in terms of *states*.. Movement from left to right indicates failure and movement from right to left indicates recovery In the model shown in Figure 4, there are 4 possible states: (1) both networks are up, (2) the UUnet network is up but the Sprint connection is down, (3) the converse, and (4) both networks are down. The second and third states are degraded in that only half the capacity is available. The fourth state is a completely failed state. The expressions next to the arrows reflect the *transition rates* among the states. The failure rate of the Sprint network is defined by the parameter λSPRINT, and the recovery rate is given by µSPRINT. The meanings of parameters such as λUUNet and µUUnet are similar. This model also considers the case when one backbone network fails and the system does not successfully make the switch to the second network. The parameter, C, or *coverage*, is the probability that there is a successful switch given that a failure in one of the networks occurs. In such a case, the quantity 1-C represents the probability that the transition is unsuccessful, that is, the Cisco switches do not handle the transition from two networks to one. As a result, the web site moves directly from the first functional state to the failure state. The final quantity in Figure 4 is the *reward*. The reward is the relative value of each state (as represented by a number from 0 to 1). We have assigned a reward of 1 to the fully functional state of both networks up, and a reward of 0.5 to the second and third states of one network up. This is to represent the loss of capacity. Specifically, if UUnet is down, the system capacity has been reduced by half (assuming that both Sprint and UUnet have equal capacity). If n fact, either network is sufficent to provide the fully capacity, then a reward of 1 would have been assigned to both states.

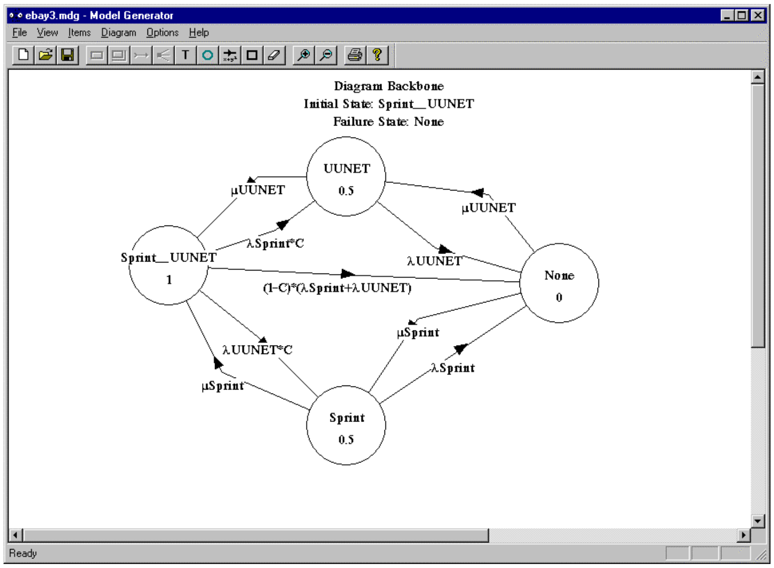


Figure 4. Markov Model for Backbone Network

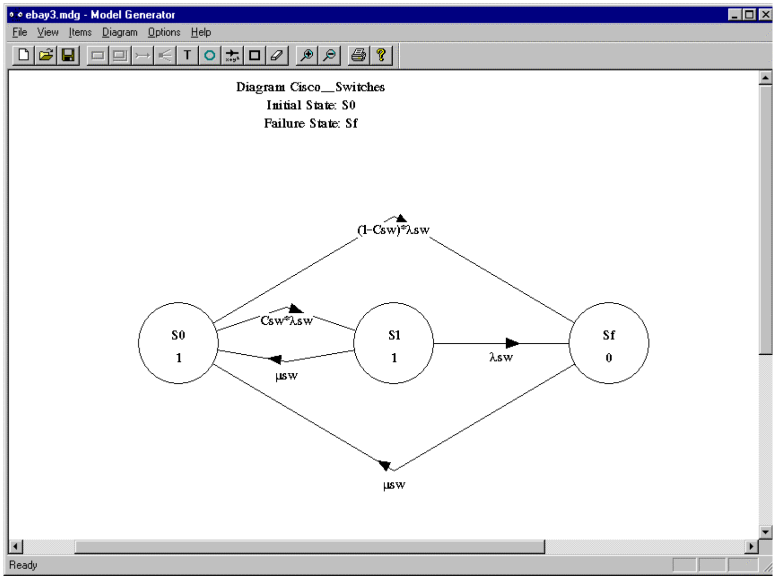


Figure 5. Gateway (Cisco Switch) Markov Model

We now move to the next block in Figure 2, the Gateway (Cisco Switches.) Figure 5 shows the corresponding lower level Markov model. This model shows a classic 1 out of 2 hardware system, and has been taken directly from the MEADEP library of reusable models. The first state on the left, designated S0, represents both switches up, the second, S1, represents one switch up, and the final state (on the right), S2, represents both switches down. As was the case in the previous model, we have a finite probability, C, that the system will successfully transition from state S0 to state S1, and the complementary probability, 1-C, that the system will transition from the functional state S0 to the failed state S2. In this model, we assign a reward of 1 to *both* the states S0 (both switches up) and S1 (one switch up) because either switch can handle the traffic of the entire network.

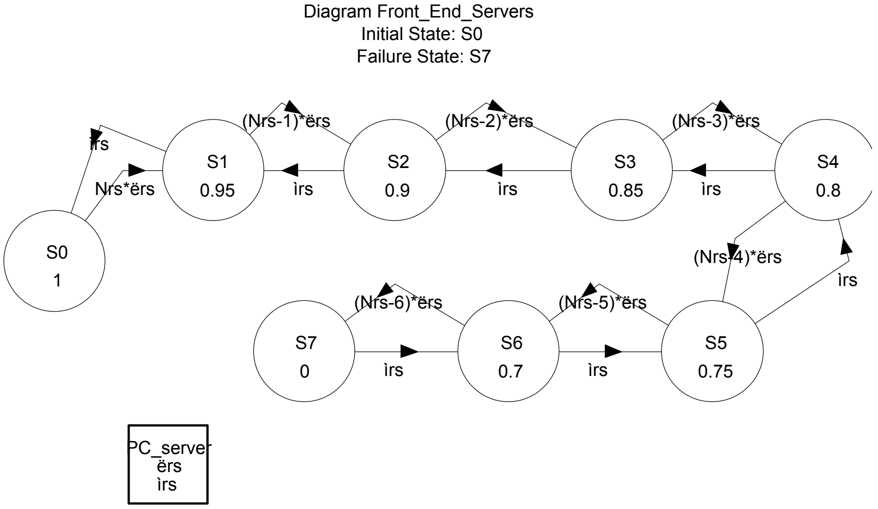


Figure 6. Portion of Front End Processor Model

The third block of Fig. 2 represents the front end servers. The model has 11 Compaq Windows NT servers. The model consists of 7 states, S0 represents all servers functional, S1 represents 1 server down, S2 represents 2 servers down, and so on. Load balancing capability in the ebay system assigns each incoming user to the most lightly loaded server. Each server is capable of handling a number of users. However, as servers fail, there is a reduction in overall capacity. We have assumed that once the system falls below 70% capacity (6 failures), a complete shutdown will occur We have assumed a 5% loss in capacity for each server failure. However, more precise estimates can be made on the basis of performance simulation modeling. Thus, we have assigned each with the appropriate reward function. For 11 servers, the reward is 1 (100% capacity), for 10 servers, it is 0.95, and so on.

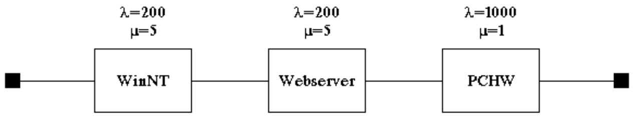


Figure 7. Server Model

Figure 7 shows a lower level model of a single PC front-end server. The model, which is used to calculated λserver as shown in Figure 6, consists of 3 elements: hardware, operating system (including the network infrastructure), and server software. This breakdown allows the uptime statistics from the Windows log files to be associated with the individual processes using the DPP and DEA modules of MEADEP (these are discussed in another application note).

The models for the database servers are shown in Figure 8 through 10. Figure 8 is a block diagram showing the dual redundant database server depicted in three segments: hardware, operating system, and DBMS software

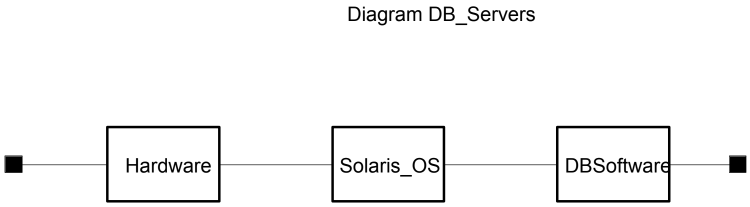


Figure 8. Database server top level diagram

Figure 9 shows the model of the dual redundant hardware platform which is similar to Figure 5. There are 3 states: both processors up (S0, i.e., no failures) , one processor up (S1, one processor down), and no processors up (Sf, failure state). State S0 has 2 transitions: the first being to state S1, and the second being to the failure state. The sum of these two transition rates is 2λhw, corresponding to the combined failure rate of the two functioning systems. State S1 also has two transitions out: restoration, i.e., µhw, and the condition of a second processor failure while the first is being restored. This occurs at the rate of failure of a single processor, or λhw.

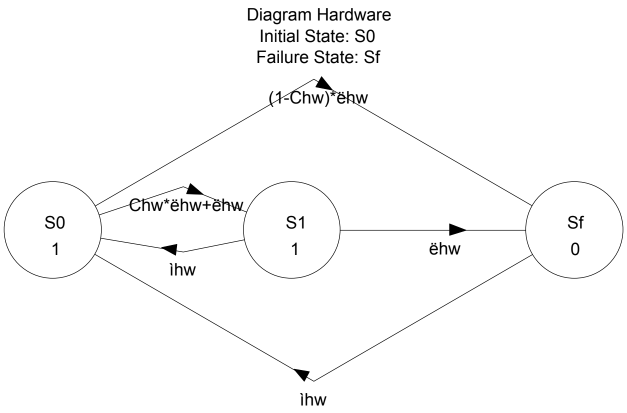


Figure 9 Database Hardware Model

Figure 10 shows the Markov model for the operating system portion of the DBMS servers, the block entitled Solaris_OS, and Figure 11 shows the corresponding diagram for the DBMS application running on the server. Both these models are similar to the hardware model shown in Figure 9. The primary difference is in the meaning of the transition parameters, which correspond to the failure rates of the DBMS and the operating system.

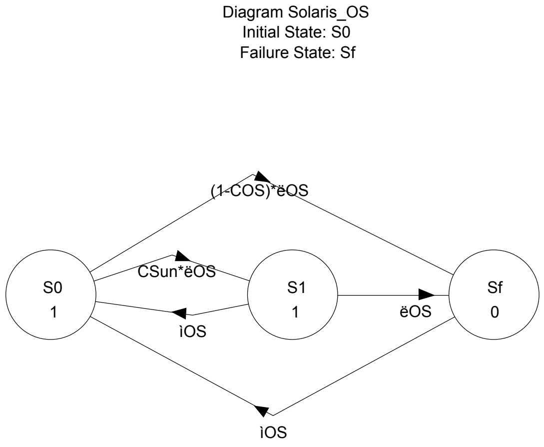


Figure 10. Solaris_OS diagram

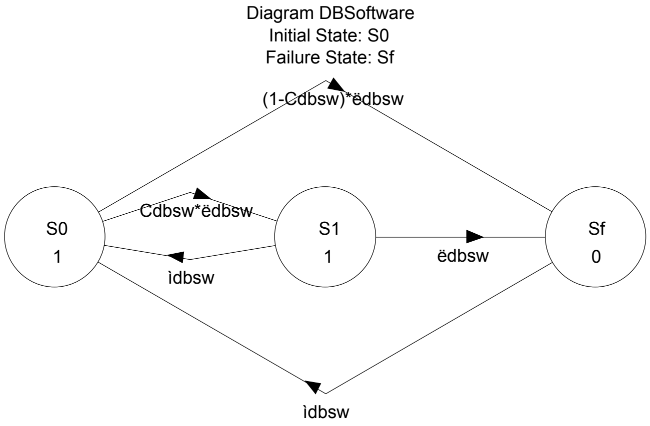


Figure 11. DBMS Block Markov Diagram

Table 1 summarizes the parameters of the model and shows representative values.

Table 1. Values of Input Parameters

Parameter	Meaning	Value
μWinNT	Reboot/restoration rate of Windows NT running on PC front end servers (corresponding to 12 minute restoration time)	5 hr^{-1}
λWinNT	Failure rate of Windows NT running on PC front end servers (corresponding to 200 hour average time between crashes)	.005 hr^{-1}
μWebserver	Reboot/restoration rate of web server software running on PC front end servers (corresponding to 12 minute restoration time)	5 hr^{-1}
λWebserver	Failure rate of web server running on PC front end servers (corresponding to 200 hour time between crashes)	.005 hr^{-1}
μPCHW	Repair rate of PC front end server hardware (corresponding to 1 hour mean repair time)	1 hr^{-1}
λPCHW	Failure rate of web server running on PC front end servers (corresponding to 200 hour MTTF)	.001 hr^{-1}
Nrs	Number of web servers	11
Csun	Failure detection and successful switchover probability of Sun Starfire server system from hardware failure	0.95
COS	Failure detection and successful switchover probability of Sun Starfire server operating systemfrom hardware failure	0.95
λOS	Failure rate of Solaris operating system on the Starfire server (corresponding to 200 hour MTTF)	0.005 hr^{-1}
μOS	Restoration time of Solaris operating system on the Starfire server (corresponding to 20 minute restoration time)	3 hr^{-1}
Cdbsw	Failure detection and successful switchover probability from DBMS failure	0.9
λdbsw	Failure rate of Oracle DBMS on the Starfire server (corresponding to 200 hour time between crashes)	.005 hr^{-1}
μdbsw	Restoration time of Oracle DBMS on the Starfire server (corresponding to 20 minture restoration time)	3 hr^{-1}
μhw	Repair time of the Starfire hardware	1 hr^{-1}
λhw	Failure rate of the Cisco Starfire hardware	.001 hr^{-1}
Chw	Failure detection and successful switchover probability from Starfire hardware failure	0.9
Csw	Failure detection and successful switchover probability from Cisco Switch failure	0.9
λsw	Failure rate of the Cisco Switch	.005 hr^{-1}
μsw	Restoration time of the Cisco Switch	3
μSprint	Restoration time of Sprint backbone (corresponding to 2 hours)	0.5
μUUNET	Restoration time of UUnet backbone (corresponding to 2 hours)	0.5
λUUNET	Outage rate of UUnet backbone (corresponding to 1000 hour time between backbone outages)	.001 hr^{-1}
C	Failure detection and successful switchover probability from a backbone failure	0.95
λSprint	Outage rate of Sprint backbone (corresponding to 1000 hour time between backbone outages)	.001 hr^{-1}

Results

Figure 12 shows the calculated results of the model described in the previous section. The subsystem with the highest failure rate is the front-end server PC subsystem followed by the WAN backbone. Table 1 shows the results in numeric form for all 9 models in the hierarchy. The overall system availability is 0.9957, corresponding to an average downtime of 38 hours per year.

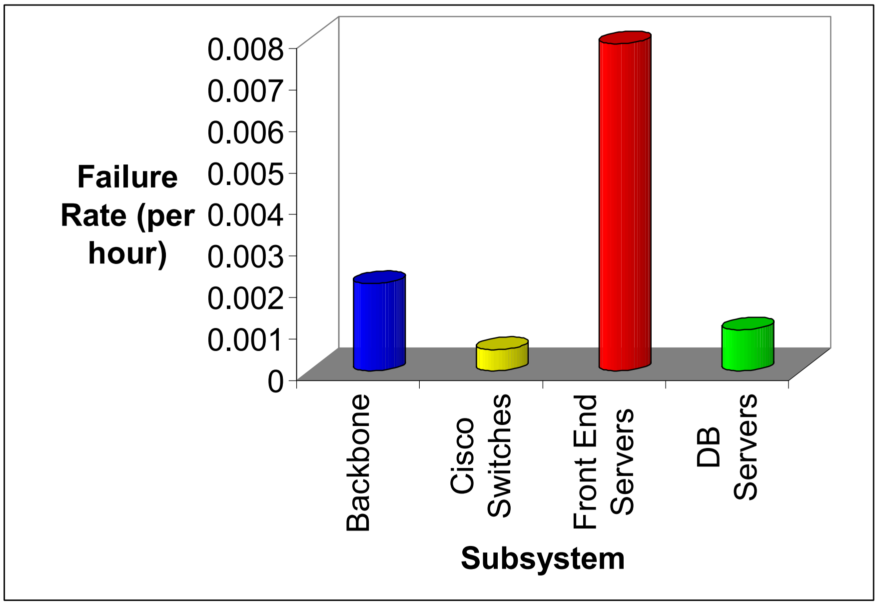


Figure 12. Failure Rates of Major Subsystems

Table 2. Model Results

Model-Name	Failure-Rate (per hour)	MTTF[6] (hours)	Repair-Rate (per hour)	MTTR[7] (hours)	Availability[8]	Unavailability
Ebay_Main	0.016189	61.8	3.74	0.268	0.995688	0.004312
Backbone	0.0021	476.2	1.00	1.000	0.997905	0.002095
Cisco_Switches	0.000507	1973.4	3.00	0.333	0.999831	0.000169
Front_End_Servers	0.007882	126.9	4.64	0.216	0.998303	0.001697
PC_server	0.013923	71.8	4.64	0.216	0.997006	0.002994
DB_Servers	0.000986	1014.5	2.77	0.362	0.999644	0.000356
DBSoftware	0.000507	1973.4	3.00	0.333	0.999831	0.000169
Hardware	0.000102	9832.4	1.00	1.000	0.999898	0.000102
Solaris_OS	0.000257	3883.5	3.00	0.333	0.999914	0.000086

These results indicate a site with a relatively high availability and reliability. However, for high volume web sites, there is always an incentive to achieve even higher levels of reliability and availability. The parametric analysis capabilities of the MEADEP tool are key strategic assets in the quest for more uptime. Their use is described in the following section.

---

[6] Mean time to failure, the inverse of the failure rate

[7] Mean time to repair/restore, the inverse of the repair rate

[8] The probability that the system or subsystem will be functional

Parametric Analyses

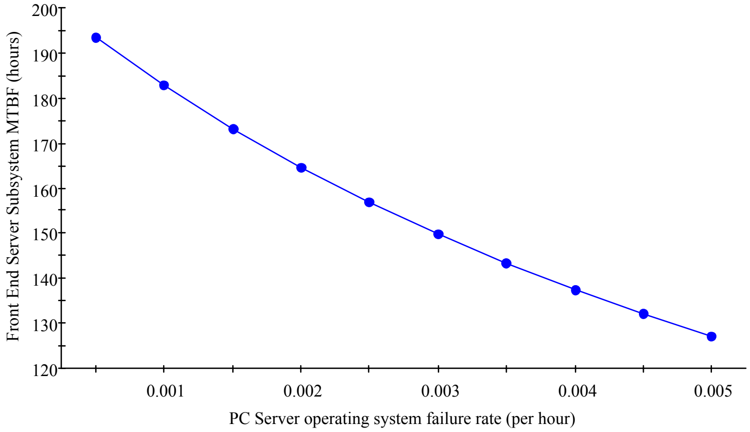


Figure 13. Impact of Operating System failure rate on Server Subsystem Reliability

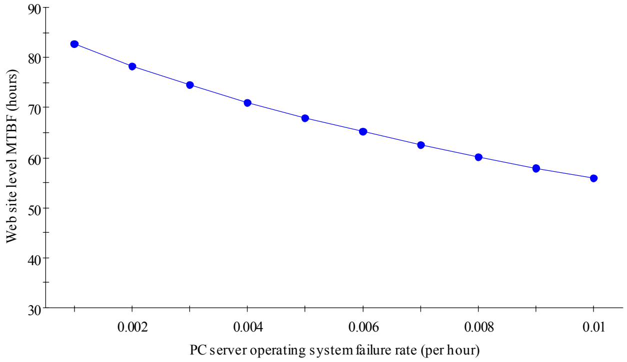


Figure 14. Impact of Operating System failure rate on Total System Reliability

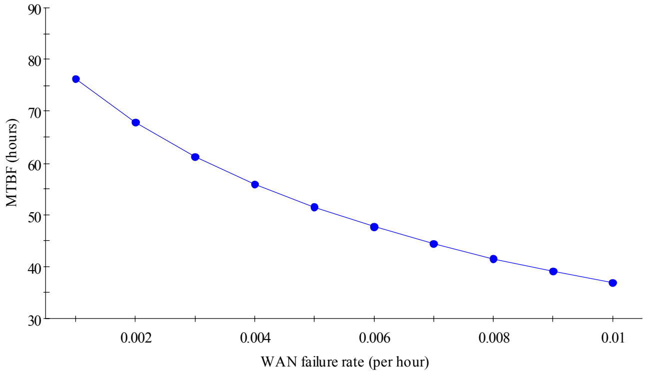
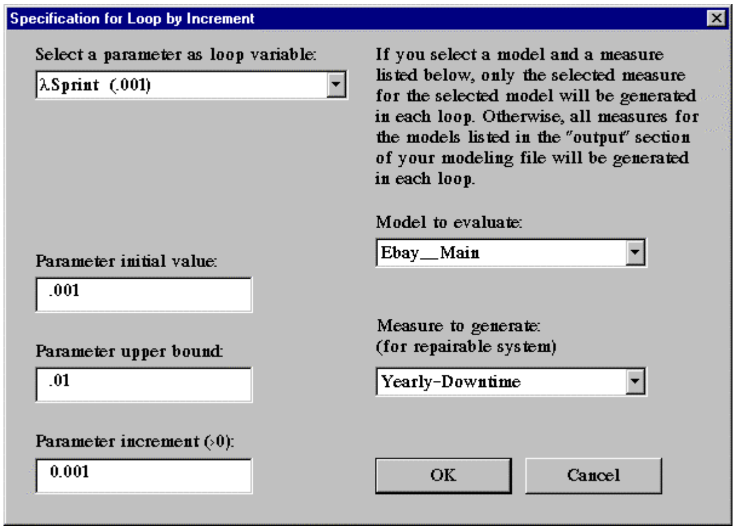


Figure 15. Impact of WAN Service Provider Failure Rates on site MTBF

Parametric analysis is a key element in defining an overall availability strategy. The Model Evaluator (ME) module of MEADEP facilitates such analyses for design and operations tradeoffs. For the model described above, the site is predicted to be off-line every 61 hours with an annual downtime of about 38 hours. With 650 auction bids per minute (as of October, 1999), this downtime could amount to more than 1.46 million lost transactions (this disregards the possible catch-up during later slack periods). This section describes how parametric analysis can be used to find the most effective means of getting more reliability and less downtime. As will be shown in the following discussion, surprising results can emerge from such analyses.

Parametric Analysis on Increasing Site Reliability

We first consider the case of increasing the site reliability, i.e., increasing its Mean Time Between Failures (MTBF). Figure 12 showed that the PC front end server subsystem has the highest failure rate. It would appear that increasing the reliability of the PC servers would therefore provide the highest benefit. However, parametric analyses show that such an effort will yield less benefits than other measures. Figure 13 shows the impact of decreasing PC server operating system failure rate from 0.005 per hour (200 hour MTBF) to 0.0005 (2000 hour MTBF), a decrease of 90% in the failure rate and a factor of 10 increase MTBF in each of the PC servers. The front server subsystem MTBF improves from 127 hours to 195 hours, corresponding to a decrease in the failure rate from 0.007882 per hour to 0.005128 per hour, a 65% decrease. This is certainly a benefit, but less than one might expect for decreasing the failure rate by 90%. The benefits on the system level are also modest. If the operating system failure rate for the PCs were decreased by 90%, the system level reliability is increased by only 20 hours as shown in Figure 14. The reason for these results can be seen in the structure of the front end server subsystem. With 11 such servers running in approximate parallel (even if there is a throughput reduction for the loss of each server), the impact of increasing the reliability of an individual computer is relatively modest.


Specification for Loop by Increment

Select a parameter as loop variable: λSprint (.001)

If you select a model and a measure listed below, only the selected measure for the selected model will be generated in each loop. Otherwise, all measures for the models listed in the "output" section of your modeling file will be generated in each loop.

Model to evaluate: Ebay__Main

Parameter initial value: .001

Measure to generate: (for repairable system) Yearly-Downtime

Parameter upper bound: .01

Parameter increment (>0): 0.001

Figure 16. ME dialog box for parametric analysis

An area where the site reliability can be more affected is is the service level agreement (SLA) with the service provider. According to a recent study of WAN service providers, the average time between hard downtime incidents was 1.7 per month, or approximately 0.00236 per hour (with an average outage time of 67 minutes)[9]. Figure 15 shows the impact of varying the service provider failure rate across a range of 90%, equivalent to the PC operating system failure rate shown in the previous paragrah. In this case, the system MTBF various from 76 hours to 36 hours, a change of 100%. Clearly, a greater benefit can be achieved by increasing WAN service provider availability over this range.

The use of the MEADEP ME module for this type of parametric analysis is illustrated by the dialog box in Figure 16. There the failure rate of the Sprint backbone is used as a parameter, varying from a failure rate of 0.001 (1,000 hour MTBF) to 0.01 (100 hour MTBF).

Parametric Analysis on Decreasing Downtime

Cumulative downtime is as important – perhaps even more important – than site reliability. Through a process similar to that described above, WAN service provider MTBF was identified as the most significant variable affecting downtime. Figure 17 shows the annual downtime as a function of the service provider as the repair rate is being varied from 0.5 per hour (corresponding to a repair time of 2 hours) to a repair rate of 2 per hour (corresponding to a repair time of 0.5

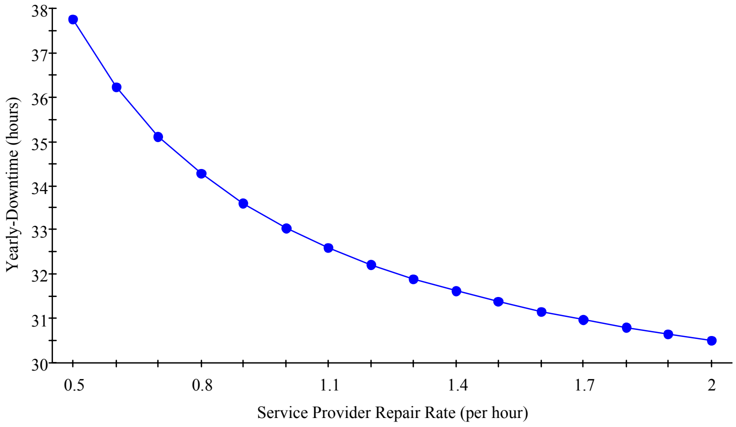


Figure 17. Yearly System Downtime as a Function of Service Provider Repair Rates

---

[9] “ WAN Downtime and SLAs”, Infonetics Research , San Jose, CA, December, 1998

hours). The ordinate shows the expected downtime decreases from approximately 38 hours to

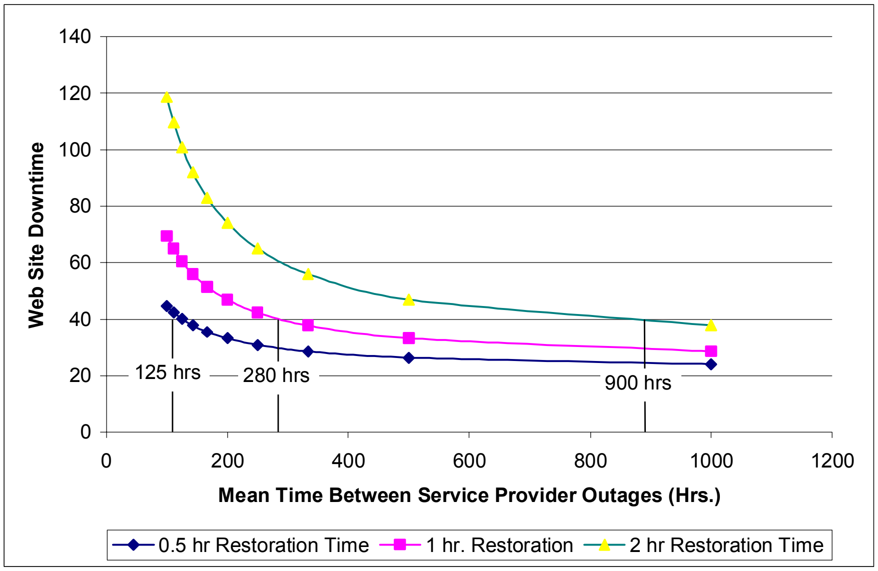


Figure 18. Impact of WAN Provider Restoration Time on Web Site Downtime

approximately 30. This result may be quite significant when negotiating a service level agreement. For example in Figure 18, we have shown how downtime is affected by mean time between outages for 3 restoration times that might be specified in a service level agreement. The figure shows that a target downtime per year for this web site of 40 hours can be achieved with a service provider who promises a 30 minute (0.5 hour) restoration time with a 125 hour MTBO, a 1 hour restoration time with a 280 hour MTBO, or a 2 hour restoration time with a 900 hour MTBO. Another way of looking at this result is that from the perspective of minimizing downtime, a service provide whose agreement specifies a rapid response time is preferable to one whose service level agreeement specifies a higher reliability (all other factors being equal).

Using MEADEP Models for Business Analyses

The MEADEP tool can make a significant contribution to maximizing the value of the web site investment. by defining the economic consequences of system decisions The section provides two examples: business analysis of a service level agreement and the tradeoff between increasing capacity or availability.

Business Analysis of a Service Level Agreement

For a continuously operating e-commerce site, maximizing uptime (i.e., availability) is the primary objective. In the case of the eBay site, uptime translates directly into transactions which in turn is the source of revenue for the enterprise. However, it is not *always* the most important attribute. For example, if a system is taken down periodically in accordance with a scheduled maintenance plan, then the *reliability*, i.e., the probablity that the system will be up during the maintenance interval is the primary figure of merit. Another example arose in a recent election when a major candidate held a fund raising event on the web site. For this limited time purpose, reliability (i.e., minimizing the probability of an outage over the fixed time that the fund raising was occurring) was also more important than continuous availability.

As was shown above, improvement of the WAN Service provider backbone restoration time provides the greatest benefit (for reliability). This section discusses the business case for two aspects of service level agreements: restoration time and reliability.

Figure 19 shows the impact of service provider restoration time on monthly revenue[10]. With a restoration time of 30 minutes, the expected number of lost transactions per year is approximately 1.2 million. However, as the restoration time approaches 5 hours, the number of lost transactions increases to 2.8 million. If the value of each transaction is $0.10, the value of a service level agreement that guarantees a restoration time of 0.5 hours or less is more than $160,000 per year.

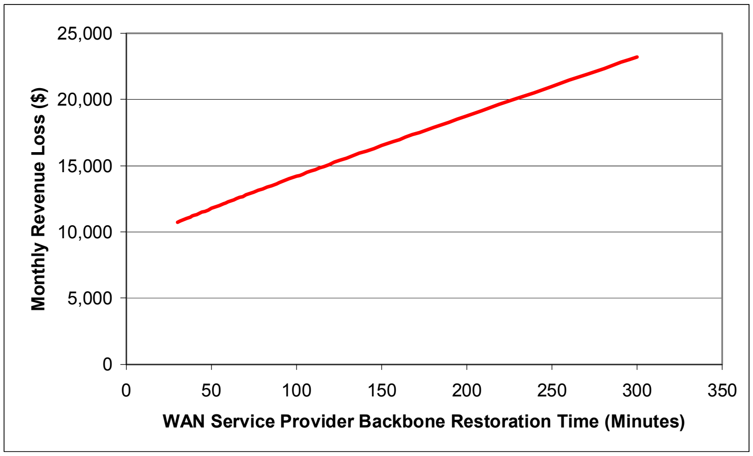


Figure 19. Impact of Service Provider Restoration Time on Monthly Revenue due to Lost Transactions

Figure 20 shows the impact of the failure rate of one of the two Internet service provider backbones on the number of lost transactions (the MTBO of the second Internet backbone is held constant). As the backbone MTBO is increased from 100 to 1000 hours, the number of lost transactions decreases from approximately 4. 2 million to just over 1 million. Under the assumption that each transaction is worth $0.10, the value of increasing reliability over this range is approximately $320,000 annually.

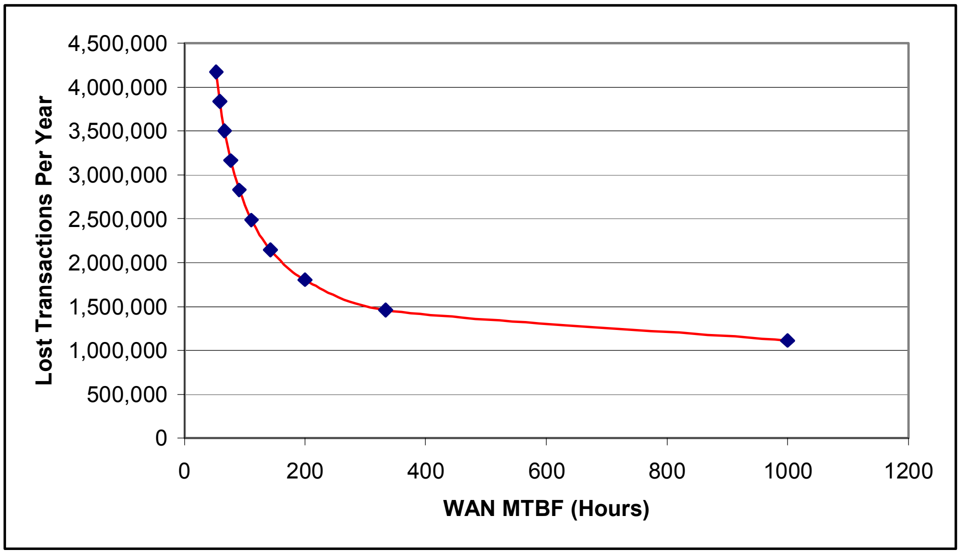


Figure 20. Impact of Backbone Outage Rate on Web Site Downtime

---

[10] We have assumed a transaction rate of 650 per minute, failure rates of 500 hours for both backbone networks and a restoration time of 1 hour for the service provider of the second network, in this case, assumed to be UUnet.

However, it may be impossible – or very expensive -- to obtain a service agreement with a guarantee of an average time between outages time of 1000 hours (approximately 6 weeks). It will probably be much more feasible to get an agreement of 200 to 300 hours. As is evident in the figure, most of the benefit can be achieved if the reliability is increased to 330 hours (a little less than two weeks). Thus, it may be more cost effective to set the reliability of the service level agreement at that level. The resulting reduction of transaction loss, approximatley 2.7 million annually, would be worth $270,000 annually.

It must be emphasized that these results are with the presence of *two independent and parallel* service providers. If a single backbone is used, the relative importance of MTBO and outage restoration time will differ.

Value of Additional Capacity

The eBay site model has two independent backbones, but neither has the capacity to carry the full traffic load (the model assumes that each has half the capacity). Given the MTBOs and restoration times of the networks, MEADEP can be used to evaluate the value the additional capacity through its *reward* function. Figure 21 compares the monthly revenue loss for two alternatives:

- *Half capacity backbones*: 2 WAN backbones each with half the capacity required for the average load, and
- *Full capacity backbones:* 2 WAN backbones each with the capacity for the entire load

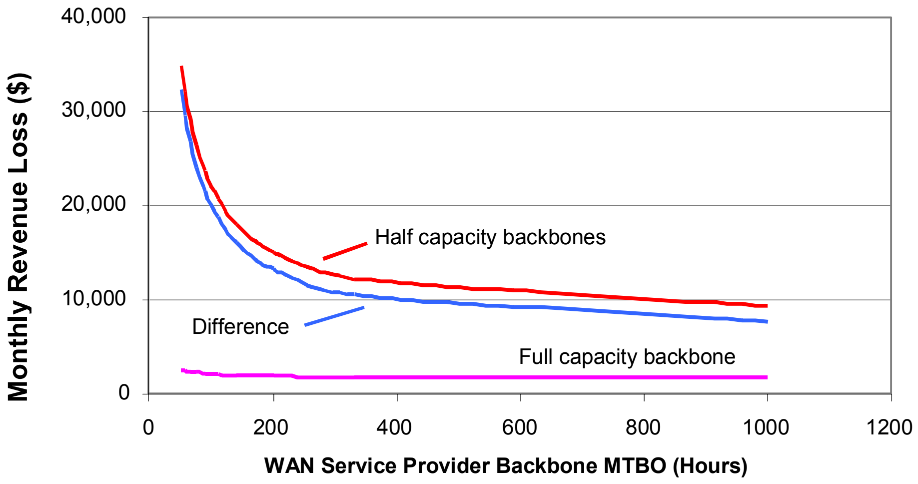


Figure 21. Revenue losses due to outages of WAN backbones

The results show that increasing the capacity of both backbone WANs so that either could handle the full load could decrease the monthly revenue loss by as much as $30,000 or as little as $9,000 depending on the reliability (as measured by mean time between outages) of the service providers[11]. The lower the reliability (MTBO), the greater the benefit of increasing the capacity.

With these results, it is also possible to assess the value of a service level agreement with the additional dimension of capacity. For example, it may be that provisioning each of the backbones with double the capacity is a lower cost option than securing service level agreements with both WAN backbone service providers. With the MEADEP reward function, it is also possible to

---

[11] Under the assumption of a uniform 650 transactions per minute and a value of $0.10 per transaction outline, the 1 hour response time, and the 500 hour MTBO on the second backbone WAN discussed above.

consider other options such as providing full capacity in the first backbone and half capacity in the second one, or providing 75% capacity in each. The optimum configuration depends on the traffic profile and the MTBO and restoration times of the WAN backbones, the load profile of the server, the value of the traffic, and other system-specific factors.

Conclusions

This application note has described the use of the MEADEP tool to analyze a web site and develop the answers to strategic questions on the configuration and operation of high availability computing systems. The methodology described here can be applied not only to on-line e-commerce sites but also to other computer architectures and configurations. For example, published MEADEP-based analyses have included air traffic control systems and nuclear reactors. These studies are available from the SoHaR webiste, at www.sohar.com. Other applications of MEADEP have been in telecommunications, process control, and HVAC.

This paper has addressed the modeling capability of the MEADEP tool but not the data analysis. Future white papers will address how to analyze data from Windows and UNIX event logs and to incorporate them into reliability models.

About SoHaR Incorporated

SoHaR is a consulting, research, and development organization which specializes in computing for critical applications (hence the name: Software and Hardware Reliability). SoHaR has participated in the development of satellite, missile, and aircraft control systems; ground-based transportation control; and nuclear reactor safety and control. For almost two decades, SoHaR has also provided reliability, maintainability, and availability support for the Federal Aviation Administration in their automation and communication, navigation and surveillance (CNS) systems. Our areas of research include distributed systems fault tolerance; Internet-based logistics and maintainability applications; safety software verification and validation; and advanced software analysis tools for reliability analysis, sneak circuit analysis, and critical software testing. Since its founding in 1978, the firm has published more than 150 articles in refereed publications and conferences. In 1990, we received the U.S. Small Business Administration Prime Contractor of the Year award in Region IX (Southwest Pacific and Hawaii).

SoHaR Incorporated • 8421 Wilshire Blvd., Suite 201 • Beverly Hills CA 90211 • Tel: (323) 653-4717 • Fax: (323) 653-3624

http://www.sohar.com