Chapter 6

Dealing with Dynamics in Modeling for Reliability Prediction

6.1 Introduction

In the previous chapter, we brought out the limitation of existing approaches for early software reliability prediction, based on Markov chain and provided our approach to address the same. A software system is made of many components, depending on the size and complexity of the system. These components are responsible to perform their predefined functions. Some functions are very critical, means the failure of such functions fails the overall system; while some functions are less critical i.e. impact of failure of such components does not fail the overall system. Instead the system continues to work in degraded fashion. Therefore the reliability requirements of all the components of a software system are not or may not necessarily be same. Reliability of the overall system is a function of reliabilities of all of its components and their arrangements. Therefore, understanding the criticality of real time systems, we focus on the fact that it is necessary to access the impact of change in reliability of any component on the reliability of other components and overall system to take some preventive action during the design phase.

The impact assessment is also very important when a software system is under operation. Reliability gets change because of environmental conditions, during system operation. Hence there is a requirement to have a mechanism to assess the reliability of all the components during operation, using operational profile data, for impact analysis. This will help the system maintainer to take preventive and corrective actions. Preventive action may include, switching to redundant healthy system while corrective action may include replacement or repair of the faulty component.

This impact assessment, during operation is also important when a software system contains one or more COTS component. Since the architectural design is not known in case of COTS component, its early reliability assessment is difficult. Hence, its impact analysis can be done during the operational phase.

Bayesian statistical frameworks have gained popularity in recent years for reliability assessment and prediction. Computational reliability methods prove to be cost-efficient for large and complex systems, and are suitable when experimental data are difficult to obtain. However unrealistic assumptions and approximations along with limited real data set impart uncertainty in the computational models. Hence the reliability-based design and operational profile should be combined to improve the reliability estimation. There are many sources of modeling errors, such as ambiguous or incomplete requirements, design defects, poor modeling of scenarios, environmental conditions, etc. Because of these uncertainties in modeling, model updating is best tackled as a Bayesian statistical inference problem [93].

Collins et al. [94] proposed an approach to update Bayesian model for mechanical systems, using identified modal parameters. Further an inclusive Bayesian framework for updating the model is described by Katafygiotis et al. [93]. They addressed the problem of updating a model and its associated uncertainties by utilizing dynamic response data. This model may not be a single “best” model, but instead updates a probability distribution over a specified set of structural and prediction-error probability models for the uncertain error between the model predictions and the corresponding actual structural response.

We extend the existing work for a software system. Software constitutes several components; each component is responsible for performing some function. The importance of every function is not necessarily to be same. Hence the reliability requirements of all the components may differ. The overall reliability of the software system is a function of the reliability of it components and their interfaces. The reliability of any component may change during the operational phase of the system, which may affect the reliability of its associated components and hence the overall reliability of the system. Hence there is a requirement of updating the estimates of reliabilities of each component and overall system whenever reliability of any components gets changed. Model updating is useful for improvement of the prediction accuracy of the system response or its current state, failure or healthy. In our work, we extend the focus on updating the reliability estimate of each software component and overall software system whenever the reliability of any component changes. Bayes' theorem, which is used for probabilistic updating, provides an appropriate framework for this purpose. Many researchers attempted to address for fatigue reliability updation using Bayes' theory [95], which are based on component level reliability.

Specifically, we illustrate our approach on a software of a safety critical CBS of a Nuclear Power Plant. Researchers have proposed several studies to update system level reliability estimates when system level test data are available, also using Bayes' theorem [96]. These approaches are based on an assumption that if the system passes, all the components of that system are healthy. But the healthiness of the system gives the assurance of the healthiness of only invoked components. In the same way if the system fails, all the components of that system do not necessarily have failed. Therefore, Martz and Waller [97] concluded that the system test data usually does not provide the complete information on the components' reliability.

6.2 Inference via Bayesian Networks

Considering a BN over $X = X_1, X_2, \ldots, X_n, X_n \in nodes$,the joint probability and marginal probability is given by equation 6.1 and 6.2 respectively.

$$P(X) = P\{X_1, X_2, \ldots, X_n\} = \prod_{i=1}^{n} P(X_i|\pi_i) \tag{6.1}$$

$$P(X_i) = \sum_{X \notin X_i} P(X) \tag{6.2}$$

Given a BN that specified the JPD in a factored form, one can evaluate all the possible inference queries by marginalization. Two types of inference support are often considered: predictive support for node X_i, based on evidence nodes connected to X_i through its parent nodes (also called top-down reasoning), and diagnostic support for node X_i, based on evidence nodes connected to X_i through its children nodes (also called bottom-up reasoning). For example, the failure of electronic IC can be represented by BN as shown in figure 6.1. It considers an IC that fails from overheating, an event represented by the variable 'Overheating' (denoted by H). Such overheating can fail the IC, represented by the variable 'IC-failure' (denoted by F). The overheating might result from a wrong voltage, represented by the variable 'wrong-volt' (denoted by V) or from environmental conditions, represented by the variable 'Environment' (denoted by E). In the latter case, it is reasonable to assume that other ICs of that electronic system will suffer and report a similar overheating syndrome, an event represented by the variable Other-ICs (denoted by O). The state of all the variables are either true (denoted by "T'') or false (denoted by $''F''$). One might consider the diagnostic support for the belief on unwanted environmental conditions at the electronic installation, given the observation that the IC fails. Such a support is formulated by equation 6.3:

$$P(E = T|F = T) = \frac{P(E = T, F = T)}{P(F = T)} \tag{6.3}$$

where,

$$P(E = T|F = T) = \sum_{V,O,H \in [T,F]} P(E = T)P(V) \times P(O|E = T)P(H|V, E = T)P(F = T|H) \quad (6.4)$$

and

$$P(F = T) = \sum_{V,O,H \in [T,F]} P(E)P(V)P(O|E)P(H|V, E)P(F = T|H) \quad (6.5)$$

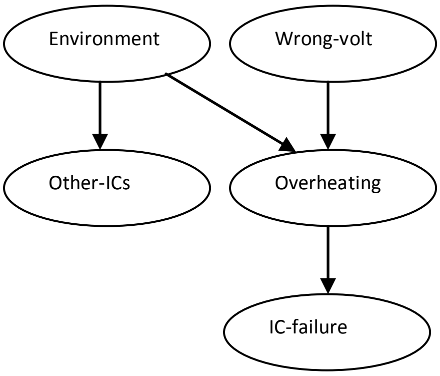

Figure 6.1: BN representation of IC.

6.3 A Case Study

We take the same case study as we have taken in section 4.2 and section 5.4 to illustrate our approach.

6.4 Reliability Estimate Updation

When the state of any software component changes, the state of other software components and hence state of overall software system can be updated using BN through backward propagation. The method is shown for the series system. The same approach can be adopted for the parallel system.

Consider a series system of the TF software. Data acquisition and Testing are the two important functions of TF, for which there are two dedicated components. The failure of either component, fails the total system. Let these components are denoted by A and B respectively, and TF is represented by C. Since it is a series system, the failure probability of TF can be computed by equation 6.6:

$$\begin{aligned} P(C = F) = 1 \times P(A = F)P(B = F) + 1 \times P(A = F)P(B = T) \\ + 1 \times P(A = T)P(B = F) \end{aligned} \tag{6.6}$$

$$= P(A) + P(B) - P(A)P(B) \tag{6.7}$$

where $P(X) = P(X = F)$ Here we presume that A & B are independent, otherwise

$$P(C) = P(A) + P(B) - P(A, B) \tag{6.8}$$

To incorporate dependency, let S_A and S_B are the strengths of A & B respectively, and applied load W be the random variables, considering this is the total load only for A & B components.

$\therefore P(A) = f(S_A, W)$ and $P(B) = f(S_B, W)$

Assuming that A takes l fraction of load, so B will take $\frac{l}{l-1}$ fraction of load. Hence, in case TF failure is observed, the failure probability of A & B and probability distributions of the corresponding random variables can be updated as:

$$\begin{aligned} P(A = F|C = F) &= \frac{P(A = F, C = F)}{P(C = F)} = \frac{P(A)}{P(C)} \\ &= \frac{S_A \leq \frac{w}{l}}{P(C)} \end{aligned} \tag{6.9}$$

$$\begin{aligned} P(B=F|C=F) &= \frac{P(B=F,C=F)}{P(C=F)} = \frac{P(B)}{P(C)} \\ &= \frac{S_B \leq \frac{\frac{w}{l}}{l-1}}{P(C)} \end{aligned} \tag{6.10}$$

$$f(s_A|C \quad = \quad F) \quad = \quad \frac{dF(s_A|C=F)}{ds_A} \quad = \quad \frac{d}{ds_A}(\frac{P(S_A \leq s_A, C=F)}{P(C=F)}) \tag{6.11}$$

From equations 6.1 and 6.2, the probability of TF failure is computed as:

$$\begin{aligned} P(C=F) = &\int\limits_{S_A}\int\limits_{S_B}\int\limits_{w} P(A=T|s_A,W)P(B=F|s_B,W)f(s_A)f(s_B)f(w)d(s_A)d(s_B)d(w)+ \\ &\int\limits_{S_A}\int\limits_{S_B}\int\limits_{w} P(A=F|s_A,W)P(B=T|s_B,W)f(s_A)f(s_B)f(w)d(s_A)d(s_B)d(w)+ \\ &\int\limits_{S_A}\int\limits_{S_B}\int\limits_{w} P(A=F|s_A,W)P(B=F|s_B,W)f(s_A)f(s_B)f(w)d(s_A)d(s_B)d(w) \end{aligned} \tag{6.12}$$

$$P(C=F) = P(A=T,B=F) + P(A=F,B=T) + P(A=F,B=F)$$

$$P(C=F) = \int\limits_{s_A \geq \frac{w}{l}, s_B \leq \frac{w}{\frac{1}{l-1}}} f(S_A)f(S_A)f(w)ds_A ds_B dw + \int\limits_{s_A \leq \frac{w}{l}, s_B \geq \frac{w}{\frac{1}{l-1}}} f(S_A)f(S_A)f(w)ds_A ds_B dw + \int\limits_{s_A \leq \frac{w}{l}, s_B \leq \frac{w}{\frac{1}{l-1}}} f(S_A)f(S_A)f(w)ds_A ds_B dw \quad (6.13)$$

$$\therefore P(C=F) = P(S_A \geq \frac{w}{l} \cap S_B \leq \frac{w}{\frac{1}{l-1}}) + P(S_A \leq \frac{w}{l} \cap S_B \geq \frac{w}{\frac{1}{l-1}}) + P(S_A \leq \frac{w}{l} \cap S_B \leq \frac{w}{\frac{1}{l-1}}) \quad (6.14)$$

MCS can be used to solve equation 6.14.

Now using equation 6.2,

$$P(S_A \leq s_A, C=F) =$$
$$\int\limits_{-\infty}^{S_A}\int\limits_{S_B}\int\limits_{w} P(A=T|s_A,W)P(B=F|s_B,W)f(s_A)f(s_B)f(w)d(s_A)d(s_B)d(w)+$$
$$\int\limits_{-\infty}^{S_A}\int\limits_{S_B}\int\limits_{w} P(A=F|s_A,W)P(B=T|s_B,W)f(s_A)f(s_B)f(w)d(s_A)d(s_B)d(w)+$$
$$\int\limits_{-\infty}^{S_A}\int\limits_{S_B}\int\limits_{w} P(A=F|s_A,W)P(B=F|s_B,W)f(s_A)f(s_B)f(w)d(s_A)d(s_B)d(w)$$

$$\begin{aligned}
&= P(S_A \geq \frac{w}{l} \cap S_B \leq \frac{w}{\frac{1}{l-1}} \cap S_A \leq s_A) + \\
&P(S_A \leq \frac{w}{l} \cap S_B \geq \frac{w}{\frac{1}{l-1}} \cap S_A \leq s_A) + \\
&P(S_A \leq \frac{w}{l} \cap S_B \leq \frac{w}{\frac{1}{l-1}} \cap S_A \leq s_A) \qquad (6.15)
\end{aligned}$$

Similarly distributions of other two variables R_B and W can be updated. Using equation 6.14 and 6.15, equations 6.9, 6.10 and 6.11 can be solved.

If input B fails, other nodes can be updated as

$$P(A = F|B = F) = \frac{P(A = F, B = F)}{P(B = F)} \qquad (6.16)$$

$$f(s_B|B = F) = \frac{dF(s_B|B = F)}{ds_B} = \frac{d}{ds_B}\frac{P(S_B \leq s_B, B = F}{P(B = F)} \qquad (6.17)$$

Again, distributions of other two variables R_A and W can be updated similarly. The estimate of system reliability can also be updated as component reliability gets change, as follows:

$$P(C = F|B = F) = \frac{P(C = F, B = F)}{P(B = F)} = \frac{P(B = F)}{P(B = F)} = 1 \qquad (6.18)$$

6.5 Experimental Validation

For the illustration of above concept, let R_A, R_B and W are independent. The failure probabilities of components A and B; and that of TF is given by Monte Carlo simulation as:

$$P(A = F) = P(S_A \leq \frac{w}{l}) = 0.000004792 \qquad (6.19)$$

$$P(B = F) = P(S_B \leq \frac{w}{\frac{l}{l-1}}) = 0.000002102 \qquad (6.20)$$

Table 6.1: Statistical Parameters of S_A, S_B and W

Statistical Parameters	S_A	S_B	w
Mean Value (requests/second)	16	22	14
Standard deviation	3.2	4	4.2

$$P(C=F) = P(S_A \geq \frac{w}{l} \cap S_B \leq \frac{w}{\frac{l}{l-1}}) + P(S_A \leq \frac{w}{l} \cap S_B \geq \frac{w}{\frac{l}{l-1}}) + P(S_A \leq \frac{w}{l} \cap S_B \leq \frac{w}{\frac{l}{l-1}}) \tag{6.21}$$

$$= 0.000001209 + 0.0000131 + 0.0000117$$

$$= 0.000026009$$

This result can be validated from the conventional reliability estimation method, which represents failure probability as the intersection of the two possible failure paths:

$$P(C=F) = P((S_A < \frac{w}{l} \cap S_B < w) \cap (S_B < \frac{w}{\frac{l}{l-1}} \cap S_A < w)) = 0.00003 \tag{6.22}$$

From equation 6.21 and equation 6.22, we see that the result of Bayesian and conventional approach are same. The mean values and standard deviations of R_A, R_B and W are given in table 6.1.

The failure probabilities of components A and B can be updated whenever TF failure probability gets observed, as shown in equation 6.23 and 6.24.

$$\begin{aligned} P(A=F|C=F) &= \frac{P(A=F, C=F)}{P(C=F)} = \frac{P(A)}{P(C)} \\ &= \frac{P(S_A \leq \frac{w}{l})}{P(C)} \\ &= \frac{0.000004792}{0.00003} = 0.1597 \end{aligned} \tag{6.23}$$

Table 6.2: Failure Probability of components A and B as a function of system C

$P(C=F)$	$P(A=F\|C=F)$	$P(B=F\|C=F)$
0.00003	0.1597	0.0701
0.00009	0.0532	0.0236
0.00015	0.0320	0.0140
0.00021	0.0228	0.0100
0.00027	0.0177	0.0078
0.00033	0.0145	0.0064
0.00039	0.0123	0.0054

$$\begin{aligned} P(B=F|C=F) &= \frac{P(B=F,C=F)}{P(C=F)} = \frac{P(B)}{P(C)} \\ &= \frac{P(S_A \leq \frac{w}{\frac{l}{l-1}})}{P(C)} \\ &= \frac{0.000002102}{0.00003} = 0.07007 \end{aligned} \tag{6.24}$$

The criticality of the software component can be known by analyzing the effect of its failure on the failure probability of the system, in case of parallel system. This is because of its infinite criticality in case of series system, like in our case study, the failure of any of A or B will fail the complete system. The same can be verified as:

$$\begin{aligned} P(C=F|A=F) &= \frac{P(C=F,A=F)}{P(A=F)} = \frac{P(A)}{P(A)} = 1 \\ P(C=F|B=F) &= \frac{P(C=F,B=F)}{P(B=F)} = \frac{P(B)}{P(B)} = 1 \end{aligned}$$

In case of the series system, the system maintainer can get a clue about the failure of a particular component or components of the system, if system failure is observed, by updating its component failure probability. This will help to bring up the system in a healthy state. Table 6.2 shows the update failure probability of components A and B, given failure probability of TF system C. We have assumed that the failure probabilities of components A and B are constant and have been known by Monte Carlo simulation, as given in equation 6.19 and equation 6.20 respectively. The analysis can be performed by drawing the chart, as shown in figure 6.2. The blue, red and green line shows P(C=F),P(A=F|C=F) and P(B=F|C=F) respectively. Interestingly, we see that on a slight increase of failure probability of system C, there is an exponentially decrease on the failure probability of components A and B, assuming the failure probability of A and B is constant.

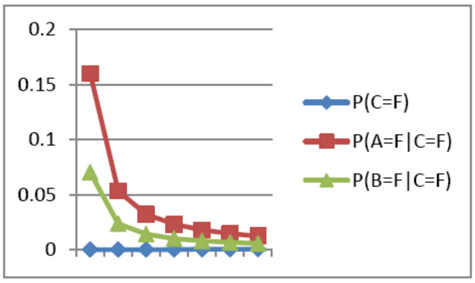

Figure 6.2: Failure probability of A, B and C.

6.6 Conclusion

We have used the BN to compute the updated estimate of reliabilities of the components of the CBS or CBS itself, whenever any of its component reliabilities or system reliability changes. Any CBS is composed of several components, which are arranged in series or parallel. Those individual components are accountable for the proper functioning of the system. Since the components are themselves coupled in some fashion, the change in any component reliability can affect the reliabilities of the components which are connected to it. We have devised an innovative method using BN and MCS, to update the estimate of individual component reliability or the system reliability. This can help the maintainer to take preventive action. We have validated our approach on a running safety critical CBS of NPP, known as Test Facility and shown the experimental results.