Combining evidence in risk analysis using Bayesian Networks

Norman Fenton and Martin Neil, 23 July 2004

Consider the following problem: You are in charge of a critical system, such as a transport system or a nuclear installation. The system is made up of many components that you buy as black boxes from different suppliers. When you need a new type of component you invite a dozen suppliers to tender. If you are lucky you might be able to get some independent test results or even operational test data on the components supplied. Your task is to accept or reject a component. One of your key acceptance criteria will be the safety of the component. This might be measured in terms of the predicted number of safety related failures that the component can cause in a ten year life-span when integrated into your system. How do you make your decision and justify it?

This is a classic risk assessment problem in which you have to come up with a quantified figure by somehow combining evidence of very different types. The evidence might range from subjective judgements about the quality of the supplier and component complexity, through to more objective data like the number of defects discovered in independent testing. In some situations you might have extensive historical data about previous similar components, whereas in other cases you will have none. Your trust in the accuracy of any test data will depend on your trust in the providence of the testers. Having little or no test data at all will not absolve your responsibility from making a decision and having to justify it. A decision based only on 'gut feel' will generally be unacceptable and, in any case, disastrous in the event of subsequent safety incidents with all the legal ramifications that follow.

Increasingly, the above type of risk assessment problem is being successfully addressed in a wide range of application domains using Bayesian Networks (BNs) [1,2,3,4]. BNs provide effective decision-support for problems involving uncertainty and probabilistic reasoning. In particular, they are uniquely effective in enabling quantitative assessments by combining the kind of diverse data above.

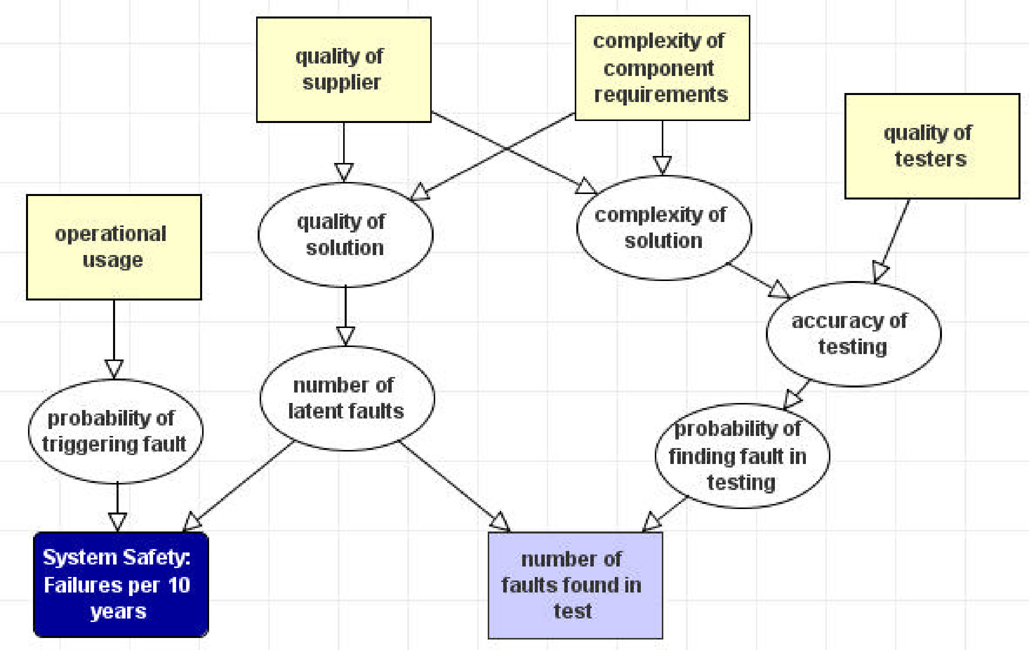

quality of supplier
complexity of component requirements
quality of testers
operational usage
quality of solution
complexity of solution
accuracy of testing
number of latent faults
probability of triggering fault
probability of finding fault in testing
System Safety: Failures per 10 years
number of faults found in test

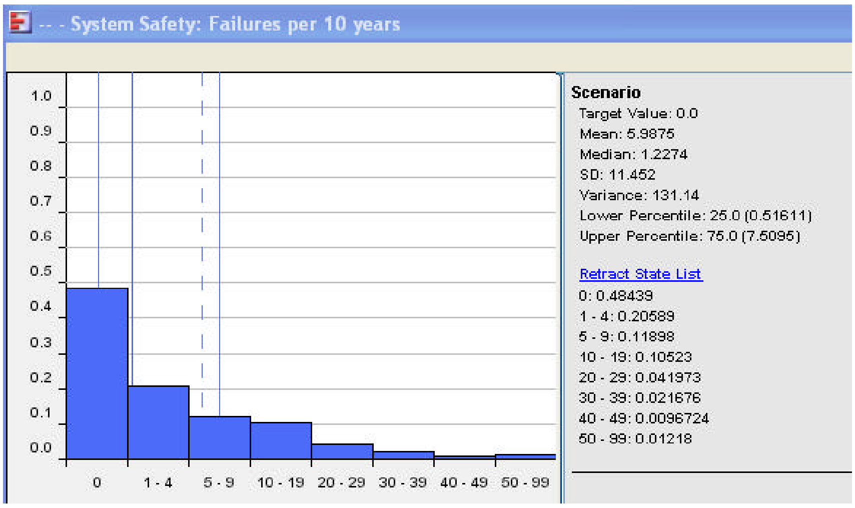
Figure 1 - BN example for component safety assessment

A BN is a directed graph, like the one shown in Figure 1, in which each node has an associated probability distribution. The nodes represent the variables relevant to your problem. In this example some nodes like *number of faults found in test* are numeric, while others like *quality of supplier* are ranked with values ranging from “very low” to “very high”. The arcs represent causal or influential relationships between variables. For example, the *number of faults found in test* is influenced by the *number of latent faults* and the *probability of finding a fault in testing*. The strength of influence of the parents is captured by the probability distribution for the node. In this case the distribution is widely accepted to be a Binomial distribution. The distribution for other nodes, such as *quality of solution*, is much less obvious and will generally need to be elicited from domain experts with or without relevant data. In the worst case this requires them to define a probability value for each possible combination of parent and child states. However, in many situations a distribution based around a single expression (such as a weighted sum of the parents) may be sufficient. Where a node, such as quality of supplier, has no parents the probability distribution is the so-called prior distribution. Thus, if you believe that 30% of suppliers have high quality then the probability associated with the state “high” is 0.3

The BN in Figure 1 is a simplified version of a model that we have used to solve exactly the kind of problem we introduced at the start. The yellow square nodes represent variables that we might expect to know for a given component. The *system safety* node will never be directly observed, but the node *number of faults found in test* might be. The round white nodes represent variables that we regard as ‘internal’ in the sense that they help us structure the model but are of no interest to the end-user of the model. Hence, we have hidden these in subsequent figures.

The BN enables us to make observations about known variables and infer the probabilities of others, which have not as yet been observed. It does this by using probability calculus and Bayes theorem throughout the model (this is called *propagation*). Thus, when we run this model in an appropriate BN tool such as AgenaRisk [6], we can view the status of any node given any number of observations made. By ‘status’ we mean the full probability distribution. For example, if we run the model with no observations we see the probability distribution for system safety as shown in Figure 2.

The resulting so-called marginal distribution for *system safety* here can be regarded as the population distribution for all components that we might assess. Thus, on average 48% of all components experience no failures in their 10-year life-time of use within our system, while about 1.2% experience more than 50 failures.



Scenario

Target Value: 0.0
Mean: 5.9875
Median: 1.2274
SD: 11.452
Variance: 131.14
Lower Percentile: 25.0 (0.51611)
Upper Percentile: 75.0 (7.5095)

0: 0.48439
1 - 4: 0.20589
5 - 9: 0.11898
10 - 19: 0.10523
20 - 29: 0.041973
30 - 39: 0.021676
40 - 49: 0.0096724
50 - 99: 0.01218

Figure 2 Marginal distribution for system safety

What we are really interested in, however, is how this distribution changes when we enter observations for a particular new component. We might have an acceptance criterion that the probability of more than 0 failures should be no less than some threshold value like 0.95. This is much higher than the starting point here of 0.48. Suppose we know that the quality of the supplier is “very high” and the complexity of the component requirements is “medium”. Then in the case where the operational usage is “high” the resulting revised distribution for safety is shown in Figure 3.

Although the prediction is better news (0.7 probability of 0 failures, compared to 0.48 before) it is clearly not good enough. Other than by lowering the operational usage of the component (the less it is used the less likely it is to fail) the only other way we can gain greater confidence in its safety is to subject it to independent rigorous testing. The best possible testing scenario is shown in Figure 4. Here, using the highest quality testing environment has resulted in 0 faults found. Consequently, our belief about the probability of 0 failures in practice increases to above the 0.95 target safety acceptance criterion.

Note, however, that this decision is reversed if we find out that the quality of testing was very poor as shown in Figure 5 .Although testing revealed 0 defects, the fact that we now know that testing quality was “very low” means that the testing information essentially tells us nothing new – the distribution for safety reverts back to more or less what it was without testing information and we cannot accept the component on this basis.

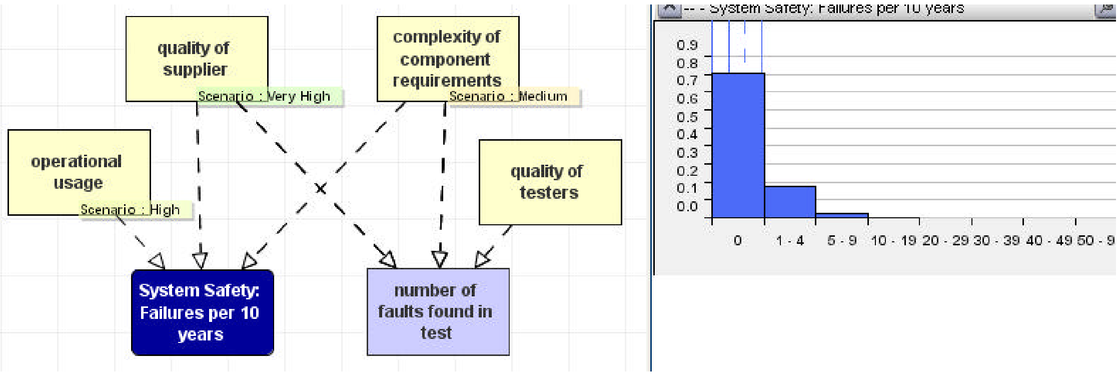


Figure 3 First revised prediction of safety

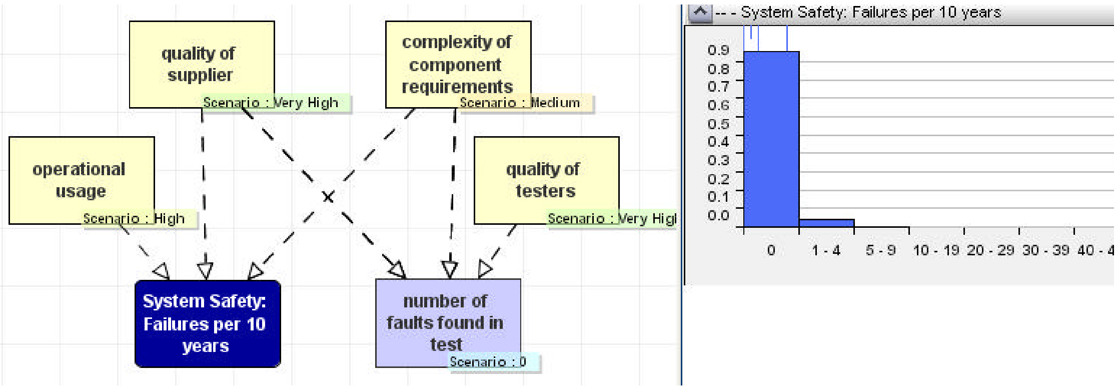


Figure 4 Second revised prediction of safety given best possible testing information

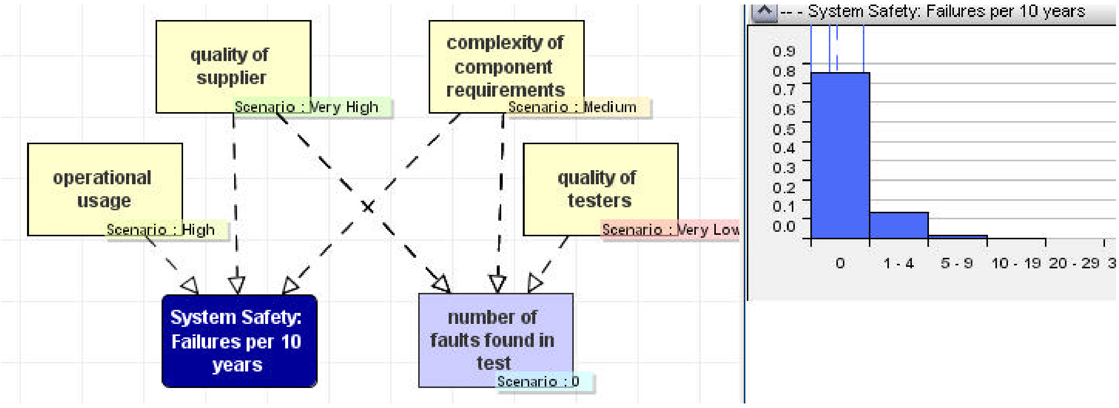


Figure 5 Third revised prediction of safety given revised testing information

In addition to the above kind of decision-support for risk assessment, we can use the BN to do various types of 'what-if' and sensitivity analysis. This is because we can enter observations into any node (including a target node such as *system safety*); the BN propagates information backward as well as forward to update the distributions of all the unknown variables. For example, suppose we set as a requirement the system safety to be 0 failures. If we also know the operational usage is "high" then in Figure 6 the BN back propagates to find explanations for the observations. In this case the most likely explanation is that the *complexity of component requirements* must tend toward low, although we also believe (less strongly) that the supplier quality must tend toward high.

Suppose we now observe that, in fact, the requirements complexity is "high" as shown in Figure 7. The model 'explains away' this observation by adjusting the belief about supplier quality, which is now almost certainly ay least 'high'. In fact, the BN model in these circumstances confirms that, given the operational and safety requirements, there is almost no chance that any supplier other than a high quality one could be used. The BN model is showing, in explicit probabilistic terms, that using anything other than the best supplier represents an intolerable risk.

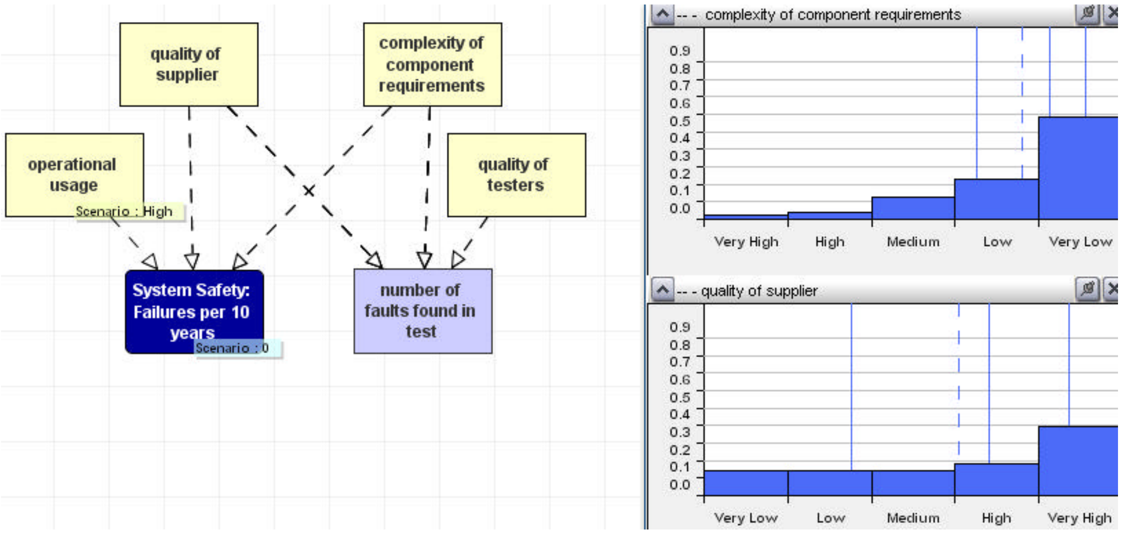


Figure 6 Example of backward inference

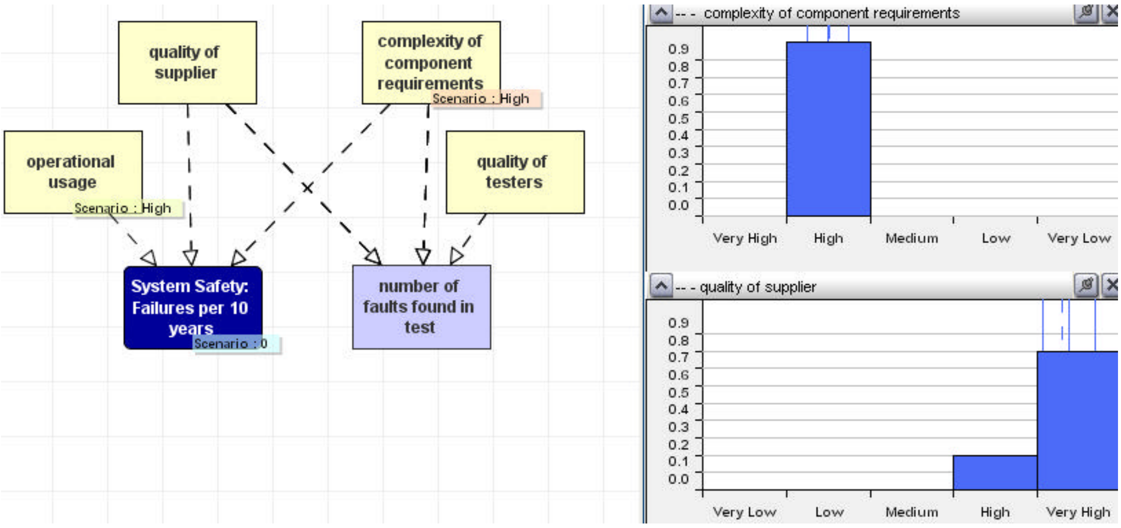


Figure 7 Revised belief about supplier quality given complexity of component requirements

The example demonstrates that BN models can be used to

- Communicate risk arguments easily and effectively
- Explicitly and rigorously quantify risk and uncertainty
- Combine diverse types of evidence including both subjective beliefs and objective data
- Reason from effect to cause and vice versa
- Overturn previous beliefs in the light of new evidence
- Make predictions with incomplete data
- Arrive at decisions based on visible auditable reasoning

All of this is good news for risk assessors, since these points address known previous impediments to rational decision-making. Even better news is that recent technological breakthroughs mean that most of the scalability problems that turned people away from using BNs have now been solved [5,6,7]. This technology has, for example, been packaged into the AgenaRisk toolset which enables:

- *Users with minimal statistical knowledge to build and edit large-scale realistic models for a range of application domains.* This has been achieved by a range of techniques including object oriented methods to support the structural model building, sets of pre-defined template models, and methods to enable large probability tables to be constructed very quickly using minimal expert involvement. For example, Philips, IAI and QinetiQ have been able to use these techniques for risk assessment in large-scale software-centric projects with acclaimed results [7].
- *Users to achieve results to arbitrarily high accuracy.* All previous generation BN tools required users to define any numeric scale as a sequence of pre-defined intervals (so, for example, instead of just specifying that the *number of faults* node ranges from 0 to 1000, you would have to specify in advance how to break up 0 to 1000 into a manageable number of intervals). The more intervals you define, the more accuracy you can achieve, but at a heavy cost of computational complexity. This process (called discretisation) was made worse by the fact that you do not necessarily know in advance which ranges require the finer intervals. The BN results were therefore necessarily only crude approximations. AgenaRisk solves this critical problem by providing so-called dynamic discretisation, enabling maximal

accuracy with no need for user intervention or set-up. For example, QinetiQ are using AgenaRisk's dynamic discretrisation to build complex, but accurate models to support critical decisions about vehicle obsolescence.

- *End users to interact via a questionnaire interface that hides all the model complexity*. Once a model is completed it can be easily packaged for many different types of users. Previous generation BN tools provided no-support for end-user decision makers; any decision support system based around a BN model had to be programmed from scratch. AgenaRisk enables non-programmers to tailor and generate attractive decision support systems based around a BN model in seconds.

Using BNs for quantitative risk assessment is therefore no longer an expensive time-consuming luxury. If you have to quantify risk using diverse information and communicate your decision to others then BNs are probably the best way to go about it.

References

1. Fenton NE, Krause P, Neil M, "Software Measurement: Uncertainty and Causal Modelling", IEEE Software 10(4), 116-122, 2002
2. Fenton NE, Marsh W, Neil M, Cates P, Forey S, Tailor T, 'Making Resource Decisions for Software Projects', 26th International Conference on Software Engineering (ICSE 2004), May 2004, Edinburgh, United Kingdom. IEEE Computer Society 2004, ISBN 0-7695-2163-0, pp. 397-406
3. Heckerman D, Mamdani A, Wellman M, "Real-world applications of Bayesian networks", Comm ACM, 38(3), 25-26, 1995.
4. Neil M, Fenton N, Forey S and Harris R, "Using Bayesian Belief Networks to Predict the Reliability of Military Vehicles", IEE Computing and Control Engineering J 12(1), 11-20, 2001
5. Neil M, Fenton NE, Nielsen L, "Building large-scale Bayesian Networks", The Knowledge Engineering Review, 15(3), 257-284, 2000.
6. www.agena.co.uk
7. www.serene.org