Using Bayesian Networks and Simulation for Data Fusion and Risk Analysis

Martin Neil, Norman Fenton and David Marquez
Queen Mary, University of London, and Agena Ltd, UK

Abstract. Bayesian networks (BNs) were pioneered to solve problems in Artificial Intelligence (AI) and have proven successful in "intelligent" applications such as medical expert systems, speech recognition, and fault diagnosis. In practical terms, one of the major benefits from using BNs is in that probabilistic and causal relationships among variables are represented and executed as graphs and can thus be easily visualized and extended, making model building and verification easier and faster. We illustrate how BNs can be used for risk analysis by introducing a novel approach modeling causal chains containing event triggers, consequences and interventions. However, if we want to incorporate continuous (as opposed to just discrete) variables in BN models the established BN tools and methods are inadequate. This paper reports on a new, unifying, approach to modeling continuous variables in BNs, called dynamic discretization, which approximates continuous variables without recourse to the traditional approach of Monte Carlo simulation methods. We illustrate the practical usefulness of the approach with an application involving the fusion of diverse sources of temporal data for fault diagnosis, classification and prediction of system behavior.

Keywords. Dynamic Bayesian Network, Simulation, Risk Analysis, AgenaRisk,

1. Introduction

Bayesian Networks (BNs) have been widely used to represent full probability models in a compact and intuitive way. In the BN framework the independence structure (if any) in a joint distribution is characterized by a directed acyclic graph, with nodes representing random variables, which can be discrete or continuous, and may or may not be observable, and directed arcs representing causal or influential relationship between variables [1]. The conditional independence assertions about the variables, represented by the absence of arcs, reduce significantly the complexity of inference and allow us to decompose the underlying joint probability distribution as a product of local conditional probability distributions (CPDs) associated to each node and its respective parents [2, 3]. If the variables are discrete, the CPDs can be represented as Node Probability Table (NPTs), which list the probability that the child node takes on each of its different values for each combination of values of its parents.

BNs were pioneered to solve problems in Artificial Intelligence (AI) and have proven successful in "intelligent" applications such as medical expert systems, speech

recognition, and fault diagnosis. In practical terms, one of the major benefits from using BNs is in that probabilistic and causal relationships among variables are represented and executed as graphs and can thus be easily visualized and extended, making model building and verification easier and faster. The power, generality and flexibility offered by BNs are now widely recognized and they are being successfully applied in diverse fields, including risk analysis and decision support.

Our own work in this area has produced solutions to a number of real world, high-stakes problems such as:

- Military vehicle reliability, [4];
- Risk of mid-air collisions in Air Traffic Control [5];
- Software defect prediction in consumer electronics products [6, 7];

As a simple illustration of the power of the BN approach we present (Section 2) a coherent and elegant method of defining and analyzing risks. This approach is based on modeling causal chains containing event triggers, consequences and interventions. This BN approach not only avoids the irrationality that characterizes many of the traditional, naïve, approaches to risk quantification, but it actually makes the whole thing much simpler.

If we want to incorporate continuous (as opposed to just discrete) variables in BN models the established BN tools and methods are inadequate. Hence (in Section 3) we describe a new, unifying, approach to modeling continuous variables in BNs, called dynamic discretization, which approximates continuous variables without recourse to the traditional approach of Monte Carlo simulation methods. In Section 4 we illustrate the practical usefulness of the approach with an application involving the fusion of diverse sources of temporal data for fault diagnosis, classification and prediction of system behavior.

2. Modeling Cause and Effect in Bayesian Networks

2.1. Risk Analysis

Our first example application of BNs is in the area of risk analysis. It focuses on exploiting cause and effect to better represent and structure risk problems and quantify risk in a simple, attractive way that is both easy to understand and communicate.

COSO [8] defines a risk as an ***event*** that can have negative impact (and conversely an event that can have a positive impact is an ***opportunity***). The RiskIT definition [9] is much more general, defining risk as:

"*a possibility of loss, the loss itself, or any characteristic, object or action that is associated with that possibility*".

We will use the COSO definition because it is clear and simple. But it turns out that the RiskIT definition can be regarded as equivalent using our approach.

Since a risk is an event that can have negative impact it follows that such events may be characterized by a causal chain involving (at least) the risk event itself and at least one consequence event (which characterizes the impact). Additionally there may be one or more *trigger* events, one or more *control* events, and one or more *mitigating* or inhibiting events. This is shown in the BN example of Figure 1 .

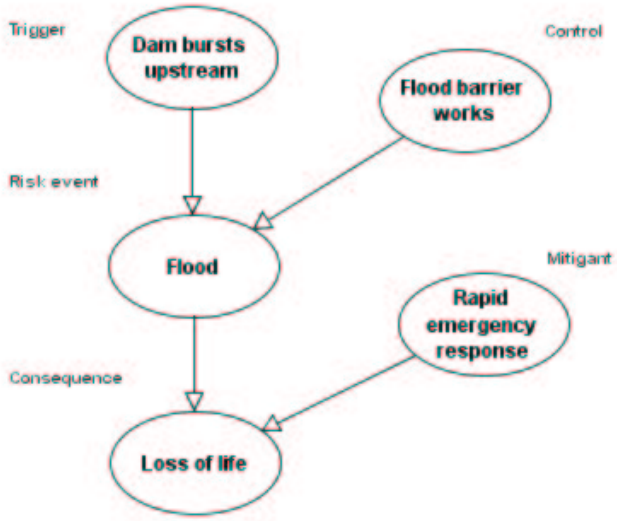

Figure 1 Causal taxonomy of risk

A risk is therefore characterized by a *set of events* as shown. These events each have a number of possible outcomes (in the simplest case you can assume each has just two outcomes *true* and *false*). Of course the 'uncertainty' associated with a risk is not a separate notion (as assumed in some approaches). Every event (and hence every object associated with risk) has uncertainty that is characterised by the probabilities of the event's outcomes.

Clearly risks in this sense depend on stakeholders and perspectives, but the benefit of this approach is that, once a risk event is identified from a particular perspective, there will be little ambiguity about the concept and a clear causal structure that 'tells the full story". For example, since "Flood" is the risk event (taking the central role in the diagram) the perspective must be of somebody who has responsibility for both the associated control and mitigant. Hence, in Figure 1 the perspective is definitely not that of, for example, a householder in the village, but rather something like the local authority responsible for amenities in the village. A householder's perspective of risk would be more like that shown in Figure 2.

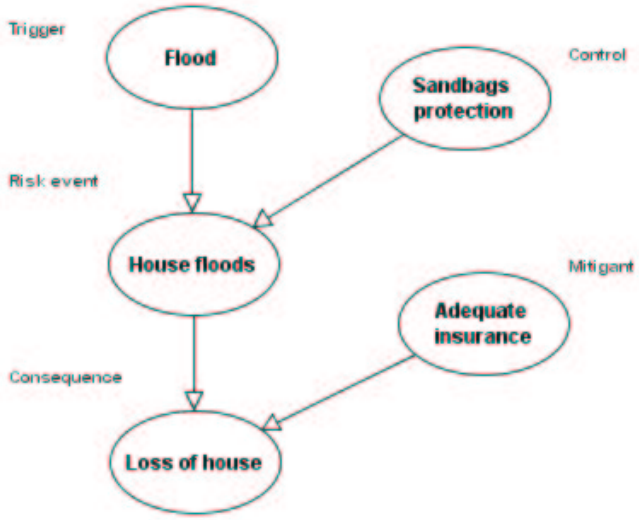

Figure 2 Flood risk from the householder perspective

What is intriguing is that the types of events are all completely interchangeable depending on the perspective. Consider the example shown in Figure 3. The perspective here might be the Local Authority Solicitor. Note that:

- The risk event now is “Loss of life”. This was previously the trigger
- “Flood” is no longer the risk event, but the trigger
- “Rapid emergency response” becomes a control rather than a mitigant

It is not difficult to think of examples where controls and mitigants become risk events and triggers. This interchangeability should be considered a benefit rather than a restriction since it stresses the symmetry and simplicity of the approach.

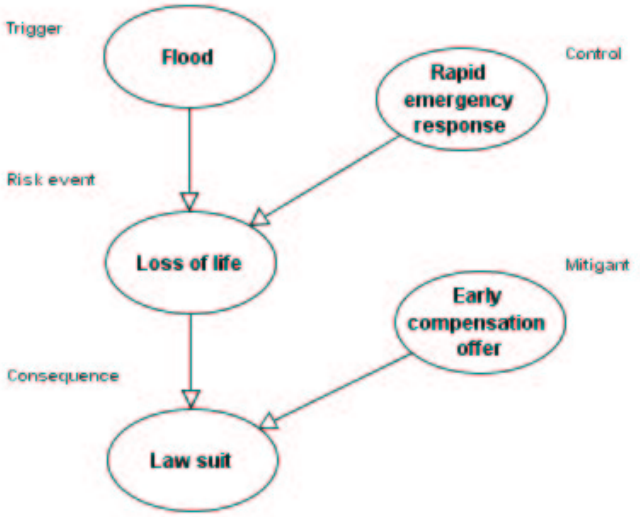

Figure 3 Interchangeability of concepts depending on perspective

2.2. Building BNs

Using software like AgenaRisk [10] it is possible to build arbitrarily large risk-based BNs based on the above approach. This involves the following steps:

1. Consider the set of risk events from a given perspective
2. For each risk event identify any triggers and/or controls
3. For each risk event identify any consequences and mitigants

By 'chaining' together different risks we can model multiple risks, risks from different perspectives, and common causes, consequences and mitigants, all within the same model. So you really can turn your risk list into a meaningful story. Next we show how to get the BN to generate quantified risk predictions and other useful functions like *simulation*, *backward reasoning* and *what-if-analysis.*

Once the topology of a BN has been constructed we need to add node probability tables (NPT) to it, which represent the quantitative specification of local dependencies between the nodes. This means we need to compute the conditional probabilities for each combination of events and states in the model. At first glance this looks easy but in practice this can prove tricky although not impossible to achieve by any means. To help in this there are means of eliciting and determining valid probability values but this is outside the scope of this paper.

2.3. Executing BNs

Once armed with the BN and the NPTs we can then run a BN propagation algorithm to calculate the marginal probabilities of any node given any configuration of valid (i.e. non contradictory evidence). A range of robust and efficient propagation algorithms has been developed for exact inference on Bayesian networks with discrete

variables [11, 12, 13]. The common feature of these algorithms is that the exact computation of posterior marginal distributions is performed through a series of local computations over a secondary structure, a tree of node clusters, which allows calculating the marginal without computing the joint distribution. See also [14].

Figure 4 shows a sample NPT for the Risk Event node "Flood?" along with the marginal probability distributions for all nodes in the model, as calculated using the AgenaRisk propagation algorithm (although in this example all nodes are Boolean but BNs can handle many valued discrete states).

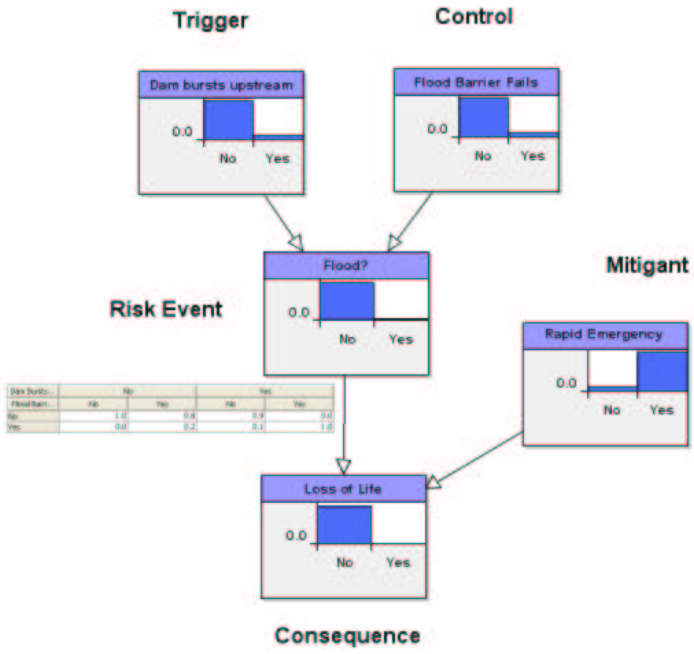

Figure 4 BN with NPT for node "Flood?"

The posterior marginal probabilities for all nodes in the model, as shown in Figure 4, quantify our degree of belief in the consequence given our uncertainties about the trigger, control and mitigation events. In this case the marginal probability of "Loss of Life" is 0.00703.

BNs are dynamic; as new data is entered the model learns or updates the prediction to take account of changed circumstances. This makes them highly suitable for sensitivity analysis, parameter learning and "what if?" analysis. Take for example the case where the Dam Bursts event has actually occurred and the Flood Barrier has failed: we enter this new data as "evidence" into the BN and recalculate the model to achieve the new marginal posterior distributions as shown in Figure 5. Now, given this new evidence, the probability of Loss of life has increased to 0.19 from the 0.00703.

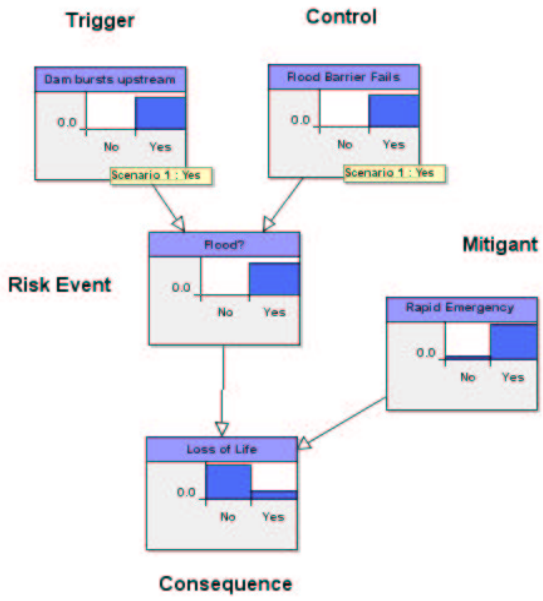

Figure 5 BN with evidence entered on Trigger and Control and Consequence prediction updated

3. Simulation by Dynamic Discretization

3.1. Overview

In the previous risk analysis problem we used nodes with discrete states only. Given that many of the quantities of interest in real problems will be continuous valued variables this presents a severe, or even terminal, constraint on using BNs for more general risk modeling purposes. Here we introduce a new approach, embedded within AgenaRisk that allows the calculation of both discrete and continuous quantities within a BN. This approach, called *dynamic discretization*, is a major step forward and presents a clear alternative to Monte Carlo simulation.

The present generation of BN software tools [15, 16] attempt to model continuous nodes by numerical approximations using static discretization Discretization allows approximate inference in a "hybrid" BN without limitations on relationships among continuous and discrete variables. However, current software implementations require users to define a discretization of the states of any numeric node (whether it is continuous or discrete) as a sequence of pre-defined intervals, which remain static throughout all subsequent stages of Bayesian inference regardless of any new conditioning evidence. The more intervals you define, the more accuracy you can achieve, but at a heavy cost of computational complexity. This is made worse by the fact that you do not necessarily know in advance where the posterior marginal distribution will lie on the continuum for all nodes and which ranges require the finer

intervals. It follows that, where a model contains numerical nodes having a potentially large range, results are necessarily only crude approximations.

Let X be a continuous random node in the BN. The range of X is denoted by Ω_X, and the probability density function (PDF) of X, with support Ω_X, is denoted by f_X. The idea of discretization is to approximate f_X by, first, partitioning Ω_X into a set of interval $\Psi_X = \{w_j\}$, and second, defining a locally constant function $\tilde{f}_X$ on the partitioning intervals. The task consists in finding an optimal discretization set $\Psi_X = \{\omega_i\}$ and optimal values for the discretised probability density function $\tilde{f}_X$. Discretization operates in much the same way when X takes integer values but here we will focus on the case where X is continuous. The approach to dynamic discretization described here searches Ω_X for the most accurate specification of the high-density regions (HDR), given the model and the evidence, calculating a sequence of discretization intervals in Ω_X iteratively. At each stage in the iterative process a candidate discretization, Ψ_X, is tested to determine whether the resulting discretised probability density $\tilde{f}_X$ has converged to the true probability density f_X within an acceptable degree of precision. At convergence, f_X is then approximated by $\tilde{f}_X$.

3.2. Dynamic discretization: An Example

To illustrate the outputs of the dynamic discretization approach let us consider the probability distribution for a sum of two independent random variables $Z = X + Y$, where $X \sim f_X$ and $Y \sim f_Y$, given by the convolution function:

$$f_Z(z) = f_X \times f_Y(z) = \int f_X(x) f_Y(z-x) dx$$

Calculating such a distribution represents a major challenge for most BN software. Traditional methods to obtain this function include Fast Fourier Transform (FFT) [17] or Monte Carlo simulation. Here we compare an example and solution using AgenaRisk with the analytical solution produced by convolution of the density functions.

Consider the case $f_X = Uniform(-2, 2)$ and $f_Y = Triangular(0, 0, 2)$. The probability density for $Z = X + Y$ can be obtained analytically by

$$f_Z(z) = \int_0^{2+z} (1/4 + x/8) dx + \int_0^2 (1/4 + x/8) dx + \int_{z-2}^0 (1/4 + x/8) dx$$

The resulting mean and variance are $E[Z] = 0.667$ and $Var(Z) = 1.555$.

Using dynamic discretization, over 40 iterations, results in the set of marginal distributions for $f_Z = f_X \times f_Y$ as shown in Figure 6. The summary statistics are

$\mu_Z = 0.667$ and $\sigma_Z^2 = 1.559$, which are very accurate estimates of the analytical solution.

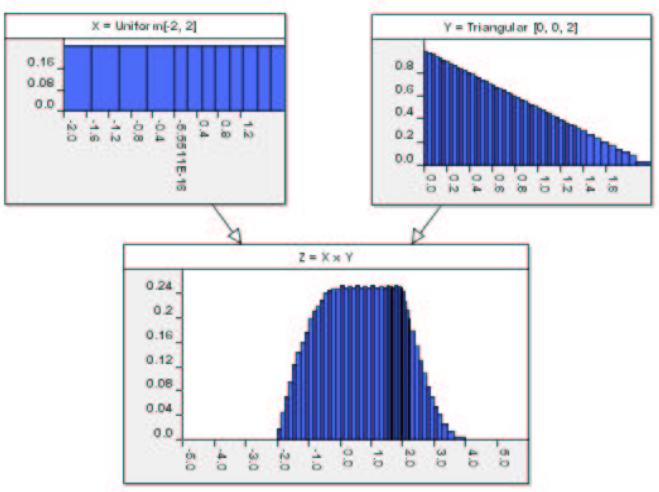

Figure 6 Marginal distributions from function $f_Z = f_X \times f_Y$ after 40 iterations as shown in AgenaRisk

4. Data Fusing using Dynamic Bayesian Networks and dynamic discretization

In this application we are extending the causal approach inherent in BNs and extending them to the temporal dimension to illustrate how:

- Temporal models can be built and constructed
- Unknown states of a system can be inferred as data is collected and this can be used to learn and predict behavior
- This approach, embedded in a Dynamic Bayesian Network (DBN), can be used for online monitoring and supervision of risky control problems.

Typically such dynamical systems are composed of mixtures of discrete and continuous nodes and hence we need methods like dynamic discretization to produce approximate solutions.

Here we wish to estimate the probability of a sensor being in a faulty state at any point in time. We are given a sequence of sensor readings taken from a system whose state is represented by a position and velocity vector (P_t, V_t) where the nodes are hidden and evolve over time. We also assume the system is self-repairing and that these repairs are random (or the fault might be transient).

We can model this as a Switching Kalman Filter Model (SKFM) and typically would have to use Monte Carlo or more complex methods to find a solution (see [18] for details). In AgenaRisk we model this as a Dynamic BN (DBN), a model containing temporal nodes, as follows.

The first element in the DBN is a double Kalman Filter to model the dynamical elements of the system. This comprises:

- *Observation model* O_t to filter sensor noise from observations:

$$p(O_t \mid P_t) = Normal(P_t, \sigma^2)$$

- *Transition model* of two difference equations for P_t and V_t:

$$P_t = P_{t-1} + V_{t-1},\ V_t = V_{t-1}$$

- *Initial conditions* for each of the system variables:

$$V_0 = N(0, \theta_1), P_0 = N(0, \theta_2)$$

To model the fact that the sensor is self-repairing we require a transition model for the sensor, S_t:

$$p(S_t = OK \mid S_{t-1} = OK) = 0.99$$
$$p(S_t = Faulty \mid S_{t-1} = OK) = 0.01$$
$$p(S_t = Faulty \mid S_{t-1} = Faulty) = 0.9$$
$$p(S_t = OK \mid S_{t-1} = Faulty) = 0.1$$

Sensor reliability is modelled in the observation model by conditioning the variance parameter, σ^2, on the sensor's state:

$$p(O_t \mid P_t, S_t = OK) = Normal(P_t, \sigma^2 = 10)$$
$$p(O_t \mid P_t, S_t = \neg OK) = Normal(P_t, \sigma^2 = 1000)$$

The DBN graph for the complete model over seven time periods is shown in Figure 7:

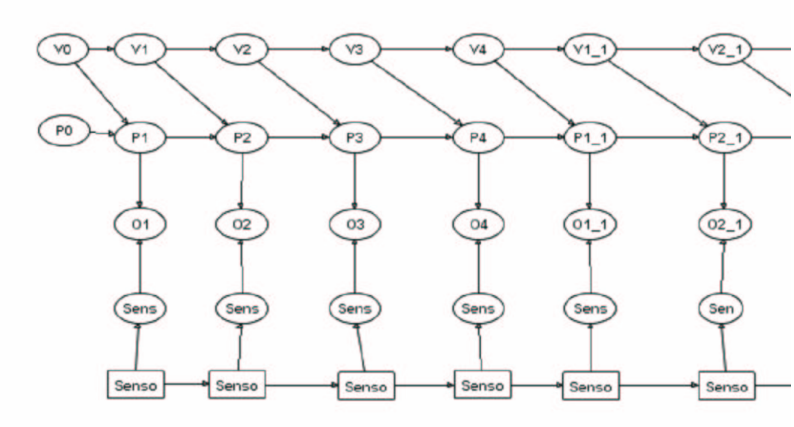

Figure 7 DBN graph for the SKFM for sensor reliability for seven time periods

We use dynamic discretization to solve the model over 25 iterations for a set of simulated actual and observed positions. At each time period we estimate the probability of the sensor being in a faulty state, represented by S_t. The results are shown in Figure 8 along with the corresponding time series plot of A_t (simulated actual), O_t (observed) and $p(S_t = OK)$ (shown as %OK).

Notice that the initial observations lead to a rapid increase in the probability that the sensor is faulty. However, after time period 8 the observed position better tracks the actual position and as a consequence the probability that it has repaired itself also increases.

time	At	Ot	%OK
0	9	10	99
1	12	20	96
2	19	17	92
3	22	30	74
4	24	20	50
5	25	20	35
6	31	10	4
7	34	25	4
8	43	40	24
9	45	47	29
10	59	55	31

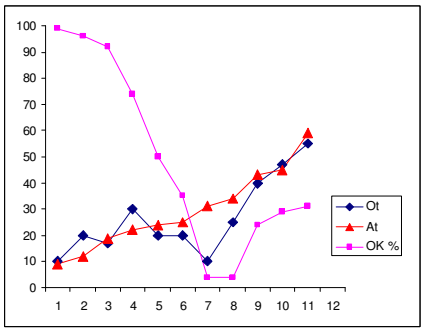

Figure 8 Table and time series plot of system position and probability of sensor failure

5. Conclusions

The power, generality and flexibility offered by BNs are now widely recognized and they are being successfully applied in diverse fields, including risk analysis and decision support. This paper reports on a new, unifying, approach to modeling

continuous variables in BNs, called dynamic discretization, which approximates continuous variables without recourse to the traditional approach of Monte Carlo simulation methods.

We have also illustrated the applicability of BN modeling in two very diverse application areas: risk assessment and sensor diagnosis. We will present the conclusions for each separately.

5.1. Risk analysis

The BN approach presented satisfies minimalist requirements described by Chapman and Ward in [19] where they recommend that any approach to risk quantification:

"should be so easy to use that the usual resistance to appropriate quantification based on lack of data and lack of comfort with subjective probabilities is overcome".

Moreover, the approach ensures that:

- Every aspect of risk measurement is meaningful in the context – the BN tells a story that makes sense. This is in stark contrast with the "risk = probability x impact" approach where not one of the concepts has a clear unambiguous interpretation.
- Every aspect of uncertainty is fully quantified since at any stage we can simply read off the current probability values associated with any event.
- It provides a visual and formal mechanism for recording and testing subjective probabilities. This is especially important for a risky event where you do not have much or any relevant data about.

Although the approach does NOT explicitly provide an overall risk "score" and prioritization these can be grafted on in ways that are much more meaningful and rigorous.

5.2. Data fusion

Typically many applications of risk take place in a context where decisions about control and intervention have to be made. Our approach, using Dynamic Bayesian Networks (DBNs), has shown how we can model and embed risk measurement in systems that evolve and change over time. One can easily imagine worthwhile and interesting applications of these ideas in numerous areas:

- Intelligent tracking of risks in complex organizational systems
- Automatic updating of risk models given new observations
- Classification of hidden risky states in agent based systems
- Automatic hypothesis selection to characterize system behavior

Acknowledgements

This report is based in part on work undertaken on the eXdecide: "Quantified Risk Assessment and Decision Support for Agile Software Projects" project, EPSRC project code EP/C005406/1, March 2005 - Feb 2008.

References

[1] Pearl J. 1993. Graphical models, causality, and intervention, Statistical Science, vol 8, no. 3, pp. 266-273.
[2] Speigelhalter D.J., and Lauritzen S.L. 1990. Sequential updating of conditional probabilities on directed graphical structures, Networks, 20, pp. 579-605.
[3] Lauritzen S.L. 1996. Graphical Models, Oxford.
[4] Neil M., Fenton N., Forey S., and Harris R. 2001. Using Bayesian Belief Networks to Predict the Reliability of Military Vehicles, IEE Computing and Control Engineering J 12(1), 11-20
[5] Neil M., Malcolm B., and Shaw R. 2003a. Modelling an Air Traffic Control Environment Using Bayesian Belief Networks, 21st International System Safety Conference, Ottawa, Ontario, Canada.
[6] Fenton N., Krause P., and Neil M. 2002. Probabilistic Modelling for Software Quality Control, Journal of Applied Non-Classical Logics 12(2), pp. 173-188
[7] Neil M., Krause P., and Fenton N. 2003b. Software Quality Prediction Using Bayesian Networks in Software Engineering with Computational Intelligence, (edited by Khoshgoftaar T. M). The Kluwer International Series in Engineering and Computer Science, Volume 73 approach', International Journal of Project Management, 18, 369-383, 2000.
[8] COSO (Committee of Sponsoring Organisations of the Treadway Commission), 'Enterprise Risk Management: Integrated Framework', www.coso.org/publications.htm, 2004.
[9] Kontio J, 'The Riskit Method for Software Risk Management', version 1.00, CS-TR-3782, 1997. Computer Science Technical Reports. University of Maryland. College Park, MD, 1997
[10] Agena Ltd. 2006. AgenaRisk Software Package, www.AgenaRisk.com
[11] Lauritzen S.L., and Speigelhalter D.J. 1988. Local Computations with Probabilities on Graphical Structures and their Application to Expert Systems (with discussion), Journal of the Royal Statistical Society Series B, Vol. 50, No 2, pp.157-224.
[12] Shenoy P., and Shafer G. 1990. Axioms for probability and belief-function propagation, Readings in uncertain reasoning, Morgan Kaufmann Publishers Inc, p.p. 575 - 610
[13] Jensen F., Lauritzen S.L., and Olesen K. 1990. Bayesian updating in recursive graphical models by local computations, Computational Statistics Quarterly, 4: 260-282
[14] Huang C., and Darwiche A. 1996. Inference in belief networks: A procedural guide. Int. J. Approx. Reasoning 15(3): 225-263
[15] Hugin. 2005. www.hugin.com
[16] Netica. 2005. www.norsys.com
[17] Brigham E. 1988. Fast Fourier Transform and Its Applications. Prentice Hall; 1st edition.
[18] Murphy K. P. 2001. A Brief Introduction to Graphical Models and Bayesian Networks. Berkeley, CA: Department of Computer Science, University of California - Berkeley.
[19] Chapman C and Ward S, 'Estimation and evaluation of uncertainty: a minimalist first pass approach', International Journal of Project Management, 18, 369-383, 2000.