Examination of Bayesian Belief Network for Safety Assessment of Nuclear Computer-based Systems

Bev Littlewood, Lorenzo Strigini, David Wright,
Norman Fenton, Martin Neil

*Centre for Software Reliability, City University,
Northampton Square, London EC1V 0HB, U.K.*

P.-J. Courtois

AIB-Vincotte Nuclear, Avenue du Roi 157, 1060 Brussels, Belgium

Abstract

Safety assessment for highly critical system differs from other performance evaluation tasks in various respects. Statistical evidence is usually insufficient for assigning model parameters with any confidence before operation of a new system, and for a long time into the operation period itself. On the other hand, a high degree of confidence is sought that the system will perform as safely as required. The assessors use disparate forms of evidence to reach this confidence, usually via their own expert judgement, a process which is poorly understood and subject to well-documented problems. Explicit, probabilistic formal reasoning is a way for the assessors to control the risks of intuitive judgement. We report on an exercise in using the formalism of Bayesian Belief Networks to support such formal probabilistic reasoning, the various difficulties encountered and methods for resolving them.

Key words: safety assessment, Bayesian belief networks, expert judgement, inference.

1 Introduction

Safety-critical equipment for regulated industries must undergo a formal safety assessment before it can be operated. This is a difficult task. The assessor must consider the possibility of design and realisation faults that would impair safety. Especially the increasing dependence on software-based systems has increased concern about safety assessment with respect to design faults. Although equipment vendors may operate to the best known standards of practice, these are known not to guarantee freedom from design faults in every single case.

Safety assessment uses disparate evidence like known conformance to standards of design and methodology, demonstrated competence of the organisations involved in producing a system, results of verification and validation activities on various products of the design process. Deriving a single judgement of satisfactory safety from all this evidence is usually an informal process of "expert judgement", which may be unreliable and is difficult to analyse and verify. The "safety argument" - the reasoning that links the evidence to the final judgement - is mostly in the assessor's mind, and its descriptions on paper are typically limited to enumerations of items of evidence, without a detailed explanation of how these are assumed to support or counter one another.

We have looked for explicit, formal ways of describing safety arguments, and chosen "Bayesian belief networks" (BBNs), a probabilistic notation for describing relationships between many variables in terms of conditional distributions and conditional independence relations.

We hope that the use of BBNs may make the hidden safety arguments visible, communicable and auditable. BBNs offer a formal mathematical language for describing reasoning in uncertain situations. The assessor can thus describe his "safety argument" in a form that he can re-examine and "debug". He can describe the causal models that he has assumed to apply to the situation considered. This description implicitly specifies the value of the evidence considered in predicting the safety of the product, and the inference process can be automatically performed by software tools. At the same time, the formal description allows experts to analyse and discuss the constituent parts of the "safety argument", devise empirical tests of their validity, and so on.

We summarise here a case study that was run as part of the European long-term research project "DeVa" (Design for Validation). A more detailed account can be found in two technical reports [1,2]. This report concentrates on issues of elicitation and validation of a BBN, as evidenced in the case study.

In section 2 we describe the essential characteristics of the BBN formalism; in section 3 we recall the context of our case study and in section 4 the BBN model we produced. In section 5, we discuss issues of validation of the model, which we tried to address 6 by developing methods for feed-back to the assessors of various implications of the BBN model. A discussion of our results and future developments follows in section 7.

2 Bayesian Belief Networks

A belief network is a directed acyclic graph, like the one in Figure 1, associated with a set of probability values. We can use a belief network to represent our uncertain, probabilistic knowledge about a real-world situation. Each node represents a set of events (a partition on the set of outcomes of an experiment or observation), e.g. the values of a numerical random variable. The events are called the possible "values" or "states" of the node. Each node has a "node probability table" (NPT) associated with it. If the node has no incoming arcs (root node), this table lists the marginal probabilities of its possible values; if it has n incoming arcs, the table lists the probabilities of its values, conditional on each possible n-tuple of values of its "parent" nodes. This information represents the fact that knowledge about a (parent) node is useful for predictions about another (child) node: either through cause-effect relationships, or via more general correlation laws. The absence of an arc between two nodes, say A and B, represents conditional independence. Roughly speaking, it means that any way the knowledge of the state of A might influence our expectations about the probabilities of states of B is already represented by other nodes in the BBN, that do have arcs joining them to B.

After building a BBN, i.e., choosing a topology and filling the NPTs, one can use an automated tool (e.g. the Hugin tool., for which information is available at http://www.hugin.dk) to:

- calculate the probabilities of the values of all nodes with incoming arcs, from the conditional and marginal probabilities of the values of their ancestor nodes;
- when an event (value of a node) is actually observed, update the (prior) probabilities given by the user to other events in the table, by repeatedly applying Bayesian inference to "propagate" the new knowledge along the arcs in the graph, and obtain (posterior) probabilities that take into account the events observed. Software tools supporting Bayesian networks have been made possible by recent, efficient algorithms for applying this rule repetitively through a large network.

A BBN can model one's (uncertain) knowledge about a situation, and also the arguments that can be built to support a thesis about the probabilities of events in the BBN itself, on the basis of the probabilities of other events. An informal judgement can be formalised into a belief network, in that one can specify a series of links of the form "the truth of statement A would support my belief in statement B", *and* can specify how much the truth of A strengthens this belief in B, compared e.g. to how much some other truth C would weaken it. On this basis, the propagation operation determines the "normatively correct" inference to be drawn from any combination of observations.

Applications of BBNs to safety and reliability issues are documented in [3,4], and have been the subject of another European research project [5].

BBNs offer the advantages of a formal probabilistic model presented in an easily assimilated visual form, together with efficient computational methods and tools for exploring model consequences. BBNs are clearly useful to summarise large models of probabilistic dependencies among variables. The model itself may be obtained from known physical and mathematical laws and statistical information, as possible, for instance, in applications to medical decision support systems. An example of such a BBN for software dependability assessment is in [3]. But another attractive feature of BBNs is that experts can represent laws that they conjecture or believe to be true, or even that they unconsciously apply. So, intuitive expert judgement can to some degree be opened to criticism, checked for consistency and challenged in terms of its constituent assumptions, and it can be integrated with other knowledge using the formal rules of probability calculus. Building, analysing and in the end trusting these representations of expert judgement procedures clearly poses serious problems, and these were the focus of this case study.

3 Context of the case study.

This case study dealt with the safety argument for a class of software-based systems used in nuclear plant for functions important to safety. The exercise involved a group of researchers on the application of BBNs to safety assessment and an expert of the class of equipment and safety problems concerned. We expect most uses of BBNs to require a similar collaboration between experts of the formalism and domain experts. For obvious reasons of professional discretion, the identity of the nuclear operators, manufacturers, systems and functions from which the expertise captured by the BBN is in part drawn has been kept confidential and not even revealed to co-authors.

To contain the effort required, we limited the exercise to describing a part of the safety argument. This BBN addresses the early part of the lifecycle of [the computer part of] a nuclear safety system, during which two documents are produced, the "System Requirements Document" and the "Computer System Specification Document", and subjected to various analyses. Its goal variable (the variable about which predictions are sought) is 'Safety Adequacy of Computer System Specification'. It is through this quality that the results of this phase of development affect successive phases. These two main documents are produced and modified through the interaction of three 'personae' (each typically consisting of a team or subset of an organisation): the *system manufacturer;* the *system licensee* (future user of the system) and the *independent assessor*, who works on behalf of a safety authority and is responsible for eventually recommending approval of the system from the safety viewpoint. In our scenario, the independent assessor has [partial] visibility (through access to documents and personnel) of the development process that produces and validates these documents, rather than being required to evaluate the finished product and documentation only.

In more detail, the two documents have these functions:–

(1) the System Requirements Document describes the environment of operation as well as the functions of the safety system. It lists the system's foreseeable failure modes, with probabilities, criticality and intended lines of defence against each of these modes, and assesses the criticality of the system;

(2) the 'Computer System Specification Document' specifies and justifies, among other things, the allocation of safety functions between hardware and software, and must demonstrate that the computer system architecture satisfies the system and safety requirements, in particular concerning adequate levels of redundancy and diversity, and barriers between safety and non-safety functions. A "failure modes and effects analysis" should also be included in terms of the software and hardware components, and the methods and mechanisms of auto-detection by the system of its own failure should be specified.

4 The BBN model and its construction.

Figure 1 shows the topology of the BBN produced. Its general structure strongly reflects the life-cycle model used, roughly divided into three subgraphs, divided by dotted lines in the figure. The three subgraphs concern (from bottom to top in the figure) the quality of the requirements document, the design process that leads from this to the computer system specification, and the quality of the specification document itself.

An important part of the safety argument is a detailed specification of the meaning of each node, which we omit for lack of space. A few conventions will help to interpret this BBN: i) the names of nodes are reasonably self-explanatory, *if read in the context* of the subgraph to which the node belongs: thus, for instance, the node named 'Completeness & Correctness' in the bottom part of the figure refers to the completeness and correctness of the requirement document; ii) we have appended an asterisk to the names of those nodes that represent observable variables; iii) when a variable is defined in terms of subjective judgement or observation, the observer or judge is the independent assessor, unless otherwise specified. Defining a node or variable as 'observable' means that we expect that at some stage of applying the BBN model to the

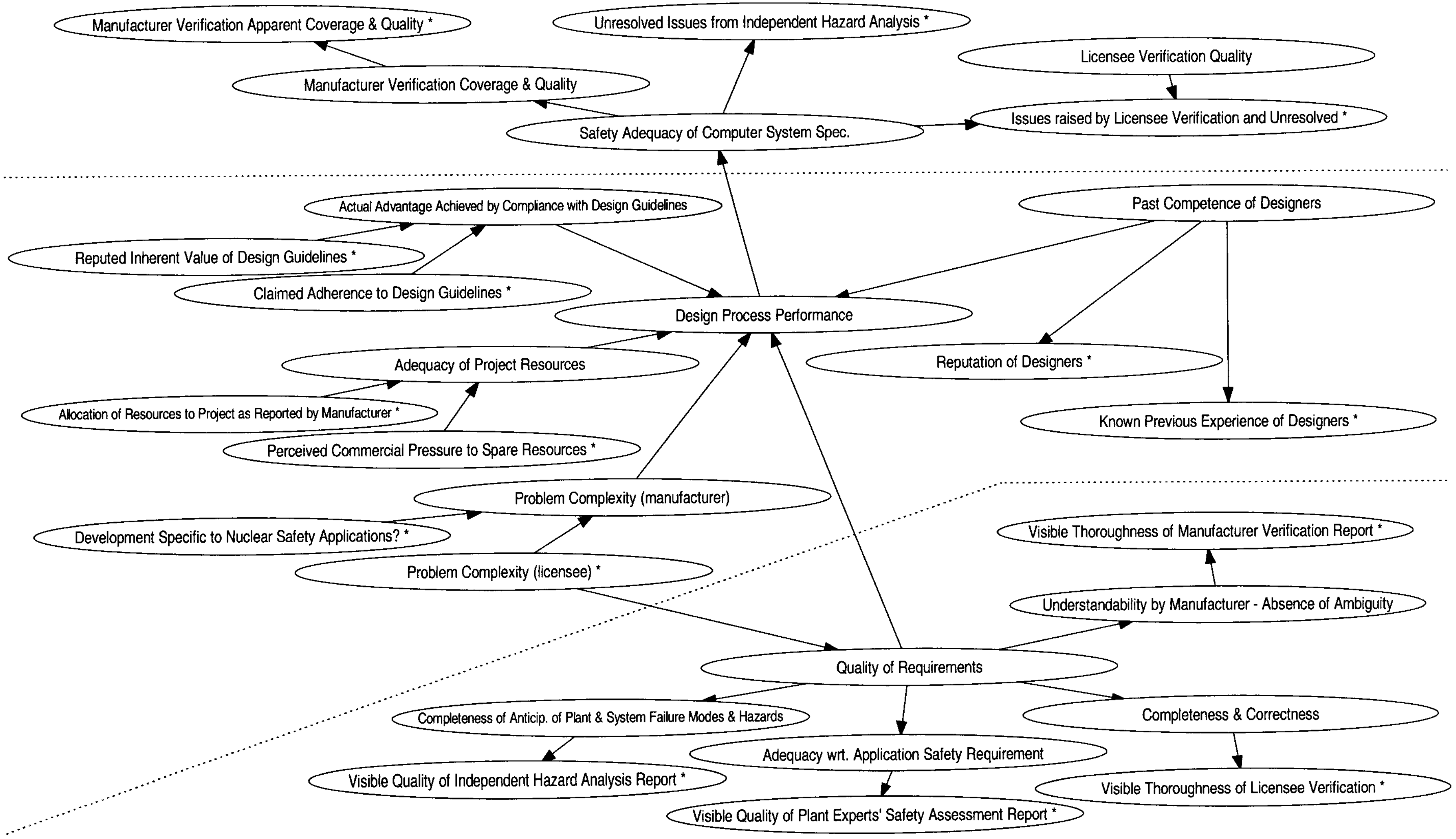

Manufacturer Verification Apparent Coverage & Quality *
Unresolved Issues from Independent Hazard Analysis *
Licensee Verification Quality
Manufacturer Verification Coverage & Quality
Issues raised by Licensee Verification and Unresolved *
Safety Adequacy of Computer System Spec.
Actual Advantage Achieved by Compliance with Design Guidelines
Past Competence of Designers
Reputed Inherent Value of Design Guidelines *
Claimed Adherence to Design Guidelines *
Design Process Performance
Adequacy of Project Resources
Reputation of Designers *
Allocation of Resources to Project as Reported by Manufacturer *
Known Previous Experience of Designers *
Perceived Commercial Pressure to Spare Resources *
Problem Complexity (manufacturer)
Visible Thoroughness of Manufacturer Verification Report *
Development Specific to Nuclear Safety Applications? *
Problem Complexity (licensee) *
Understandability by Manufacturer - Absence of Ambiguity
Quality of Requirements
Completeness of Anticip. of Plant & System Failure Modes & Hazards
Completeness & Correctness
Adequacy wrt. Application Safety Requirement
Visible Quality of Independent Hazard Analysis Report *
Visible Thoroughness of Licensee Verification *
Visible Quality of Plant Experts' Safety Assessment Report *

Fig. 1. BBN topology. The dotted lines separate the three sub-graphs described in the text.

assessment of a system, the user will enter a value for that node (or even possibly an assignment of probabilities to its possible values - a 'likelihood observation' in BBN jargon). For this BBN, one can see that most *leaf nodes* (the nodes having one arrow attached to them) will normally be observable to the independent assessor: and most of the other nodes will normally be unobservable to the independent assessor.

The meanings of the nodes must be carefully defined, and will be specific to the particular context in which the expert assessors operate, and may not be obvious to non-experts. For instance, there are two nodes labelled **Problem Complexity**, indicating the inherent difficulty of the development tasks: **Problem Complexity (manufacturer)** and **Problem Complexity (licensee)** because the manufacturer may replace the licensee's problem by a different problem, of a greater or lesser complexity - e.g. they may want to make the equipment marketable for a more general set of uses than required in the application for which it is now being assessed.

The possible states of a node in this BBN are usually ordered on a scale of increasing or decreasing 'quality', e.g., in the 'Requirements Document' subgraph the node **Quality of Requirements** may take the values *('Poor' 'OK' 'Good')*.

The BBN model was built by an iterative process: the construction broadly proceeded in 'top-down' fashion from the definition of the nodes to that of the BBN topology to that of the NPTs, but elicitation at a later stage in this sequence often prompted reconsideration of previous stages and changes in the information that had been elicited at an earlier stage.

Eliciting the multi-dimensional NPT for the Design Process Performance node posed serious problems. The sheer size of the table is enough to make it very difficult for experts to describe their beliefs as a complete, consistent probability table. To elicit this NPT, we obtained a smaller NPT by fixing the states of a pair of parent nodes, identified as least significant among the five parents. We then varied the states of the most significant nodes until every combination had been covered.

To represent the effects of variation in the states of the two parent nodes believed to be least significant, we used a simple parametric formula to produce a linear displacement consistent with a short list of rules which the domain experts believed to govern the influence of these two less significant parents.

5 Validation & Sensitivity Issues

Ideally, the language of BBNs allows experts to express their beliefs about a complex problem within a formal probabilistic framework. In practice we expect both that the experts may find the BBN they produced inadequate, and that the process itself of building and validating it may change these beliefs. This may be a positive effect - they may face questions that they had not previously thought of, and thus be led to deeper analyses, confutation of previous beliefs, etc. On the down side, the experts may be led to give inaccurate descriptions of their beliefs, simply through the need to express themselves in an unfamiliar or inappropriate language.

A BBN is a way of specifying a joint probability distribution for all the variables corresponding to the BBN nodes, i.e., a complex set of probabilistic dependencies among all these variables. The expert builds it by breaking this complex 'global' set of dependencies into an equivalent set of 'local' dependencies which he/she can handle, describing the detailed constituent laws governing these 'local' dependencies among variables. These laws – detailed beliefs – may have various origins, from accepted laws (physical or mathematical) to the expert's attempt to describe intuitive, experience-based laws that he *believes* he applies when producing safety judgements without the assistance of formal mathematics.

Multiple validation issues thus arise, which are particularly pertinent in the nuclear application we treated, because empirical data are relatively sparse. In other fields in which BBNs have been used successfully, such as in medicine, there are large empirical data bases and the dependence upon the unaided expert (for selecting topologies and especially for specifying NPTs) is less.

A first set of validation issues concerns whether the BBN represents the expert's initial understanding of the way he applies judgement to safety assessment. Issues at this stage may include:

- slips and other errors of execution in using the formalism. Some but not all of these will be detected by the BBN support tools;
- lack of self-consistency of the experts' intuition. The set of 'local' dependencies specified may contradict some other aspect of their global beliefs, as evidenced in their judgements;
- insufficient familiarity with the subtleties of the BBN formalism, leading to errors in using it;
- 'normatively incorrect' reasoning by experts. A BBN can only describe a process of judgement that is a correct application of Bayesian reasoning. Human judgement seldom approaches such formal perfection. Mismatches must to be resolved by the experts diagnosing errors in their previous intu-

itive judgement and/or their specification of the BBN;

- ambiguities in the definitions of the variables and their states, making it difficult for an expert to interpret them consistently over time, and to communicate their meanings to other experts to allow meaningful dicussion of the safety argument.
- spurious precision in the BBN, as an artefact of demanding that the expert specify numerical probability values.

A second order of concerns is whether the experts, once satisfied that a BBN represents their current beliefs, will change these beliefs once they have a chance to analyse them thoroughly, as made possible by the BBN itself, and compare them to additional knowledge/beliefs, not represented in the BBN. Last, another interesting question is whether the captured intuition of the expert really does accurately express the real-world uncertainty. In cases where there is a lot of data it may be possible to address this question - it has been examined in some detail, for example, in software reliability growth modelling using tools such as Prequential Likelihood [6]. Such an investigation is, however, beyond the scope of the present work.

6 Support for Validation and Sensitivity Analysis

To check the validity of a BBN model they produced, experts need feedback from it. They need to see various (non-obvious) implications of the model to decide whether any of these are counter to their intuition and deal with any surprising differences. We see two main forms of feed-back: exploring the direct consequences of the BBN "as is", and sensitivity analysis with respect to observations or changes in the NPTs.

Mathematically, a BBN model can be seen as a function which maps vectors of *observations* onto finite vectors of *probabilities* of the states of un-observed nodes. Feed-back to the experts may consist in showing them instances of this function: what the BBN "would infer" if given facts were observed, or what it implies in terms of prior probabilities in the absence of observations. The experts can compare these results from alternative NPTs, among which they are undecided. In particular, the NPT values may have been defined as parametric functions, as we did for the NPT of the Design Process Performance node, and the experts can "tune" the parameters by observing their effects on the results of the BBN model. Many approaches are possible here. We show some examples.

6.1 Feedback from Numerical Calculation using the Hugin tool

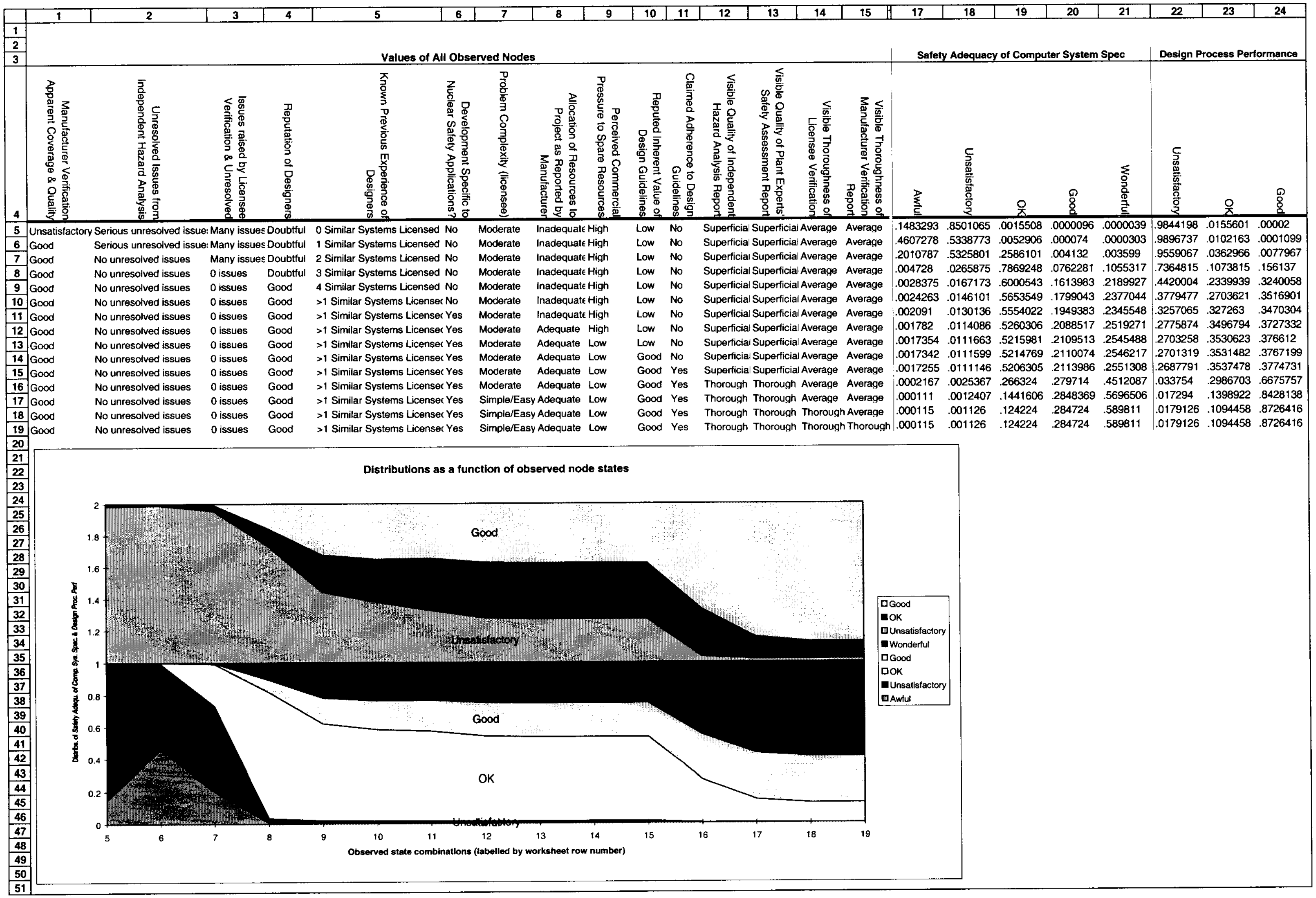
As an illustration of how evidence affects model conclusions, we traversed the space of possible complete observations of the 15 variables which we intend typically to be observable, and plotted in Figure 2 the resulting distribution of the main goal node *Safety Adequacy of Computer System Specification.* The path chosen is one of the possible paths that lead from an intuitively 'least favourable' to a 'most favourable' combinations of observations, by changing one node at a time to an intuitively more favourable state.

This graphical format is clearly appropriate for the viewer to note general trends and exceptions to them, especially appropriate for nodes whose states can be considered as ordered on a scale. Then, any line in the graph represents the probability of a node being in a 'higher' state (or 'better' or 'worse', depending on the meaning of the particular ordering) than a chosen threshold. For instance, the line under the region labelled "Good" in the upper half of the figure, if taken as a function plot in isolation, represents the probability that the state of the node is worse than "Good", and the dependency of this probability on observed evidence (within the particular set of observations represented on the x-axis).

In this particular figure, one would naturally expect every curve to be monotonically non-increasing. The "spike" in the left-hand side of the bottom graph is an obvious "irregularity", which would prompt an expert to re-analyze the pertinent NPTs. The expert may conclude that the irregularity is due to an error in building the NPTs, and correct this error; on the other hand, the expert may conclude that the NPTs are a correct representation of his beliefs, and the perceived irregularity indicates a previously ignored, counterintuitive consequence of these beliefs. So, graphs like these can be a powerful visual aid. Experts may even be surprised by features that do not violate any rule they may have specified beforehand, and yet require a re-analysis of the NPTs.

6.2 Complementary Symbolic Analysis Using Polytree Propagation Algorithm

Plots such as that in Figure 2 raise many questions about the systematic relationships which must exist for the mathematical function which our model embodies. We wished therefore to give the experts feedback in an analytical form, more susceptible to systematic analysis than individual numerical results. An "analytical propagation engine" for BBNs would be extremely complex, but luckily the special topology of this BBN allowed a simpler solution. This topology can be treated as a *polytree* [7] provided we block one of the nodes in the only cycle present, by assuming its value has been



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	17	18	19	20	21	22	23	24
1																							
2																							
3	Values of All Observed Nodes															Safety Adequacy of Computer System Spec					Design Process Performance		
4	Manufacturer Verification Apparent Coverage & Quality	Unresolved Issues from Independent Hazard Analysis	Issues raised by Licensee Verification & Unresolved	Reputation of Designers	Known Previous Experience of Designers	Development Specific to Nuclear Safety Applications?	Problem Complexity (licensee)	Allocation of Resources to Project as Reported by Manufacturer	Perceived Commercial Pressure to Spare Resources	Reputed Inherent Value of Design Guidelines	Claimed Adherence to Design Guidelines	Visible Quality of Independent Hazard Analysis Report	Visible Quality of Plant Experts' Safety Assessment Report	Visible Thoroughness of Licensee Verification	Visible Thoroughness of Manufacturer Verification Report	Awful	Unsatisfactory	OK	Good	Wonderful	Unsatisfactory	OK	Good
5	Unsatisfactory	Serious unresolved issue	Many issues	Doubtful	0 Similar Systems Licensed	No	Moderate	Inadequate	High	Low	No	Superficial	Superficial	Average	Average	.1483293	.8501065	.0015508	.0000096	.0000039	.9844198	.0155601	.00002
6	Good	Serious unresolved issue	Many issues	Doubtful	1 Similar Systems Licensed	No	Moderate	Inadequate	High	Low	No	Superficial	Superficial	Average	Average	.4607278	.5338773	.0052906	.000074	.0000303	.9896737	.0102163	.0001099
7	Good	No unresolved issues	Many issues	Doubtful	2 Similar Systems Licensed	No	Moderate	Inadequate	High	Low	No	Superficial	Superficial	Average	Average	.2010787	.5325801	.2586101	.004132	.003599	.9559067	.0362966	.0077967
8	Good	No unresolved issues	0 issues	Doubtful	3 Similar Systems Licensed	No	Moderate	Inadequate	High	Low	No	Superficial	Superficial	Average	Average	.004728	.0265875	.7869248	.0762281	.1055317	.7364815	.1073815	.156137
9	Good	No unresolved issues	0 issues	Good	4 Similar Systems Licensed	No	Moderate	Inadequate	High	Low	No	Superficial	Superficial	Average	Average	.0028375	.0167173	.6000543	.1613983	.2189927	.4420004	.2339939	.3240058
10	Good	No unresolved issues	0 issues	Good	>1 Similar Systems Licensed	No	Moderate	Inadequate	High	Low	No	Superficial	Superficial	Average	Average	.0024263	.0146101	.5653549	.1799043	.2377044	.3779477	.2703621	.3516901
11	Good	No unresolved issues	0 issues	Good	>1 Similar Systems Licensed	Yes	Moderate	Inadequate	High	Low	No	Superficial	Superficial	Average	Average	.002091	.0130136	.5554022	.1949383	.2345548	.3257065	.327263	.3470304
12	Good	No unresolved issues	0 issues	Good	>1 Similar Systems Licensed	Yes	Moderate	Adequate	High	Low	No	Superficial	Superficial	Average	Average	.001782	.0114086	.5260306	.2088517	.2519271	.2775874	.3496794	.3727332
13	Good	No unresolved issues	0 issues	Good	>1 Similar Systems Licensed	Yes	Moderate	Adequate	Low	Low	No	Superficial	Superficial	Average	Average	.0017354	.0111663	.5215981	.2109513	.2545488	.2703258	.3530623	.376612
14	Good	No unresolved issues	0 issues	Good	>1 Similar Systems Licensed	Yes	Moderate	Adequate	Low	Good	No	Superficial	Superficial	Average	Average	.0017342	.0111599	.5214769	.2110074	.2546217	.2701319	.3531482	.3767199
15	Good	No unresolved issues	0 issues	Good	>1 Similar Systems Licensed	Yes	Moderate	Adequate	Low	Good	Yes	Superficial	Superficial	Average	Average	.0017255	.0111146	.5206305	.2113986	.2551308	.2687791	.3537478	.3774731
16	Good	No unresolved issues	0 issues	Good	>1 Similar Systems Licensed	Yes	Moderate	Adequate	Low	Good	Yes	Thorough	Thorough	Average	Average	.0002167	.0025367	.266324	.279714	.4512087	.033754	.2986703	.6675757
17	Good	No unresolved issues	0 issues	Good	>1 Similar Systems Licensed	Yes	Simple/Easy	Adequate	Low	Good	Yes	Thorough	Thorough	Average	Average	.000111	.0012407	.1441606	.2848369	.5696506	.017294	.1398922	.8428138
18	Good	No unresolved issues	0 issues	Good	>1 Similar Systems Licensed	Yes	Simple/Easy	Adequate	Low	Good	Yes	Thorough	Thorough	Thorough	Average	.000115	.001126	.124224	.284724	.589811	.0179126	.1094458	.8726416
19	Good	No unresolved issues	0 issues	Good	>1 Similar Systems Licensed	Yes	Simple/Easy	Adequate	Low	Good	Yes	Thorough	Thorough	Thorough	Thorough	.000115	.001126	.124224	.284724	.589811	.0179126	.1094458	.8726416

Distributions as a function of observed node states
Distribs. of Safety Adequ. of Comp. Sys. Spec. & Design Proc. Perf
2
1.8
1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
0
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
Observed state combinations (labelled by worksheet row number)
Good
Unsatisfactory
Good
OK
Unsatisfactory
□ Good
■ OK
□ Unsatisfactory
■ Wonderful
□ Good
□ OK
■ Unsatisfactory
▨ Awful

Fig. 2. Updated Distributions Plotted as a Function of Some Observation Combinations

observed. This way of conditioning on values of certain nodes to break loops in the BBN topology is discussed in [7, §4.4.2]. In our example, we expect the 'Problem Complexity (licensee)' node to be observed by the assessor. We simply condition all our reasoning on its observed state. This creates a polytree topology.

We then devised a "symbolic propagation algorithm" for polytree BBNs [2], implemented with the Maple mathematical software. This gives us:

- Arbitrary (i.e. user specified) numerical precision.
- The ability to substitute any particular observation, NPT, or part of an NPT, by an arbitrary parametric function, and to observe the resulting functional form of any selected model output.
- Greater ease of obtaining plots of functional relationships.
- Potentially better intuitive understanding from access to algebraic, as well as visual topological, representations of model assumptions and their consequences. We found that these different forms of representation of model output complemented each other well.

Analytic expressions for model outputs make it easier to investigate the effects on some important model output of different values of the model inputs by analytic differentiation to find maxima and minima.

7 Discussion and Conclusions

This special session is meant to compare different forms of "validation", and specifically evaluation performed at the end of system development against early prediction followed eventually by evaluation of the completed system. From this viewpoint, safety validation in general has some peculiar characteristics. The comparison above suggests that evaluation "at the end of system development" is much more reliable than preliminary predictions for the as-yet unbuilt system. And indeed predictions about, for instance, throughput under typical loads, can usually be validated as soon as a system is built, by running it with appropriate test loads. But safety requirements are of a negative nature, requiring some system behaviours to be extremely unlikely. Demonstrating that they are really so via statistical observation is usually considered unaffordable and is in many cases infeasible [8]. So, even at the end of system development, the system's conformance to the user's safety needs is difficult to ascertain with great confidence. Much "safety validation" activity deals with checking other qualities that are generally believed to be good indicators of safety. In many markets, this safety validation must not only convince the vendor and the buyer, but an independent regulatory authority (like the F.A.A.) and/or independent certification agency (like a TUeV) as well. To reduce the risk of products being rejected at this last validation stage, after long and expensive development processes, interactions have developed between the parties involved. Many design and management decisions during development are in practice pre-negotiated with the regulators, or dictated by guidelines they have approved. In this sense, safety validation is seldom performed on the finished product only. Even so, important unresolved issues remain: there is little evidence that the checks being performed actually deliver the required levels of safety. For instance, it seems self-evident that HAZOP (HAZard and OPerability analysis, essentially a systematic procedures for checking the ways that failures might cause accidents - several such systematic methods are in common use), will decrease the risk of the produced system causing accidents, and yet no-one knows how much the probability of such accidents will be reduced by having performed a HAZOP.

BBNs are an ideal tool for this continuing assessment. At the beginning of a project, a BBN like that in Figure 1, without any observation having yet been entered, can describe the prior probability of the project producing acceptable results. "What if" analysis - entering various possible combinations of observations - can answer questions like: "What would I need to observe, at each successive stage of the project, in order to show that the chances of success are still acceptable - that the project is on track?", or "After observing some damning evidence, what kind of reassurance would be needed to believe that the project still has a good chance of success?". As the project proceeds, new evidence is entered and predictions evolve accordingly.

So, a BBN model, once built, is an extremely powerful tool. It solves a daunting problem in safety analysis, i.e., the complex "propagation" process from the multifaceted evidence that can be collected to a judgement about the qualities that the assessor must judge. However, this is just the third one of three serious problems in safety assessment: the first problem is *obtaining* evidence that can demonstrate the required qualities with the required confidence. This evidence may not be available, which is not a problem for BBN models: the model will just show that the evidence is insufficient for the positive judgement that is sought. The second, essential problem is correctly describing the relationship between the evidence and the required qualities, i.e., building a correct BBN. This is why we put such emphasis on the validation of a BBN model. Assessors need to be clear about how much trust they put in the scientific accuracy of a BBN model. We believe, and this case study gives moderate reassurance in this sense, that the process itself of building and validating the BBN model is helpful to this end. Like other formal methods, BBNs help their users to examine their own thinking, and to seek for inconsistencies and factual errors. If this process converges to a BBN that the users trust as "scientifically accurate" - an accurate enough description of their best understanding of the real situation - then the BBN model can be used directly for decision support. Otherwise, the value of BBNs for seeking insight and for producing alternate predictions under different theories may still be valuable enough to justify their use.

In this case study, we have reached the point of building a complete first version of the BBN, with complete NPTs. The validation exercise is still in progress.

In conclusion, in this case study we have explored some of the issues that arise in building and validating BBNs for safety assessment, and produced some useful tools to support these phases. Developing the BBN has been useful for the expert to question and analyse his criteria of judgement. Further experimentation with feedback methods and support tools will also help us to develop improved guidelines to assist choices about the topology and complexity of a BBN, so as to facilitate its refinement and its use in communication between experts.

References

[1] N.E. Fenton, B. Littlewood, M. Neil, L. Strigini, D.R. Wright, and P.-J. Courtois. Bayesian belief network model for the safety assessment of nuclear computer-based systems. Technical Report 52, ESPRIT DeVa project 20072, January 1998. Available at http://www.newcastle.research.ec.org/deva/trs/.

[2] B. Littlewood, L. Strigini, D.R. Wright, and P.-J. Courtois. Examination of Bayesian belief network for safety assessment of nuclear computer-based systems. Technical Report 70, ESPRIT DeVa project 20072, December 1998. Available at http://www.newcastle.research.ec.org/deva/trs/.

[3] K. A. Delic, F. Mazzanti, and L. Strigini. Formalising engineering judgement on software dependability via belief networks. In *DCCA-6, Sixth IFIP International Working Conference on Dependable Computing for Critical Applications, "Can We Rely on Computers?"*, Garmisch-Partenkirchen, Germany, 1997.

[4] N.E. Fenton and Neil M. A critique of software defect prediction models. *IEEE Transactions on Software Engineering*, 1999.

[5] The serene method manual version 1.0 (d), 1999. EC Project No. 22187 Project Doc Number SERENE/5.3/CSR/3053/R/1. Available from ERA Technology.

[6] A. A. Abdel-Ghaly, P. Y. Chan, and B. Littlewood. Evaluation of competing software reliability predictions. *IEEE Transactions on Software Engineering*, 12:950–67, 1986.

[7] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference.* Mathematics and Its Applications. Morgan Kaufmann, San Mateo, California, 1988. Revised 2$^{\text{nd}}$ printing 1991.

[8] B. Littlewood and L. Strigini. Validation of ultra-high dependability for software-based systems. *Comm. Assoc. Computing Machinery*, 36(11):69–80, November 1993.

Vitae

Bev Littlewood holds degrees in mathematics and statistics, and a PhD degree in computer science and statistics. He founded the Centre for Software Reliability, of which he is director, and is professor of software engineering at City University. Dr. Littlewood has worked for many years on problems associated with the modeling and evaluation of software dependability (i.e., reliability, safety, and security). He has published many papers in international journals and conference proceedings, and has edited several books. He leads several current research projects on the modeling of dependability, involving collaboration with partner institutions throughout Europe. Dr. Littlewood is a member of IFIP Working Group 10.4 on Dependable Computing and Fault Tolerance, the British Computer Society's Safety-Critical Systems Task Force, the IEEE, and a member of the United Kingdom's Nuclear Safety Advisory Committee.

Lorenzo Strigini holds a Laurea degree in electronic engineering from the University of Pisa, Italy. He is a professor of systems engineering at the Centre for Software Reliability, City University, London. Previously, he has been a researcher with the National Research Council of Italy and a visiting scientist at the University of California at Los Angeles and at Bell Communication Research Laboratories, Morristown, New Jersey. His past research work has addressed issues of fault-tolerant design, high-speed networks and software dependability. He has led several research projects and been a consultant to industry on fault-tolerant design and on software reliability His current main interests are practical, rigorous methods for assessing the reliability and safety of software and other systems subject to design faults, and for guiding the application of design diversity and testing. He is a member of the IEEE and of IFIP Working Group 10.4 on Dependable Computing and Fault Tolerance.

David Wright is a Research Fellow at CSR and is completing a PhD thesis on software reliability prediction. He holds a BSc degree in Mathematics from Royal Holloway College, University of London. He previously worked as a Scientific Officer at the Defence Operational Analysis Establishment and as a Research Assistant in non-linear filtering theory during a three year collaborative research project between Royal Holloway College, and Ferranti Computer Systems Ltd. His research interests include reliability prediction based on failure-count data, the incorporation of 'explanatory variables' in reliability predictions, the elicitation and combination of experts' systems-dependability judgements, the application of Bayesian nets to systems dependability assessment, and the question of the extent to which empirical information about achieved reliability levels can legitimately be transferred, as a basis for prediction, either from one system to another, or from one execution environment to another.