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

N.E. Fenton, B. Littlewood, M. Neil, L. Strigini, D.R. Wright

Centre for Software Reliability
City University

P.-J. Courtois

AV Nuclear
Brussels, Belgium

Abstract

The formalism of Bayesian Belief Networks (BBNs) is being increasingly applied to probabilistic modelling and decision problems in a widening variety of fields. This method provides the advantages of a formal probabilistic model, presented in an easily assimilated visual form, together with the ready availability of efficient computational methods and tools for exploring model consequences. Here we formulate one BBN model of a part of the safety assessment task for computer and software based nuclear systems important to safety. Our model is developed from the perspective of an *independent safety assessor* who is presented with the task of evaluating evidence from disparate sources: the requirement specification and verification documentation of the system licensee and of the system manufacturer; the previous reputation of the various participants in the design process; knowledge of commercial pressures; information about tools and resources used; and many other sources. Based on these multiple sources of evidence, the independent assessor is ultimately obliged to make a decision as to whether or not the system should be licensed for operation within a particular nuclear plant environment. Our BBN model is a contribution towards a formal model of this decision problem. We restrict attention to a part of this problem: the safety analysis of the *Computer System Specification* documentation. As with other BBN applications we see this modelling activity as having several potential benefits. It employs a rigorous formalism as a focus for examination, discussion, and criticism of arguments about safety. It obliges the modeller to be very explicit about assumptions concerning probabilistic dependencies, correlations, and causal relationships. It allows sensitivity analyses to be carried out. Ultimately we envisage this BBN, or some later development of it, forming part of a larger model, which might well take the form of a larger BBN model, covering all sources of evidence about pre-operational life-cycle stages. This could provide an integrated model of all aspects of the task of the independent assessor, leading up to the final judgement about system safety in a particular context. We expect to offer some results of this further work later in the DeVa project.

Keywords — Combining Evidence, Safety Assessment, Safety Critical Software, Bayesian Belief Networks

1 Introduction

The problem of safety assessment for complex systems involves a large number of variables and different potential sources of evidence. Many of these variables are inter-related by dependencies of an experimental, probabilistic and even subjective nature which are not well understood formally. These two aspects of the safety assessment application make it typical of the classes of problems to which BBNs have recently been profitably applied. For detailed previous discussions of the potential of BBNs as a tool for modelling software safety assessment see e.g. [Delic *et al.* 1995, Delic *et al.* 1997], [Neil and Galliers 1997]. In this paper we apply the formalism of Bayesian Belief Networks (BBNs) to the problem of safety assessment of nuclear computer-based systems. The BBN formalism, with its easily assimilable graphical representation for probabilistic 'dependency models' [Pearl 1988], and with its amenability to efficient computational tools such as Hugin [Andersen *et al.* 1989], provides a relatively new and attractive method of constructing Bayesian probability models of many judgement and decision problems.

Our approach to this modelling task involves direct interactions with experts who have experience in assessing safety critical software on behalf of safety authorities. Our primary aim has been to explore BBN formalisations of the subjective expertise of an independent safety assessor. This has involved eliciting some objective representation of certain aspects of the subjective understanding of this generic class of problems. The safety assessor's subjective understanding stems from theoretical knowledge, accumulated experience in a range of particular, similar safety assessment tasks, and more general experience of related situations. As with any other BBN modelling exercise, the more objective representation which we are attempting to construct consists of:-

- a set of carefully defined variables (inevitably including some of a rather abstract and subjective nature);
- a statement using a graphical language of a number of formal, probabilistic conditional independence relations (the *dependency model*) which are assumed to apply for these variables; and, lastly,

- a set of numerical conditional probability distributions[1] which is minimally sufficient, given the dependency model, to determine a complete joint probability distribution over the full set of defined variables.

Having completed all three of these stages, the unique joint probability distribution that results can be regarded as a complete description of all uncertainty concerning all of the variables of the model and their stochastic interactions.

We do not intend here to describe the theory of BBN models itself—there is an extensive existing literature on this (see e.g. [Pearl 1993, Pearl 1988, CACM 1995, Jensen 1996]). We intend to present the conclusions obtained to date of an ongoing empirical investigation into the benefits and effects of applying a BBN modelling approach to the safety assessment of a class of nuclear computer-based systems. In Section 2 we describe this application domain in some detail, and outline what we expect will be the scope of application of our particular model. In Sections 3 and 4 we describe the BBN model, as it has evolved up till now. The process of constructing, criticising, and validating a BBN model can consume a considerable amount of effort, partly *because* of the flexibility of such models using current tools, and the high 'visibility' of the modelling assumptions. These models are well suited to a kind of incremental development - involving experimentation with changes to both the topology and the conditional probability tables - which is easily achieved using modern BBN tools such as Hugin. In this sense, we regard the model on which we are reporting here as having the status of 'work in progress'. Section 3 concentrates on the variable definitions (nodes) and conditional independence assumptions used (network topology). Section 4 discusses the elicitation of precise numerical conditional probability distributions for the values of the model variables, and gives examples of the numerical posterior distributions output by the BBN method. In Section 5 we provide some information about the process whereby we came to arrive at the current model, we discuss some general advantages and outstanding problems with the BBN approach, and mention the possible directions in which this work might be taken further. Section 6 contains a summary of our main conclusions and discusses our immediate intentions for pursuing this modelling approach within DeVa.

2 Licensing of Nuclear Computer-Based Systems

The BBN we present in this paper applies to a class of software based systems used in nuclear plant for functions important to safety. For obvious reasons of professional discretion, the identity of the licensees, 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. It is only the precise delineation of the class of nuclear safety

[1] We will term the tabulation of such a conditional distribution function a CPT (conditional probability table) throughout this document. In all BBN models there is one such table required for each variable of the model, i.e. for each *node*. It may be that 'NPT (node probability table)' is a more appropriate term, since the distributions attached to certain nodes of the net (the root nodes) will be unconditional (see Section 4 and the general literature on BBNs for further explanation) but, throughout this document, we have standardised on the term CPT, made applicable to all nodes, including root nodes.

functions to which the BBN is intended to apply which is necessary to preserve the usefulness of the results. These safety functions are all of the highest criticality and under the responsibility of systems called ‘safety systems’ by international nuclear safety guides.

These functions include those of the reactor protection system, whose role is to detect the fact that the state of a nuclear reactor has moved outside its allowed safe limits, to trip the reactor in that eventuality, and to assist the operators to keep it safe after the trip. In this context, safety means low probability of dangerous releases of radiation, either internally within the plant or externally to the wider environment and the public.

The class includes also other functions such as the protection of safety engineering features’ power supplies, the monitoring of critical plant and core parameters in post-accidental situations, the display of alarms and their masking in certain phases of operations, etc.

The inputs to these systems include sensor values of physical parameters such as reactor core temperature, coolant pressures and levels, and possibly various inputs from instrumentation and micro-processors embedded in the reactor hardware. The output data include commands given to devices such as actuators, valves, pumps, information for control room displays and alarm systems.

In preparation for the construction of our formal BBN model, we discussed informally the typical development life-cycle for a system of the kind we are concerned with here, and the activity of the safety authority in observing and reacting to this development through the role of the independent safety assessor.

2.1 Safety System Development Phases

A software-based nuclear safety system, or component system, is typically developed using a sequence of development phases. The licensee and the manufacturer produces a particular piece of safety-critical software in the form of a sequence of intermediate products produced by successive phases. These are termed in this paper:-

System Requirements Document

↓

Computer System Specification Document

↓

Software Specification Document

↓

Code Document

↓

Internal Test Document

It is the task of the independent safety assessors to examine all of these intermediate products, and to gain what evidence they can obtain about the quality of all the development processes used to produce them.

In this paper, as an initial step towards analysing the larger safety assessment problem, we concentrate our attention on modelling those parts of the activities of the safety authority's independent assessors which address evidence arising from the initial phase of system development, up to the production of the Computer System Specification document. Thus the partial net as formulated here models the completion of only the first stage (first arrow above) of the typical sequence of development steps. As far as this net is concerned, the goal variable is 'Adequacy of Computer System Specification'. We regard the BBN produced in this paper as representing only a part of a considerably larger modelling exercise which covers the entire development life-cycle, and which we have not attempted to formulate here. As far as it is possible to generalise about the typical process by which nuclear instrumentation systems are developed, a typical scenario for the first stages of the development of such a system, on which we shall now concentrate our attention, seems to be as described in the next three paragraphs.

The development of a system important to nuclear safety normally begins with the elicitation and specification of a System Requirements Document, accompanied by an assessment of the system's foreseeable failure modes, an enumeration of intended lines of defence against each of these modes, and an analysis in terms of anticipated frequencies of classes of different postulated initiating events related to these failure modes. Thus the two processes of hazard analysis and system requirements specification take place in an integrated way at the beginning phase of the development process. The requirements document consists essentially of a statement of requirements written with classical engineering terminology and in natural language. This document would describe the tasks the system is required to perform, together with a detailed description of the environment in which it is required to operate, including the constraints and the values that must be maintained by various physical environmental parameters.

As part of this phase of requirements formulation and failure modes analysis, the level of criticality of the system is assessed. This may be done, for instance, in terms of postulated initiating events (PIE) corresponding to well defined types of faults (e.g. a pipe break). These events are classified according to their anticipated frequencies of occurrence, these frequencies being estimated from past operational experience data from similar plant designs. To each class are associated upper limits on the radiological consequences of any single PIE. The basic principle is that the most frequent events must yield little or no radiological risk.

For every postulated initiating fault, including other potential failures which may directly result from the PIE, and possibly under other conservative assumptions, it is required to demonstrate that following this fault sequence no unacceptable radiological risk can occur. These analyses are done for every PIE. They may imply probabilistic arguments. As a result of these analyses for instance upper limits may be required on the rate of failure per demand of the equipment and systems in charge of controlling and mitigating the consequences of the postulated initiating events. The independent assessor may affect the system specifications at this stage by formulating objections if the plant safety requirements or the safety design criteria are not judged complete, correct or precise enough. The subsequent decision as to acceptance or rejection of the system and safety requirements will be influenced by the licensee's and developer's responses to these specific objections by the safety assessor.

The second documentation entity chronologically following the Requirements Specification is termed in our discussion the 'Computer System Specification Document'. Ideally this should take the form of a clear, understandable document, specifying in particular, and justifying, decisions taken relating to the allocation of safety functions to hardware and software respectively. It must be demonstrated that the system architecture satisfies the system and safety requirements, in particular that adequate levels of redundancy and diversity have been introduced. Walls and barriers between safety and non-safety functions must also be clearly specified. A failure modes and effects analysis should be presented in terms of the software and hardware structural decomposition into system components, and the methods and mechanisms of auto-detection by the system of its own failure should be specified.

The Computer System Specification Document would normally make greater use than the Requirements Document of diagrammatic forms of representation such as data flow diagrams and flow charts. The independent safety assessor affect the development at this level by formulating objections to the Computer System Specification. (e.g. "It is unacceptable to us that the system fails to tolerate faults of the kind ....") The subsequent decision as to acceptance or rejection of the system will be influenced by the licensee's and developer's responses to these specific objections of the independent safety assessor.

The BBN model in this paper - see Figure 1 - represents evidence regarding the direct safety assessment of the two preceding artefacts of our life-cycle model : the System Requirements document, and the Computer System Specification document. In future extensions of the work, we envisage that there will be an arrow out of the Adequacy of Computer System Specification node - which is our goal node here - to other variables further along the system life-cycle. Such extensions will lead to a BBN-based model of the entire assessment task of the independent assessor. It is our intention to continue further towards this objective within the DeVa project.

2.2 The Role of the Independent Assessor

There are three key institutional roles involved in the interactions modelled by this net: the *independent assessor* (who works on behalf of a safety authority); the *system manufacturer*; and the *system licensee*. Our particular net is a representation of the assessment activities of the independent safety assessor. This role is usually occupied by a department of the nuclear authority with responsibility for assessing the safety of the instrumentation, control, and protection systems of nuclear power plants, and should normally employ personnel with expertise in nuclear instrumentation and computer systems and expertise in nuclear plant safety engineering.

The independent assessor's responsibility is to assess the safety of the system and ultimately to recommend approval (or not to) for its use in the plant by the licensee. The licensee remains responsible for the safety of the plant, and has the duty of demonstrating this safety to the independent assessor. The independent assessor is usually not prescriptive, i.e. does not impose specific methods and techniques of design, implementation, verification or validation to the licensee and the manufacturer. However, he or she will request evidence that the safety and system requirements are adequate, and that they are satisfied by the system design, its implementation, and its use in operations.

Therefore, over the entire development process of the system there are many diverse factors influencing the likely safety of the system, which will be taken into account by the independent safety assessor. Among others, they may be related to:-

- the ability to demonstrate the validity of the system requirements, to validate their implementation, and in particular to demonstrate an adequate coverage of the verifications and the validation tests;
- productivity pressures and the effects of these and of other factors on both complexity and general quality of design and code, and specifically on their likely fault content;
- the competence of staff used for all the design and verification intermediate tasks from the requirements specification to the final integrated system.

However, as we have stated above, for the purposes of this paper we will concentrate on modelling the influence of evidence concerning one of the earlier development phases on the judgement processes of the independent safety assessor. We envisage that, at a later stage, this work will ultimately be expanded to cover assessment of evidence relating to all the later development phases, up to the completion of the operational system (or system component) whose safety is to be assessed. However, the completion of such a model, in which we could have real confidence, would be a very large task indeed. Within the DeVa project, we aim only to make some further well-documented progress towards such a goal.

We should point out an issue about the time at which a judgement is made and the effects of this on determining the kind of evidence that is likely to be available on which to base the judgement. Depending on circumstances, an independent assessor may become involved in the development of a protection system or component at an earlier or a later stage. For example, in judging whether to grant a license for operation of a component sub-system, the information available to the regulator differs depending on whether the licensee has chosen to select from pre-existing software systems available on the market, or has instead decided to design a new software system. In the latter case, more information about the sub-system will normally be available to the safety authority, because the independent assessor will expect to be in a position to follow and observe the design process directly as it progresses, rather than having to rely on recorded past documentation about it. A BBN model is well suited to being repeatedly applied as more evidence accumulates, because there is no built-in formal distinction between observable and unobservable variables. Different subsets of the model's variables may be observable at different times, allowing predictions or estimations to be made at any particular time concerning any of the remaining variables. So we would expect a BBN model such as ours to be applicable to several different situations, with availability of different items of evidence for the particular system or sub-system to which it is applied. Equally it might be applied repeatedly to different stages of an evolving situation, as more sources of observable evidence become available.

In assessing the System Requirements Document, the assessor's job may be to assess an already completed Requirements Document. But the safety authority might, in the case of some systems, be involved at an earlier stage - able to observe the process of preparing successive versions of this document, and hence able to take account of additional kinds of evidence. The BBN includes this second approach. In this case, the assessor judges the quality of the system requirements through the quality of the elicitation and validation process, and through the quality of the contents of produced documents, in consultation with the domain experts (nuclear engineers) who are employed by the safety authority. The System Requirements document normally consists primarily of an understandable natural language specification of the system, augmented by some semiformal diagrammatic representations familiar to nuclear instrumentation and control engineers. This document will be examined by the independent assessor specifically with a view that it should be understandable, without ambiguities and that the control functions and safety functions are completely and correctly specified. The safety assessor attaches great importance to the assessment of the quality of the failure modes assessment exercise done at this stage. The assessor needs to verify that the criticality allocation of the system is correct, and then to assess whether the system specifications achieve the safety requirements stipulated for systems of this criticality category. For this kind of safety-critical application the logic and mathematical safety-related functions should consist of rather simple, easily understandable operations like: compare a value to a threshold; take a maximum or a minimum of a finite number of values; evaluate an integral, or a derivative.

For the assessment of both the Computer System Specification and sometimes the Requirements documents, it is sometimes possible to directly assess the competence of the staff who created the document (i.e. through direct means apart from inferring backwards from examination of the document itself). This direct staff-competence factor is not always taken into account by the assessor. Where it is done, this would tend to be in terms of the staff's (or their employer's) reputation for previous, similar work and an assessment of whether this previous work is generally held to be of a high quality. In the BBN as presented here we have allowed for this possibility with the Computer System Specification document, but not for the Requirements document. This was somewhat of an arbitrary decision: We could easily have done otherwise.

3 Node Definitions and Structure of the Network Topology.

The network topology we elicited is shown in Figure 1. We see that its general structure strongly reflects the life-cycle model used. The ‘backbone’ of the graph consists of three nodes: ‘Quality of Requirements’, ‘Design Process Performance’, and, the *goal node* of this BBN, ‘Adequacy of Computer System Specification’. These three can be thought of as the foci of three sub-graphs, or ‘clusters’, of nodes arranged in a linear, approximately chronological and causal sequence. These three sub-graphs are indicated by the dotted lines (which are provided to aid the reader’s appreciation of the graph structure, and have no meaning in terms of the BBN syntax): at the bottom we have a cluster of nodes providing evidence relating to the requirements specification. These nodes all branch off from the central node, for this bottom cluster, which represents the unknown ‘truth’ regarding a general Quality of Requirements attribute of the requirements specification documents. At the top of the net is a cluster of nodes representing the quality of the resulting Computer System Specification document, with nodes for the various key sources of evidence relating directly to this. Mediating between these two is a central ‘process’ cluster representing evidence bearing on the quality of the development process connecting these two artefacts. We will define the node variables below in three sections, according to these three component clusters of our BBN topology. In each case, the proposed state space of the node is shown in brackets following its name. In the majority of cases, the values in the state-space of a node represent categories defined by the independent assessor so as to correspond to aspects of his or her subjective safety assessment activity. In most cases this scale is an ordinal scale [Fenton 1991], though we do not propose this as an absolute requirement of interpretation. For example, in the case of the 5 point scale for the ‘Adequacy of Computer System Specification’ node, there are two underlying dimensions of assessment involved in the specification of this state-space: the presence or absence of actual safety problems with the document; and the ease with which the assessor is able to judge this by examining the document.

3.1 ‘Requirements Document’ sub-graph

Quality of Requirements ***(‘Poor’ ‘OK’ ‘Good’)*** The requirements document consists essentially of a statement of requirements written in natural language. This document would describe the task the system is required to perform, together with details of the environment in which it is required to do this, such as the values of various physical environmental parameters, etc. Attributes such as completeness, correctness, consistency, understandability, verifiability are normally thought of as comprising the quality of such a document.

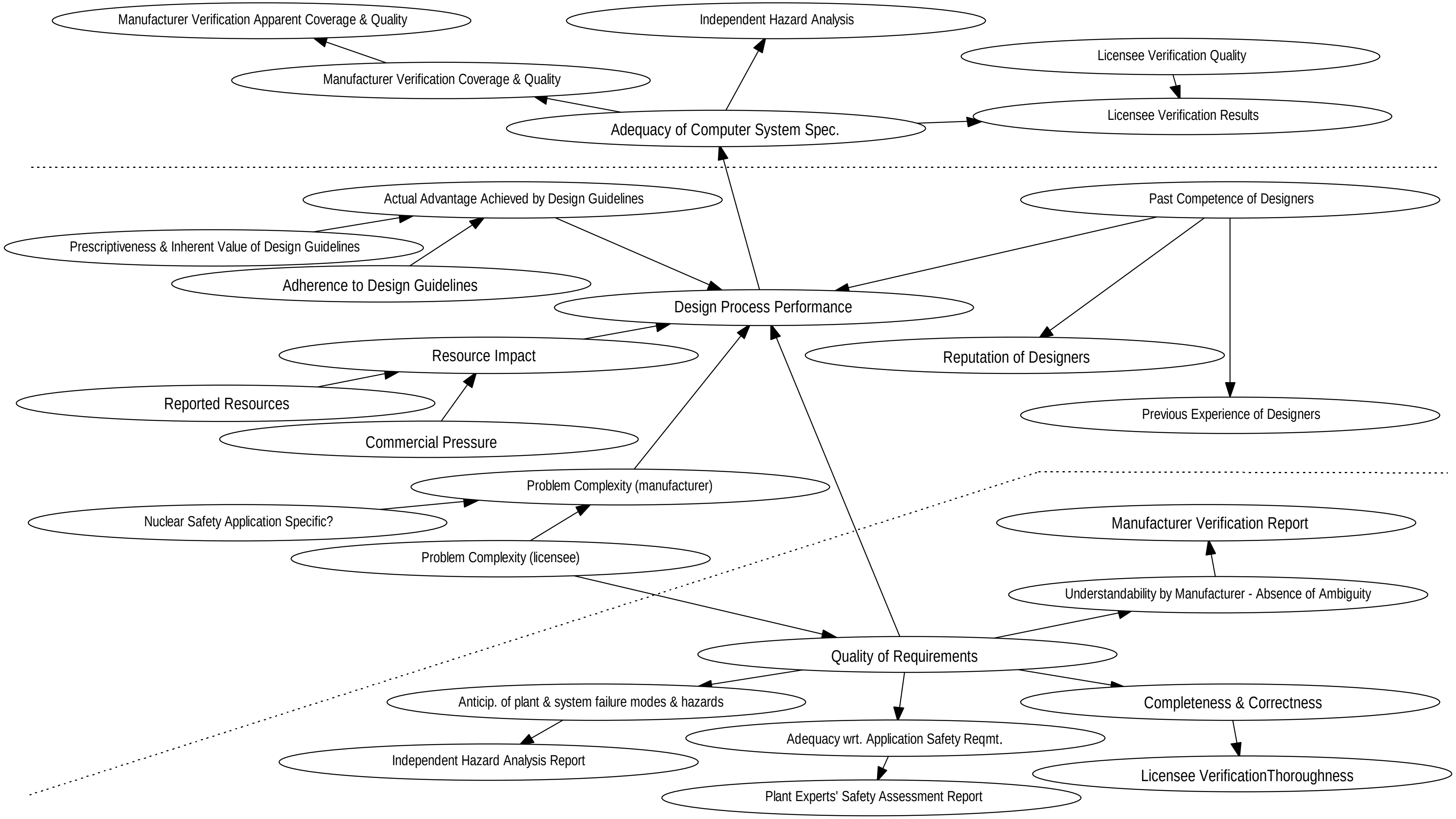

Manufacturer Verification Apparent Coverage & Quality
Independent Hazard Analysis
Licensee Verification Quality
Manufacturer Verification Coverage & Quality
Licensee Verification Results
Adequacy of Computer System Spec.
Actual Advantage Achieved by Design Guidelines
Past Competence of Designers
Prescriptiveness & Inherent Value of Design Guidelines
Adherence to Design Guidelines
Design Process Performance
Resource Impact
Reputation of Designers
Reported Resources
Previous Experience of Designers
Commercial Pressure
Problem Complexity (manufacturer)
Nuclear Safety Application Specific?
Manufacturer Verification Report
Problem Complexity (licensee)
Understandability by Manufacturer - Absence of Ambiguity
Quality of Requirements
Anticip. of plant & system failure modes & hazards
Completeness & Correctness
Adequacy wrt. Application Safety Reqmt.
Independent Hazard Analysis Report
Licensee VerificationThoroughness
Plant Experts' Safety Assessment Report

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

Perhaps a metaphor for how we have treated this node in our topology - with a number of more specific quality aspects represented as child nodes - would be the attribute of ‘general intelligence’ of a person. The role of the central node here is essentially one of conceptual simplification and presentation of what is actually just a model assumption about the stochastic association between more concrete attributes of the requirements document[2]. It represents a single amalgamation of a number of associated, but distinct, quality attributes. Compare asking human subjects to perform tasks of various different kinds for the purposes of assessing specialised kinds of mental ability. In understanding a relationship between these abilities, and associated performance levels, which we believe we should expect to observe (and/or feel that we have actually observed), we can choose to introduce a notion of a *general mental ability* to which all these other abilities are assumed to be positively associated. Similarly in our BBN topology, the general property of ‘requirements quality’ sits in the middle of (is composed of) four more specific attributes of requirements. For each of these four, we have a related observable node, providing a more or less accurate measurement instrument (the specific accuracy being represented by a CPT). But the link of this cluster to ‘the rest of the net’ is simplified by requiring that all connections are via the central, generic ‘requirements quality’ node. Equally, it sometimes simplifies various human/social/management tasks to think along the lines “this person is quite intelligent”, etc. Bearing in mind that BBNs can be developed and tested in an evolutionary fashion, this structure seemed to our independent assessor to be a reasonably accurate starting position for his or her beliefs about what can be inferred from specific requirements quality evidence.

Anticipation of Plant & System Failure Modes & Hazards *(‘Sketchy’, ‘Satisfactory’, ‘Detailed’)* This, typically unobservable, node represents “the truth” about one of the six facets of general ‘requirements quality’ (its parent). As part of the safety demonstration, the system requirements should be subject to a failure analysis. Potential failures of the system should be identified, their impact on the application should be evaluated, and the presence of adequate defences should be confirmed. In particular, potential failures at the interface with the plant, such as sensor defects, inputs out of range, etc., should be identified and adequate system reactions should be defined.

Independent Hazard Analysis Report *(‘Superficial’ ‘Average’ ‘Thorough’)* The independent assessor may discover, by inspection, defects or inadequacies, of greater or lesser seriousness, in the previous hazard analysis which will have been provided, usually by the licensee. The idea is that this node will typically be the observable consequence of its non-observable parent. The number and nature of inadequacies in this report are indications of the thoroughness of the independent hazard analysis. From the pertinence of these inadequacies, values for the parent node may be inferred.

[2] That is, the position of the *Requirements Quality* node within the topology provides a mathematical model of the form of the association between its child nodes.

Adequacy with respect to Application Safety Requirements ***('Unsatisfactory' 'Satisfactory')*** Usually unobservable. The system must be demonstrated to comply with the safety requirements of the application. For instance, one may identify a set of initiating events to which the system is supposed to react with protective actions so as to keep the frequencies and the consequences of these events within predetermined limits

Plant Experts' Safety Assessment Report ***('Superficial' 'Average' 'Thorough')*** Observable consequences of its non-observable parent. See description of 'Independent. Assessor Hazard Analysis Report'

Completeness & Correctness ***('No' 'Yes')*** Obvious meaning, but usually non directly observable node.

Licensee Verification Thoroughness ***('Superficial' 'Average' 'Thorough')*** Here 'licensee' means the licensee's system architect. Thoroughness may be observed and characterised by the nature and number of the comments and defects noted in the verification report. From the criticality of these observations, values of the parent node can be inferred.

Understandability by Manufacturer - Absence of Ambiguity ***('Inadequate' 'Satisfactory' 'Good')*** This, typically unobservable, node represents "the truth" about one of the six facets of general 'requirements quality' (its parent). It is essential for the quality of the implementation that the manufacturer's designers have a thorough understanding of and familiarity with the system requirements and their implications.

Manufacturer Verification Report ***('Superficial' 'Average' 'Thorough')*** Could include the reports on results from the execution of prototypes. Quality of test plan, test coverage, & test reports.

3.2 'Design Process' sub-graph

Design Process Performance ***('Awful' 'Unsatisfactory' 'OK' 'Good' 'Wonderful')*** This refers only to that part of the total design activity that takes place chronologically up till the life-cycle stage of designing a 'Computer System Specification' has been completed, working from the Requirements Document.

Actual Advantage Achieved by Design Guidelines ***('No' 'Yes')*** Advantages resulting from complying with the recommendations pertaining to specifications, design, verification, validation, such as those within specific standards such as the IEC 880.

Prescriptiveness & Inherent Value of Design Guidelines ***('Low' 'Good')*** Pragmatic guidelines, not limited to general principles and motherhood recommendations, can have a concrete impact on the quality of the design, for instance its verifiability.

Adherence to Design Guidelines ***('No' 'Yes')*** Non adherence may be observed when guidelines are simply ignored. Designers may claim adherence, but this may be formal only. This possibility is taken into account in the CPT of 'Actual Advantage Achieved by Design Guidelines'

Problem Complexity (manufacturer) ***('Complex/Difficult' 'Moderate' 'Simple/Easy')*** The manufacturer may replace the licensee's problem by a different problem, of a greater or lesser complexity, or may transform the original problem into a different problem - e.g. for reasons to do with their longer-term commercial intentions for marketing solutions to applications related to this one.

Nuclear Safety Application Specific? ***('No' 'Yes')*** . System modules used may, or may not, have originally been intended for use with safety-critical systems, or may not be intended to be used in future only in this application. This requirement for portability to applications of other kinds can create added system complexity. For example additional system configuration variables might have to be introduced into the design for this purpose.

Problem Complexity (licensee) ***('Complex/Difficult' 'Moderate' 'Simple/Easy')*** This node refers to the complexity or difficulty of the original problem owned by the system licensee which motivates the licensee to commission the system. Note that the small cluster of three nodes (on the right hand side) to which it belongs are currently the only nodes having separate direct paths to both the 'Quality of Requirements' and the 'Design Process Performance' nodes.

Past Competence of Designers ***('Doubtful' 'Average' 'Good')*** Competence of the designers in the realisation of similar systems (usually non observable). It is inferred from its two children nodes, which represent the objective record ('Previous experience') and the other evidence of competence ('Reputation') which may indicate that the visible record may be misleading.

Previous Experience of Designers ***('0 Similar Systems Licensed' '1 Similar System Licensed' '>1 Similar Systems Licensed')*** Number of success stories with 'similar systems' for the state variable here—success stories meaning that the system was licensed and has not since manifested safety defects in operation

Reputation of Designers ***('Doubtful' 'Average' 'Good')***This variable represents a general subjective assessment of competence. Observable effect of its non-observable parent.

Resource Impact ***('High' 'Low')*** Extent to which scarcity of resources may have hindered 'Design Process Performance'. It is not observable, but its two parents are. The assessor has the manufacturer's report of the resources dedicated to the design process, but must consider that this report may overestimate the resources actually used, especially in case of problems within the manufacturer's organisation (cf. 'Commercial Pressure').

Commercial Pressure ***('High' 'Low')*** Pressures (e.g. development deadlines) within the manufacturer's organisation, which may have created scarcity of resources. The scale is subjective, representing the assessor's own judgement of the evidence available to him or her (observable node).

Reported Resources ***('Inadequate' 'Adequate' 'More than Adequate')*** This node represents the adequacy of the resources that the manufacturer *claims* to have dedicated to the design process. The scale is subjective: the assessor uses the manufacturer's reports and compares it with the assessor's own estimate of the amount of resources that would be adequate for the task, but ignores any evidence that the reports are inaccurate.

3.3 ‘Computer System Specification’ sub-graph

Adequacy of Computer System Specification ***(‘Awful’ ‘Unsatisfactory’ ‘OK’ ‘Good’ ‘Wonderful’)*** Since we are focusing on safety assessment, this node refers to ‘adequacy’ from a safety point of view. I.e. its value is not intended to reflect non-safety-related quality attributes such as the likely impact of this specification on the efficiency of designing the system, in terms of design effort required, or cost of the hardware components and software tools needed. Neither should the value of this node be affected by ‘performance’, nor any other such system attribute whose level is likely to result from this specification, except to the extent that the attribute in question has some impact on safety. The node states have the following meanings:

- *Awful* = It is quite clear from the presentation that there are real problems with the specification

- *Unsatisfactory* = It appears that there may be real problems here but they are not easy to identify or diagnose with confidence.

- *OK* = We are fairly sure there are no safety problems but this was not obvious, nor can we now state it with the very highest levels of certainty, because of inadequacies in the presentation.

- *Good* = There are no safety problems; and the high quality presentation makes this apparent

- *Wonderful* = The presentation is very clear and it is quite obvious that there are no safety problems with this specification document.

In these state definitions ‘problems’ means *outstanding* problems, and does not include problems which were raised, but were later explained away satisfactorily by the manufacturer.

Note that it is not correct to think of the state space described here as a one-dimensional five-point ordinal scale. We see it rather as a ‘collapsed’ version of two distinct dimensions. There is a dimension concerning the true presence or absence of safety-related problems with this specification document. There is a second dimension relating to the clarity of the presentation of the specification - its traceability - the extent to which it exemplifies successful ‘design for validation’; rather than only design for correct (system safety-related) functionality. Although aspects of the node such as clarity of the presentation may - in a general sense - be directly assessable by the independent assessor, this node itself is not observable in our restricted sense of the footnote on page 17. One of its two dimensions - namely the genuine presence or absence of remaining safety issues - cannot be directly observed in this sense. Neither is it entirely possible for the independent assessor to observe the degree to which the specification is understandable and amenable to detailed analysis by independent verifiers at the manufacturer and licensee sites. These latter qualities are important in view of the fact that the computer system specifications will be the basis of the documentation given to the programmers.

Manufacturer Verification Coverage & Quality ***('Unsatisfactory' 'OK' 'Good')*** Unobservable node representing the truth about how well the manufacturer verified the system. The evidence from the manufacturer role in the verification of the system will generally - at this phase of system development - tend to carry less weight than the licensee-verification contribution.

Manufacturer Verification Apparent Coverage & Quality ***('Unsatisfactory' 'OK' 'Good')*** The purpose of this node is to represent the Independent Assessor's knowledge that he/she only sees *reports* of manufacturer verification activity - i.e., does not observe that activity itself. Further, it is generally a foregone conclusion that the *findings* of the reports presented by the manufacturer to the Independent Assessor will look reasonably good. So we think of this observable node merely as evidence as to the possible state of its parent, whose value it is one of the tasks of the assessor to infer.

Licensee Verification Results ***('0 issues' 'A few issues' 'Many issues')*** This is designed to be an observable node representing the results and the observable records and indications as to the thoroughness and quality of the licensee system architect's verification of the computer system specification. (Here 'licensee' means licensee system architect.) At this particular development phase, this 'licensee side' of the top part of the net is of greater significance than the manufacturer-verification part.

For state variable here, we use an indication of the number of 'issues' raised. The licensee will raise with the manufacturer issues of concern about the computer system specification. Often some of these will be satisfactorily resolved by the manufacturer providing some satisfactory explanation, or will be judged by the independent assessor to be groundless or inconsequential, for one reason or another. Having eliminated any such 'false issues', the remaining issues, judged by the independent assessor to be 'valid issues raised' are the ones which are counted here.

Licensee Verification Quality ***('Low' 'OK' 'High')*** This variable implicitly depends on an array of variables which have not been differentiated explicitly, such as the competence of the licensee verification team, the resources they have available, the time-pressure they are placed under, etc. It is not necessarily observable.

Independent Hazard Analysis ***('No unresolved issues' 'Minor unresolved issues' 'Serious unresolved issues')*** The independent hazard analysis refers to an activity of the regulatory authority itself. In contrast to the other two (manufacturer and licensee) verification branches of the top part of the net, there is no corresponding second node here for the 'quality' of this (independent assessor's) analysis. To introduce a performance-quality self-assessment to be carried out by the independent assessor's team was thought to be unnecessarily convoluted.

3.4 Further Comments on this Topology

The network, as presently formulated in Figure 1, is topologically slightly different from what Pearl calls a '*causal polytree*' [Pearl 1988] structure. It contains a single cycle[3] which if broken by the removal of a single arrow would leave a causal polytree. In the expected application of this network, the assumption regarding observability[4] is *almost* simply that all the *leaf nodes* (the nodes having one arrow attached to them) will normally be observable to the independent assessor: and that the other nodes will normally be unobservable to the independent assessor. (The two exceptions are: (i) that 'Licensee Verification Results' is observable whilst 'Licensee Verification Quality' may well not be observed; and (ii) that 'Problem Complexity (licensee)' will normally be observable.) These features of the topology of our model are mentioned as a way of enabling the reader to digest the model in terms of its topology, and are features peculiar to this BBN model. The same features will not necessarily apply (nor necessarily fail to apply) to other BBN models (for this or for different application contexts). Likewise it is possible that future criticism of the structure of this net - whether in the light of output it produces, or simply by inspection - may result in changes to the topology which will change the character of the net. It is our understanding that such BBN modelling exercises as this should be expected to result in models which evolve through a (perhaps long-term) process of criticism and review. To facilitate such processes of human communication is an important function of the graphical language and presentation.

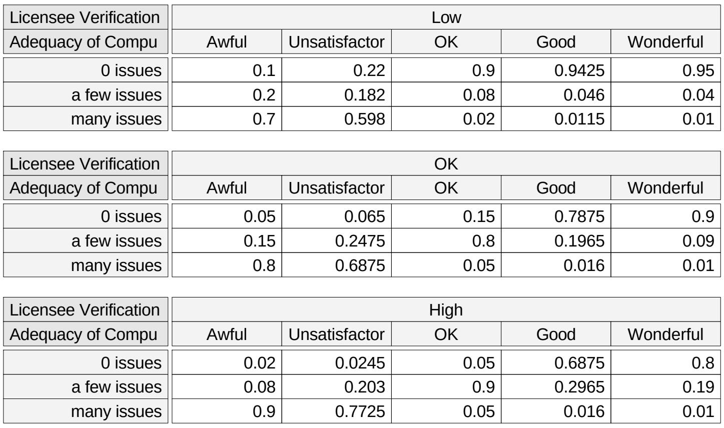
4 Conditional Probability Tables (CPTs) and Numerical Model Output

From a mathematical point of view, after the definition of node state-spaces and the construction of the dependency model (expressed by the graph topology), there is an important final stage required if we are to complete the construction of a full joint probability distribution over the product state space of the model variables. This stage consists of the specification of a conditional probability table (CPT) for each node: a table of conditional probabilities of each of that node's values, given each combination of its parent node. Further theoretical explanation of how these CPTs together define a joint probability distribution is given in [Pearl 1988], [Lauritzen and Spiegelhalter 1988] and [Jense 1996].

[3] -undirected: directed cycles are forbidden in BBN models [Pearl 1988].

[4] **Important note on our use of the term "observable**": By "observable" for a node in this model we mean merely that we expect to input, into the BBN model when we apply it to the assessment of some system, a precise value (or perhaps in some circumstances a "likelihood observation", see []) for that node. So for our purposes here, "observation" includes subjective assessment of a node's value (or even in some cases subjective assessment as the *definition* of the node's meaning) using expert judgement. If it is intended that the expert should make such a judgement of the value of a particular node when assessing a particular system, then, for purposes of brevity in this paper and of consistency with terminology used in the formal theory of BBNs, we classify this act as an "observation" and term the node "observable".

In practice, we found with our BBN that for small, relatively low dimensioned CPTs, it is practical to ask the 'expert' (in our case the independent safety assessor) to simply fill in the numbers of the table by reference to his/her domain knowledge and experience. This was the technique used for most of the nodes in our net. For a few nodes we used linear interpolation (see 2nd and 4th bullet points of Section 5.1) as a first approach to filling in some values. For example, the CPT for the 2-parent 'Licensee Verification Results' was constructed as follows, where the names of the top two rows of each of the three parts of the table are the (truncated) names of the two parent nodes. For each 'Licensee Verification Quality' value, distributions were elicited directly in the form of numbers at the central (ok) and extreme (awful, and wonderful) values of 'Adequacy of Computer System Specification'. Then 'first-cut' distributions for the two intermediate states (unsatisfactory and good) of 'Adequacy of Computer System Specification' were obtained by specifying a single numerical linear interpolation point, resulting in the CPT of Figure 2.



Licensee Verification	Low				
Adequacy of Compu	Awful	Unsatisfactor	OK	Good	Wonderful
0 issues	0.1	0.22	0.9	0.9425	0.95
a few issues	0.2	0.182	0.08	0.046	0.04
many issues	0.7	0.598	0.02	0.0115	0.01

Licensee Verification	OK				
Adequacy of Compu	Awful	Unsatisfactor	OK	Good	Wonderful
0 issues	0.05	0.065	0.15	0.7875	0.9
a few issues	0.15	0.2475	0.8	0.1965	0.09
many issues	0.8	0.6875	0.05	0.016	0.01

Licensee Verification	High				
Adequacy of Compu	Awful	Unsatisfactor	OK	Good	Wonderful
0 issues	0.02	0.0245	0.05	0.6875	0.8
a few issues	0.08	0.203	0.9	0.2965	0.19
many issues	0.9	0.7725	0.05	0.016	0.01

Figure 2 Elicited CPT for 'Licensee Verification Results' node (printed from Hugin tool)

On a point of terminology, when we speak about CPTs in this paper we have a standard layout in mind (which would be obtained from Figure 2 by aligning the three sections of the table horizontally, instead of vertically as they are show) in which the CPT has one *row* for each possible state of the node in question, and one *column* for each possible combination of values of the states of that node's parents. Each *column* should therefore consist of numbers whose total is 1.

For the few larger CPTs, most notably the CPT of the 'Design Process Performance' node, it becomes unrealistic to set the assessor this task. Some more systematic method seems called for to address the sheer scale of the problem. For example, even with the rather coarse grading into variable values that we have used in the current version of the net, the 'Design Process Performance' CPT is required to be a six dimensional table of dimensions 3×3×3×3×2×5, i.e. requiring 810 numerical entries. Admittedly, since these must represent probability distributions, there are only 648 independent entries in this table. But still this is a large number. There are a number of possible responses to this problem. Firstly, we might ask: Do we really need a node having this number of parents (i.e. five)? Is it possible to argue for a modified dependency model which would win some advantage here, by some strategy such as to introduce an extra node in a position which reduces the numbers of parents of individual nodes? This may be a matter for profitable future discussion. For the present, however, our independent assessor preferred to persist with the current topology of Figure 1. Having made this decision, there are strategies, besides attempting to change the topology, for approaching the resulting combinatorial problem of the scale of certain of the CPTs. The elicitation of multivariate joint and conditional probability distributions is a significant research area in its own right [Keeney & von Winterfeldt 1991, Chhibber *et al.* 1992, Vanlenthe 1993, Chaloner *et al.* 1992], which obviously has great potential application in BBN construction. Apart from existing systematic methods and tools, it is likely that there may be others not yet devised which would help here. Some of the methods which we tried during our CPT elicitation to deal with multiple parent nodes are discussed in Section 5.

We have yet to complete the CPT for the Design Process Performance node in a way that is satisfactory to the independent assessor. - Although progress does seem to be achievable with this task using the methods outlined in the list of Section 5.

However, sufficient progress has been made with CPT elicitation to enable us to show here a few examples indicating the kind of numerical output that can be produced from such a model. We will concentrate on examining the effect of various combinations of assumptions and observations about the values of other nodes of the BBN on beliefs about the main *goal node* 'Adequacy of Computer System Specification'. Let us agree on a *notation* $(p_a, p_u, p_o, p_g, p_w)$ to represent current probabilistic beliefs about the value of this node, where the subscripts identify the five states, from most unfavourable ('awful') on the left, and where the *p*s are expressed in percentage terms. We begin by making the artificial assumption that all parent nodes of the 'Design Process Performance' node other than 'Quality of Requirements' are assigned hypothetical fixed, known states. This assumption relieves us from the necessity of completing the CPT of 'Design Process Performance' entirely (although some columns of this CPT will still be required in order to proceed: we did elicit these.). Throughout the results that follow, the two parent nodes which were ranked by our assessor as the least significant of the five parents will be assumed fixed in states 'Actual Advantage Achieved by Design Guidelines' = 'yes' and 'Resource Impact' = 'adequate'.

We begin by assigning the values 'Past Competence of Designers' = 'average' and 'Problem Complexity (manufacturer)' = 'moderate'. This yields the distribution (7.09, 28.63, 50.83, 12.46, 1.00) for our goal node. Adding the observation 'Manufacturer Verification Apparent Coverage & Quality' = 'unsatisfactory' lowers, as one might expect, the confidence in the Computer System Specification, giving a distribution (4.41, 57.06, 33.77, 4.14, 0.33). A favourable observation on the same node, 'Manufacturer Verification Apparent Coverage & Quality' = 'good' gives instead the distribution (7.30, 17.89, 57.53, 15.99, 1.28) for the goal node.

We can also enter evidence at the nodes corresponding to the requirements quality sub-net of the BBN. Keeping the assumptions of the last paragraph, but now retracting any 'Manufacturer Verification Apparent Coverage & Quality' observation, and instead making the observations 'Manufacturer Verification Report' = 'average', 'Plant Experts' Safety Assessment Report' = 'thorough', and 'Licensee Verification Thoroughness' = 'average' results in the distribution (5.30, 25.95, 54.4, 13.35, 1.00) which seems to be a slight improvement on the initial beliefs. If we now combine this evidence with the observation 'Manufacturer Verification Apparent Coverage & Quality' = 'unsatisfactory' that we used before from the top part of the net, we obtain (3.66, 53.79, 37.59, 4.61, 0.35) and it seems reasonably clear that this last unfavourable observation has had a larger negative effect on our beliefs than the small increase in confidence that the requirements quality observations alone had appeared to bring.

We finish this illustrative set of numerical outputs by adding to this situation (the one represented by the last figures in the previous paragraph) the effect of more optimistic assumptions for the middle, 'design process performance' part of the net. If we replace our assumptions about designer past competence and problem complexity by the following: 'Past Competence of Designers' = 'good' and 'Problem Complexity (manufacturer)' = 'simple/easy', our consequent improved confidence in the 'Adequacy of Computer System Specification' node is represented by the distribution (2.69, 36.25, 35.12, 22.87, 3.07). Of course, the size of this improvement is partly a consequence of the fact that, due to the current incomplete state of our numerical CPT elicitation exercise for the 'Design Process Performance' node, we have - unrealistically - made assumptions directly about some of its adjacent nodes, rather than requiring longer paths of inference from the nodes truly intended to represent observables for input to the model (such as the nodes: 'Reputation of Designers', 'Previous Experience of Designers', 'Problem Complexity (licensee)', 'Nuclear Application Specific'). When we reach a stage of model development allowing us to do the latter, then we anticipate that the effects of the more realistic 'process evidence' this will allow us to supply will be weaker than the effects of the rather strong process assumptions we have made here.

At this stage, this kind of hypothetical inference is mostly useful for the assessor in exploring and 'debugging' the net. After the BBN has reached its final version, similar inference would be applied to decision problems.

5 Model Elicitation and Development: Problems Encountered, Some *ad hoc* Solutions, Future Work

5.1 General Procedure and some Shortcuts

The elicitation from an expert of all the information needed to define a BBN is, as we have indicated, a complex task in all except the most trivial cases. We found, for example, that it was not possible to sensibly carry out this exercise in a single sequence from nodes, through states and structure, ending with CPTs. Instead, there was extensive iteration: elicitation at a later stage in this sequence resulted in back-tracking and changing the information that had been elicited at an earlier stage. We feel that such an iteration is probably inevitable, if only because the domain expert will often not be an expert on BBNs and will thus be learning this methodology as the exercise proceeds. In such a case, it would be unwise for the elicitors to impose a simple progression, even if this were possible. The assessor and the knowledge elicitor will begin to understand better the full consequences (for later modelling stages) of modelling decisions they take at an early stage only after having gained some experience of these consequences.

The difficulties of elicitation varied through the different stages. By far the hardest problems were posed by the need to elicit the multivariate CPTs required by the net topology. The node ‘Design Process Performance’, as we have indicated, presented the largest problem and this could have been a candidate for the kind of iteration mentioned above. When we realised how difficult the elicitation problem was for this part of the net topology, a possible solution might have been to go back and alter the topology: essentially introducing extra, intermediate nodes to reduce the dimension of this distribution. After discussion with our subject, we chose not to follow this route, and instead sought means of easing the task of eliciting the required six-dimensional probability table.

Listed below are some ways of proceeding that were suggested to the assessor - based on grounds of intuitive reasonableness - as ways of easing this elicitation process, with some success:

- Order, or partially order, the parent nodes (the nodes themselves: not their states, or state combinations) according to their significance, then fix the values of the least significant nodes, and begin by numerically constructing the resulting smaller dimensioned CPT. This CPT could then later be expanded upon by fixing now the states of the most significant nodes and attempting to generalise about the perturbations produced from these probabilities by the weaker parent node influences.

- Order, or partially order, the parent node state combinations in accordance with some believed stochastic ordering of the child node conditional distributions. Then fill in actual numerical probabilities for these parent node value combinations in a table with its columns first ordered from left to right according to this expected stochastic ordering of the distributions.

- Amalgamate certain states of the child node and produce first the conditional distributions for this ‘coarser grained’ child variable.

- Provide probability distributions at ‘extreme’ cases, for example at the extremes of ordered sequences of parent node value combinations (2nd bullet in this list), and then examine intermediate distributions defined at specified linear interpolation points between these extreme distributions (i.e. discrete mixtures or weighted sums of the extreme distributions). The assessor can then make fine adjustments from these values, where it is felt necessary to do so. In some cases, the extreme values between which we interpolated here had first been obtained by stating a distribution for one particular parent value combination which the assessor felt to be a ‘middle point’ in terms of the associated distribution of the child node. This middle distribution was defined as a reference point, to begin the process, with balancing extreme distributions then constructed on either side.

Apart from the use of linear interpolation between distributions obtained as mixtures of extreme distributions, we did not enlist existing theory on parametric distribution families or their estimation. However, there is ongoing research in this area in other current projects (see e.g. [Neil and Galliers 1997]) which we believe may have application to nuclear safety assessment BBNs based on the one described here.

All these methods essentially allow the assessor to assume that the probabilistic dependencies among the random variables of interest enjoy some form of ‘natural’ regularity, so that the CPT can be described, within reasonable approximation, by comparatively few parameters rather than the whole set of its entries. Of course, this is not without risks: if the problem were simple in the first place, the use of a BBN would not be felt necessary. However, we believe it to be more cost-effective to obtain a rough description through these shortcuts, and then eliminate any serious problems in a subsequent revision, than attempting the impossible task of enumerating all values in one heroic session.

5.2 Feedback to the Assessor

We also made use of various forms of visual feedback to the assessor of the probability distributions as they were elicited. There are innumerable ways that this can be done, each useful for different purposes, and in slightly different situations, and capable of usefully becoming quite sophisticated (using similar techniques to those used in graphical exploratory data analysis tools attached to modern statistics packages). As an example of a relatively simple form of visual feedback, we noticed that, where a child node has only three possible states, the conditional distribution of that node, for each particular combination of the values of its parent nodes (i.e. any one column of the node's CPT) can be plotted as a single point in an equilateral triangle (rather than using a column of three numerical values, or a histogram, or pie or bar chart to represent the distribution). This is the triangle $\left\{(p_1, p_2, p_3)\, ; p_i \geq 0\, , p_1 + p_2 + p_3 = 1\right\}$ transformed into 2-dimensional co-ordinates. We mention this representation as one of the many possibilities which might, in principle, be collected in a support toolbox to provide a rich variety of forms of feedback to the analyst of the 'shape' of the beliefs which he or she has entered in numerical CPT form. In the case of this particular triangle-plot example, the three numerical probabilities associated with a point, are proportional to its perpendicular distance from each of the three sides of the triangle. The fact that we have then reduced the representation of a scalar distribution to just a single point's position, allows the effect of changing parent node values to be presented visually without the diagram becoming too crowded for comprehension. We can represent changes in one parent node, by joining a series of these points in the triangle by straight line segments; and the effect of a changes in a second parent node, in combination with the first, can be indicated by a plot consisting of one of these lines for each value of that second node, ideally using a different colour or line format for each line. The resulting plot for the CPT of Figure 2 is shown in Figure 3 to illustrate this idea.

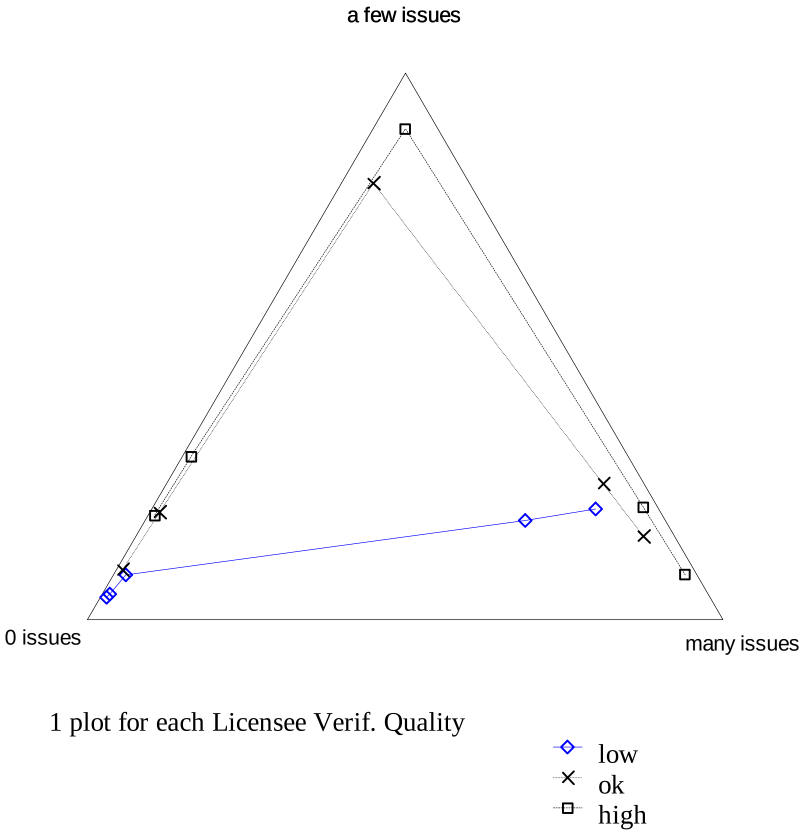


Figure 3 Plot of CPT of 'Licensee Verification Results'

5.3 General Problems in Elicitation

Whilst multi-dimensionality is a major difficulty, it would be wrong to imply that elicitation of probabilities for simpler nodes is a trivial exercise. One method of generating the required distributions that was used by the assessor for several nodes was to derive a first approximation using conditional frequencies from the values taken by the parent and the child states in the previous safety cases in which the assessor had been involved. Then, the assessor would state conditional probability distributions from these frequencies by making small corrections to remove peculiarities resulting from the small sample size or the specific problems with the sampled data. One way of performing these corrections is to estimate the relative likelihood of these peculiar features in practice. This was perceived as a feasible approach to deriving first-cut probabilities. However, it needs to be followed by thorough re-analysis as it appears liable to well known systematic biases [Chhibber *et al.* 1992, Strigini 1994].

A more formal approach when there is empirical data would be to use Dirichlet distributions for each column of CPT as Bayesian conjugate priors, and this is something that we plan to investigate: there are already techniques and automated tools for this [Cooper and Herskovits 1992, Heckerman and Geiger *et al* 1995].

In the absence of abundant empirical data, eliciting the contents of the conditional probability tables in a BBN presents several practical problems, apart from those of scale arising with large multidimensional BBNs. Since the BBN is supposed to represent the independent assessor’s subjective probability distribution, the analyst must ask the assessor for the probability values. However, it is well known that the way elicitation is conducted greatly affects the quality of the results. The assessor may experience several problems due to unfamiliarity with the method, to the complexity of the task, or to basic limitations of the human mind:

- unease at having to express uncertainty in terms of probabilities. This affects even people who are expert in using probabilities in other contexts. It may be true that everyone has subjective probabilistic beliefs, as the Bayesian methods assume, but we are certainly not trained to make them explicit. This unease may make it difficult for the assessor to choose sound procedures for choosing the probabilities;
- strain on one’s attention and concentration, due to the complexity of the dependencies to be modelled and the sheer size of the tables to be filled;
- difficulty in describing a complex structure that one has never explicitly described in detail before, for lack of a language like BBNs;
- dependency on simplified ways of describing one’s beliefs, as the one we have applied here, which may hide from the assessor some complexity of the situation;
- difficulty in deriving proper inference from one’s experience.

The CPTs may thus be unsatisfactory in at least two ways. Firstly there is the obvious risk that they may be inconsistent (i.e. the probabilities input for some conditional distribution may not add[5] up to 1). Secondly, and more problematic, they may not represent a set of beliefs that the assessor would find satisfactory after mature reflection. We would normally expect that the first set of probability values produced by an expert may exhibit this defect, so iteration will be necessary. Many kinds of checks can be applied to detect defects and prompt corrections, though of course there is no certain criterion for certifying the absence of defects. The practical problems arise in deciding which checks to apply, and in which order, to achieve a reasonably effective combination with reasonable effort by the assessor. In particular, we would expect that someone who is familiar with using BBNs could apply checks in a systematic order that attempts to minimise the iterations necessary before the CPTs appear satisfactory. If instead an assessor is using a BBN for the first time, a more exploratory procedure may be necessary: the assessor may find it necessary to practice using the BBN for inference and prediction, even with unsatisfactory CPTs, before attempting to rectify the CPTs.

[5] It can easily be shown that the risk of other forms of inconsistency of the marginal and conditional probability tables used in model construction is removed by the simple requirement that the network topology should be a-cyclic. It follows from the *directed, a-cyclic graph* (DAG) topology that the set of CPTs used to define the BBN model is precisely minimally sufficient to define a coherent joint probability distribution over the complete node set.

Internal inconsistencies such as cycles in the topology or improper CPTs can be detected automatically. Actually, tools like Hugin can automatically scale the probabilities entered so they add up to 1. Of course there is a risk in this kind of functionality: when an inconsistency occurs, it may be useful to check with the assessor whether the inconsistency was really due to some error that he or she would wish to rectify. There is evidence that entering probabilities in the form of odds ("5 to 1 against") is less subject to errors, for many people, than entering them as numbers, though we are uncertain as to whether we should expect this to apply to the elicitation of discrete probability distributions, rather than to only the probability of a single event.

For all other aspects of poor quality in elicited probabilities, an ample literature has developed both about the origins of errors in expert judgement and in reasoning with probabilities, and on ways to correct these errors (see [Strigini, 1996] for some pointers). The assessor needs to be made aware of possible errors, and given feedback in the form of different presentations of the implications of his or her supposed beliefs. There is an essential role to play for an analyst/elicitor, who needs to challenge the expert's first expression of probabilities, so as to prompt the expert to see them with fresh eyes, without forcing the analyst's own beliefs (beyond the analyst's beliefs in coherent Bayesian probabilistic reasoning) on the assessor. Essentially, the assessor is shown some of the implication of his or her statements, and, if these implications are unsatisfactory, is able to acknowledge an inconsistency in his or her first description of beliefs and to revise the description to represent the "true" beliefs. "Truth" need not imply that these beliefs existed in the assessor's mind before the elicitation and revision exercise.

In the case of BBNs, it seems that the information that needs checking spans the whole range from a simple marginal distribution for a root node, to the whole multidimensional distribution expressed by the BBN. The set of all possible checks is clearly redundant (an inconsistency between the CPTs and the assessor's "true" beliefs may be noticed in many alternative ways). However, it seems that the essential needs are still feedback and prompting to re-examine the implications of the CPTs. For complex, multidimensional CPTs graphical feedback appears very useful, as well as dynamic and interactive forms of presentation, like live electronic documents, maybe even animation of a variety of forms of graphical projections, an ability to rotate 3D presentations, etc. There are many alternative ways of presenting specific aspects of a complex distribution, and all of them should in principle be available.

An assessor may obtain useful feedback from observing:-

- distributions that he or she would consider true *before* observing some variables (in formal terms, distributions that are not conditioned on any value of these nodes).
- predictions that the BBN would allow one to derive if certain events were observed (in formal terms, distributions that are conditioned on the observed event)

However, it is clear that the sheer number of possible forms of feedback is in itself a threat to effective checks. An analyst should be able to assist the assessor in choosing a few crucial checks at a time, during the development and successive revision iterations on a BBN. In principle, it is possible that every new form of feedback will prompt the assessor to revise his or her previous description of the CPT. This would imply that the BBN was too complex in the first place. Rather than allowing the assessor to describe a judgement process in manageable, rigorous terms, the availability of the BBN language has prompted him or her to seek a level of detail that makes the task mentally unmanageable again.

6 Discussion and Conclusions

The material in this report discusses work in progress within Task 5.1 of DeVa. Our future plans include both further work on this particular BBN, and work on some of the more general outstanding problems associated with BBNs. Concerning the present BBN, we intend to complete the construction exercise. The main outstanding tasks are to complete the first-pass filling of CPTs for all nodes (by dealing with the Design Process Performance node and with its two ‘least significant’ parent nodes) and to perform a thorough internal validation, by assisting the expert in re-examining the completed BBN and its implications, and on this basis to rectify any discrepancies between the BBN and his or her beliefs (as updated during this exercise).

For completing the CPTs, our present intention is to retain the current topology of the net, and tackle the dimensionality problem via some of the techniques mentioned earlier. In particular, some of the graphical techniques that we have tried look promising. In the event that this approach does not succeed, we would have to reconsider the topology, perhaps introducing another node which might allow us to adduce further conditional independence assumptions. When we have a complete net, including all CPTs, we shall be in a position to obtain some numerical results using a BBN tool such as Hugin.

We plan to restrict our study of this nuclear example to the present net, and not to attempt a full BBN for a complete nuclear ‘system important to safety’, such as might form a part of a safety case: this seems impractical in the time available in the project. However, the DeVa work will continue to be closely associated with, and benefit from, CSR’s other projects on BBNs - SERENE and IMPRESS - which are currently examining a number of different safety-critical applications. A number of general problems in constructing BBNs have arisen from the these studies, and we intend to investigate these in some detail.

One of the most important issues concerns the *validation* of a BBN. This has three distinct aspects. In the first place there are issues such as consistency that are open to automatic checking, and probably present the least difficulty. Then there are issues concerning the confidence of the assessor that the BBN actually does capture his or her beliefs about the system, or class of systems, under examination. Clearly this is not simply a question of asking whether this is so: instead, the assessor needs support for these judgements at all stages of the elicitation process. Thus, for example, it would be valuable to have tools that allow implications of CPTs to be generated easily, so that an assessor can judge whether these are compatible with his or her beliefs. Some of these tools are currently being produced by the SERENE project, and DeVa will take advantage of their availability. Finally, there are the very hard problems concerned with *external* validation - the question of whether the ‘final’ BBN really does capture the ‘truth’ about the external world (e.g. it makes accurate predictions of system behaviour). Even leaving aside the issue of experts with false world pictures, there is the possibility for the process of generating a net to fail, resulting in a net that imperfectly captures the expert's *correct* beliefs without the expert knowing this.

It was clear from an early stage in our work that there is a need for more general guidance and tool support at all stages in building BBNs. Whilst we have tended to concentrate upon the difficulties of eliciting complex multivariate probability distributions, even the earlier task of defining the net topology can be difficult. There is a possibility of progress here through the use of 'idioms' [Neil and Galliers 1997] - essentially small subnet structures that have been found to be common to BBNs in many application domains**.** The idea is that an assessor's progress in building a net structure can be helped by suggesting to him or her the possibility of using these building blocks. The assessor can then save the effort that would otherwise be spent in learning by trial and error convenient representations for these commonly occurring reasoning patterns.

A major problems in building BBNs, however, tends to be in elicitation of complex CPTs. We shall continue to examine the approaches that have been outlined briefly earlier, including means of graphical presentation.

A long term view of the usefulness of BBNs should not only be concerned with the accuracy of the quantitative results that they produce, important though this is. In many cases BBNs will be used in circumstances where an assessor currently has available only quite informal ways of reasoning about a system, based on evidence of disparate types, and using his or her own expert judgement when direct evidence is missing. A major potential strength of the BBN approach is that it provides a formal framework in which this reasoning can take place, facilitating argument and criticism about assumptions and reasoning. The BBN approach will have been made a useful contribution, for example, if the assessor feels that he or she has learned something from the exercise that would not have been possible with a less formal approach; or if the assessor has been able to make explicit, and open to discussion by other experts, the detailed assumptions and beliefs that went into his or her reasoning. There is some evidence that this is the case in our example net, even though it is still incomplete.

7 Acknowledgements

This work was supported by the ESPRIT Long Term Research Project 20072 on 'Design for Validation' (DeVa). It has also benefited from the contribution of colleagues working on the ESPRIT Framework IV Information Technology Programme SERENE Project (22187).

References

[Andersen *et al.* 1989] S. K. Andersen, K. G. Olesen, F. V. Jensen and F. Jensen, "HUGIN-A Shell for Building Bayesian Belief Universes for Expert Systems", in *11th International Joint Conference on Artificial Intelligence,* (Detroit 1989), pp.1080-84, 1989.

[CACM 1995] CACM, "Real-World Applications of Bayesian Networks", *Communication of the ACM, Special Issue*, 38 (3), pp.24-57, 1995.

[Chaloner *et al.* 1992] K. Chaloner, T. Church, T. A. Louis and J. P. Matts, “Graphical Elicitation of a Prior Distribution for a Clinical Trial”, in *Conference On Practical Bayesian Statistics,,* (Univ. of Nottingham, Nottingham, United Kingdom), pp.341-53, 1992.

[Chhibber *et al.* 1992] S. Chhibber, G. Apostolakis and D. Okrent, “A taxonomy of issues related to the use of expert judgements in probabilistic safety studies”, *Reliability Engineering & System Safety,* 38 (1-2), pp.27-45, 1992.

[Cooper and Herskovits 1992] G. F. Cooper and E Herskovits, “A Bayesian Method for the Induction of Probabilistic Networks from Data”, *Machine Learning*, vol 9, pp309-87, 1992.

[Delic *et al.* 1995] K. A. Delic, F. Mazzanti and L. Strigini, *Formalising a software safety case via belief networks,* SHIP Project, Technical Report, N°SHIP/T/046, July 1995.

[Delic *et al.* 1997] 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), p.to appear, 1997.

[Fenton 1991] N. E. Fenton, *Software Metrics : A rigorous Approach*, Chapman & Hall, London 1991.

[Heckerman and Geiger *et al* 1995] D. Heckerman and D. Geiger and D. M. Chickering, “Learning Bayesian Networks: The Combination of Knowledge and Statistical Data”, *Machine Learning,* vol 20, pp197-243, 1995.

[Jensen 1996] F. V. Jensen, *An introduction to Bayesian networks,* 208p., UCL Press (Published in North America by SpringerVerlag NewYork Inc.), 1996.

[Keeney & von Winterfeldt 1991] R. L. Keeney and D. von Winterfeldt, “Eliciting probabilities from experts in complex technical problems”, *IEEE Transactions on Engineering Management,* 38 (3), pp.191-201, 1991.

[Lauritzen and Spiegelhalter 1988] .S. L. Lauritzen and D. J. Spiegelhalter, “Local Computations with Probabilities on Graphical Structures and their Application to Expert Systems”, *Journal of the Royal Statistical Society, Series B*, vol 50(2), pp 157-224, 1988, (with discussion).

[Neil and Galliers 1997] M. Neil and J. Galliers, “Task 1.3 Report”, ESPRIT Framework IV IT Program Project SERENE 22187, Doc. No. SERENE/1.3/CSR/3010/R/2, 1997.

[Pearl 1988] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference,* Morgan-Kaufmann, 1988.

[Pearl 1993] J. Pearl, “Belief Networks Revisited”, *Artificial Intelligence*, 59, pp.49-56, 1993.

[Strigini 1994] L. Strigini, *Engineering judgement in reliability and safety and its limits: what can we learn from research in psychology?*, SHIP Project, Technical Report, N°SHIP/T/030, July 1994.

[Vanlenthe 1993] J. Vanlenthe, “A Graphically Oriented Technique For Eliciting Subjective Probability Distributions”, in *5th Workshop on Computers In Psychology - Applications, Methods, and Instrumentation*, (F. J. Maarse, A. E. Akkerman and M. l. Brand_an, Vanderstelt_mj, Eds.), (Nijmegen Univ, Nijmegen, Netherlands), pp.178-92, Taylor & Francis, 1993.