School of Computer Science and Software Engineering
Monash University

Bachelor of Computer Science Honours (1608), Clayton Campus

Literature Review - 2003

Knowledge Engineering a Bayesian Network for an Ecological Risk Assessment (KEBN-ERA)

Owen Grant Woodberry
ID: 12093009

Supervisors:
Ann Nicholson
Kevin Korb
Carmel Pollino

July 31, 2003

Contents

Abstract

This thesis develops upon existing research in the field of knowledge engineering Bayesian networks for application domains. The application to be studied in this project is a risk assessment in the ecological domain.

In this chapter I review existing literature regarding knowledge engineering Bayesian networks in general, for this particular application domain and for this particular risk assessment. I start by providing background on Bayesian networks, which are based on probability and graph theory, accounting for their construction and use. I then provide background on the knowledge engineering field, and also review the existing methods for evaluating Bayesian networks. I then introduce the ecological risk assessment that is the basis of my project, defining its scope, approach and objectives.

1 Bayesian Networks

Bayesian networks (BNs) are graphical tools for reasoning with probabilities. BNs have a qualitative and quantitative component. The qualitative component represented by a graphical structure composed of nodes with relationships between them, representing random variables and causal influences. The quantitative component represented by conditional probability tables (CPT) for each node, representing the effects of its parent nodes on it [25, Chapter 3], [6]. The power of Bayesian networks lies in its ability to calculate the probability of an event given a set of evidence from only a small set of probabilities defined in the CPT's. It does this by using Bayes rule and conditional independence relationships between variables to reduce the number and form of conditional probabilities required to represent the problem.

In this section I summarise the literature regarding Bayesian networks, first reviewing the underlying theory on which BNs are based before going on to talk about their use and identify some applications. The construction and evaluation of BNs are discussed in later sections.

1.1 Underlying Theory

Because BNs are graphical tools for reasoning with probabilities it is necessary to introduce both the probability and graph theory that are the basis of BNs before introducing BNs.

1.1.1 Probability Theory

When dealing with small, theoretical problems it is often enough to consider these predictions as an application of a set of IF-THEN statements, this is called classical logical inference. Although in many real world applications this type of inference is not useful due to uncertainty [12]. Uncertainty arises with real world applications because our understanding of the world is either incomplete or incorrect. For even mildly complex problems the amount of logical rules required to explain a domain is considered to be much too large to be useful [27, Chapter 14]. For these reasons, when dealing with real world problems it to often best to limit talk to that of degrees of belief or probabilities. When we talk in terms of probabilities we need to define a formal language for representing and reasoning with probabilities.

Prior Probability

The notation, $pr(\mathrm{A})$, is used to represent the prior probability distribution of a random variable, A [25, Chapter 2]. For example the result of a single coin toss, C, has a domain of possible values, heads and tails, {h, t}. So we could write:

$$pr(\mathrm{C}) = \{0.5, 0.5\}$$

to represent the prior probability distribution of a coin toss C for a fair coin, and:

$$pr(\mathrm{C} = \mathrm{t}) = 0.5$$

to represent the prior probability, 0.5, of the event tails, t, occurring.

Joint Probability

The notation, $pr(\mathrm{A}, \mathrm{B})$, is used to represent all combinations of values of a set of random variables called the joint probability distribution (JPD) [25, Chapter 2]. In the case of two coin tosses, we have the domain of possible values, {hh, ht, th, tt}, therefore the resulting probability distribution has 2^2 entries and will continue to grow at an exponential rate as the number of variables increases.

Conditional Probability / Posterior Probability

The notation, $pr(\mathrm{A}|\mathrm{B})$, is used to represent the conditional, or posterior, probability distribution of the random variable A given some evidence B [25, Chapter 2]. Using the notation defined above with the joint probability distribution and prior probability we can define the conditional probability as,

$$pr(\mathrm{A}|\mathrm{B}) = \frac{pr(\mathrm{A}, \mathrm{B})}{pr(\mathrm{B})} \tag{1}$$

provided that $pr(\mathrm{B}) > 0$. Returning to the example of the two coin tosses,

$$pr(\mathrm{C2}|\mathrm{C1}) = pr(\mathrm{C2}) \tag{2}$$

because the variables C1 and C2 are conditionally independent of each other, that is, knowledge about the result of the first coin toss, C1, doesn't change the probabilities for the second coin toss, C2:

$$pr(\mathrm{C1}, \mathrm{C2}) = pr(\mathrm{C1}) \times pr(\mathrm{C2}) \tag{3}$$

Bayes' rule

Using the notation for conditional probability (1) combined with the symmetry rule:

$$pr(\mathrm{A}, \mathrm{B}) = pr(\mathrm{B}, \mathrm{A}) \tag{4}$$

we can define Bayes' rule [25, Chapter 2]:

$$pr(\mathrm{A}|\mathrm{B}) = \frac{pr(\mathrm{B}|\mathrm{A}) pr(\mathrm{A})}{pr(\mathrm{B})} \tag{5}$$

1.1.2 Graphical Models

The joint probability distribution for a problem captures the probability information of every possible combination of a set of variables, and their states. Once a joint probability distribution has been defined for a problem domain then it is possible, using it along with the axioms of probability, to answer any probabilistic query regarding any the variables. This includes their value given additional evidence, that is, their conditional probability. Although, as was stated earlier, the space, and consequently time, complexity required to represent and manipulate the joint probability distribution is exponential in the number of variables to be considered [8]. For example the joint probability distribution required to represent a system with 20 binary values would have 2^{20} (1,048,576) values. This causes such problems in the elicitation, storage and manipulations of these values to make the use of joint probability distribution unfeasible for any practical use. Fortunately, when modelling most real systems, we can take advantage of any inherent structure the system has by modelling the system as a graph [8]. When we talk in terms of graphs we need to define some terms of the formal language of graphs.

Graph Theory

Some relevant graph theory definitions:

A graph is defined as a pair (V,E), where V is a finite set and E is a binary relation on V [25, Chapter 3]. That is, a graph is a finite set of vertices or nodes with a set of edges or relationships connecting these nodes.

A graph is a directed acyclic graph (DAG) if the all relationships within the graph are directed and the graph contains no cycles [25, Chapter 3].

D-separation

The inherent structure of a system can be defined in terms of dependency/independency assumptions between variables. A graphical model can greatly simplify the representation of the joint probability distribution capturing any dependences, independences, conditional independences and marginal independences between variables. To understand and identify these dependency/independency assumptions it is useful to first understand the concept of d-separation.

Direction dependent separation or d-separation [27, Chapter 15] is used to determine if two nodes are conditionally independent given evidence of some other node. Formally, if every undirected path from a node in set X to a node in set Y is d-separated by a node in set E, then X and Y are conditionally independent given E. The set of nodes E, d-separates sets X and Y if every undirected path between X and Y is **blocked** given evidence E. A path is blocked given the set E, if there is a node Z on the path for which one of three conditions hold (see Figure 1),

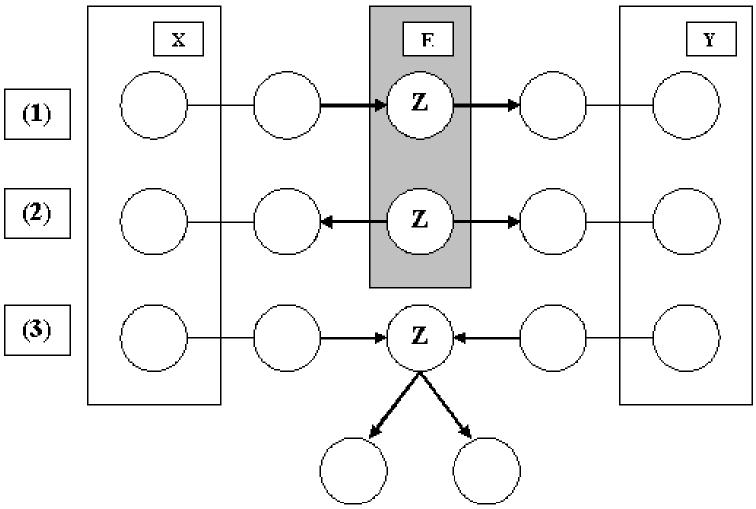

Figure 1: D-Separation [27, Chapter 15]

(1) Z is in set E and Z has a relation from either set X or Y directed in and a relation from the remaining set directed out.

(2) Z is in set E and Z has both relations from sets X and Y directed out.

(3) Neither Z nor any of it children are in set E and Z has both relations from sets X and Y directed in.

Using this concept of d-separation helps us to understand, and represent, the following relationships between variables from the topology of the graph.

Independent: nodes A and B are not connected by any path in the graph. Therefore nodes A and B have no influence on each other.

Dependent: nodes A and B are directly connected in the graph. Therefore nodes A and B have direct influence on each other.

Conditionally independent: nodes A and B are connected via a third node C as in cases 1 and 2. Therefore nodes A and B have influence on each other if nothing is known about the state of node C and d-separated from each other given knowledge of the state of node C.

Marginally independent: nodes A and B are connected via a third node C as in case 3. Therefore nodes A and B are d-separated from each other if nothing is known about the state of node C or any of it's children and do have influence on each other given knowledge of the state of node C or any of it's children.

1.1.3 Bayesian Networks

It is now possible to give a formal definition for Bayesian networks.

A Bayesian network is a directed acyclic graph that represents a joint probability distribution. It has nodes that represent random variables, and arcs that represent probabilistic relationships, or correlations, between the variables. Qualitative information in the types of paths, or lack of paths, between variables indicates dependence/independence relationships. Quantitative probability information in the conditional probability table for each node specifies the probability of each possible state, and relative uncertainty, given the possible states of its parents [25, Chapter 3],[10].

1.2 Reasoning with Bayesian Networks

When reasoning with BNs a user will want to find the probability of an event given some or no knowledge of the system state. The nodes in the BN can be broken into three categories; those nodes that we wish to gain knowledge of, which we call the query nodes, those nodes that we already have knowledge of, which we call the evidence nodes, and the remaining nodes [27, Chapter 15]. We enter our knowledge of the system state by selecting values for the evidence nodes, there is then some inference algorithm used to update the posterior probability distributions of the query nodes.

The inference algorithm involves four types of inference [27, Chapter 15] (see Figure 2):

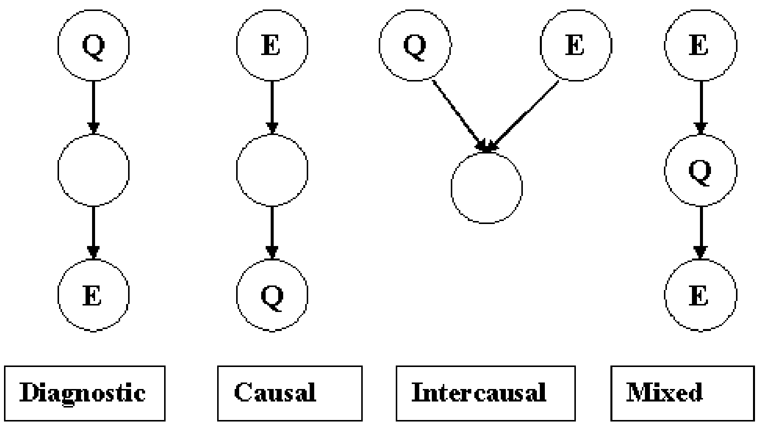

Figure 2: Types of Inference [27, Chapter 15]

1.**Diagnostic inference**: involves updating beliefs from effects to causes.

2.**Causal inference**: also called prediction, involves updating the beliefs from causes to effects.

3.**Intercausal inference**: also called explaining away, involves updating the beliefs between causes of a common effect.

4.**Mixed inference**: involves updating beliefs using a mixture of the inferences listed above.

All these different types of inference can be made with the CPT's for each node. For example, $pr(\mathrm{A}|)$, $pr(\mathrm{B}|\mathrm{A})$, $pr(\mathrm{D}|\mathrm{B},\mathrm{C})$ and bayes' rule allows the computation of $pr(\mathrm{A}|\mathrm{B})$ and $pr(\mathrm{B},\mathrm{C}|\mathrm{D})$. Hence, a BN together with the inference algorithm allows any of these types of reasoning with the system represented by the BN.

1.3 Bayesian Network Applications

Bayesian networks have been applied to a great variety of problems in medical diagnosis [22, 11, 24, 19], Microsofts Office Assistant [14], Bayesian Poker [17, 5], Seabreeze prediction [15], intelligent tutoring [23] and many other applications. Some applications that have particular relevance to this project are ecosystem modelling [2, 3], natural resource management [4] and species-environment relations [20, 21].

2 Knowledge Engineering

In this section I summarise the literature regarding knowledge engineering Bayesian networks, first reviewing the problems of elicitation, the solutions provided by the knowledge engineering field and then going on to talk about the spiral development model and its applicability to the development of BN software systems.

2.1 Modelling Tasks

The process of knowledge engineering a Bayesian network can be broken up into three tasks [16, chapter 10]. The first two tasks relate to defining the graphical structure of the network. The last task is to define the conditional probability tables for each node [9]. These tasks as listed here:

1. Identifying the set of variables and their states the make up the system.

2. Identifying the graphical structure, that is, the qualitative structural assumptions of the system.

3. Identifying the conditional probability tables of each variable, that is, the quantitative effects of variables parent nodes on it.

Along with properly defining the problem domain, an objective of the first two steps is to express the problem in its simplest yet sufficiently complete form. This is done to reduce the number and form of probabilities to be entered into the CPTs which is to be done in the third task, and is often considered the most difficult task.

In this section I will address each of the tasks in eliciting knowledge from domain experts from the knowledge engineering perspective, identify some automated methods available and talk about the potential issues created by integrating these methods.

2.1.1 Eliciting Variables and their States

When selecting variables to be modelled, it is important to limit their number to keep the knowledge engineering task tractable. This can be done by only including the more important variables, identified by determining whether it falls into one of these four categories [16, Chapter 10], [4]:

1. **Query variables**: also called objective or target variables, are those variables that are to be the 'output' of the network, i.e. the variables the end-user wants to know about.
2. **Evidence variables**: also called observation or controlling variables, are those variables that are to be the 'input' of the network, i.e. the variables that would be potentially useful in inferring the states of the query variables.
3. **Context variables**: also called intermediate variables, are those variables that link the query and evidence variables.
4. **Controllable variables**: also called intervention variables, are those variables that that could potentially be interventions to the domain system.

After selecting the set of variables to be used in the model the next task is to decide on the states, or values, that each variable can take. As with selecting the variable set it is useful to limit the number of states to minimise the size of the network. To decide what states to include for a variable a possible simple guide is [4],

1. The state it is currently in.
2. The state/s toward which you think it may move under possible management plans.
3. Any intermediate states.

Only those states that are possibly of interest to the end-user should be included. When selecting states it is necessary is ensure that the variable states are exhaustive and exclusive, this means that a variable must take, at any particular point in time, exactly one of these states. Although not required, it is usually simplest to represent continuous variables as discrete, this can be done by converting the original range of continuous values into a finite set of sub-ranges.

2.1.2 Eliciting Graphical Structure

As with selecting variables to be modelled, it is also important to limit the number of relationships between variables to keep the knowledge engineering task tractable. It is also important because it keeps the BN understandable. When determining the structure of the network the key is to focus on the relationships between the variables. As was stated in the preceding section there is four types of relationships between variables independent, dependent, conditionally independent and marginally independent.

These relationships can be determined by asking direct questions about these relationships, such as, what can cause variable 'A' to take on this state? Any answers would suggest a causal relationship between the variable answered and variable 'A'. A support tool for this type elicitation, called MATILDA, is discussed in the preceeding evaluation section.

2.1.3 Eliciting Conditional Probability Tables

Eliciting the CPT's is often considered the most difficult task when creating Bayesian networks. The CPT's for each node specify the values of the node given the values of its parents. For each possible instantiation of parent values there is a probability distribution, this means that the probability elicitation task is exponential in the number of parent variables. This is one of the reasons that the aims of the previous elicitation tasks were to keep the graphical structure as simple as possible whilst properly representing the system modelled. There are three possible elicitation sources [16, Chapter 10], these are domain experts, experimental data and literature:

Domain experts: the obvious approach using domain experts is to directly elicit the values to be entered into the CPT's, by asking questions such as, what is the probability that variable 'A' takes this state given these parent variable values? However it may be better to work in terms of frequencies, odds or even qualitative assessment using terms such as the probability is 'high' or 'unlikely', a support tool, called VE [13], aids in this type of elicatation by mapping verbal terms to probabilities. This source of elicitation may be the only source available in some domains, although it is often difficult to find experts who have the time and interest to go though the elicitation process and there will, almost certainly, be problems with bias in the estimation of CPT's by humans.

Experimental data: if there is enough data then the process of training the CPT's can be entirely automated [16, Chapter 10], although there may be problems of noisy data, missing values, bias created from the collection of the data and the values of the data may not match the variables in the model.

The literature: many of the CPT's required may already be specified the literature concerning the application domain, although it is very unlikely that this elicitation source will cover all the probability distributions that need to be specified. There may also be bias in the information selected in the literature.

It is also possible to combine information from these different sources, although it would be necessary to specify the confidence or weighting of the probabilities acquired from each source. There is also risk in combining probabilities from different sources of accumulating the biases, which won't necessarily cancel each other out.

2.2 Development Model

As research into BNs increases the understanding of and therefore the applications of BN networks, the problems and systems they are being applied to are becoming larger in size and greater in complexity [18]. As with other large software systems, the development of these large

BNs requires an engineering process. There is much literature particular to the BN elicitation process and other aspects of the knowledge engineering process can be drawn from the general software engineering literature.

It often useful to consider the software engineering process as having a lifecycle, it is born during the requirements phase, it matures through design and development and eventually it is retired. This view of a softwares lifecycle is standard to the waterfall development model popular in software engineering. Another way of viewing this lifecycle is to see the software as continually growing organism [16, Chapter 10], which at any point in time is a functional, even if limited, implementation of the final system. This organism is continuously trialled, or evaluated, and grows according to the outcomes of these life trials. This is the view standard to the spiral development model, the software goes through a repeating cycle of design, development, operation and evaluation (see Figure 3). The early iterations of the lifecycles are generally called prototypes and at some point these prototypes eventually become system versions.

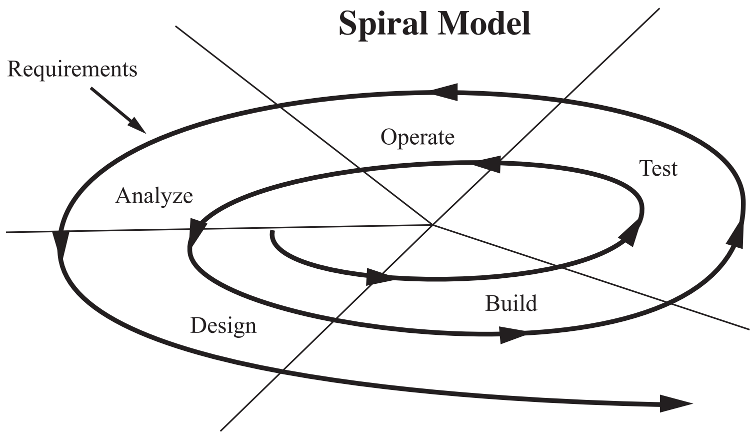

Figure 3: Spiral Development Model [18]

The main reason for preferring the spiral to the waterfall lifecycle model is risk management. One of the prime difficulties in software engineering is concerning the requirements phase [18], this is mostly due to the communication gap between the knowledge engineer and the domain expert. With the spiral development model requirement development parallels the prototype development, as each prototype is created and evaluated, the requirements are refined and the potential risks are better understood. The prototype does this by providing an effective communication tool that bridges the gap between the knowledge engineer and the domain expert. At any point in time the system can be trialled in the domain by the users for which it is intended for to gain valuable feedback from early on in its development. This prototyping development model is particularly useful in the development of BNs because BNs are usually built using visual aids and with graphical interfaces, making a prototyping approach even more relevant [16, Chapter 10].

2.3 Knowledge Engineering for Domain Specific Problems

It is considered important to consider this engineering process in the domain for which the system is to be used. This is to account for differences in the viewing of the concepts of the domain, by experts and users, differences in what the system is to be used for and differences in the type of data available to support the elicitation process. Some of the different application domains that BNs are being used in were overview in a previous section.

3 Evaluation Methods

As identified in the preceding section of development models, the evaluation of the Bayesian network is essential. Evaluation methods are useful to grade the Bayesian network, identify errors and possible improvements. When the spiral development model is used we can start evaluating the BN as soon as the first prototype is finished and minimise any risks of an invalid model being produced.

In this section I summarise the literature regarding methods for evaluating Bayesian networks, first talking about evaluation using feedback from domain experts and then evaluation using statistical methods that can be automated.

3.1 Domain Expert Evaluation

When evaluating a BN using feedback from experts, it is necessary to find domain experts that were not involved in the BN creation process. The problems involved with the elicitation process are also common with this type of evaluation. It is often difficult to find experts who have the time and interest to go though a BN evaluation process, although this is not as time consuming as the elicitation process.

In this section I summarise the literature regarding four types of BN evaluation using feedback from domain experts. These are using an elicitation review, MATILDA, sensitivity analysis and case based evaluation.

3.1.1 Elicitation Review

This is a formal structured review of the elicitation process, providing a global overview of the decisions made during the development of the model [16, Chapter 10]. Primarily focused on the qualitative component of the model, that is the variable values and the graphical structure of the model. First the selections of variables are reviewed checking for clarity, consistency and clearness of definitions. Then the graphical structure is reviewed checking whether the structure and the implications of d-separation in the model violate any prior knowledge of causality and independencies in the actual system. A tool developed to support this type of evaluation, called MATILDA, is discussed in the proceeding section.

3.1.2 MATILDA

This support tool [1] was developed to aid in communicating and explaining the graphical component of a BN to the domain expert who may not immediately understand the implications of dependencies and d-separation in the model. It is useful in evaluating, and developing, the network, as it supports the comparison of structural assumptions of the domain and the qualitative modelling decisions, without the need specify the quantitative component of the model.

3.1.3 Sensitivity Analysis

Sensitivity analysis is primarily used to evaluate the quantitative component of the model. There are two different types of sensitivity analysis, these are testing how sensitive the network is to changes in the findings and changes in the parameters. In the former case the influence of each of the nodes in the network on a query node can be measured, using a measure such as entropy, and ranked. This is useful to prioritise the portions of the model for later iterations in the development cycle [18]. In the latter case the influence of each of the parameters on the network can be tested, determining whether more precision in estimating them would be useful in later iteration of the development cycle. Contrary to previous belief [26] it can be shown that although a network may be insensitive to many of its parameters it can also be very sensitive to certain parameters [28].

3.1.4 Case Based Evaluation

It is also often useful to conduct a walkthrough, running the network though a set of various cases, ideally, exhaustively testing all situations allowed by the model. The cases are entered as findings in the network and the posterior values of remaining nodes can then be evaluated based on the prior knowledge of the domain expert indicating potential errors in the network. A potential difficulty with this approach is that the domain expert may not have the experience to make a judgement on all such cases. The evaluation methods discussed in the proceeding sections draw upon this type of evaluation using automated statistical methods rather than relying on the human domain expert's judgement.

3.2 Automated Evaluation

If there is a large body of data available for the system domain then this can be used to evaluate the network. As with evaluation from domain experts it is undesirable to use the same information source used for the evaluation process as was used for the creation process. Even less so in the case of data as it does not change its perspective, to address this problem it is standard to divide the data into 90/10 or 80/20 split and use the 10 or 20 per cent to evaluate the BN.

In this section I summarise the literature regarding two of the popular types of BN evaluation using automated methods. These are evaluating the predictive accuracy and expected value.

3.2.1 Predictive Accuracy

As with the case based evaluation via a human domain expert, in testing the predictive accuracy of the network, various cases are entered as findings in the network. The predicted value, the most probable state, of the remaining nodes can then be compared with the actual values withheld in the case data, giving a measure of the predictive accuracy, or conversely the error rate, of the network. Such a measure is relatively simple to obtain and is a good initial grade of the model, although it disregards the confidence of the prediction, that is a prediction of 51% versus 99.9% are treated the same [16, Chapter 11]. A better measure would include and reward correct calibration, such a measure is discussed in the proceeding section.

3.2.2 Expected Value

This is a cost sensitive evaluation method, instead of only rewarding the model for predictive accuracy, this will prefer a model with the best weighted average prediction. It does this by assigning utility values to correct and incorrect classifications. This means that instead of treating, for example, a false positive the same as a false negative, they will have different weights associated to them. This could be illustrated by an example from a medical diagnosis [16, Chapter 11] with a false positive diagnosis of a cancer versus a false negative diagnosis, which would have very different consequences. The main difficulty with this type of evaluation is obtaining the utility values which may not be obvious to any domain expert.

4 A Bayesian Network for an Ecological Risk Assessment

The application of this thesis is an ecological risk assessment (ERA). The objective of this ERA is to develop and test a generic framework to be used in the assessment of ecological risks associated with Australian irrigation activities. This project is funded by the National Program for Sustainable Irrigation and is currently being conducted at the water studies centre at Monash University, under Dr Carmel Pollino. The overall objective of this project is to develop a model to determine the effects of alternative management actions on the native fish abundance and diversity. It was considered by the group that Bayesian network technology would provide the best model to do this and one is currently in production.

In this section I first review the conceptual model that the current BN is based on. I then talk about the study area and time scales to be considered and proposed approach to be taken.

4.1 Conceptual Model

During Phase 1 of the ERA a conceptual model was created to show possible factors influencing the native fish population and diversity, a simplified version of this model is shown here (see Figure 4), the double arrow links are included to demonstrate possible interaction between factors:

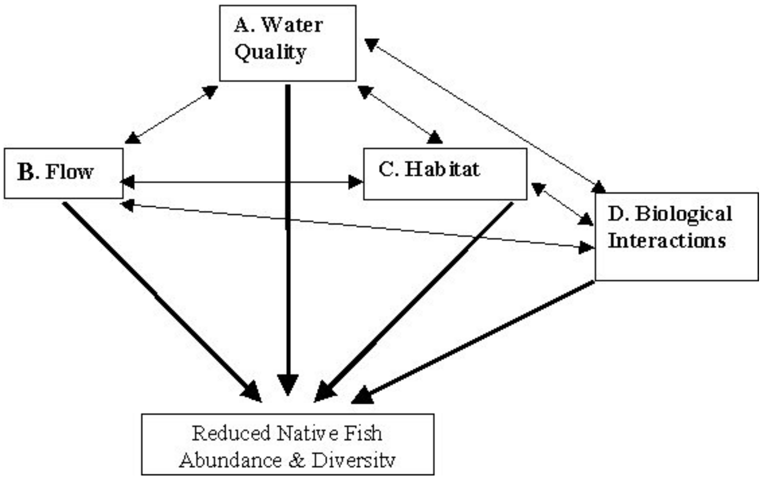

Figure 4: Conceptual Diagram

4.2 Study Areas and Time Scales

The study is being conducted in the Goulburn Broken Catchment with geographical areas to be considered broken into regional and local scales. The local scale chosen was the Goulburn Weir/Lake Nagambie, and the regional scales are the Goulburn River reaches from Eildon to Seymour and Murchison to the Murray River. Time scales to be considered are 1, 5, 10 years, and 30 to 50 years; these were picked to properly reflect the life history of the fish species in the area.

4.3 Approach

Phase 1, the problem formulation phase, of the ERA identified the native fish abundance and diversity as being at risk due to irrigation activities in the area [7]. Phase 2 of the ERA is to identify and quantify these hazards and develop a model to measure the probability of increasing or decreasing the population abundance and diversity in response to varied management interventions. Due to the incompleteness and incorrectness of field data most of the elicitation and evaluation process will rely heavily on feedback gained from running workshops with domain experts and stakeholders. What information can be gained from the data is to be incorporated into the model quantitative component. It is yet to be determined whether the field data can be used for automated evaluation.

References

[1] T. Boneh. Support for graphical modelling in Bayesian network knowledge engineering: a visual tool for domain experts. Master's thesis, Dept. of Computer Science, University of Melbourne, 2003.

[2] Mark Borsuk, Robert Clemen, Lynn Maguire, and Kenneth Reckhow. Stakeholder values and scientific modeling in the neuse river watershed. *Group Decision and Negotiation*, 10:355–373, 2001.

[3] M.E. Borsuk, C.A. Stow, and K.H. Reckhow. Integrative environmental prediction using bayesian networks: A synthesis of models describing estuarine eutrophication. In Andrea E. Rizzoli and Anthony J. Jakeman, editors, *Integrated Assessment and Decision Support, Proceedings of the First Biennial Meeting of the International Environmental Modelling and Software Society*, pages 102–107. iEMSs, June 2002.

[4] J Cain. Planning improvements in natural resources management. Technical report, Centre for Ecology and Hydrology, Crowmarsh Gifford, Wallingford, Oxon, UK, 2001.

[5] J. Carlton. Bayesian poker. Unpublished Honours thesis, School of Computer Science and Software Engineering. www.csse.monash.edu.au/hons/projects/2000/Jason.Carlton/, 2000.

[6] Eugene Charniak. Bayesian networks without tears. *Artificial Intelligence Magazine*, 12:50–63, 1991.

[7] P. Cottingham, R. Beckett, P. Breen, P. Feehan, M. Grace, and B. Hart. Assessment of ecological risk associated with irrigation systems in the goulburn broken catchment. Technical report, ACT, Cooperative Research Centre for Freshwater Ecology, 2001.

[8] B. D'Ambrosio. Inference in Bayesian networks. *Artificial Intelligence Magazine*, 20(2):21–36, 1999.

[9] M.J. Druzdzel and L.C. van der Gaag. Building probabilistic networks: Where do the numbers come from? Guest editors introduction. *IEEE Trans. on Knowledge and Data Engineering*, 12(4):481–486, 2001.

[10] P. Haddaway. An overview of some recent developments in Bn problem-solving techniques. *Artificial Intelligence Magazine*, 20(2):11–19, 1999.

[11] David Heckerman. Probabilistic similarity networks. *Networks*, 20:607–636, 1990.

[12] M. Henrion, J.S. Breese, and E.J. Horvitz. Decision analysis and expert systems. *Artificial Intelligence Magazine*, 12:64–91, 1991.

[13] L. Hope, A.E. Nicholson, and K.B. Korb. Knowledge engineering tools for probability elicitation. Technical report, School of Computer Science and Software Engineering, Monash University, 2002. 2002/111.

[14] E. Horvitz, J. Breese, D. Heckerman, D. Hovel, and K. Rommelse. The Lumiere project: Bayesian user modeling for inferring the goals and needs of software users. In *Proc. of the 14th Conf. on Uncertainty in AI*, pages 256–265, 1998.

[15] R.J. Kennett, K.B. Korb, and A.E. Nicholson. Seabreeze prediction using Bayesian networks. In *PAKDD'01 – Proc. of the 4th Pacific-Asia Conf. on Knowledge Discovery and Data Mining*, pages 148–153, Hong Kong, 2001.

[16] Kevin B. Korb and Ann E. Nicholson. *Bayesian Artificial Intelligence*. CRC Press, 2003.

[17] Kevin B. Korb, Ann E. Nicholson, and Nathalie Jitnah. Bayesian poker. In Laskey and Prade, editors, *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence*, pages 343–350, Sweden, 1999.

[18] K.B. Laskey and S.M. Mahoney. Network engineering for agile belief network models. *IEEE: Transactions on Knowledge and Data Engineering*, 12(4):487–498, 2000.

[19] L. Leibovici, M. Fishman, H.C. Schonheyder, C. Riekehr, B. Kristensen, I. Shraga, and S. Andreassen. A causal probabilistic network for optimal treatment of bacterial infections. *IEEE: Transactions on Knowledge and Data Engineering*, 12(4):517–528, 2000.

[20] B. G. Marcot, R. S. Holthausen, M. G. Raphael, M. Rowland, and M. Wisdom. Using bayesian belief networks to evaluate fish and wildlife population viability under land management alternatives from an environmental impact statement. *Forest Ecology and Management*, 153(1-3):29–42, 2001.

[21] B.G. Marcot. A process for creating bayesian belief network models of species-environment relations. Technical report, USDA Forest Service, Portland, Oregon, 1999.

[22] S. Monti and G. Carenini. Dealing with the expert inconsistency in probability elicitation. *IEEE: Transactions on Knowledge and Data Engineering*, 12(4):499–508, 2000.

[23] A. Nicholson, T. Boneh, T. Wilkin, K. Stacey, L.Sonenberg, and V. Steinle. A case study in knowledge discovery and elicitation in an intelligent tutoring application. In *Proc. of the 17th Conf. on Uncertainty in AI*, pages 386–394, Seattle, 2001.

[24] D. Nikovski. Constructing Bayesian networks for medical diagnosis from incomplete and partially correct statistics. *IEEE: Transactions on Knowledge and Data Engineering*, 12(4):509–516, 2000.

[25] Judea Pearl. *Probabilistic Reasoning in Intelligent Systems.* Morgan Kaufmann, San Mateo, Ca., 1988.

[26] M. Pradhan, M.Henrion, G. Provan, B. Del Favero, and K. Huang. The sensitivity of belief networks to imprecise probabilities: An experimental investigation. Technical Report KSL-95-66, Knowledge Systems Laboratory, Medical Computer Science, 1995.

[27] Stuart Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach.* Prentice Hall Series in Artificial Intelligence. Prentice Hall, Englewood Cliffs, New Jersey, first edition, 1995.

[28] L. C. van der Gaag and S. Renooij. Analysing sensitivity data from probabilistic networks. In Breese and Koller, editors, *Proceedings of the Seventeenth Conference on Uncertainty in Artificial Intelligence*, pages 530–537, Seattle, 2001.