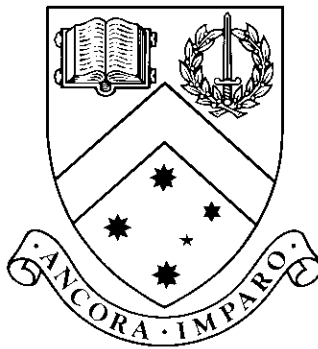


Knowledge Engineering a Bayesian Network for an Ecological Risk Assessment (KEBN-ERA)

by

Owen Grant Woodberry



Thesis

Submitted by Owen Grant Woodberry

in partial fulfillment of the Requirements for the Degree of
Bachelor of Computer Science with Honours (1608)

in the School of Computer Science and Software Engineering at

Monash University

Monash University

November, 2003

© Copyright

by

Owen Grant Woodberry

2003

Contents

List of Tables	vi
List of Figures	vii
Abstract	ix
Acknowledgments	xi
1 Introduction	1
2 Background	3
2.1 Bayesian Networks	3
2.1.1 Probability Theory	3
2.1.2 Graphical Models	5
2.1.3 Defining Bayesian Networks	7
2.1.4 Reasoning with Bayesian Networks	7
2.1.5 Bayesian Network Applications	8
2.2 BN Knowledge Engineering	8
2.2.1 Eliciting Variables and their States	9
2.2.2 Eliciting Graphical Structure	9
2.2.3 Eliciting Conditional Probability Tables	10
2.2.4 Development Model	10
2.2.5 Automated methods	11
2.3 Evaluation Methods	13
2.3.1 Domain Expert Evaluation	13
2.3.2 Automated Evaluation	15
3 Preliminary Prototype Bayesian Network for the Ecological Risk Assessment 16	
3.1 Conceptual Model	16

3.2	Spatial and Temporal Scales	16
3.3	Approach	17
3.4	Reverse Engineering of the Prototype Model	17
3.4.1	Water Quality Portion	18
3.4.2	Flow Portion	18
3.4.3	Habitat Portion	19
3.4.4	Biological Interaction Portion	19
3.4.5	Species Diversity Portion	19
3.4.6	Discretisation of the variables	19
4	Phase 1	20
4.1	Methods	20
4.1.1	Stakeholder Workshop	20
4.1.2	The Domain Expert Developer	21
4.1.3	Spatial and Temporal Components	21
4.2	Evaluation	22
4.2.1	Stakeholder Workshop	22
4.2.2	Domain Expert Evaluation	25
4.3	Outcomes	25
4.3.1	Changes to Network Ontology	25
4.3.2	Changes to Network Structure	25
5	Phase 2	27
5.1	Methods	27
5.1.1	Sensitivity To Findings	27
5.1.2	Sensitivity To Parameters	28
5.1.3	MATILDA Evaluation	31
5.1.4	End User Evaluation	31
5.2	Evaluation	31
5.2.1	Sensitivity to Findings	31
5.2.2	Sensitivity to Parameters	31
5.2.3	MATILDA Evaluation	32
5.2.4	End User Evaluation	33
5.3	Outcomes	33
5.3.1	Sensitivity Analysis Support Tools	33
5.3.2	MATILDA Evaluation	34

5.3.3	End User Evaluation	34
6	Phase 3	35
6.1	Methods	35
6.1.1	Automated learning of Model Components	35
6.1.2	Predictive Accuracy	35
6.2	Evaluation	36
6.2.1	Data Quality Analysis	36
6.2.2	Learning Parameters from the Data	36
6.2.3	Learning Structure from the Data (CaMML)	38
6.2.4	Predictive Accuracy	38
6.3	Outcomes	39
6.3.1	Bhattacharyya Distance Support tool	39
6.3.2	Changes to Quantitative Component	40
6.3.3	Predictive Accuracy	40
7	Conclusions and Further Work	42
7.1	Review	42
7.2	Further Work	43
	References	45
	Appendix A	49

List of Tables

- 4.1 Site Information 24
- 4.2 Site Children Identified 24
- 5.1 Sensitivity to Findings Analysis performed on **Future Abundance** 32
- 6.1 Bhattacharyya distance for node **Natives Biological Potential Descriptor**
with experience value 5 38
- 6.2 Prediction Confusion Matrix for **Future Abundance** 39
- 6.3 Experience Values Assigned to Missing Variables 40
- 1 Methodology used to discretise nodes and resulting states [39] 50
- 2 Changes to Network Ontology 54

List of Figures

- 2.1 D-Separation [42, Chapter 15] 6
- 2.2 Types of Inference [42, Chapter 15] 7
- 2.3 Spiral Development Model [29] 11

- 3.1 Conceptual Diagram used to develop preliminary BN prototype 17
- 3.2 The Prototype Bayesian Network 18

- 4.1 Proposed improvement to spatial representation on the habitat portion (new nodes circled) 21
- 4.2 Proposed temporal representation on the biological interaction portion (new node circled) 22
- 4.3 Proposed dynamic temporal representation on the biological interaction portion (arrows indicate dynamic links added) 23

- 5.1 Sensitivity to Findings algorithm 28
- 5.2 Sensitivity to Parameters algorithm 30
- 5.3 Algorithm to find Sensitivity Functions 30
- 5.4 Example of an Insensitive Parameter Function 33
- 5.5 Example of a Sensitive Parameter Function 34

- 6.1 Variables that have Case File Entries (Circled) 37
- 6.2 Structure created by CaMML when run on data case file 39
- 6.3 Relative Abundances of Native Fish at a site (≥ 1970 fisheries data) vs. BN Predicted Abundances of ‘High’ (≥ 54) Native Fish at the same site [40] 41
- 6.4 Total Number Species of Native Fish at a site (≥ 1970 fisheries data) vs. BN Predicted Diversity of ‘High’ Native Fish at the same site [40] 41

- 1 The Water Quality Portion of the Prototype Model 49
- 2 The Flow Portion of the Prototype Model 51
- 3 The Habitat Portion of the Prototype Model 51

4	The Biological Interaction Portion of the Prototype Model	51
5	The Species Diversity Portion of the Prototype Model	52
6	Flow Portion Resulting from Phase 1	53
7	Biological Interaction Portion Resulting from Phase 1	53
8	Species Diversity Portion Resulting from Phase 1	55
9	Complete Bayesian Network Resulting from Stakeholder workshop and Inclusion of Spatial and Temporal Components	55

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

Owen Grant Woodberry, BCompSc(Hons)
Monash University, 2003

Supervisor: Ann Nicholson, Kevin Korb and Carmel Pollino

Abstract

This thesis develops upon existing research in the field of knowledge engineering of Bayesian Networks (BN's), specific for a complex application domain. The application studied in this project is an Ecological Risk Assessment (ERA), which is being conducted in the Goulburn Broken Catchment, Victoria, Australia. The objective of this ERA is to model the effects of various management interventions on native fish populations. The system being modelled is complex, with multiple variables and interactions. Relationships between variables within the system are poorly understood and data is limited. Although the knowledge engineering of BNs can be an exceptionally difficult task when developing complex models, their representative power justifies their use.

To assist in the building of complex BNs, formal knowledge engineering techniques and tools are required. This thesis used the ERA model to identify existing and potential techniques and tools that are useful for development of the model ontological, qualitative and quantitative components. Tasks of elicitation, evaluation and implementation using both domain expert and automated methods were investigated and techniques/tools were applied or developed where none existed. The results of these investigations were an increased representational power and validity of the model. In addition to the development of tools and methodologies for evaluating and improving the model, constructive evaluation suggestions for future model development were identified.

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

Declaration

I declare that this thesis is my own work and has not been submitted in any form for another degree or diploma at any university or other institute of tertiary education. Information derived from the published and unpublished work of others has been acknowledged in the text and a list of references is given.

Owen Grant Woodberry
November 5, 2003

Acknowledgments

I would like to thank my supervisors: Ann Nicholson, Kevin Korb, Carmel Pollino, for their continued support throughout this project. I would also like to thank Charles Twardy for his feedback in meetings, and Sophie Martin for being my end user evaluation victim.

Owen Grant Woodberry

Monash University
November 2003

Chapter 1

Introduction

Knowledge engineering is the process of building expert systems, such as Bayesian Networks, for application domains. It involves investigating an application domain, identifying important concepts and expressing these concepts as objects and relationships in the formal representation of the expert system. The knowledge engineer is, usually, either trained in the formal representation of the expert system or is an expert in the application domain, but often not in both. For this reason the knowledge engineering task can be seen to address two issues. The first, how can someone, especially someone who isn't an expert in the application domain, properly identify the important concepts? And second, how can someone, especially someone who isn't trained in the formal representation of the expert system, express the concepts that they have identified as important? To address these issues the field of knowledge engineering emerged, its objective, to formalise the process of building expert systems, ensuring they are created correctly and used to their maximum potential.

Bayesian Networks (BNs) are graphical expert systems for reasoning with probabilities. BNs are used to identify the posterior probability of an event given observations of current system state. BN's are composed of ontological, qualitative and quantitative components [38, Chapter 3]. The ontological component is represented by a set of variables, also referred to as nodes, which can take on various values, also referred to as states. The qualitative component is represented by a graphical structure composed of nodes with directed links representing causal influences between parent and child variables. The quantitative component is represented by conditional probability tables (CPT's) which quantify the effects of causal variables. Bayes' theorem is central to the inference mechanisms used to update the posterior probabilities of the *variables* using the probabilistic information of the *CPT's* and dependency information of the *casual structure*.

When developing BNs, knowledge engineering techniques need to be applied to the development of each of these components, separately and collectively. Although there is no strict necessity to use *formal* knowledge engineering techniques, experience has shown that these tasks can be extremely difficult, especially when modelling complex systems. As the theory of BNs is reasonably well understood the emphasis of current research involving BNs is on overcoming their development difficulties, via the development of better support tools, and investigating their applicability to certain domain areas. The modelling shell Netica (Norsys, Inc.) was used to construct the BNs identified in this thesis unless otherwise stated.

The BN, which is the subject of this thesis, is in the ecology domain and is part of a broader Ecological Risk Assessment (ERA) specific to the Goulburn Broken Catchment. 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 (Monash University), under Dr Carmel Pollino. The model component of this project incorporates physical and fishery knowledge to assess the native fish abundances and diversities resulting from management interventions in the Goulburn river system. The model is to be used in an adaptive management scheme and will continue to undergo development, using the methods identified in this thesis.

This thesis investigates and applies knowledge engineering techniques to three development phases of this model. The thesis begins with an overview of the relevant research areas in chapter 2, namely BNs, knowledge engineering and evaluation methods. Background on the ERA and the prior model are given in Chapter 3. The following three chapters describe the three different development phases undertaken. These phases are loosely chronological, with some overlap and parallel development. The first phase (Chapter 4) is concerned with domain expert elicitation and evaluation, both during a stakeholder workshop and through collaboration with the domain expert developer. The second phase (Chapter 5) is concerned with the development and application of computer support tools, and includes an external expert evaluation. The third phase (Chapter 6) is concerned with incorporating the available experimental data, using automated methods to extend and evaluate the model. Each phase is described in terms of the methods used, the evaluation undertaken, and its outcomes. Conclusions and future work are presented in chapter 7.

Chapter 2

Background

2.1 Bayesian Networks

Bayesian networks (BNs) are graphical tools for reasoning with probabilities. BNs have ontological, qualitative and quantitative components. The ontological component is represented by a set of variables which can take on various values. The qualitative component is represented by a graphical structure composed of nodes with relationships between them, representing random variables and causal influences. The quantitative component is represented by conditional probability tables (CPT) for each node, which describe the relative likelihood of each value of the child node, conditional on every possible combination of values of its parents [38, Chapter 3], [10]. 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.

Given that BNs are graphical tools for reasoning with probabilities, it is necessary to introduce both the probability and graph theory which are the underlying theory on which BNs are based. Because the superficial aspects of BNs are relatively easy to read, further in-depth understanding of the technology is sometimes neglected. The consequences of these misunderstands can result in wasted valuable domain expert resources. In order to guide the reader through the important properties of BNs, sections have been provided on their underlying theory to lay the foundations for a comprehensive understanding of the BN technology.

2.1.1 Probability Theory

When dealing with small, theoretical problems it is often enough to consider predictions as an application of a set of IF-THEN statements, this is called classical logical inference. However, in many real world applications this type of inference is not useful due to uncertainty [20]. 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 [42, Chapter 14]. For these reasons, when considering real world problems it is often best to limit reasoning 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, $pr(A)$, is used to represent the prior probability distribution of a random variable, A [38, 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:

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

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

$$pr(C = t) = 0.5$$

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

Joint Probability

The notation, $pr(A, B)$, is used to represent all combinations of values of a set of random variables called the joint probability distribution (JPD) [38, 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, $pr(A|B)$, is used to represent the conditional, or posterior, probability distribution of the random variable A given some evidence B [38, Chapter 2]. Using the notation defined above with the joint probability distribution and prior probability we can define the conditional probability as,

$$pr(A|B) = \frac{pr(A, B)}{pr(B)} \quad (2.1)$$

provided that $pr(B) > 0$.

Independence

Returning to the example of the two coin tosses we can see that the probability of one coin toss given another,

$$pr(C2|C1) = pr(C2) \quad (2.2)$$

because the variables C1 and C2 are 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:

$$pr(C1, C2) = pr(C1) \times pr(C2) \quad (2.3)$$

Bayes' rule

Using the notation for conditional probability (2.1) combined with the symmetry rule:

$$pr(A, B) = pr(B, A) \quad (2.4)$$

we can define Bayes' rule [38, Chapter 2]:

$$pr(A|B) = \frac{pr(B|A)pr(A)}{pr(B)} \quad (2.5)$$

2.1.2 Graphical Models

The JPD for a problem captures the probability information of every possible combination of a set of variables, and their states. Once a JPD 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 of the variables. This includes their value given additional evidence, that is, their posterior probability. Although, as was stated earlier, the space, and consequently, time complexity required to represent and manipulate the JPD is exponential in the number of variables to be considered [14]. For example the JPD required to represent a system with 20 binary values would have 2^{20} (1,048,576) values. This causes problems in the elicitation, storage and manipulations of these values, thus making the use of JPDs 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 [14]. 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 [38, 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 [38, 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 JPD 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 [42, 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 2.1),

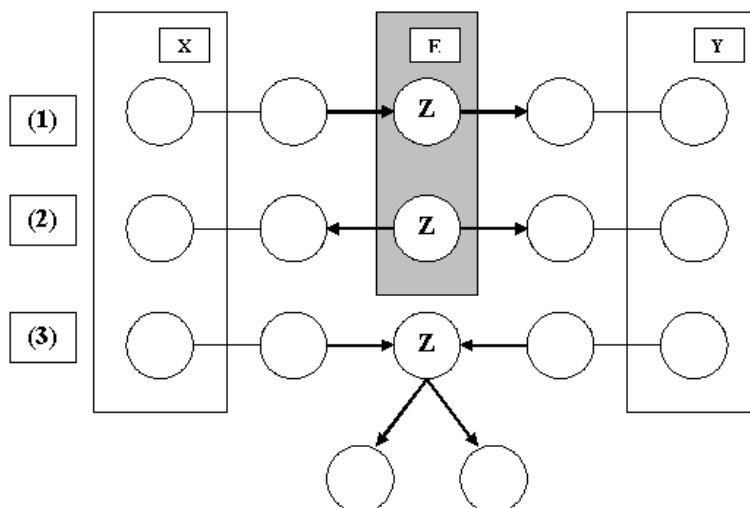


Figure 2.1: D-Separation [42, Chapter 15]

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 its 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 its children and do have influence on each other given knowledge of the state of node C or any of its children.

2.1.3 Defining Bayesian Networks

It is now possible to give a formal definition for Bayesian networks. A Bayesian Network (BN) is a directed acyclic graph (DAG) that represents a joint probability distribution (JPD). 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, or relative uncertainty, of each possible state given the possible states of its parents [38, Chapter 3],[18].

2.1.4 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 for, which we call the query nodes, those nodes that we already have knowledge for, which we call the evidence nodes, and the intermediate nodes between them[42, Chapter 15]. We enter our knowledge of the system state by selecting values for the evidence nodes, and an inference algorithm is used to update the posterior probability distributions of the query nodes.

There are four types of inference algorithms: [42, Chapter 15] (see Figure 2.2):

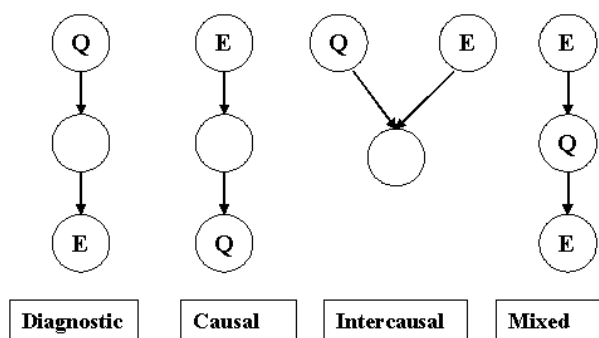


Figure 2.2: Types of Inference [42, Chapter 15]

- **Diagnostic inference:** involves updating beliefs from effects to causes.
- **Causal inference:** also called prediction, involves updating the beliefs from causes to effects.
- **Intercausal inference:** also called explaining away, involves updating the beliefs between causes of a common effect.
- **Mixed inference:** involves updating beliefs using a mixture of the inferences listed above.

All these different types of inference can be made using the CPT's for each node. For example, $pr(A)$, $pr(B|A)$, $pr(D|B, C)$ and Bayes' rule allows the computation of $pr(A|B)$ and $pr(B, C|D)$. Hence, a BN together with the inference algorithm allows any of these types of reasoning with the system represented by the BN. Even though BNs are more efficient than the JPDs, inference in BNs is NP-hard, in both exact and approximate inference. Although in practice there exist fast and efficient algorithms to do this inference. The networks generated in this thesis use the Netica application which uses a version of the Jensen join-tree algorithm [23] to perform exact inference.

2.1.5 Bayesian Network Applications

BNs have been applied to a great variety of problems, including medical diagnosis [34, 19, 36, 31], Microsoft's Office Assistant [22], Bayesian Poker [28, 9], Seabreeze prediction [24], intelligent tutoring [35] amongst others. Applications that have particular relevance to this project are ecosystem modelling [6, 7], natural resource management [8] and species-environment relations [32, 33].

2.2 BN Knowledge Engineering

In this section the literature regarding knowledge engineering BNs is summarised, first reviewing the difficulties of elicitation, the solutions provided by the knowledge engineering field and then the spiral development model and its applicability to the development of BN software systems. The process of knowledge engineering a BN can be broken up into three tasks [27, 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 [16]. These tasks are listed here:

1. Identifying the set of variables and their states, which composes the ontological component of the system.
2. Identifying the graphical structure, which composes the qualitative structural assumptions of the system.
3. Identifying the conditional probability tables of each variable, which composes the quantitative effects of the system.

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 in the third task. The third task is often considered to be the most difficult.

In this section I address each of the tasks in eliciting knowledge from domain experts from the knowledge engineering perspective, identifying selected automated methods and the potential issues created by integrating these methods.

2.2.1 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 including only the important variables. These can be identified by determining whether the variable falls into one of these four categories [27, Chapter 10], [8]:

- **Query variables:** also called objective or target variables, which are the *output* of the network, i.e. the variables the end-user wants to know about.
- **Evidence variables:** also called observation or controlling variables, which are the *input* of the network, i.e. the variables that would be potentially useful in inferring the states of the query variables.
- **Context variables:** also called intermediate variables, which link the query and evidence variables.
- **Controllable variables:** also called intervention variables, which 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 to identify [8]:

- The state it is currently in.
- The state/s toward which you think it may move under possible management interventions.
- Any intermediate states.

Only those states that are possibly of interest to the end-user should be included. When selecting states it is necessary to 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.

2.2.2 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. When determining

the structure of the network the key is to focus on the relationships between key variables. As was stated in the preceding section there is four types of relationships between variables independent, dependent, conditionally independent and marginally independent.

The relationships can be determined by asking direct questions, such as, what can cause variable **A** to take on the value **a**? Any answers would suggest a causal relationship between the variable identified and the variable **A**. A support tool for this type of elicitation, called MATILDA, is discussed in the following evaluation section (Section 2.3.1).

2.2.3 Eliciting Conditional Probability Tables

Elicitation of CPT's is often considered the most difficult task when creating BNs. The CPT's for each node specify the values of the node conditional on 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 [27, 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**. This approach is utilised in the support tool, VE [21]. VE aids in elicitation by mapping verbal terms to probabilities. This source of elicitation may be the only type available in some domains. However, it is often difficult to find experts who have the time and interest to go through 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 available within a domain, then the process of training the CPT's can be entirely automated [27, Chapter 10]. However, 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. Methods for learning BNs from the data are discussed in greater depth in section 2.2.5.

The literature: many of the CPT's required may already be specified in 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 when doing this, it is necessary to specify the confidence or weighting of the probabilities acquired from each source. There is also problems in combining probabilities from different sources due to accumulation of biases, which won't necessarily cancel each other out.

2.2.4 Development Model

As research into BNs increases, the understanding of and therefore the applications of BN networks, are becoming larger in size and greater in complexity [29]. 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 is 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 phase and eventually it is retired. This view of a software lifecycle is standard to the waterfall development model popular in software engineering. Another way of viewing this lifecycle is to see the software as a continually growing organism [27, 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 process standard to the spiral development model, the software goes through a repeating cycle of design, development, operation and evaluation (see Figure 2.3). The early iterations of the lifecycles are generally called prototypes and at some point these prototypes eventually become system versions.

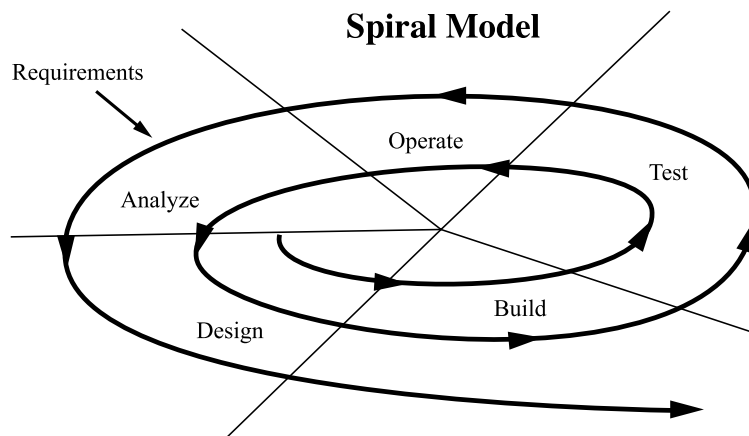


Figure 2.3: Spiral Development Model [29]

The main reason for preferring the spiral model to the waterfall lifecycle model is risk management. One of the prime difficulties in software engineering is the requirements phase [29]. 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, thus providing valuable feedback. This prototyping model development is particularly useful in the BN development as visual aids and with graphical interfaces are utilised [27, Chapter 10]. The model developed for the ERA follows this development model.

2.2.5 Automated methods

There are many algorithms for automated learning of BN parameters and structures using case data. The Netica environment provides methods to learn parameters which were used in this thesis. The CaMML learning program, developed here at Monash University, was also applied to learn the causal model structure describing the data. In this section I review the parameter

learning methods available in the Netica environment and the CaMML program used to learn causal structure.

Lauritzen & Spiegelhalter Method (L&S)

The Lauritzen & Spiegelhalter Method (L&S) [30] is a traditional one-pass learning algorithm. It only modifies parameters that have a complete set of parent values for the probability distribution. The Netica implementation of this algorithm requires that each case is given a weighting (degree). It works by first assigning an initial experience value of 1 to all probability distributions in the CPT. When an instance of case occurs in the data file the corresponding probability distribution experience value e is incremented by the degree d assigned to each case file entry.

$$e' = e + d \quad (2.6)$$

where e' is the new experience value, e is the old experience value and d is the degree assigned to each case file entry. The new parameter value is then assigned by,

$$pr'_j = (pr_j \times e + d) / e' \quad (2.7)$$

where pr' is the new parameter value and pr is the old. The remaining parameters are then renormalised by the function,

$$pr'_i = \frac{(pr_i \times e)}{e'}, i \neq j \quad (2.8)$$

where pr'_i are the new probabilities and pr_i the old. This method does not perform well in situations where there are hidden and/or missing variables values as only nodes with a full complement of parents are updated. The next two methods deal with this problem.

Expectation Maximisation (EM)

The Expectation Maximisation (EM) [15] method optimises the BN CPT's by maximising the probability, or alternatively minimising the negative log likelihood, of the data given the BN. The Netica implementation of this method requires that any nodes, which already have CPT values, be allocated experience values, after which they are treated as part of the data. This experience represents the equivalent data case weighing of the probability distribution, the same as the S&L experience values. It ensures that knowledge already in the network is combined with the knowledge in the data with an appropriate weighting. Any nodes that either have findings in the case data file or are an ancestor to one of these nodes will be modified. This method converges on local maxima, which may not be global. The algorithm is described by [26];

1. Set θ arbitrarily

2. Compute the probability distribution over missing values:

$$P(E^*|E, \theta) = \frac{P(E|E^*, \theta)P(E^*|\theta)}{\int_{E^*} P(E|\theta)dE^*} \quad (2.9)$$

3. Compute the new maximum likelihood estimate θ' given $P(E|E^*, \theta)$, repeat until θ and θ' converge.

Gradient Descent (GD)

This method is similar to the EM method described in the preceding section. It differs in using a conjugate gradient descent to maximize the probability of the data given the BN. This algorithm converges faster than the EM method, but tends to be more susceptible to local maxima.

Learning Structure from the Data (CaMML)

CaMML [45] is a tool for the automated learning of the underlying causal model of data. CaMML uses an inference technique called Minimum Message Length, MML, to compare different potential models. MML applies an information penalty, in the Shannon sense, based on a Bayesian prior reward for simpler models which describe the data equally well. CaMML will attempt to find a model that describes the data well without over-fitting. As an exercise, it is interesting, and most likely informative, to compare the CaMML generated model based on the experimental data with the human developed model.

2.3 Evaluation Methods

As identified in the preceding section, the evaluation of the BN is an essential component in their development. Evaluation methods are useful to grade the BN to 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 methods for evaluating BNs are summarised. Methods used are evaluation using feedback from domain experts and evaluation using statistical methods that can be automated.

2.3.1 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 because developers will tend to overlook errors of not identified at first. Once again, it is often difficult to find experts who have the time and interest to go through a BN evaluation process, although the evaluation phase is not as time consuming as the elicitation process.

Four types of BN evaluation methodologies that can be used to gain feedback from domain experts are an elicitation review, MATILDA, sensitivity analysis and case based evaluation. All of these evaluation methods were applied at various development phases during this thesis.

Elicitation Review

Elicitation review is a formal structured review of the elicitation process, which provides a global overview of the decisions made during the development of the model [27, Chapter 10]. The review is primarily focused on the ontological and qualitative components 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 and checking to determine 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 is MATILDA.

MATILDA

Matilda is a support tool [5] that 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.

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 testing how sensitive the network is to 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 development in later iterations in the development cycle [29]. 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 iterations of the development cycle. Contrary to previous belief [41] 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 [44]. It is these parameters where attention is needed the most.

Case Based Evaluation

It is also often useful to conduct a walkthrough, running the network though a set of various cases in an attempt to exhaustively test 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 identifying possible errors in the network. A potential difficulty with this approach is that the domain expert may not have the experience to make a judgment 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 domain expert's judgment.

2.3.2 Automated Evaluation

If there is a large body of data available for the system domain, then this can be used to evaluate the network. If data has already been incorporated into the node then we do not want to use same data for evaluation. To address this problem it is standard to divide the data into 90/10 or 80/20 split, and use the 10 or 20 percent to evaluate the BN.

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, which is 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 [27, Chapter 11].

Bhattacharyya Distance

In order to identify changes in the CPTs the Bhattacharyya distance [4] could be employed. Tools to compare distances between probability distributions can be used to evaluate the resulting changes of applying automated learning methods (see section 2.2.5). The Bhattacharyya distance between two probability distributions is given by,

$$D_B(P, Q) = -\log \sum_i \sqrt{P_i Q_i} \quad (2.10)$$

This distance measure is more appropriate, for comparing probability distributions generated by automated learning, than the more commonly used Kullback -Leibler distance as it is symmetric.

Chapter 3

Preliminary Prototype Bayesian Network for the Ecological Risk Assessment

The BN application for this thesis is an ecological risk assessment (ERA). The objective of the 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 being conducted by the Water Studies Centre at Monash University, under by Dr Carmel Pollino. The objective of this part of the project is to develop a model to determine the effects of alternative management actions on the native fish population abundance and diversity in the Goulburn Broken Catchment. It was considered by the group that BN technology would best meet the needs for modelling.

In this section the conceptual model that the BN is based on is reviewed, and the study spatial scales and temporal scales to be considered are identified before a reverse-engineering analysis of the prototype model is given.

3.1 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 3.1), the double arrow links are included to demonstrate possible interaction between factors:

3.2 Spatial and Temporal Scales

The study is focussed on the Goulburn Broken Catchment. The geographical areas within the catchment are divided 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. Temporal scales to be considered are 1, 5, 10

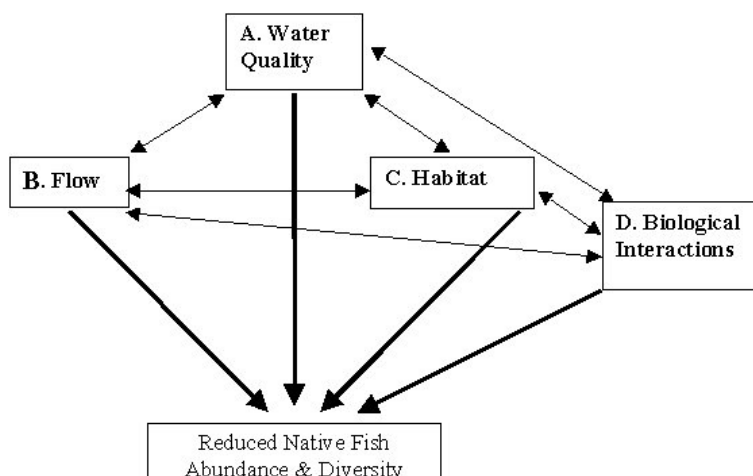


Figure 3.1: Conceptual Diagram used to develop preliminary BN prototype

years, and 30 to 50 years; these were selected as they accurately reflect the life history of the different fish species in the area.

3.3 Approach

In the problem formulation phase, of the ERA, native fish abundance and diversity was identified as being at risk due to irrigation activities in the Goulburn Broken Catchment [12]. This phase of the ERA was initiated to identify and quantify these hazards and develop a model to measure the probability of increasing or decreasing the fish population abundance and diversity in response to various management interventions. Due to the incompleteness of existing field data, most of the elicitation and evaluation process for building the qualitative and quantitative model components relied heavily on feedback gained from running workshops with domain experts and stakeholders. It had not yet been determined whether the field data could be used for automated evaluation.

3.4 Reverse Engineering of the Prototype Model

The four portions identified in section 3.1 were built into the prototype causing the query variables named **Native Population Abundance** and **Native Population Diversity**. Due to the multiple data nodes in each of the model portions, the results were integrated into a descriptor node designed for the model. These descriptor nodes for each portion were named **Water Quality**, **Overall Change in Flow Regime**, **Structural Habitat Quality** and **Competition**. These descriptor nodes were based purely on expert elicitation. The only nodes to be trained via data were those that were identified as the variables and for which data was available. The L&S (see section 2.2.5) method of learning parameters was used to do this.

Apart from through the descriptor nodes, the actual physical and fishery variables represented in the prototype had little or no interaction between them. By representing the system in such a

manner, the overall known effects in the system could be quantified with what little was known. The model could be later used to identify where further investigation of relationships between variables would be most beneficial. In addition to the four portions identified (see section 3.1) another, species diversity, portion was created which incorporated separate species abundances information. The overall model structure can be seen in figure 3.2, each of the BN portions are discussed further in the proceeding sections.

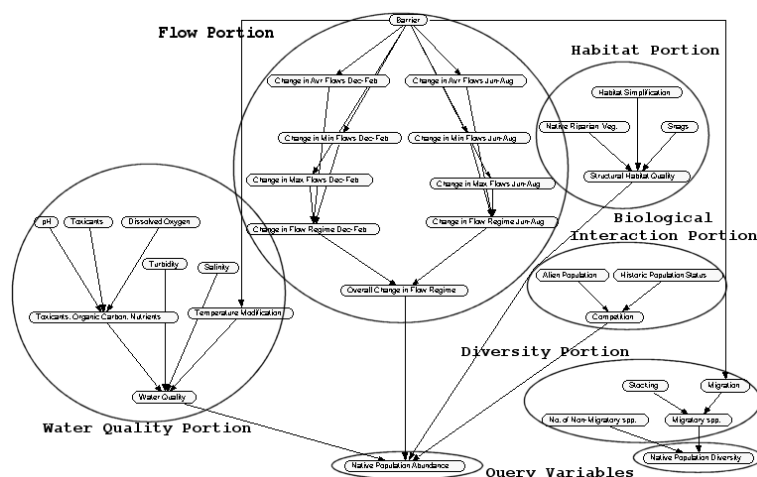


Figure 3.2: The Prototype Bayesian Network

3.4.1 Water Quality Portion

The key variables identified as influencing **Water Quality** in the Catchment were: **Toxicants**, **pH**, **Dissolved Oxygen**, **Salinity**, **Turbidity** and **Temperature Modification**. The structure of this portion can be found in Appendix A (figure 1). The **Barrier** node, which forms part of the flow portion of the model, is included to show the link between these portions via the **Temperature Modification** node. The nodes **Toxicants**, **pH** and **Dissolved Oxygen** are divorced from the descriptor node by **Toxicants, Organic Carbon, Nutrients** node.

3.4.2 Flow Portion

The key variables identified as influencing **Overall Change in Flow Regime** in the Catchment were: **Barrier**, **Change in Avr Flows Dec-Feb**, **Change in Min Flows Dec-Feb**, **Change in Max Flows Dec-Feb**, **Change in Avr Flows Jun-Aug**, **Change in Min Flows Jun-Aug** and **Change in Max Flows Jun-Aug**. The structure of this portion can be found in Appendix A (figure 2). The nodes **Change in Flow Regime Dec-Feb** and **Change in Flow Regime Jun-Aug** are specific descriptors for the seasonal flow regimes.

3.4.3 Habitat Portion

The key variables identified as influencing Structural Habitat Quality in the Catchment were: **Native Riparian Veg.**, **Habitat Simplification** and **Snags**. The structure of this portion can be found in Appendix A (figure 3).

3.4.4 Biological Interaction Portion

The key variables identified as influencing **Competition** in the Catchment were: **Historic Population Status** and **Alien Population**. The structure of this portion can be found in the Appendix A (figure 4).

3.4.5 Species Diversity Portion

The key variables identified as influencing the **Native Population Diversity** in the Catchment were: **Migration**, **Stocking**, **Migratory spp.** and **No. of Non-Migratory spp.** The structure of this portion can be found in the Appendix A (figure 5). The **Barrier** node which forms part of the flow portion of the model is included to show the interaction between these portions of the BN via the **Migration** node. The **Stocking** and **Migration** nodes were identified as causing the **Migratory spp** variable.

3.4.6 Discretisation of the variables

The nodes in the network were discretised according to the protocols of different monitoring programs and agencies [39]. Table 1 in Appendix A summarises the methodology used and the resulting states for each node.

Chapter 4

Phase 1

In this development phase, evaluation, and subsequent implementation was centred around a two-day stakeholder workshop. Problems with the model identified before and during the workshop were addressed and possible solutions decided upon. Following the workshop, an in-depth investigation of issues raised was conducted, identifying where possible improvements could be implemented within the time constraints of the project. Improvements identified included better representation of spatial and temporal components of the model.

In this chapter, I summarise the methods used to, evaluate the BN and the subsequent outcomes. I include the activities conducted to lay the foundational material for subsequent development phases, focusing on improvements to the spatial component of the model and introduction of a temporal component, which are among the main outcomes of this thesis.

4.1 Methods

4.1.1 Stakeholder Workshop

In order to evaluate and improve the existing model a two-day workshop was held involving experts in fish ecology, management and modelling. The workshop was conducted at the Water Studies Centre (Monash University) in May 2003 and was run by the domain expert developer, Dr Carmel Pollino. The objectives of the workshop were:

- To briefly review the progress of the project to date.
- To review the structure of the Bayesian network produced for native fish abundance and diversity in the Goulburn Broken catchment.
- To incorporate the knowledge of the expert panel into the network.
- To identify key knowledge gaps in the model and specify how these gaps might be filled.

On the first day the set of variables and structure of the model were reviewed and discussed. The focus of the second day was on the conditional probability tables. Possible improvements to the spatial and temporal representation in the model and incorporation of experimental data

were also addressed. These improvement methods were central to this thesis' involvement in this development phase and are the subject of a later section (see section 4.1.3).

4.1.2 The Domain Expert Developer

As with all iterative phases of development, improvements to the model were identified, implemented and evaluated in collaboration with the domain expert developer. In the evaluation phase, the domain expert was consulted before and after the stakeholder workshop, and throughout the implementation of improvements.

4.1.3 Spatial and Temporal Components

Originally, separate networks were prepared for each spatial scale of interest, (see chapter 3). Each of the networks represented the physical and fishery data particular for each site, making the site dependent nodes the root nodes of the networks. In order exploit the inter-site redundancies possible methods of incorporating networks into a single global network where examined. One possible method identified was to include an additional **Site** and/or **Type** variable, which could be a parent node to all **Site/Type** dependent nodes (see figure 4.1). In doing this all the data cases could be incorporated into a single case file with an additional **Site/Type** field.

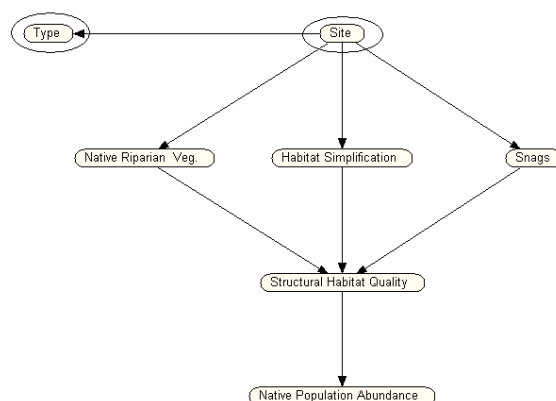


Figure 4.1: Proposed improvement to spatial representation on the habitat portion (new nodes circled)

A Bayesian Network normally represents a single time step. This was the case with the original networks. In order to consider changes in nodes over multiple time steps the network needed to be extended. One possible method identified was the addition of a new time node influencing the query nodes directly, in much the same way as the spatial representation alteration proposed above (see figure 4.2). Another possible method was to repeat the Bayesian Network over multiple time steps, called a Dynamic Bayesian Network [42, Chapter 17]. Links between variables in different time steps, called dynamic relationships, need to be identified to use this type of extension (see figure 4.3). In order to extend the model by either of these methods it was necessary to define the time scales of changes and predictions. By matching temporal differences, the case data could be extended to include cases for the different time scales.

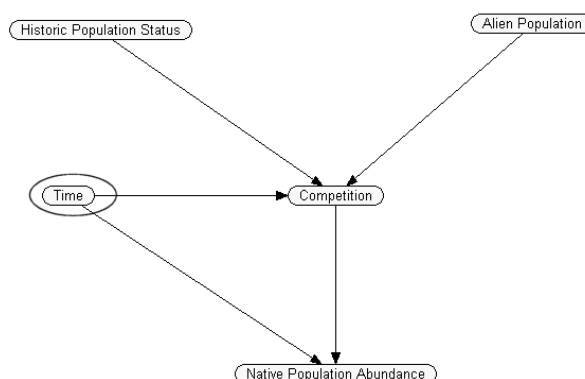


Figure 4.2: Proposed temporal representation on the biological interaction portion (new node circled)

4.2 Evaluation

4.2.1 Stakeholder Workshop

The first day of the stakeholder workshop, which was focused on the model structure, generated much discussion on possible improvements to the existing model. At the end on day 1, the recommended model structure was far too complex to be implemented in the project time frame. On the second day a simpler model was agreed upon. When the discussion focused on the conditional probability tables, progress again proved to be difficult as a complete review of the parameters was not possible in the time frame allocated. It was decided that further expert elicitation of the parameters was to be done on an individual basis by the domain expert developer, using the workshop discussions as a basis for model improvements. The difficulties encountered during the stakeholder workshop are symptomatic of the complex and tedious nature of eliciting the information required in developing Bayesian Networks. They further underline the requirement for formal knowledge engineering techniques in the development of Bayesian networks.

The discussion during the workshop also focused on improvements to the spatial and temporal components of the model identified during this thesis' investigations on the prototype model. Stakeholders strongly supported the proposed changes, which are reviewed here in more depth.

Spatial Components

During the stakeholder workshop it was agreed that the methods used for spatial representation of the sites, being separate networks, could be improved upon. The potential methods for incorporating the separate spatial networks into a single global network (see Section 4.1.3) were prepared and presented as part of this thesis. It was agreed that the inclusion of a **Site** node to represent local scales and a **Type** node to represent global scales would improve the utility of the existing model. It was also acknowledged that the improvements would be useful in increasing

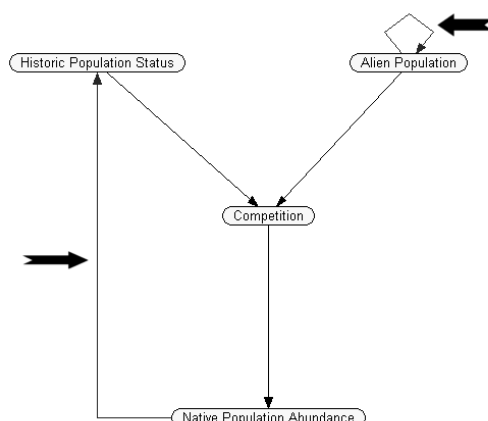


Figure 4.3: Proposed dynamic temporal representation on the biological interaction portion (arrows indicate dynamic links added)

the representational power of future model development phases. This permitted the inclusion of important factors that had, up to that point, been considered outside the capacity of the model.

In order to identify the states of the new **Site** and **Type** variables existing local and global scales used in the separate networks were reused. Table 4.1 shows how the sites are organised into region and type groups. The first two columns represent the new **Type** state and the following three columns display the site details and the proposed **Site** state names.

The stakeholders were also asked to identify the new causal dependencies that needed to be represented in the new model. The **Site** node was identified as a root node with particular following child nodes, given in Table 4.2.

The **Type** node, which was made a child of the **Site** node, was kept childless making it casually independent of the rest of the network given evidence of the **Site** node. It was recognised that along with its informative power under this situation, it could also be used to represent other possible factors of type dependent relationships in the model identified in later development iterations.

Temporal Components

In preparation for the stakeholder workshop two different methods for dealing with temporal representation were identified and investigated. The first method identified was to include an additional **Time Scale** node, which could be done in a similar fashion to the **Site** and **Type** nodes introduced in the preceding section. The second method identified was to use a dynamic Bayesian Network, although further investigation showed that such a method would be beyond the time constraints of an honours project.

During the stakeholder workshop both methods were presented and it was agreed that the inclusion of a new **Time Scale** node to represent the temporal component would be sufficient. Again, as with the spatial improvements proposed, it was acknowledged that the new **Time**

Table 4.1: Site Information

Region	Type	Stream	Nearest landmark	Site state
Upper	Main	Goulburn River	Elidon	G Eild
			Alexandra	G Alex
			Yea	G Yea
			Trawool	G Trawool
	Tributary	Rubicon River	Rubicon	Rubi
		Taggerty River	Lady Talbot Dve	Tagg
		Acheron River	Taggerty	Ach
		Murrindindi River	Above Colwells	Murrin
		Yea River	Devlins Bridge	Yea
		King Parrot Creek	Flowerdale	King
		Sunday Creek	Tallarook	Sunday
		Hughes Creek	Tarcombe Rd	Hughes
Mid	Main	Lake Nagambie	Nagambie	G LkNag
Lower	Main	Goulburn River	Murchison	G Murch
			Shepparton	G Shep
			McCoys Bridge	G McCoy
			Undera	G Undera
			Echuca	G Echuca
	Tributary	Pranjip Creek	Moorilim	Pranjip
		Brankeet & Creighton Creeks	Brankeet	Crei Bran
		Castle Creek	Arcadia	Castle
		Seven's Creek	Polly McQuinn Weir	Sevens

Table 4.2: Site Children Identified

Component	Site Dependent nodes
Water Quality	Dissolved Oxygen, pH, Anthropogenic Inputs, Turbidity, Salinity
Flow	Barrier, Change in Avr Flows Summer-Autumn, Change in Avr Flows Winter-Spring, Change in Min Flows Summer-Autumn, Change in Max Flows Winter-Spring
Structural Habitat	Native Riparian Veg, Snags, Habitat Simplification
Biological Potential	Stocking Rate, Current Abundance, Loss of Fish, Alien Threat, Macroinvertebrates
Diversity	Migratory spp, Non-Migratory spp
Spatial	Type

Scale node would improve the utility of the existing model. It would be also be useful in increasing the representational power of resulting future model development phases.

In selecting the states of the **Time Scale** node the stakeholders were asked to consider the life history of the various fish species in the area. The stakeholders had previously agreed that time scales of 1, 5, 10 and 30 to 50 years would be necessary to properly model fish abundance and diversity. However In the subsequent discussion it was agreed that time scales of 1 and 5 years would suffice for immediate development, with potential for extensions in future iterations.

The stakeholders were then asked to identify the time dependent causal relationships that needed to be represented in the network. The **Time Scale** node, like the **Site** node, was identified as a root node, with the child nodes **Natives Biological Potential Descriptor** and **Future Abundance**.

4.2.2 Domain Expert Evaluation

In post-workshop evaluation carried out with the domain expert further changes to the species diversity portion were proposed. Due to the inclusion of temporal scales it was decided that a separate **Future Diversity** reflecting the differences in time scale should be included. This new node was identified as a third child of the **Time Scale** node.

4.3 Outcomes

4.3.1 Changes to Network Ontology

Table 2, in Appendix A, summarises the changes to the model ontology resulting from the stakeholder workshop and subsequent evaluation conducted, with consultation, by the domain expert developer.

4.3.2 Changes to Network Structure

The stakeholder workshop and subsequent evaluation with the developer resulted in changes to the flow, biological interaction and species diversity portions of the model. Following the model evaluation some variables were removed and added to the network. In the flow portion of the model the nodes **Change in max flows Dec-Feb** and **Change in min flows Jun-Aug** were removed, as mentioned already. The node **Floodplain Inundation** was added as a child of **Change in Max Flows Winter-Spring**, (see Figure 6 in Appendix A).

The most dramatic changes occurred in the biological interaction portion of the model. The nodes **Macroinvertebrates**, **Zoobenthos** and **Loss of Fish** were added as children of the **Site** node. The node **Potential Recruitment** was added as a child of **Water Quality Habitat Descriptor**, **Macroinvertebrates**, **Zoobenthos** and **Change in Min Flows Summer-Autumn** nodes. The node **Community Change** was added as a child of **Loss of Fish** and **Alien Threat** nodes and as an additional parent to **Natives Biological Potential Descriptor**. A new causal dependency between **Stocking Rate** and **Natives Biological Potential Descriptor** was included, (see Figure 7 in Appendix A).

In the species diversity portion the node **Future Diversity** was added as a child of **Current Diversity**, **Future Abundance** and **Time Scale**. The link between the nodes **Connectivity** and **Migratory spp** was removed and a new link between **Connectivity** and **Current Diversity** was added (see Figure 8 in Appendix A).

Spatial and Temporal Components

Following the model evaluation at the stakeholder workshop, and subsequent consultation with the domain expert, the complete structure (see Figure 9 in Appendix A), including **Site**, **Type** and **Time Scale** nodes were generated.

These improvements lay the foundations for incorporating data related development using automated evaluation, parameter learning and causal discovery methods. These methods and the resulting model improvements, were crucial for next model developmental phases (see Chapter 6).

Chapter 5

Phase 2

In this development phase methods to assist in the evaluation and elicitation of the quantitative component with sensitivity analysis programs were investigated. Methods to assist in the evaluation of the qualitative component using the support tool MATILDA were investigated. The end useability of the model was also assessed. The methods identified in this development phase were to be combined with stakeholder evaluation and are to be implemented in future development phases, although preliminary application, testing applicability, was performed and is included here. This development phase ran in parallel with phase 3 resulting in some overlap in development. The investigations of these phases were based on evaluation from reverse engineering the prototype model and issues identified during the stakeholder workshop.

5.1 Methods

5.1.1 Sensitivity To Findings

The properties of d-separation (see Section 2.1.2) can be used to identify if a variable will be affected by evidence from another variables in the network. Sensitivity analysis can be used to identify where evidence of a variable will push the posterior probabilities of our query nodes to higher certainty levels, 0 or 1. This information could be used to identify the variables where further evidence entered would be most informative. The measure, **entropy**, $H(\mathbf{X})$, is commonly used to evaluate the uncertainty, or randomness, of a probability distribution. Its formula is given by:

$$H(\mathbf{X}) = - \sum_{x \in X} \text{pr}(x) \times \log \text{pr}(x) \quad (5.1)$$

Measuring the effect of one variable on another is referred to as the **mutual information**, $I(\mathbf{X}|\mathbf{Y})$:

$$I(\mathbf{X}|\mathbf{Y}) = H(\mathbf{X}) - H(\mathbf{X}|\mathbf{Y}) \quad (5.2)$$

where,

$$H(X|Y) = - \sum pr(x, y) \times \log pr(x|y) \quad (5.3)$$

This method was employed in this project to prioritise future improvements and provide detailed user feedback. The Netica Application Programming Interface contains functions to assist in the development of this type of sensitivity analysis application.

Sensitivity to Findings Support Tool

Two text interface programs were generated to assist in performing sensitivity analysis to findings. Both of the programs used similar algorithms and differed mainly in giving the user control on the direction of the sensitivity analysis. In the first program the user was required to give a *Netica* network name and a query variable from this network on which to perform the sensitivity analysis. The second program required a *Netica* network, on which it did an exhaustive search. The Netica API [2] provides functions to compute the entropy of a node and the mutual information of one node given another. Using these functions the algorithm (see figure 5.1) was followed.

Loop

 Compute entropy of query node and mutual information values

 Display entropy of query node

 Rank and display mutual information values

Prompt user for action

If Set Node Finding selected **Then**

Prompt user for node name and value

 Enter node finding

If Back up selected **Then**

 Remove last node finding

If Save Report selected **Then**

 Save sensitivity analysis results to file

break Loop

End Loop

Figure 5.1: Sensitivity to Findings algorithm

5.1.2 Sensitivity To Parameters

There is high amount of inherent inaccuracy in eliciting the quantitative component of a BN. This inaccuracy could be caused by incompleteness of data used to train the model or partial knowledge of domain experts, both of which are problems identified during the development of this model. It has been shown that a BN can be extremely sensitive to changes in certain parameter estimates. Analysis of these sensitive parameters allows attention to be directed on identifying and improving them.

Sensitivity analysis can be performed using an empirical approach, by altering the parameters of query node and observing the related changes in the posterior probabilities of the target node. However, such a straightforward analysis can be extremely time consuming, especially on larger networks, such as the one developed in this project. Coupe and Van der Gaag [13] address this difficulty by identifying a sensitivity set of a variable given evidence. This set can be found using an adapted d-separation algorithm and can be used to eliminate variables from further analysis. Coupe and Van der Gaag subsequently demonstrate that the probability of a state given evidence can be given a functional representation. Using Bayes Theorem,

$$pr(Q|E) = \frac{pr(Q, E)}{pr(E)} \quad (5.4)$$

they show that both $pr(Q, E)$ and $pr(E)$, where Q is the query value and E is the evidence set, can be related linearly to a parameter under study, x . Hence the functional relationship can be represented by a hyperbolic function,

$$pr(Q|E) = \frac{ax + b}{cx + d} \quad (5.5)$$

or a linear function,

$$pr(Q|E) = ax + b \quad (5.6)$$

if the node, Q , has no observed descendents. It is just a matter of empirically determining these functional relationships, which can be found with two or three parameter values, and finding the derivative at the current parameter value. This method is further extended by also considering the deviation to the hyperbola vertex where the slope greatly increases [17].

This method was employed in this project to aid in the elicitation of parameters. Again the Netica Application Programming Interface contains functions to assist in the development of this type of sensitivity analysis application.

Sensitivity to Parameters Support Tool

A text/graphical interface program was generated to assist in performing sensitivity analysis to parameters. The program required the user to give a *Netica* network name, an interest and test node and set of evidence nodes from this network on which to perform the sensitivity analysis. The user was asked to identify the minimum variation of interest in the interest node and maximum variation expected in the test node. These were to compute the gradient and deviation, to the hyperbola vertex, of interest so the program could be selective on what it displayed. The Netica API [2] provides functions to compile the network and find the belief of an event given evidence. A gnuplot API [1] was also used to generate and display plots to the screen. Using these functions the algorithm (see figure 5.2) was followed.

```

Compute interest gradient grad and deviation devi
For all combinations of evidence states
  If test node is in sensitivity set Then
    Determine if a linear or hyperbolic function
    For each parameter in table
      Compute function coefficients
      If (function gradient  $\geq grad$ ) or (parameter deviation  $\leq devi$ ) Then
        Display plot to screen
    End For Loop
  End For Loop

```

Figure 5.2: Sensitivity to Parameters algorithm

To determine whether a node was in the sensitivity set, a parent auxiliary variable of the test node is imagined that causes uncertainty in this node. The path between this auxiliary node and the interest node can be tested for d-separation (see section 2.1.2) and therefore used to determine influence or sensitivity. When a particular evidence instantiation is set the type of sensitivity function for the parameters in the query node can be identified. This is done by checking to see if the query node has any observed descendants (see section 5.1.2). This information is used to determine which algorithm to follow. As these algorithms only differ in how many parameter-belief pairs they need to solve for coefficients they are considered together here. Once the sensitivity function is determined for a parameter the gradient and the deviation from the hyperbola vertex can be computed. If either of these values are within the interest values identified by the user the plot is displayed to the screen. To compute the sensitivity function coefficients this algorithm (see figure 5.1.2) was followed.

```

Loop 2 times for linear, 3 times for hyperbolic
  Set new normalised probability distribution in test node
  Compile network
  Get node belief in interest node
  Store parameter and belief pair
  Restore old probability distribution
end loop
Solve for coefficients of sensitivity function
End For Loop

```

Figure 5.3: Algorithm to find Sensitivity Functions

The new normalised probability distribution of the test node is set by first selecting a new value for the parameter under investigation. The remaining parameters were normalised to retain relative values by the updating function,

$$pr_i \leftarrow pr_i \times (1 - pr_{new}) / (1 - pr_j), i \neq j \quad (5.7)$$

before the parameter under study was updated,

$$pr_j \leftarrow pr_{new} \quad (5.8)$$

5.1.3 MATILDA Evaluation

MATILDA (see section 2.3.1) is a support tool that was developed to aid in communicating and explaining the graphical component of a BN. To aid the domain expert developer in understanding the concepts of d-separation, and to trial this program, evaluation of the model was conducted using MATILDA.

5.1.4 End User Evaluation

As well as ensuring that the model is a valid representation of the system, it is also important to evaluate the usability of the model. In order to evaluate usability of the model an external domain expert evaluation was needed, as experts involved in creating the model will tend to overlook these considerations if not identified at first. This evaluation was performed by conducting an elicitation review (see section 2.3.1) with the model end user, Sophie Martin, from Goulburn Murray Water. This evaluation was focused on the ontological component of the model.

5.2 Evaluation

5.2.1 Sensitivity to Findings

The results of the sensitivity to findings analysis reflect the quantitative elicitation carried out with the domain experts. The results of sensitivity to findings analysis enabled the ranking of variables according to the capacity of evidence to change the posterior probability of the query node. Thus identifying the management interventions that will have the greatest effect on the posterior probability of interest. If used correctly, these results provide yet another method of prioritising where changes in the system will be most influential.

Results from preliminary analysis show the following ranking of variables for the query variable, **Future Abundance** (see Table 5.1). These results indicate that **Future Abundance** is most sensitive to **Future Diversity**, followed by **Water Quality**.

When findings for **Water Quality** are entered into the network, the sensitivities change and a new ranking is obtained. These results show that when the **Water Quality** is **Low** changes to other variables will have less effect. This demonstrates the dominating effect of a low water quality in the system. On the other hand when **Water Quality** is **High** changes in the some of the remaining variables will be even more influential. These observations agree with the domain expert evaluation of the system.

5.2.2 Sensitivity to Parameters

In order to test the program measuring sensitivity to parameters, the results from [13] were replicated. In the study, sensitivity analysis was conducted on the well-known ALARM-network

Table 5.1: Sensitivity to Findings Analysis performed on **Future Abundance**

	No Evidence		WQmain = Low		WQmain = High	
Entropy of Future Abundance	0.826839		0.439008		0.945721	
Node	MI	Rank	MI	Rank	MI	Rank
FutureDiversity	0.091180	1	0.040395	1	0.105656	1
WQmain	0.067922	2	-	-	-	-
Site	0.029672	3	0.000767	4	0.005396	5
OverallFlow	0.028498	4	0.002151	2	0.024601	3
BiolPoten	0.025654	5	-	-	0.049423	2
Barrier	0.023502	6	0.000400	8	-	-
Temp	0.023492	7	0.000394	10	-	-
AvrSummer	0.022253	8	0.000486	7	0.005037	6
Type	0.022216	9	0.000399	9	0.002369	10
OverallSummer	0.020825	10	0.000642	5	0.008157	4
OverallWinter	-	-	0.000865	3	0.004022	9
AvrWinter	-	-	0.000495	6	-	-
PopStat	-	-	-	-	0.004818	7
MinSummer	-	-	-	-	0.004569	8

[3]. Figure 5.4 gives an example of a typical insensitive function. The conditional probability being altered, on the x-axis, is $pr(\text{Shunt}=\text{Normal}|\text{plum}=\text{True})$ and the observed posterior probability, on the y-axis, is $pr(\text{Shunt}=\text{Normal}|\text{PAP}=\text{High})$. The gradient of this linear function is 0.136, which is a relatively low sensitivity value.

Figure 5.5 gives an example of a highly sensitive plot. Here it can be seen that if the parameter value under study is close to the vertex of the hyperbola then the corresponding observed posterior probability will be altered dramatically. In this figure this would be near the parameter value of 0.08 on the x-axis.

5.2.3 MATILDA Evaluation

In a session with the domain expert, the relationships between selected variables and single variables were examined using MATILDA. Unfortunately, due to time constraints and the complexity of the model, it was impossible to do a complete evaluation of the model. Instead the objective of the session was to aid the domain expert in better understanding the concepts of d-separation in their model.

Using Matilda, the d-separation relationships between variables were mostly anticipated by the domain expert. However, there were some unexpected results when describing relationships between the different portions of the model. Due to the complex nature of the model, some inference paths were difficult to identify without MATILDA assistance. For example, **Future Abundance**'s influence on **Current Diversity** via the various paths though the **Barrier** node. Casual and diagnostic relationships were readily understood whilst some intercausal and mixed

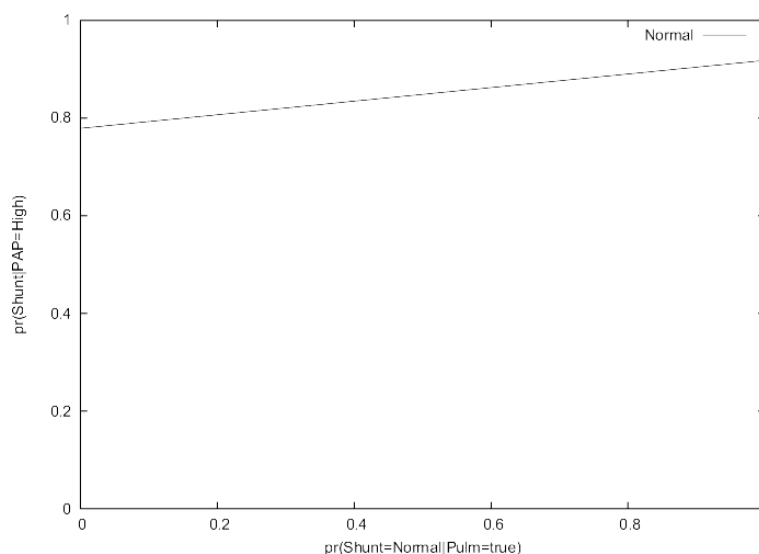


Figure 5.4: Example of an Insensitive Parameter Function

relationships were not immediately clear. This result was expected as these types of relationships are often considered the most difficult to understand.

5.2.4 End User Evaluation

This evaluation covered all components of the model but was mainly focussed on the ontological components of the model. For each of the variables that had a non-trivial meaning the end user was asked to give their interpretation of what they thought it's meaning was. For the variables that had an unclear meaning the actual meaning was given and possible naming changes were identified. This type of evaluation methodology was also applied to identifying the variable state names that were unclear. Issues with the consistency of state spaces were also addressed where the naming and ordering conventions differed across nodes. More general aspects of the model were then discussed identifying missing variables and links in the model.

5.3 Outcomes

5.3.1 Sensitivity Analysis Support Tools

The results of the sensitivity to findings investigations provide a useful extension to the user interface of the model. The sensitivity analysis tools developed in this study will be used to assist in making decisions regarding management interventions, and identifying where future studies will be most beneficial. The results of sensitivity analysis will also be used for further developing the model, prioritising and assisting in the elicitation of conditional probabilities.

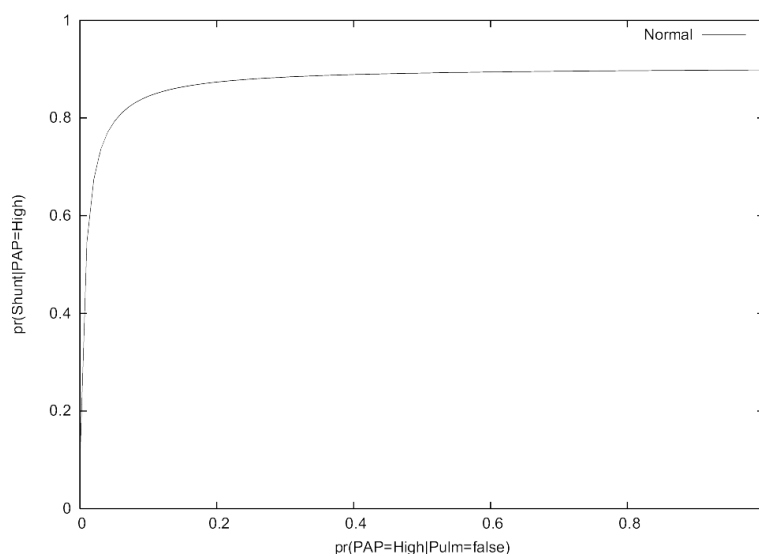


Figure 5.5: Example of a Sensitive Parameter Function

5.3.2 MATILDA Evaluation

The MATILDA evaluation exercises were useful in aiding the domain expert to gain better understanding of the model. It also helped to identify where the implications of d-separation were not immediately obvious in the model. This was useful, not only for understanding the model, but also in guiding future structural elicitation sessions with experts who do not **any** experience with the technology.

5.3.3 End User Evaluation

The end user evaluation session helped to evaluate the usability of the model. Suggestions were generated identifying aspects of the model that were unclear or were not represented. These suggestions are to be incorporated into future stakeholder evaluation sessions.

Chapter 6

Phase 3

This development phase was concerned with incorporating automated methods of model learning and evaluation from the available data, into the model development. To assess the suitability of the data to these tasks, a data quality analysis was conducted, identifying missing variables and assessing the completeness of coverage of variable states. Based on the results of this analysis, methods for incorporating the data information into the parameters and causal structure were identified and implemented. In order to test the validity of the model predictive accuracy tests using the data were also performed.

6.1 Methods

6.1.1 Automated learning of Model Components

In order to incorporate improved parameter learning into the model, different support tools offered by Netica were investigated and tested for suitability. The Netica environment supports a number of different methods for learning parameters from case data files, these were the Lauritzen & Spiegelhalter, EM and gradient descent methods (see section 2.2.5). Both the EM and gradient descent methods allowed the combination of data with expert knowledge already in the CPTs, via the assigning of an experience weight. All of these Netica methods were investigated to determine how the data could best be incorporated into the existing model CPTs.

CaMML (see section 2.2.5) a tool for automated learning of the underlying causal model of data was also applied. In order to gain a better understanding of the data, and perhaps create useful feedback, the CaMML generated model was created for comparison with the expert generated model.

6.1.2 Predictive Accuracy

To evaluate the model, its predictive accuracy (see section 2.3.2) on the case data was determined. The predictive accuracy is determined by entering data case information, whilst withholding values, to see how often the model predicts the missing value/s correctly. When combining this evaluation with models learnt from data it is standard to divide the data into

two sets, one for learning and one for testing, so that the same data is not used twice. This measure does not access how close, or distant, the prediction is from the correct value, however it is a good initial grade of the model.

6.2 Evaluation

6.2.1 Data Quality Analysis

In order to evaluate the suitability of the data to automated learning tasks a data quality analysis was undertaken. To generate the case file, physical and fisheries data cases for each of the sites were matched by sample date and incorporated into a single case file. Data cases of native fish abundance were time stamped and matched to corresponding cases of 1 and 5 year time scales to incorporate values for the **Future Abundance** node. The resulting case file had approximately 1500 case entries, but as most of the sites have low native fish populations there were only 8 cases with a high future abundance.

The first of the potential problems identified during this analysis was the amount and locations of missing variables. Many of the variables in the model were created exclusively for the model and the case data file consequently did not have entries for them. Variables that did occur in the case files are identified in figure 6.1. As can be seen from this figure the missing variables are clustered near each other, leaving chains of missing variables in the model. The consequences of such chains of missing variables are hard to quantify, but one would presume that as expectation maximisation and gradient descent methods find local maxima, this could be a problem.

The second problem identified was the limited coverage of variable values. The node **Water Quality Habitat Descriptor** had extremely limited coverage with variation in only the **Temperature Modification** node as each of the other parent variables remained in the same state across all the cases. For these reasons it was excluded from any learning via data. The site dependent nodes had a complete coverage of parent states and therefore included in learning. The remaining nodes had a reasonable coverage of about 50%-90% of parent states. For these nodes it was decided that learning would be feasible although this was still done with caution.

Early experiments with learning parameters produced odd results for the **Water Quality Habitat Descriptor**. When experience values for the CPTs in the network were altered the resulting changes in this nodes CPTs were flipped. At low experience values the CPTs changed in one direction and as the experience values were increased the resulting changes moved to the opposite direction. Whether this was caused by a bug in the Netica algorithm or because this node, in particular, suffered from the problems identified above was not determined.

6.2.2 Learning Parameters from the Data

In the original model the data was only incorporated into the nodes which had data values present in the case file (see figure 6.1) minus the **Future Abundance**, **Site**, **Type** and **Time Scale** nodes which were included during the stakeholder workshop. This was performed on each of the BNs representing separate spatial scales using the S&L method with the case data for that spatial scale. Inclusion of temporal and improved spatial representations allowed the incorporation of all spatial case files into a single global file with values for the **Future Abundance** node (see section 4.1.3). These improvements meant that the learning capacity of the model was

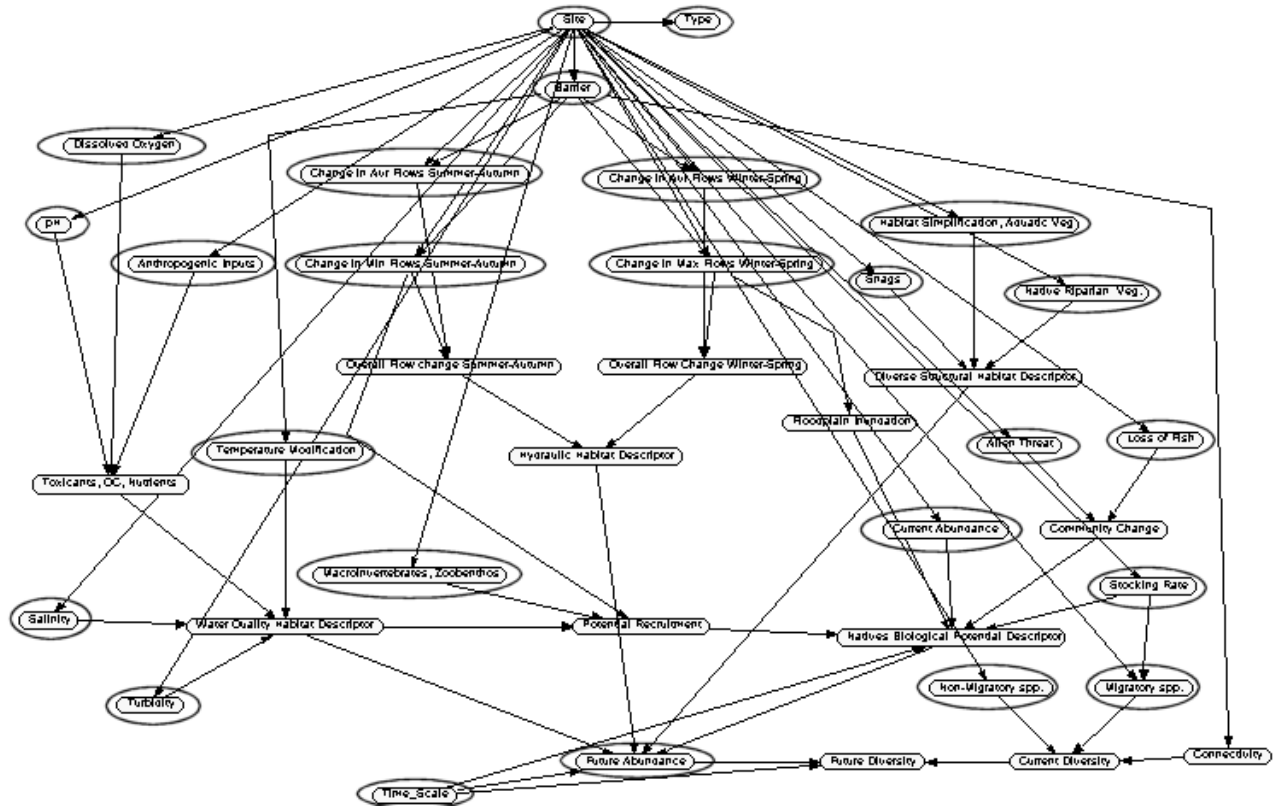


Figure 6.1: Variables that have Case File Entries (Circled)

increased to include the **Future Abundance** node and all of its ancestors. As no values could be generated for the **Future Diversity** node no automated learning could be done on it or its ancestor nodes **Current Diversity** and **Connectivity**.

Due to the sparse and incomplete nature of the data the GD method converged quickly on deterministic CPT's, which was undesirable, and therefore the method was no longer considered. Results from the EM method were more promising and an investigation of the weighting of the expert CPT's was pursued. In order to summarise the changes to the model's CPTs the Bhattacharyya distance between the pre and post learning BNs was identified (see section 2.3.2). These measures were then used to guide the domain expert in evaluating the changes produced so an appropriate weighting could be given to the expert CPT's. If the domain expert believed the resulting parameter value was outside the error range identified then the experience value for the node was increased. The variables for which this process was applied were: **Hydraulic Habitat Descriptor**, **Diverse Structural Habitat Descriptor**, **Natives Biological Potential Descriptor**, **Temperature Modification**, **Community Change**, **Floodplain Inundation**, **Future Abundance**, **Potential Recruitment**.

Table 6.1 shows an example of the Battacharyya distance feedback for node **Natives Biological Potential Descriptor** with an experience value of 5.

Table 6.1: Bhattacharyya distance for node **Natives Biological Potential Descriptor** with experience value 5

Parent State Values						D_B
Time Scale	PopStat	LowFlowSp	CommCh	Stocking	Floodplain	
One year	Low	Low	Yes	None	no	0.0023
One year	Low	Low	Yes	None	yes	0.0002
One year	Low	Low	No	None	no	0.0014
One year	Low	High	Yes	None	no	0.0019
One year	Low	High	No	None	no	0.0005
Five year	Low	Low	Yes	None	no	0.0001
Five year	Low	Low	Yes	Low	yes	0.0001
Five year	Low	Low	No	Low	yes	0.0002
Five year	High	Low	Yes	None	no	0.0001
Five year	High	High	Yes	None	no	0.0003

6.2.3 Learning Structure from the Data (CaMML)

A CaMML structure was created, concentrating only on variables where case data existed (see circled variables in figure 6.1). Figure 6.2 shows the network structure resulting from the trial CaMML run. The structure, although incomplete and in some parts nonsensical (**Future Abundance** causing **Time Scale**), identified some interesting causal relationships in the data. Some of these causal relationships could be related to other current studies in river systems, such as the effect of habitat simplification on native fish. Unfortunately, this model did not say much about the causes of **Future Abundance** except that it was best determined by the **Barrier** type. Barriers, including dams and weirs, have a well known major effect on native fish populations which has been identified by CaMML from the data.

6.2.4 Predictive Accuracy

As case values were generated for the **Future Abundance** node (see section 6.2.1) the predictive accuracy of the expert CPTs for this node could be assessed with predictive accuracy measures. The confusion matrix generated when testing the predictive accuracy of **Future Abundance** is displayed in table 6.2. The confusion matrix shows the number of cases for each actual versus predicted pair with the diagonal elements representing correct matches. The error rate is the percentage of cases predicted incorrectly. As can be seen from these results the model did not distinguish well between the **High** and **Low Future Abundance** cases, instead predicting **Low** all the time. Further investigation through manually entering cases for the observation nodes revealed that a **High** prediction was only predicted in near optimal states.

After the training of this node with the EM method the predictive accuracy tests were reapplied. Preliminary results, not performing the split on the data, showed that there was no improvement in this node. Although this same preliminary analysis *did* show improvements to the nodes previously trained using the L&S method. A proper predictive accuracy analysis, by splitting

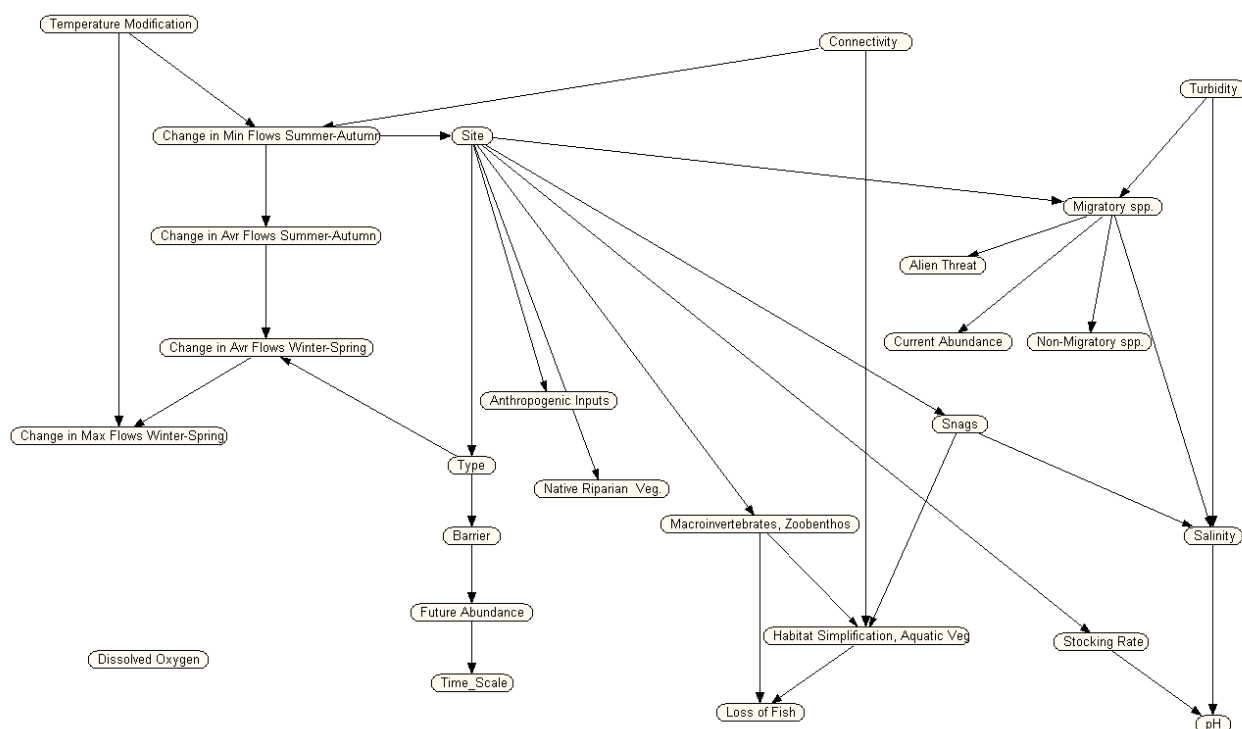


Figure 6.2: Structure created by CaMML when run on data case file

Table 6.2: Prediction Confusion Matrix for **Future Abundance**

Low	High	Actual
103	0	Low
8	0	High
Error rate	7.207%	

the data into two sets and using one for evaluation exclusively, was not done but was identified as a potential for future work.

6.3 Outcomes

6.3.1 Bhattacharyya Distance Support tool

The Bhattacharyya distance support tool provides a useful extension to the Netica programming environment. Providing a useful and powerful interface to compare BNs with differing CPTs. In this application the Bhattacharyya distance tool was used to compare the pre and post models in automated parameter learning with data. It was useful to review the overall changes to the model identifying which parameters were changing and by how much.

Table 6.3: Experience Values Assigned to Missing Variables

Node Name	Experience Value Assigned
Hydraulic Habitat Descriptor	5
Diverse Structural Habitat Descriptor	5
Natives Biological Potential Descriptor	5
Temperature Modification	10
Community Change	5
Floodplain Inundation	5
Future Abundance	10
Potential Recruitment	5

6.3.2 Changes to Quantitative Component

As a result of the investigations into the effects of incorporating the case data files the variables with data case entries were given experience values of 1. The variables with expert elicited CPTs, if included in training, were given experience values between 5 and 10 (see table 6.3).

6.3.3 Predictive Accuracy

The results of the predictive accuracy tests were hardly surprising given the status of the project. The poor quality of the data (see section 6.2.1), having only 8 cases with a high future abundance meant that expectations of the value of predictive accuracy tests on this node were low. These results also reflect the poor knowledge of the fish communities in the catchment and suggest that the domain experts are biased toward giving pessimistic predictions. Using an earlier model iteration, plots showing actual fish abundance data versus predicted fish abundance data [39], further suggest that this would be the case (see figures 6.3 & 6.4). In these plots there is a positive correlation between the actual and predicted values but the model poorly predicted sites with high fish abundances.

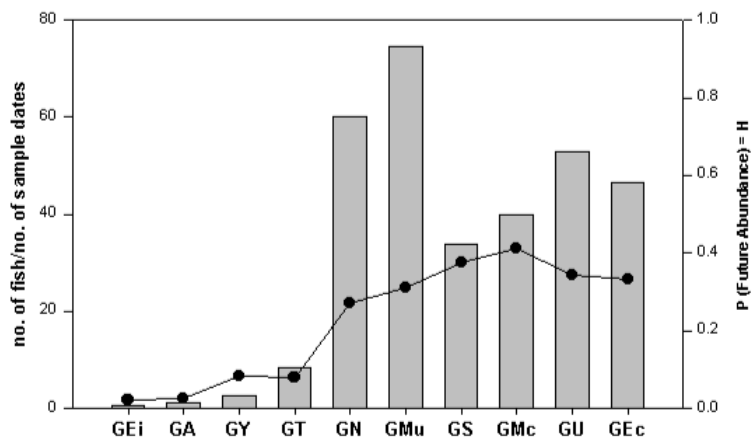


Figure 6.3: Relative Abundances of Native Fish at a site (≥ 1970 fisheries data) vs. BN Predicted Abundances of 'High' (≥ 54) Native Fish at the same site [40]

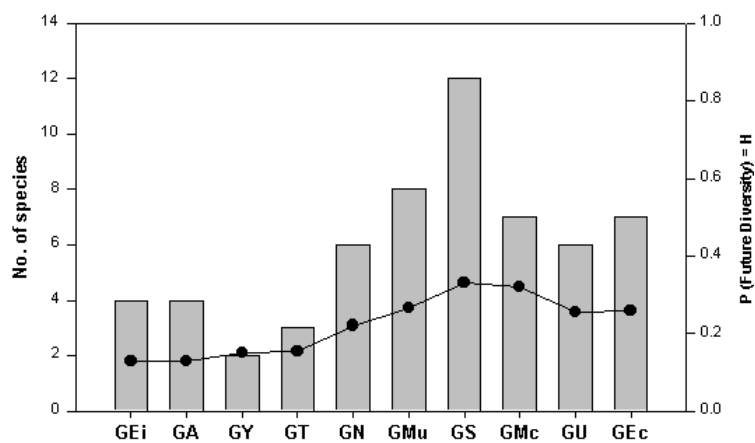


Figure 6.4: Total Number Species of Native Fish at a site (≥ 1970 fisheries data) vs. BN Predicted Diversity of 'High' Native Fish at the same site [40]

Chapter 7

Conclusions and Further Work

The research in this thesis was motivated by the requirements for formal knowledge engineering techniques in the development of Bayesian Networks for application domains. This thesis was conducted by involvement in a practical application, an Ecological Risk Assessment, giving a first hand experience of the potential difficulties in the development tasks involved. The project underwent three development phases in the duration of this thesis covering aspects of elicitation, evaluation and implementation tasks involving human and automated data learning. Any potential knowledge engineering methods/support tools identified for assisting in the development of this model were investigated and either applied, if existing, or generated and applied, if not. In this chapter, I review the achievements of this thesis in both the BN knowledge engineering research field and the ecological project. I then provide directions for future work identified.

7.1 Review

The first task of this thesis was a reverse engineering analysis of the ERA prototype model (chapter 3). This analysis included identification of knowledge engineering tasks that had not been undertaken by the model developers or that could be improved. The prototype model was evaluated and preliminary improvements were identified. These included the incorporation of temporal and improvements to the spatial representation of the model, which had a secondary advantage of an improved potential for learning from the data. It was also identified that elicitation and evaluation support tools would be useful in assisting the development tasks.

In the first development phase (chapter 4) improvements identified during the reverse engineering of the model were investigated and presented at a stakeholder workshop held to evaluate the model. These included improvements to the spatial and temporal representation of the model. During the workshop, a range of improvements to the ontological, qualitative and quantitative components of the model were identified. Stakeholder suggestions were evaluated and implemented where possible. The temporal and spatial representation improvements were accepted and identified as powerful improvements to the representational power of the model. The workshop process highlighted the difficulties in the elicitation and evaluation of the quantitative component of the model. It was suggested that development of sensitivity analysis support tools would be useful for identifying where greater effort could be directed for improving the quantitative component of the BN.

In the second development phase (chapter 5) various support tools were created and/or applied to evaluating the quantitative and qualitative components of the model. The end usability of the model was assessed and qualitative evaluation was conducted using structural evaluation program MATILDA. The development of the sensitivity analysis support tools took two directions. In order to identify which variables had dominating effects on the query variables, analysis on sensitivity to findings was conducted. Although support for this type of analysis was provided in the Netica environment, Netica did not enable a complete sensitivity analysis to be carried out efficiently. In order to improve upon the existing Netica tools, an automated program was written using Netica's programming interface. This enabled the generation of improved and informative reports.

In order to identify which parameters had dominating effects on the query variables, analysis on sensitivity to parameters was conducted. No existing support tools for this task could be identified, although existing literature describing methods to do this efficiently were found. Again using Netica's programming interface, support tools were generated to guide the elicitation and evaluation of the quantitative components.

The structural evaluation support tool MATILDA was used to improve the domain expert's understanding of the network structure. This was useful, as this improved knowledge would aid in directing future structural elicitation, evaluation and implementation. The end usability of the model was also assessed and suggestions were made for ontological improvements. The resulting tools and evaluation outcomes of this phase are to be incorporated with stakeholder evaluation in future development phases.

In the third development phase (chapter 6) methods for evaluating and learning parameters and structure were investigated and applied. The existing data quality was assessed to determine its suitability to the learning tasks. Due to the poor quality of the data, methods were developed allowing the domain expert to supervise the automated learning tasks. Investigation into each of the learning methods revealed that the expectation maximisation algorithm would be most appropriate to learn the parameters. Support tools to do EM learning were included in the Netica environment. In order to guide this learning task a program applying the Bhattacharyya distance measure was developed to provide feedback on which parameters were changed by learning and by how much.

In addition to the learning of parameters an automated model-learning tool, CaMML, was used. Even though the resulting structure was not useful in improving the model, it did offer interesting insight into other properties of the data. The predictive accuracy was also assessed in this phase, using support tools in the Netica environment. Results from this evaluation showed that the model was not a very good predictor of the data. This was not very surprising as the knowledge of the domain system is poor and there is a suspected negative bias in parameters elicited from experts.

7.2 Further Work

Further evaluation and elicitation of parameters needs to be done to improve the predictive accuracy of the model. This process could be supported using the sensitivity analysis tools created in this thesis and by improving the data collection techniques. The sensitivity analysis tools can be used to identify where improvements would be most beneficial. Improvements to the nodes can be identified with the sensitivity to findings tools, and improvements to the

parameters can be identified with the sensitivity to parameters tools. Data collected from sites where management interventions are being implemented could be used to train the model, improving its utility as a management support tool. Also the suggestions generated during the end user evaluation need be investigated to determine where incorporation into the model would be beneficial.

During the stakeholder workshop it was noted that the methods applied to the spatial and temporal representation of the model could also be applied to improving the fish species representation of the model. The inadequacy of the current species representation was apparent during the evaluation of the model, as many of the relationships are dependent on the species being considered. An improvement could be an additional variable like the **Site** variable which has values for each of the species in the river system. This new node could be linked to all species dependent variables in the model allowing different relationships to be identified depending on the species.

The model contains much information that could be used to drive further investigation in the ecology research area. The nodes identified by the sensitivity to finding tools as having dominating effects could be used to direct where such study would be most beneficial. CaMML identifies causal relationships in the data and can be used to identify possible relationships between variables in the system worthy of investigation.

References

- [1] gnuplot interfaces in ansi c. <http://ndevilla.free.fr/gnuplot/>.
- [2] Netica bayesian network software from norsys. <http://www.norsys.com/>.
- [3] I.A. Beinlich, H.J. Suermondt, R.M. Chavez, and G.F. Cooper. The ALARM monitoring system: A case study with two probabilistic inference techniques for belief networks. In *Proceedings of the Second European Conference on Artificial Intelligence in Medicine*, pages 247–256, London, 1989.
- [4] A. Bhattacharyya. On a measure of divergence between two statistical populations defined by their probability distributions. *Bull. Calcutta Math Soc.*, 35:99–110, 1943.
- [5] 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.
- [6] 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.
- [7] 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.
- [8] J Cain. Planning improvements in natural resources management. Technical report, Centre for Ecology and Hydrology, Crowmarsh Gifford, Wallingford, Oxon, UK, 2001.
- [9] 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.
- [10] Eugene Charniak. Bayesian networks without tears. *Artificial Intelligence Magazine*, 12:50–63, 1991.
- [11] P. Clunie, K. James T. Ryan, and B. Cant. Implications for rivers from salinity hazards: Scoping study. Technical report, Victoria, Arthur Rylah Institute, DNRE, 2002.
- [12] 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.

- [13] V.M.H. Coup and L.C. van der Gaag. Properties of sensitivity analysis of bayesian belief networks. *Annals of Mathematics and Artificial Intelligence*, 36:323–356, 2002.
- [14] B. D’Ambrosio. Inference in Bayesian networks. *Artificial Intelligence Magazine*, 20(2):21–36, 1999.
- [15] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society, Series B (Methodological)*, 39:1–38, 1977.
- [16] 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.
- [17] L.C. van der Gaag and S. Renooij. Analysing sensitivity data. In J. Breese and D. Koller, editors, *Seventeenth Conference on Uncertainty in Artificial Intelligence*, pages 530–537. Morgan Kaufmann Publishers, 2001.
- [18] P. Haddaway. An overview of some recent developments in Bn problem-solving techniques. *Artificial Intelligence Magazine*, 20(2):11–19, 1999.
- [19] David Heckerman. Probabilistic similarity networks. *Networks*, 20:607–636, 1990.
- [20] M. Henrion, J.S. Breese, and E.J. Horvitz. Decision analysis and expert systems. *Artificial Intelligence Magazine*, 12:64–91, 1991.
- [21] 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.
- [22] 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.
- [23] Frank Jensen, Finn V. Jensen, and Søren L. Dittmer. From influence diagrams to junction trees. *Uncertainty in Artificial Intelligence: Proc. of the Tenth Conf.*, 1994.
- [24] 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.
- [25] J. D. Koehn and W. G. O’Connor. Biological information for management of native freshwater fish in victoria. Technical report, Victoria, Parks, Flora and Fauna Division, State Government of Victoria, 1990.
- [26] Kevin B. Korb. Parameter learning. Causal Discovery Lecture Notes.
- [27] Kevin B. Korb and Ann E. Nicholson. *Bayesian Artificial Intelligence*. CRC Press, 2003.
- [28] 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.

- [29] 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.
- [30] S. L. Lauritzen and D. J. Spiegelhalter. Local computations with probabilities on graphical structures and their application to expert systems. In G. Shafer and J. Pearl, editors, *Readings in Uncertain Reasoning*, pages 415–448. Kaufmann, San Mateo, CA, 1990.
- [31] 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.
- [32] 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.
- [33] B.G. Marcot. A process for creating bayesian belief network models of species-environment relations. Technical report, USDA Forest Service, Portland, Oregon, 1999.
- [34] 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.
- [35] 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.
- [36] 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.
- [37] Department of Sustainability and Environment. Victoria water resource data warehouse. <http://www.vicwaterdata.net>.
- [38] Judea Pearl. *Probabilistic Reasoning in Intelligent Systems*. Morgan Kaufmann, San Mateo, Ca., 1988.
- [39] C.A. Pollino. Ecological risk associated with irrigation in the goulburn-broken catchment phase 2: Milestone 4 fish component. npird internal report. Npird internal report, Water Studies Center, Monash University, 2003.
- [40] C.A. Pollino, P. Feehan, M. Grace, and B. Hart. Quantifying the risks to fish in the goulburn broken catchment (victoria, australia) using bayesian networks. *Society of Environmental Toxicology and Chemistry (SETAC)*, 26, 2003.
- [41] 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.
- [42] 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.

- [43] T. Ryan, A. Webb, R. Lennie, and J. Lyon. Status of cold water releases from victorian dams. Technical report, Victoria, Arthur Rylah Institute, DNRE, 2001.
- [44] 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.
- [45] Chris S. Wallace and Kevin B. Korb. Learning linear causal models by MML sampling. In A. Gammerman, editor, *Causal Models and Intelligent Data Management*. Springer-Verlag, 1999.

Appendix A

Prototype Model Details

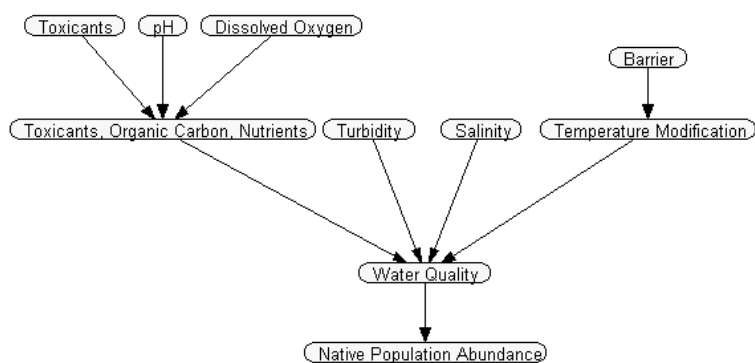


Figure 1: The Water Quality Portion of the Prototype Model

Table 1: Methodology used to discretise nodes and resulting states [39]

Node	Discretisation methodology	States
Barrier	Based on classification in ISC (Index of Stream Condition)	None, Complete Deep, Complete Shallow and Inundated
pH	EPA draft SEPP (State Environment Protection Policy) Guidelines	alkaline, neutral and acidic
Dissolved Oxygen	EPA draft SEPP Guidelines	Low and Normal
Turbidity	EPA draft SEPP Guidelines	Low, Medium and High
Salinity	Salinity data contained in [11] - fish data only	Low, Medium and High
Toxicants	Based on expert knowledge	Low, Moderate and High
Temperature modification	[43] Modelling of natural temperatures, and relating to temperatures required for spawning [25]	None, Minor, Moderate and Major
Change in Avr Flows Dec-Feb, Change in Min Flows Dec-Feb, Change in Max Flows Dec-Feb, Change in Avr Flows Jun-Aug, Change in Min Flows Jun-Aug, Change in Max Flows Jun-Aug, Change in Flow Regime Dec-Feb and Change in Flow Regime Jun-Aug	Collection of pre dam and post dam data, calculating % change in means Classification in ISC [37]	None, Low, Moderate, High and Extreme
Native Riparian Veg	Classification in ISC [37]	Low, Medium and High
Habitat Simplification		None, Some and Complete
Snags		Low, Medium and High
Historic Population Status, Alien Population and Native Population Abundance	Percentiles of populations based on abundances of fish at each site of interest in the Goulburn catchment	Low, Medium and High
No. of Migratory spp, No. of Non-Migratory spp and Native Population Diversity	Percentiles of populations based on abundances of fish at each site of interest in the Goulburn catchment	Low and High
Stocking	Information from [37]	Yes and No
Migration	Based on the type of barrier present	Yes and No
Structural Habitat Quality, Water Quality and Toxicants, Organic Carbon, Nutrients	Based on stakeholder workshop outcomes	Low, Medium and High
Overall Change in Flow Regime, Change in Flow Regime Dec-Feb and Change in Flow Regime Jun-Aug		None, Low, Moderate, High and Extreme
Competition		Low and High

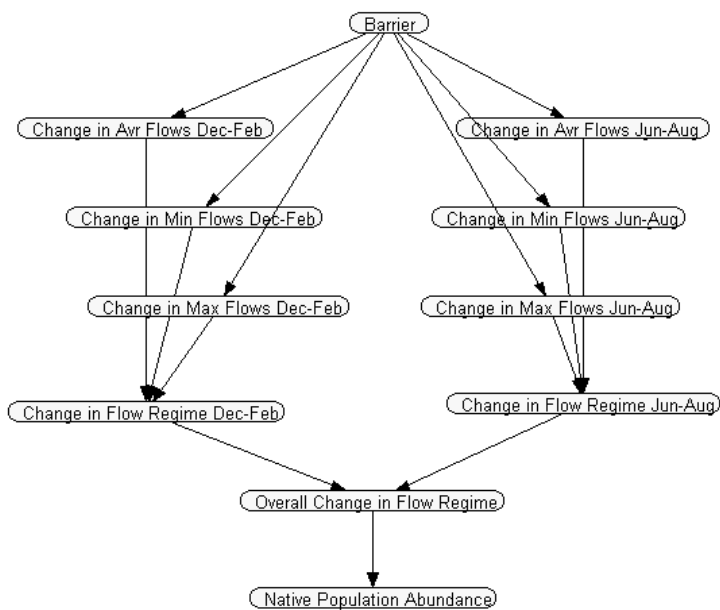


Figure 2: The Flow Portion of the Prototype Model

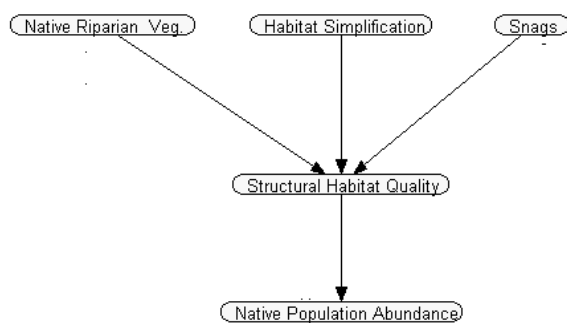


Figure 3: The Habitat Portion of the Prototype Model

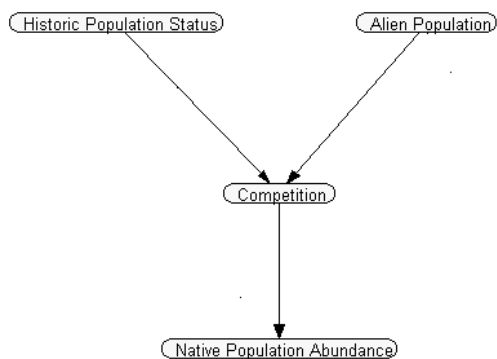


Figure 4: The Biological Interaction Portion of the Prototype Model

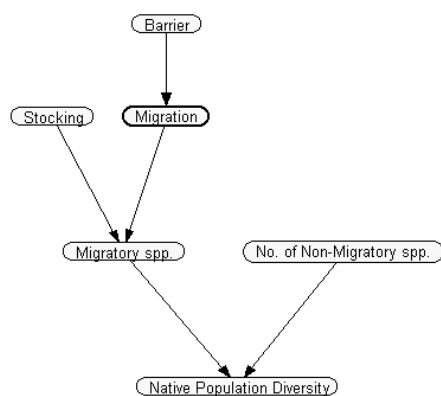


Figure 5: The Species Diversity Portion of the Prototype Model

Post Stakeholder Workshop Model

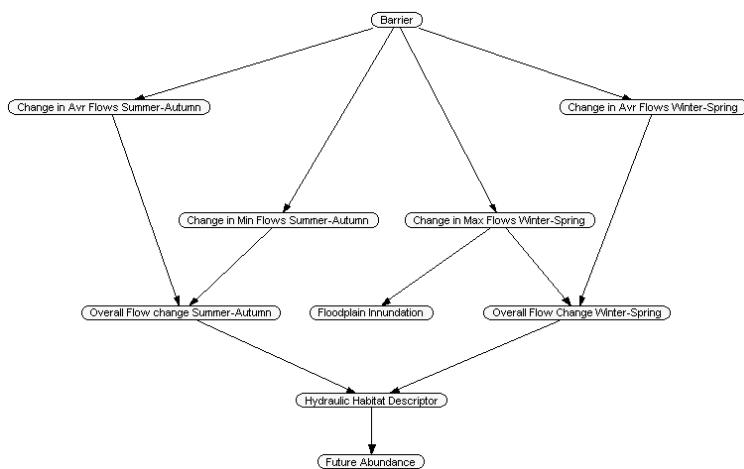


Figure 6: Flow Portion Resulting from Phase 1

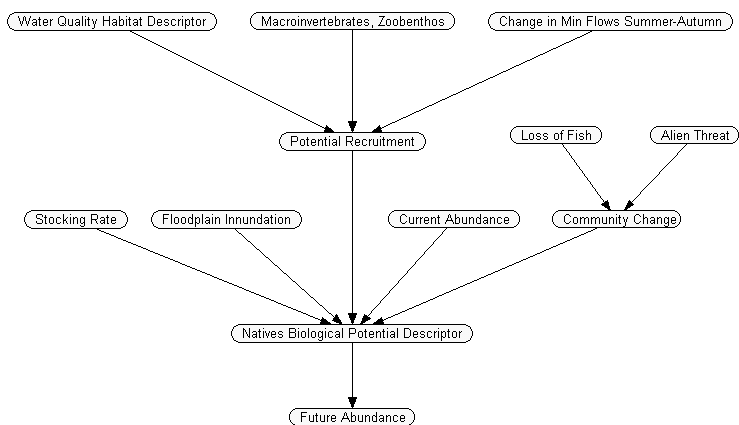


Figure 7: Biological Interaction Portion Resulting from Phase 1

Table 2: Changes to Network Ontology

Variable	New State Space	New Variable Name
Removed		
Change in max flows Dec-Feb		
Change in min flows Jun-Aug		
Added		
Floodplain Inundation	No and Yes	
Macroinvertebrates, Zoobenthos	Low, Medium and High	
Potential Recruitment	Low and High	
Loss of Fish	Low and High	
Community Change	Yes and No	
Future Diversity	Low and High	
Site, Type and Time Scale	(see Sections 4.2.1&4.2.1)	
Renamed		
Migration		Connectivity - Migration
Change in min flows Dec-Feb		Change in min flows summer-autumn
Change in max flows Jun-Aug		Change in max flows winter-spring
Change in flow Regime Dec-Feb		Overall flow change summer-autumn
Change in flow Regime Jun-Aug		Overall flow change winter-spring
Toxicants		Anthropogenic Inputs
Water Quality		Water Quality Habitat Descriptor
Stocking		Stocking Rate
Native Population Diversity		Native Fish Diversity
Overall Change in Flow Regime		Hydraulic Habitat Descriptor
Historic Population Status		Current Abundance
Natives Population Abundance		Future Abundance
Alien Population		Alien Threat
Competition		Natives Biological Potential Descriptor
Habitat Simplification		Habitat Simplification, Aquatic Veg
Structural Habitat Quality		Diverse Structural Habitat Descriptor
Rediscretised		
Change in avr flows summer-autumn Change in avr flows winter-spring Change in min flows summer-autumn Change in max flows winter-spring Overall flow change summer-autumn Overall flow change winter-spring Hydraulic Habitat Descriptor	ExtDecrease, Decrease, NoChange, Increase and ExtIncrease	
Temperature Modification	NoChange, Moderate and Major	
Dissolved Oxygen	ExtremeLow, Normal and ExtremeHigh	
Stocking Rate	None, Low and High	
Current Abundance Future Abundance Alien Threat	Low and High	
Natives Biological Potential Descriptor	Low, Medium and High	
Native Riparian Veg	Degraded, Moderate and Intact	

