The Knowledge Engineering Review, Vol. 15:3, 2000, 233–256
Printed in the United Kingdom

Decision-model construction with multilevel influence diagrams

X. WU[1] and K. L. POH

Department of Industrial & Systems Engineering, National University of Singapore, 10 Kent Ridge Crescent, Singapore 119260.
Email: wuxi@krdl.org.sg, isepohkl@leonis.nus.edu.sg
[1]*Also at Kent Ridge Digital Laboratories, 21 Heng Mui Keng Terrace, Singapore 119613*

Abstract

We propose an approach to knowledge-based decision-model construction. The domain knowledge is represented in the form of multilevel influence diagrams. At each level, a regular influence diagram representing a decision model may be generated and the influence diagrams between the levels are related through a series of operations. We defined eight such operations and show that these are sufficient at the structural level to construct any target influence diagram in any practical situation. We provide a series of examples drawn from different domains on the use of multilevel influence diagrams for model construction. We also propose suggestions on providing users with guidance that can direct the model-construction procedure towards the best target model.

1 Introduction

In recent years there has been a growing interest among researchers in the artificial intelligence and uncertain reasoning communities in automated probabilistic reasoning and decision modelling. One approach, known as the knowledge-based model construction (KBMC) method, encodes general knowledge about the domain in an expressive language and dynamically constructs a decision model for each particular situation or problem instance (Breese et al., 1994; Wellman et al., 1992). Core issues in KBMC include knowledge representation, triggering and elaboration, control of construction and knowledge-base development.

Graphical decision-modelling formalisms such as influence diagrams (Howard & Matheson, 1981) are often used for the target decision model representation in KBMC systems since they can describe probabilistic relationships as well as support the inference process. Influence diagrams (IDs) are graphical representations of decision problems. They were first defined by Howard and his colleagues more than twenty years ago (Howard & Matheson, 1981; Miller et al., 1978; Owen, 1978). An influence diagram is a directed acyclic graph with four types of node. These are chance nodes (oval), decision nodes (rectangle), value nodes (diamond) and deterministic nodes (double oval), representing random quantities, decisions, utilities and deterministic quantities, respectively. The directed arcs in an influence diagram represent relations between the nodes connected. An arc ending at a chance node represents probabilistic relevance. An arc into a decision node represents information availability at the point of decision. An arc between two decision nodes represents the chronological order of the decisions. An arc into the value node indicates preferences over possible outcomes. Figure 1 shows an example of a influence diagram representing the "bring umbrella" problem. The problem is to decide whether or not to take an umbrella along to work. The weather is uncertain and the decision-maker is concerned with its outcome and the decision taken. Weather forecast information is available to the decision-maker and this information is dependent on the actual weather.

Many knowledge representation schemes have been developed for use with KBMC systems.

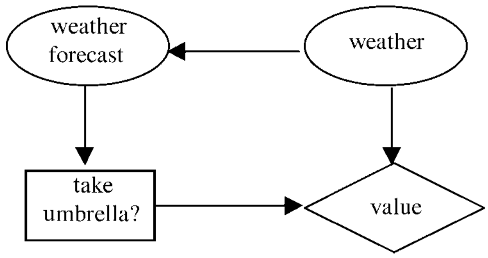

Figure 1 An influence diagram representing the "umbrella problem"

These include first-order predicate logic (Breese, 1992; Goldman & Charniak, 1990; Holtzman, 1989), similarity networks (Heckerman, 1991), fluid multilevel representation (Leong 1991; Wellman 1988), and probabilistic concept networks (Poh & Fehling, 1993, Poh et al., 1994). All these methods can facilitate the dynamic construction of probabilistic and decision models. Methods for model construction may be query-driven, decision-driven, value-driven or data-driven (Breese et al., 1994; Wellman et al., 1992).

In this paper we will first review a number of existing KBMC systems and then introduce a decision-modelling and construction system based on the concept of multilevel influence diagrams. We extend the standard influence diagram representation (Howard & Matheson, 1981) to a multilevel structure in order to facilitate decision-model construction at multiple levels of abstraction. We have developed formal relations between influence diagrams that exist at different levels of abstraction. These relations are represented by a set of multilevel influence diagram operations that enable one to transform an influence diagram at one level to another level of abstraction while preserving all probabilistic relevance information. By using an appropriate set of operations, a non-domain expert user can use multilevel influence diagrams to construct a target decision model, according to his or her specifications, with a domain knowledge base.

Multilevel influence diagrams serve the roles of both decision-model representation and knowledge representation. This unification of the target model language and knowledge-base language increases the efficiency of model-construction processes. Our work complements a number of previous works on model-construction techniques as well as some ongoing works. These include those of Holtzman (1989), Breese (1992), Goldman and Charniak (1990), Heckerman (1991), Wellman (1988), Egar and Musen (1993), Provan (1994), Poh et al. (1994), Yuan (1993) and Mahoney and Laskey (1998).

This paper is organized as follows: in the next section, we give a review of some related work. In Section 3 we define the operations for multilevel influence diagrams, specify conditions for soundness and show their sufficiency. In Section 4 we provide an algorithm for performing KBMC using multilevel influence diagrams. In Section 5 we describe an implementation of our system. In Section 6 we describe some real-world applications of our system. Finally, in Section 7, we conclude by discussing possible extensions or enhancements to the system.

2 Related work

In this section, we present a review of four existing representative KBMC systems that have been reported in the literature. We will provide a brief description of how the systems work and mention their contributions to research in this area.

2.1 RACHEL: An Intelligent Decision System for Infertile Couples (Holtzman, 1989)

RACHEL was a system developed for the application of intelligent decision systems in a medical domain. It was designed to aid infertile couples and their physicians in selecting a medical treatment. In RACHEL, there are three types of knowledge base. The first type is concerned with developing a value model. The second type aims to indirectly assess the value node by using medical knowledge.

The third type embodies the knowledge about the decision-analytic process. To construct models for individual patients, three steps are necessary. First, the system generates alternatives through a user–system dialogue. Next, the system develops a model of the individual patient's preferences. Finally, the system assesses the necessary probabilities. Model evaluation is carried out by the system's influence diagram processing system which can construct, evaluate and analyse influence diagrams. By computing the maximum expected utility, the system yields a recommended decision, an optimal treatment choice and the certainty equivalent. RACHEL was implemented in LISP and there is an online explanation mode to help users understand the operation of the system. The inference engine was designed to perform depth-first backward reasoning. To build the model, the system uses standard first-order prefix–predicate calculus-style interactive queries. It permits user input of special data structures such as the range of possible outcomes and probability distributions for variables that are directly assessed.

The advantages of RACHEL are its ability to report the status of its reasoning and its ability to perform expected value-of-information computation (Howard, 1966; 1967) and other sensitivity analyses. It has also been found to be effective for a particular class of decision problem, namely the construction of special models for couples with different infertility problems, and providing recommendations for the problems.

A major limitation of RACHEL is its restriction to building decision models with only a single decision node and no predecessors. It also requires the users to be familiar with first-order prefix–predicate calculus as well as with the basic concepts of decision analysis. It is also difficult to correct erroneous data entry. Finally, it does little to help the user appraise the decision model.

2.2 ALTERID (Breese, 1987)

ALTERID is an experimental prototype running on LISP workstations. The aim of ALTERID is to provide the user with an environment for both constructing and developing a decision-domain knowledge base and using the knowledge base to answer logical, possible outcome, probabilistic and decision-theoretic queries. The system has three modules, namely the knowledge base, the inference engine and the interaction environment. The system's knowledge-base language is based on first-order predicate calculus. The knowledge-base structure comprises propositions and influences which are well-formed logical, probabilistic and informational influences. The inference engine of ALTERID uses and manipulates sentences in the knowledge-base language. Methods of reasoning include logical inference, probabilistic inference, alternative outcome inference, and decision-theoretic inference. The system's target model representation is an influence diagram and the dynamic model-construction approach can be classified as query-driven. The system has an implicit control structure guiding the search for models based on the principle of minimising uncertain representations. A single query could generate several probabilistic or decision-theoretic models that are consistent with a decision-domain knowledge base.

A major feature of ALTERID is its graceful integration of probabilistic reasoning with logical deterministic inference. Furthermore, the expressiveness of the language it uses does not impose assumptions of conditional independence on the probabilistic representation. Another advantage of ALTERID is the availability of its knowledge base to the user for making changes. The system also provides explanation and heuristic search for the construction and solution of models. As a domain-independent tool, it can support different classes of decision problems by constructing the appropriate knowledge bases. Finally, ALTERID is capable of constructing multiple models for the same phenomena thus enabling the system to investigate the outcomes and results of different models within the same system environment.

A limitation of ALTERID is the possibility that the multiple models it is capable of generating from a query may be inconsistent. Also, by allowing the user to freely modify the knowledge base, there is a danger that the knowledge base may become inconsistent. Another limitation is that it is not suitable for building autonomous agents for decision-making under uncertainty as it can only respond to a given query. Finally, ALTERID cannot deal with continuous variables.

2.3 *PATHFINDER and similarity networks (Heckerman, 1991)*

PATHFINDER is a normative expert system developed to assist surgical pathologists with the diagnosis of lymph-node disease. It uses the hypothetical-deductive approach to make differential diagnosis, and has the capability to recommend the most cost-effective tests using the concept of expected value of information from decision analysis. The most recent version of PATHFINDER was constructed using similarity networks. Similarity networks are tools for the capture and representation of probabilistic knowledge in single-hypothesis diagnosis problems. Similarity networks have provided an efficient approach for the construction of normative expert systems in domains of larger scope and greater complexity than those previously thought to be amenable to the decision-theoretic approach.

Similarity networks comprise a similarity graph and a set of knowledge maps. The similarity graph is used to indicate pairs of possible diagnoses or hypotheses that are considered by the domain expert to have some similarity. The knowledge maps are special and small-sized influence diagrams where the impact of only two possible diagnoses are considered and modelled in the diagram. These knowledge maps represent the expert domain knowledge about the problem when the problem is narrowed down to only two possibilities at a time. To construct the decision-theoretic single-hypothesis diagnosis model, the knowledge maps from the similarity networks are graphically combined.

An advantage of the similarity network is that it makes the construction of very large and complex systems practical. For the single-disease case, it can provide cost-effective recommendations. It can also be used for inference, explanation and utility assessment. Finally, it is a useful heuristic tool in knowledge acquisition. A limitation of the similarity network is that it can only be used to build a system for the diagnosis of a single disease.

2.4 *SUDO PLANNER (Wellman, 1988)*

This system was developed to employ decision-theoretic principles to formulate trade-offs in domains characterized by partially satisfiable goals and actions with uncertain effects. SUDO PLANNER formulates trade-offs for a medical decision problem by proving decision-theoretically that certain classes of plan are dominant based on qualitative relations in the domain. SUDO PLANNER consists of two discrete elements. One is a dominance-proving architecture for planning with partially satisfiable goals, which is used to characterize the space of admissible plans. The other is an effective representation formalism for the effects of actions and the desirability of outcomes. The latter is called a qualitative probabilistic network (QPN) which was specially designed to support the knowledge required for trade-off formulation. QPNs serve as the knowledge-base representation language for SUDO PLANNER. An algorithm is available for the construction of target decision models from a knowledge base which describes the effects of actions and relations among the events at multiple levels of abstraction. The target model representation is also a QPN.

The SUDO PLANNER approach to model formulation is abstract, robust and modular. It is capable of dealing with the uncertain effects of actions and unpredictable goals by combining the methodologies of classical AI planning and decision theory. Finally, SUDO PLANNER can effectively exploit knowledge at multiple levels of abstraction.

SUDO PLANNER suffers from the following limitations: first, it assumes that the current strategy is optimal given the current knowledge, and the planning process must be grounded in situation changes. Next, its plan-constraint language has limited expressive power. In particular, its representations for plan classes, actions and events are entirely atemporal. Finally, its inference mechanisms may mismatch our intuitive notion of trade-off.

3 Multilevel influence diagram representation

One of the most difficult aspects of KBMC in decision-analytic terms is composing a diagram of relevant relations from a set of domain concepts (Egar & Musen, 1993). It is therefore essential that the representation scheme for the domain knowledge be sufficiently expressive for the domain and, at the same time, it must be capable of supporting the construction of different classes of decision model. Multilevel influence diagrams have been developed to satisfy both of these requirements.

The multilevel influence diagram approach extends the standard influence diagram scheme so as to enable a KBMC system to represent a decision situation using different influence diagrams at different levels of abstraction. In multilevel influence diagrams each level of representation corresponds to a standard influence diagram and the diagrams at various levels are linked via a set of operators.

In multilevel influence diagrams, the influence diagram at the highest-detailed level includes every possible detail about the domain. We shall refer to this highest-detailed level influence diagram as the *global influence diagram* for the domain. As we move down from the highest-detailed level to a lower level, each level consists of a valid influence diagram that can be generated to describe the domain in more abstract terms. We shall refer to the most abstract model as the *top-level influence diagram*. By using multilevel influence diagrams, we can selectively construct a target decision model of our choice. The relations between the various levels in multilevel influence diagrams are characterized by a set of operations which we shall define in the following subsections.

3.1 Operations for multilevel influence diagrams

3.1.1 The extend and retract operations

We shall first consider the operation *extend* and its inverse, *retract*. As illustrated in Figure 2, the extend operation takes a chance node C and adds a set of direct predecessor nodes to it, thereby extending the influence diagram to a more refined level. The inverse operation retract, on the other hand, takes a chance node C with direct predecessors and removes all of them, thereby retracting the influence diagram to a more abstract level.

In general, given a chance node C without any predecessor node, the extend operation may be applied in more than one way since any number of predecessors can be added to C. Let $\pi(C)$ represent the set of all potentially possible direct predecessors C. The extend operation can expand the influence diagram based on any non-empty subset of $\pi(C)$. For example, if C has three potentially possible predecessors, i.e. $\pi(C) = \{X_1, X_2, X_3\}$, then there are seven possible ways in which the extend operation can be carried out. The actual set of nodes to be added depends on the actual situation faced by the decision-maker. This feature of allowing the user to freely extend the model in different ways and directions to satisfy a wide range of different requirements and situations provides a high degree of flexibility for the model-construction process. An idea similar to the extend operation was briefly outlined in Provan (1991). We provided here a more detailed description and application of the operation, and also defined the inverse operation, retract, which will allow the user to move between levels of abstraction in both directions during the model-construction process.

3.1.2 The refine and abstract operations

The next two operations, *refine* and *abstract*, are similar to those used in concept refinement and generalization in model construction using probabilistic conceptual network (Poh & Fehling, 1993;

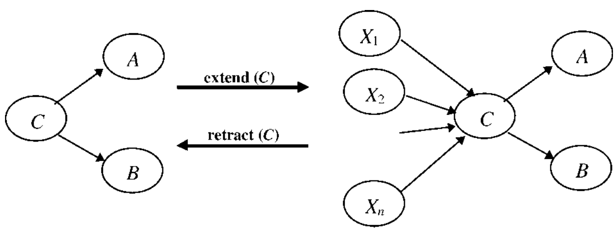

Figure 2 The extend and retract operations

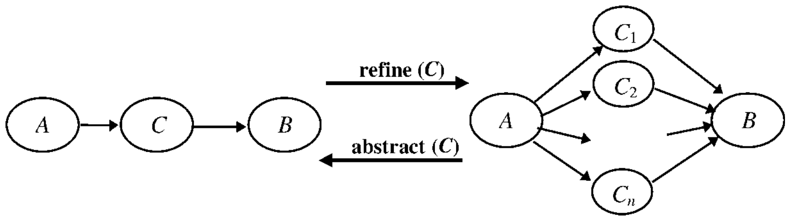

Figure 3 The refine and abstract operations

Poh et al., 1994), PE-subnet (Lehner et al., 1994), state-space abstraction (Liu & Wellman, 1995; Wellman and Liu, 1994) and belief network refinement and coarsening (Chang & Fung, 1990). These operators are designed to change the precision or detail of some chance nodes. As shown in Figure 3, the refine operation replaces a chance node C by a set of refined chance nodes C_1, C_2, ... C_n. We assume that after the operation, all the refined chance nodes have same outcome set $\varphi(C)$ as C. They also have, as direct predecessors, all the previous direct predecessors of C, and have, as direct successors, all the previous direct successors of C. The inverse operation abstract, on the other hand, takes a set of chance nodes C_1, C_2, ... C_n, with same outcome set as well as common direct predecessors and successors, and replaces them with an integrated or abstract chance node C. After the operation, the integrated chance node C has the same outcome set as C_1, C_2, ... C_n. It also has, as direct predecessors, all the common direct predecessors of the chance nodes being abstracted, and has, as direct successors, all the common direct successors of the set of chance nodes being abstracted.

As in the case of the extend operation, the refine operation can be performed flexibly in many different ways. Let $\Phi(C)$ represent the set of possible refined chance nodes for C. It is possible to apply the refine operation based on any non-empty subset of $\Phi(C)$. Since the integrated node C is an abstraction of the sub-nodes C_1, ... C_n, we can specify the relationships between the probability distribution for C and those for C_1, C_2, ... C_n depending on the domain. Our refine operation is similar to those used in state-space abstraction/refinement formalisms (Chang & Fung, 1990; Wellman et al., 1994) and node generation/specification formalisms (Lehner et al., 1994; Poh et al., 1994). In those works, the probability distribution for the more abstract or aggregated concept is often a weighted sum of the probabilities of the elementary concepts it constitutes.

We will assume that for each $C_i \in \Phi(C)$, there exists a number w_i $(0 \leq w_i \leq 1)$ that signifies the relative importance or likelihood of the sub-node C_i with respect to C such that

$$p(C|A) = \sum_{C_i \in \Phi(C)} w_i p(C_i|A), \tag{1}$$

where $\sum_i w_i = 1$. If only a subset of $\Phi(C)$ is chosen, then the set of chosen weights can be re-normalised. It should be noted that in the refine and abstract operations, each of the sub-nodes plays a different role in relation to the integrated node. If we let the relative weights indicate their respective contributions to the integrated node, then it is reasonable to approximate the probability of the integrated node by considering it as the weighted sum of the probabilities of the sub-nodes. Since the integrated node and the sub-nodes refer to the same distinction, we will assume that the integrated node and the sub-nodes have the same influence on their successor nodes, i.e.

$$p(B \mid C) = p(B \mid C_i), \qquad \forall\, i = 1, 2, \ldots n \tag{2}$$

We will show later in this section that such conditions are essential for maintaining the soundness of the operation.

3.1.3 *The deterministic-extend and deterministic-retract operations*

The fifth and sixth operations, *deterministic-extend* and *deterministic-retract*, are very useful for concept refinement and structure refinement. In particular, they improve the expressiveness of multilevel influence diagrams. As illustrated in Figure 4, the *deterministic-extend* operation takes a

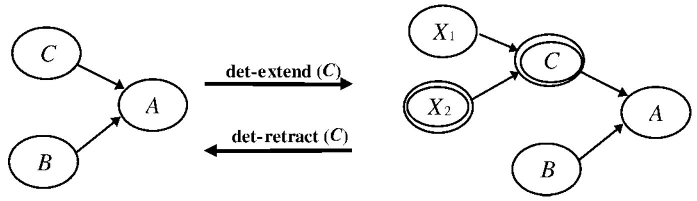

Figure 4 The deterministic-extend and deterministic-retract operations

chance node C, converts it into a deterministic node and adds a set of direct predecessors to it. The corresponding inverse operation *deterministic-retract*, on the other hand, takes a deterministic node with direct predecessors, converts it into a chance node and removes all its previous direct predecessors.

The difference between the deterministic-extend operation and the regular extend operation is that the operand node becomes deterministic after deterministic-extend whereas for the regular extend, the operand node remains probabilistic. Since a deterministic node is a special kind of chance node, the deterministic-extend and deterministic-retract operations are merely special cases of the extend and retract operation respectively.

3.1.4 The extend-function and retract-function operations
The final two operations are mainly used to extend or simplify the number of factors that affect a deterministic node. As shown in Figure 5, the *extend-function*, operation takes a deterministic node T and adds to it additional direct predecessors. The corresponding inverse operation *retract-function*, on the other hand, takes a deterministic node and removes one or more of its direct predecessors. In the example shown in Figure 5, node T is a function of C_1 and C_2 before the extend-function operation. After the operation, T becomes a function of C_1, C_2 and C_3. Conversely, the *retract-function* operation removes C_3 from the function T.

3.2 The soundness of the operations

The eight operations defined in the previous subsections basically perform model transformations. An important issue in model transformations is the validity of such operations. In general, the validity of any model-construction procedure depends on the soundness of its construction operations (Poh & Horvitz, 1993). Heckerman and Horvitz (1991) suggested that the soundness of a construction process should be characterized by the preservation of the joint-probability distribution of the variables involved across the process.

In multilevel influence diagrams, let M_1 and M_2 represent the influence diagrams before and after a construction operation, respectively, and let X_{M_1} and X_{M_2} denote the sets of chance nodes in M_1 and M_2 respectively. In general, an operation changes only certain part of a diagram while preserving the other part. Hence $X_{M_1} \cap X_{M_2} \neq \varnothing$. We will use the convention whereby probability distributions that are based on general domain knowledge are denoted by $p(\ )$, i.e. without any subscript, and probability distributions in a derived influence diagram M_i be denoted by $p_i(\ )$, i.e.

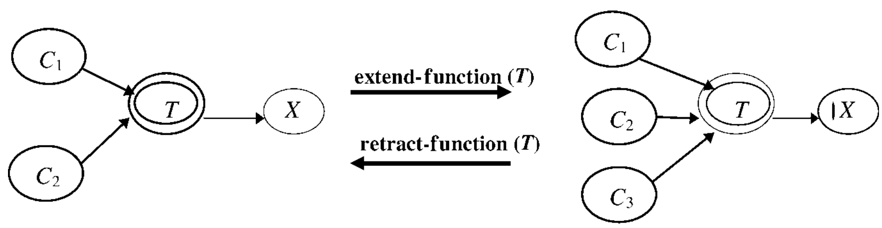

Figure 5 The extend-function and retract-function operations

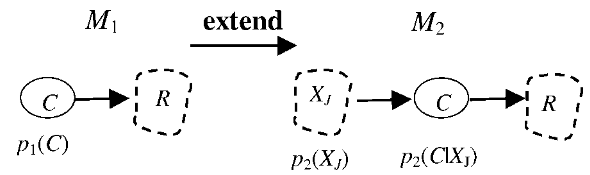

Figure 6 Execution of the extend operation on node C in a general influence diagram

with a subscript i. We describe an operation which transforms an influence diagram M_1 to another influence diagram M_2 as *sound* if, and only if,

$$p_1(X_{M_1} \cap X_{M_2}) = p_2(X_{M_1} \cap X_{M_2}). \tag{3}$$

We shall now consider the conditions under which our eight operations are sound.

3.2.1 The soundness of the extend and retract operations

Consider a general extend operation on a chance node C in a generic influence diagram M_1 as shown in Figure 6 where regions enclosed by dotted lines denote clusters or sets of nodes. Let $X_J \subseteq \pi(C)$ represent the set of possible direct predecessors to be included by the operation, and let M_2 be the resulting influence diagram. We shall show that if

$$p_1(C) = \sum_{X_J} p_2(X_J) p_2(C|X_J) \tag{4}$$

then the extend operation is sound.

As shown in Figure 6, let R denote the set of chance nodes in influence diagram M_1 that excludes C. In M_1, the joint probability distribution is $p_1(C, R)$ whereas in M_2, the joint probability distribution is $p_2(C, R)$. For the extend operation to be sound, we just need to show that $p_1(C, R) = p_2(C, R)$. In M_1, we can write $p_1(C, R) = p_1(R \mid C)\, p_1(C)$. However, in M_2, since X_J is conditionally independent of R given C, and suppose that Equation (4) holds, we have

$$\begin{aligned} p_2(C, R) &= \sum_{X_J} p_2(X_J) p_2(C|X_J) p_2(R|C) \\ &= p_2(R|C) \sum_{X_J} p_2(X_J) p_2(C|X_J) \\ &= p_1(R|C) p_1(C) \\ &= p_1(C, R). \end{aligned}$$

Hence Equation (4) provides the condition under which the extend operation is sound. It can also be shown that the same condition applies for the soundness of the retract operation which transforms M_2 to M_1.

3.2.2 The soundness of the refine and abstract operations

We consider next the refine and abstract operations. Figure 7 shows the application of the refine operation on an integrated chance node C in a general influence diagram M_1 where A is the set of common predecessors of C, and B is the set of common successors of C. The resulting influence

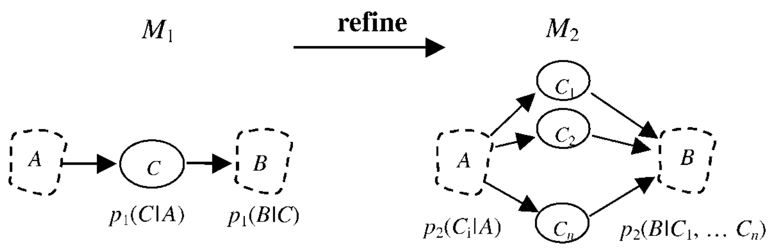

Figure 7 Execution of refine operation on node C in a general influence diagram

diagram is M_2. We shall show that if the refine operation replaces the chance node C in M_1 with a set of refined nodes $C_J \subseteq \Phi(C)$ to derive M_2 such that

$$p(C|A) = \sum_{C_i \in C_J} w_i p(C_i|A), \tag{5}$$

$$w'_i = \frac{w_i}{\sum_{C_i \in C_J} w_i}, \text{ and} \tag{6}$$

$$p(B \mid C) = p(B \mid C_i) \, \forall \, C_i \in C_J, \tag{7}$$

then the refine operation is sound.

First, we note that if Equations (5) to (7) are all satisfied, then

$$\begin{aligned} p(A, B, C) &= p(B|C)p(C|A)p(A) \\ &= p(B|C) \sum_{C_i \in C_J} w'_i p(C_i|A)p(A) \\ &= \sum_{C_i \in C_J} w'_i p(B|C_i)p(C_i|A)p(A) \end{aligned}$$

Hence

$$p(A, B, C) = \sum_{C_i \in C_J} w'_i p(A, B, C_i) \tag{8}$$

Next, we observe that the set of common variables across the transformation is B, and that $p_1(B) = \sum_{A, C} p_1(A, C, B)$. From Equation (8) it follows that

$$\begin{aligned} p_1(B) &= \sum_{A, C} \sum_{C_i \in C_J} w'_i p_2(A, C_i, B) \\ &= \sum_{C_i \in C_J} w'_i \sum_{A, C_i} p_2(A, C_i, B) \\ &= \sum_{C_i \in C_J} w'_i \sum_{A, C_i} \sum_{\Phi(C) \backslash C_i} p_2(A, C_i, \Phi(C) \backslash C_i, B) \\ &= \sum_{C_i \in C_J} w'_i \sum_{A, \Phi(C)} p_2(A, \Phi(C), B) \\ &= \sum_{A, C_1, C_2, \dots C_n} p_2(A, C_1, C_2, \dots C_n, B) \\ &= p_2(B) \end{aligned}$$

where $\sum_i w'_i = 1$.

The same result is also applicable for the soundness of the abstract operation which transforms M_2 to M_1.

3.2.3 The soundness of the deterministic-extend and deterministic-retract operations

We now consider the conditions under which the deterministic-extend and deterministic-retract operations are sound. Figure 8 shows an application of the of deterministic-extend operation on node C in a general influence diagram M_1, where R is the set of successors for C and $\pi(C)$ is the set of direct predecessors added to C. The resulting diagram is M_2. We shall show that if

$$p_1(C) = p_2(f(\pi(C))) \tag{9}$$

then the deterministic-extend and deterministic-retract operations are sound.

In Figure 8, let R denote the set of chance nodes in M_1 that exclude C and let $\pi(C)$ represent the set of all potentially possible direct predecessors for C. The set of common variables across the transformation is $\{C\} \cup R$. In diagram M_1, $p_1(C, R) = p_1(C)\,p_1(R \mid C)$. In diagram M_2, we introduce a function $f(\pi(C))$ for C and convert it to a deterministic node. If we view C as a special chance node,

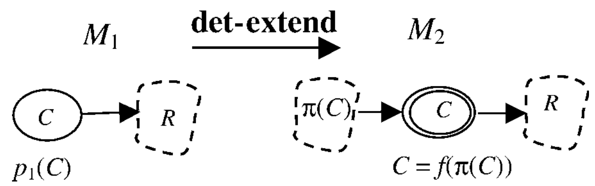

Figure 8 Execution of deterministic-extend operation on node C in a general influence diagram

then its probability is $p_2(C) = p_2(f(\pi(C)))$. But since $\pi(C)$ is conditionally independent of R given C, it follows that if Equation (9) is satisfied then

$$\begin{aligned} p_2(C,R) &= p_2(C)p_2(R|C) \\ &= p_2(f(\pi(C)))p_1(R|C) \\ &= p_1(C)p_1(R|C) \\ &= p_1(C,R). \end{aligned}$$

The same result is also applicable for the case of deterministic-retract, which transforms M_2 to M_1.

3.2.4 The soundness of the extend-function and retract-function operations

Finally, we consider the soundness of the extend-function and retract-function operations. Figure 9 shows an application of the extend-function operation on deterministic node T in a general influence diagram M_1. Let R be the set of successors of T, and C_J be the set of predecessor nodes of T in M_1. Suppose that after the operation, the set of nodes C_I is added as additional predecessors to T, where $C_I \cap C_J = \phi$ and $C_I \cup C_J = \pi(T)$, resulting in diagram M_2. We shall now show that this operation is sound if the added set of nodes C_J is conditionally independent of R given C_I.

We note from Figure 9 that the set of common variables in M_1 and M_2 which are affected by the operation is $R \cup \{T\}$. In diagram M_1, $p_1(T,R) = p_1(R)p_1(R|T) = p_1(R|T)\sum_T p_1(R,T)$.

Since $T = f(C_J)$, it follows that $p_1(T,R) = p_1(R|T)\sum_{C_J} p_1(R,C_J) = p(R|T)\sum_{C_J} p(R,C_J)$.

In diagram M_2, $p_2(T,R) = p_2(R)p_2(R|T) = p_2(R|T)\sum_T p_2(R,T)$.

Since $T = f(C_I, C_J)$, $C_I \cap C_J = \phi$, and $C_I \cup C_J = \pi(T)$, it follows that

$$\begin{aligned} p_2(T,R) &= p_2(R|T)\sum_{C_I C_J} p_1(R,C_I,C_J) \\ &= p_2(R|T)\sum_{C_I}\sum_{C_J} p_2(R,C_I,C_J) \\ &= p(R|T)\sum_{C_I}\sum_{C_J} p(R,C_I,C_J) \\ &= p(R|T)\sum_{C_I}\sum_{C_J} p(C_I,C_J)p(R|C_I,C_J). \end{aligned}$$

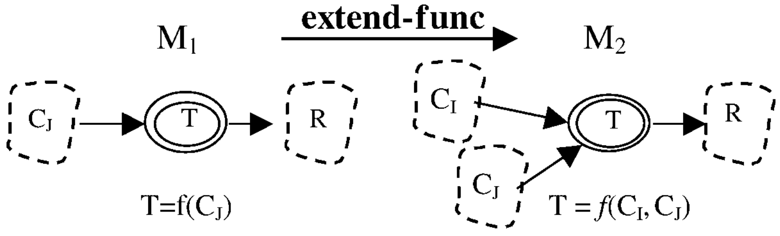

Figure 9 Execution of extend-function operation on node C in a general influence diagram

Since R is conditionally independent of C_I given C_J, it follows that

$$\begin{aligned} p_2(T,R) &= p(R|T)\sum_{C_I}\sum_{C_J} p(C_I,C_J)p(R|C_J) \\ &= p(R|T)\sum_{C_J} p(C_J)p(R|C_J) \\ &= p(R|T)\sum_{C_J} p(R,C_J). \end{aligned}$$

Hence $p_1(T,R) = p_2(T,R)$ and the extend-function operation is sound. It is also obvious that the retract-function operation is sound under the same conditions.

3.3 The multilevel structure

Given any problem domain, the multilevel influence diagrams representation induces a set of possible target influence diagrams that can be constructed. The use of multilevel influence diagrams as the knowledge-representation scheme for KBMC systems offers an expressive and concise way of representing any general domain knowledge. At the same time, it provides multilevel abstractions that help users manage the complexities of the modelling process. Furthermore, multilevel influence diagrams combine the formalism of classical influence diagrams with hierarchical modelling techniques, thereby allowing the construction of the decision model at the right level of detail, an important feature essential for real-world decision-making under time and resource constraints.

Given a problem domain, we may construct a multilevel influence diagram as our knowledge base through the operations we have defined. When faced with a particular problem instance, we can build a specific decision model at the most appropriate level of abstraction with all relevant factors. We have developed an algorithm for performing this task in real time with possible extension to time-critical conditions. In Section 6, we provide some real examples to illustrate how multilevel influence diagrams can be used in knowledge-based model construction.

3.4 The sufficiency of the operations

In this subsection, we examine the sufficiency of the eight operations under the multilevel influence diagram formalism from the structural point of view. We show that these eight operations are sufficient to construct a target influence diagram at the structural level for any practical domain. Let M_0 represent the top-level influence diagram, M_G the global influence diagram, M_i an intermediate-level influence diagram at some level i, and $\mathcal{C}$ the set of all possible chance nodes in the domain.

We say that a node $X \in \mathcal{C}$ is *reachable* if, and only if, X is included in a decision model which is generated by a finite sequence of operations performed on M_0. At the model level, we say that a decision model M_i is *constructible* if, and only if, M_i is the outcome of a finite sequence of operations performed on M_0. Finally, we describe a set of operations as *structurally sufficient* if, and only if, all the potential target decision models M_i are constructible. We shall first show that every node $X \in \mathcal{C}$ is reachable by our set of eight operations and then show that the set of eight operations is structurally sufficient.

First, we note that by definition any node included in M_0 is reachable. Next, let $\mathcal{C}_o$ represent the node set of M_0 and let $\mathcal{C}'_o$ represent the set $\mathcal{C}\backslash\mathcal{C}_o$. Then for $\forall X \in \mathcal{C}'_o$, $\exists Y \in \mathcal{C}'_o$ which is linked to X by a path. M_0 is the most abstract decision model which includes the value node, the decision nodes and all the essential chance nodes used to describe a decision problem. So if any node, i.e. any factor, in this domain plays a role in the decision situation, it should either have already been included in M_0 or it will have influence in some way on at least one node in M_0, which is assured when we assess the node set in the domain knowledge base. For example, consider a general influence diagram in Figure 10 which includes all possible topological connectivities for an influence diagram. Suppose that in this case M_0 comprises V_0, D_{11}, C_{12} and D_{21}, then any other nodes will be connected with these nodes in some way. Using the operations we have defined as indicated in the diagram, a node in the

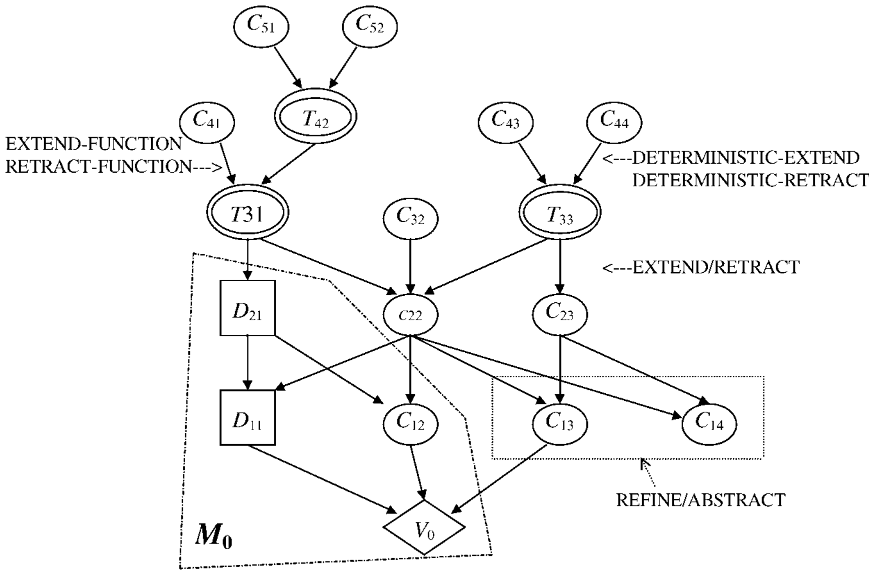

Figure 10 An example general influence diagram

diagram is always related to another via some operation. Thus, starting from nodes in M_0 we can derive a finite sequence of operations to reach any other node not already included in M_0. Since the influence diagram enumerates all kinds of nodes and relationships, we conclude that all nodes $X \in \mathcal{C}$ are reachable using the eight operations.

To show structural sufficiency, we let $\mathcal{C}_i$ represent the node set of M_i, and $\mathcal{C}_G$ represent the node set of M_G. Since each of $X_i \in \mathcal{C}_i$ is reachable, we can always find a finite sequence of operations to reach each X_i. Suppose that M_i^* is the outcome decision model which includes all the nodes tracked by these operations, then $M_i^* \supseteq M_i$. If we remove all the nodes not included in M_i via the corresponding inverse operations then any potential target decision model M_i can be successfully constructed. Hence the set of eight operations is structurally sufficient.

Based on the above analysis, we conclude that given any general influence diagram which represents a particular domain or problem, once we have assessed the top-level influence diagram, we can construct the multilevel influence diagram structure between them using only the eight operations. Subsequently, we may get a set of decision models at different levels of abstraction or details based on these operations.

4 Model construction using multilevel influence diagrams

4.1 The domain knowledge base

The domain knowledge base required to support model construction using multilevel influence diagrams comprises the following components:

- A set of all the *chance nodes* $\mathcal{C}$.
  For each chance node $C \in \mathcal{C}$, let $\varphi(C)$ be the set of possible outcomes of C, $\Omega(C)$ be the set of operations which are applicable to C, $\pi(C)$ be the set of all the direct predecessors of C, $p(C|\pi(C))$ be the conditional probabilities of C given its predecessors, and $\forall x \in \pi(C)$, $p(C|x)$ be the conditional probability for C given x.
- A set of all the *decision nodes* $\mathcal{D}$.
  For each decision node $D \in \mathcal{D}$, let $\alpha(D)$ be the set of alternatives for D.
- A *value node* V in the form of a utility function.
- A set of *directed arcs* $\mathcal{A} \subseteq (\mathcal{C} \cup \mathcal{D}) \times (\mathcal{C} \cup \mathcal{D} \cup \{V\})$ such that $\forall x \in (\mathcal{C} \cup \mathcal{D})$ and $\forall y \in (\mathcal{C} \cup \mathcal{D} \cup \{V\})$, $(x, y) \in \mathcal{A}$ if, and only if, there exists an arc from node x to node y.

```
MODEL CONSTRUCTION ALGORITHM
• Input:  Decision Problem
          Domain Knowledge

• Output: Decision Model

• Procedure:
1.  Select problem domain DOM.
2.  Load the knowledge base for domain DOM.
3.  Display the top-level influence diagram, M0, of domain
    DOM.
4.  Let the current model M = M0.
5.  While the user is not satisfied with model M do
6.     Select a node X in M.
7.     Display the set of valid operations, Ω(X) applicable to
          node X.
8.     Select an operation ω ∈ Ω (X).
9.     Perform operation ω on M producing a new model ω(M).
10.    Let M = ω(M).
11. End While
12. Output target model M.
```

Figure 11 An algorithm for interactive model construction

- A *top-level influence diagram* $\boldsymbol{M}_0$.
  $\boldsymbol{M}_0$ includes the decision nodes in $\mathcal{D}$, an initial subset of chance nodes of $\mathcal{C}$, and the value node V, all of which are connected according to $\mathcal{A}$.

4.2 The decision model construction algorithm

We propose a model-construction algorithm using multilevel influence diagrams as shown in Figure 11. This algorithm is decision-driven according to the classification of Wellman et al. (1992). The algorithm provides an interactive tool for a user to explore and add relevant nodes in order to arrive at a final decision model. Starting with the top-level influence diagram, which includes the decision to be made, the user may select any node in the current influence diagram to operate on. For any selected node, the user is always guided by a list of operations that are applicable to that node. The model is continuously updated during the modelling process until the user is satisfied.

A unique feature of our model-construction approach compared with many others (for example Breese, 1992; Goldman & Charniak, 1990; Holtzman, 1989) is that our knowledge-base representation and decision-model representation are based on the same formalism. This is similar to other model-construction procedures, such as Mahoney and Laskey (1998), which involve "picking out" the appropriate pieces of components already in the knowledge base and "assembling" them to form the situation-specific target model. This method of pick and assemble provides a highly flexible and effective way of model construction.

5 System implementation

In this section we describe the implementation of the multilevel influence diagrams (MLID) modelling system.

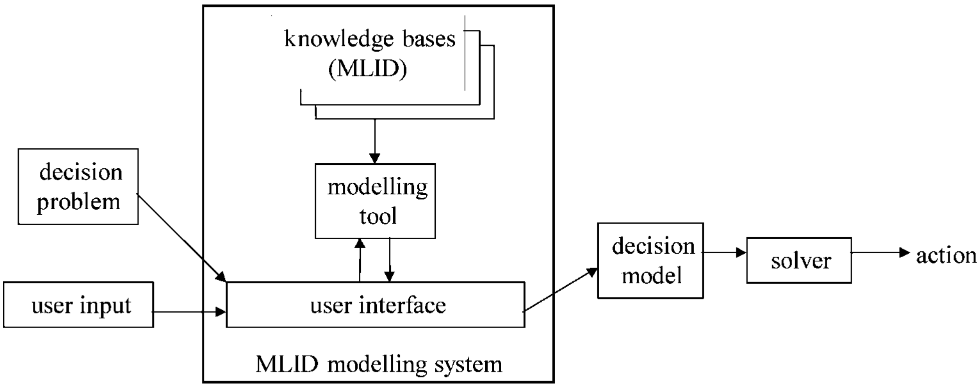

Figure 12 The MLID modelling system architecture

5.1 System architecture

The architecture of our decision modelling system is shown in Figure 12. As seen from Figure 12, the MLID modelling system works as a bridge connecting real decision problems with model-analysis tools. A prototype of the system has been developed under the Microsoft Windows environment using MS Visual C+ +. The system displays the decision model that is being constructed as an influence diagram in the main window. A user can select and operate on any node in the current model by simply clicking on it, and a dialog toolbox listing all the valid operations associated with that node will be displayed. The user can either select a valid operation or cancel his or her selection and select another node. The user is always guided by the system throughout the entire modelling session.

5.2 Elicitation of knowledge bases

A major effort in building the system is the elicitation of the domain knowledge bases. For each domain that is supported, a knowledge base with components specified in subsection 4.1 is required. The components needed comprise mainly (1) the set the uncertain variables, their outcomes and their probabilistic relationships in the form of conditional probabilities; (2) the decision alternatives; (3) the value model; and (4) the operators that are applicable. The elicitation process is largely similar to that of building decision models using regular influence diagrams. Hence regular tools for building graphical decision models are applicable. These include techniques for structuring values and objectives from value-focused thinking (Keeney, 1992), probability assessment protocols such as the Stanford/SRI protocol (Spetzler & Stael von Holstein, 1972), the Wallsten/EPA protocol (Wallsten & Whitfield, 1980) and Morgan and Henrion's protocol (1990).

Based on our experience of eliciting the knowledge bases for the prototype system, we have found the following procedures to be useful. Start the knowledge elicitation with the construction of an influence diagram that is sufficiently detailed. Very often such a model already exists as a result of a previous one-off decision analysis and it may have been constructed via direct elicitation from experts, or through automated machine learning from data, or a combination of both. Next, proceed to ask the experts if the any parts of model can be simplified, such as via abstraction of some of the nodes. If so, which sets of nodes can be abstracted and the applicable operations are identified. This can done recursively until the top-level influence diagram is reached. Similarly, we can also ask the experts if any parts of the influence diagram may be refined and the applicable operations are identified. The process can be repeated until the global influence diagram is reached. During the elicitation process, the experts are also asked to provide the weightings for the abstract and refine operations.

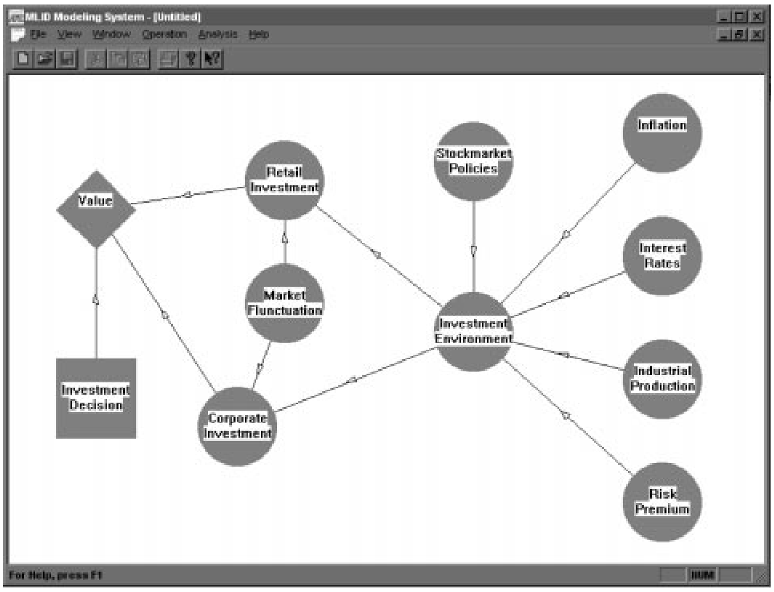

Figure 13 The global influence diagram for a stock market investment problem

5.3 An example model-construction session

The effectiveness of the MLID modelling system in performing knowledge-based model construction has been tested with different problems drawn from different domains. We provide here an example of the modelling process performed by the MLID modelling system constructing an investment decision model. The global influence diagram for the investment decision problem is shown in Figure 13 (Chen, 1996). This diagram represents the most detailed decision model we could construct and it reflects the entire knowledge that we have on this financial problem.

At the start of the modelling session, a window similar to that shown in Figure 14 will be displayed. This is the top-level influence diagram from the knowledge base. This model is the most abstract decision model from the knowledge base and it has a structure of the basic risky problem (Clemen, 1996), comprising a single decision node and a single uncertain node. We will show how a

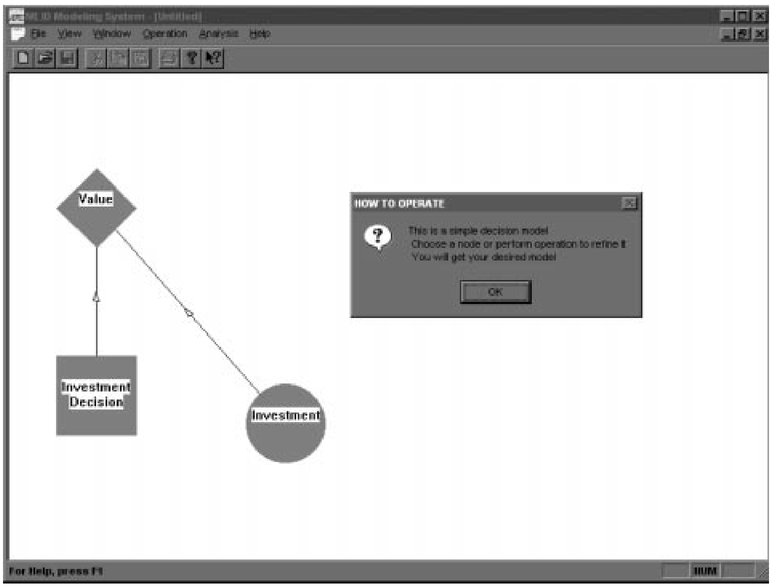

Figure 14 Top-level influence diagram for the stock market investment problem

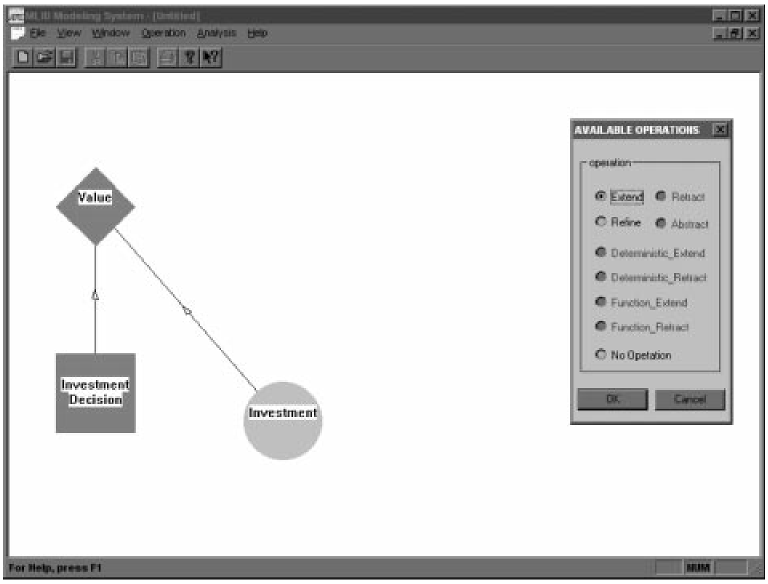

Figure 15 Choosing the extend operation for the "investment" node

suite of influence diagrams between the lowest and highest levels may be generated by the user as the modelling process proceeds.

Suppose that we choose the "investment" node in the current model by clicking on it. An operation dialog box will immediately appear to indicate all the available valid operations. Using any of these operations, we can transform the current model to another one which may be either more detailed or more abstract relative to the current model. In this particular case, we select the EXTEND operation as shown in Figure 15. By choosing the EXTEND operation, we have the intention to add direct predecessors for "investment" node so as to include more details into the current model. For the "investment" node, the set of potential predecessors includes "market fluctuation" and "investment environment". In Figure 16, we show the selection of both nodes for inclusion in the extended model. The result of the operation is the extended decision model as shown in Figure 17. Note that two new nodes have been included in the model.

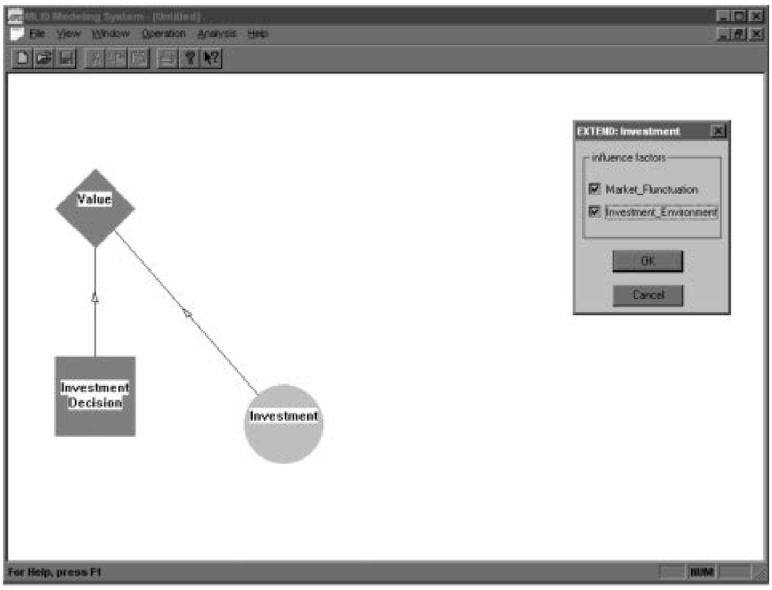

Figure 16 User selects the nodes to be added

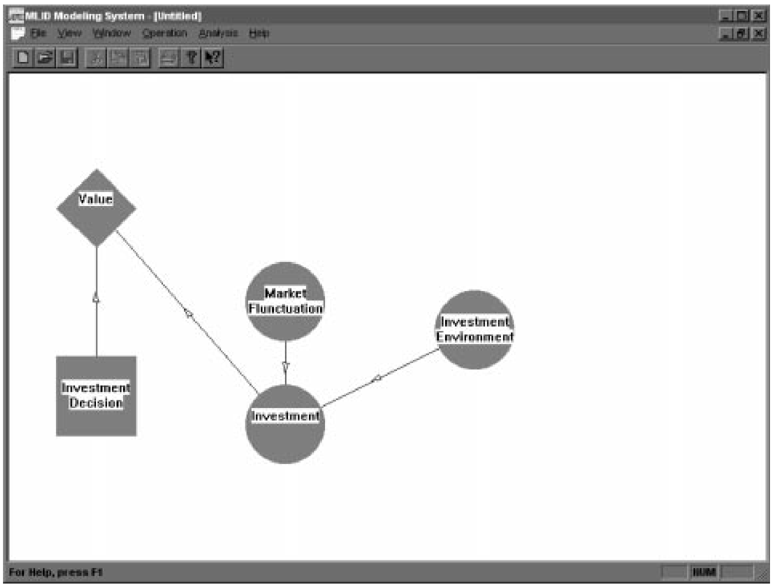

Figure 17 The model after the extend operation

We may continue to modify the diagram. Suppose that, after a while, we have reached the decision model with eight nodes shown in Figure 18. Suppose that we now click on the ‘‘general economic indicators’’ node, and then select the REFINE operation. Our intention here is to replace the integrate node ‘‘general economic indicators’’ with a group of more refined nodes. A dialog box shown in Figure 19 appears, to indicate to the user the set of available refined nodes that may be selected. The available nodes are ‘‘inflation’’, ‘‘interest rates’’, ‘‘industrial production’’ and ‘‘risk premium’’. As indicated in Figure 19, we choose ‘‘interest rates’’ and ‘‘risk premium’’ for inclusion, and the new decision model is as shown in Figure 20. Compared with Figure 19, the new model in Figure 20 represents the situation with more detailed knowledge and hence provides the user with a more detailed representation of the decision situation. Finally, assuming that the user is satisfied with the model constructed, he or she can terminate the modelling process. The constructed model

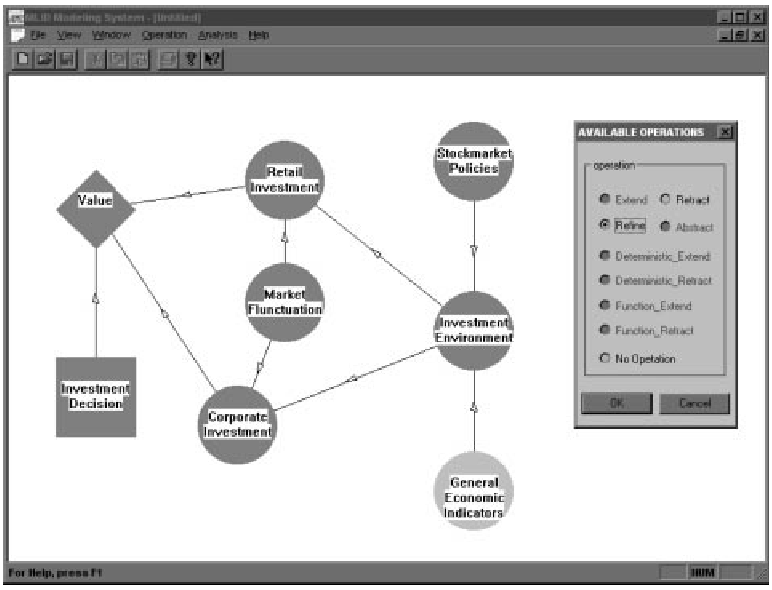

Figure 18 User chooses the refine operation for the ‘‘general economic indicators’’ node

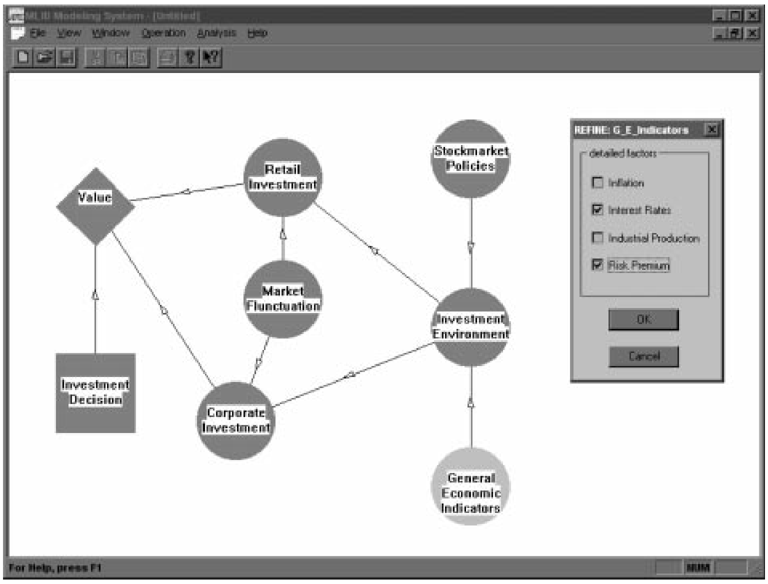

Figure 19 User selects the refined nodes

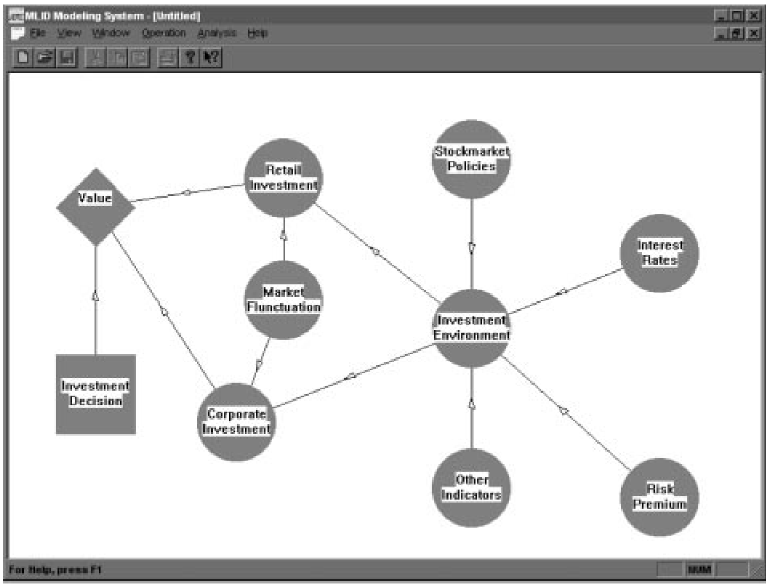

Figure 20 The model after the refine operation

can now be saved and then submitted to any standard influence diagram solver for the optimal decision to be computed.

The above example has demonstrated the ease with which knowledge-based decision-model construction may be accomplished using multilevel influence diagrams. It has also shown how a non-domain expert may use the system to construct situation-specific decision models. A user may start from the most abstract level of the problem and, using the tools provided by the MLID modelling system, navigate his or her way to a desired model. At any time, the user may also "backtrack" and take a different model-construction path.

6 Real-world applications

In this section, we look at two more real-world examples to further illustrate the use of multilevel

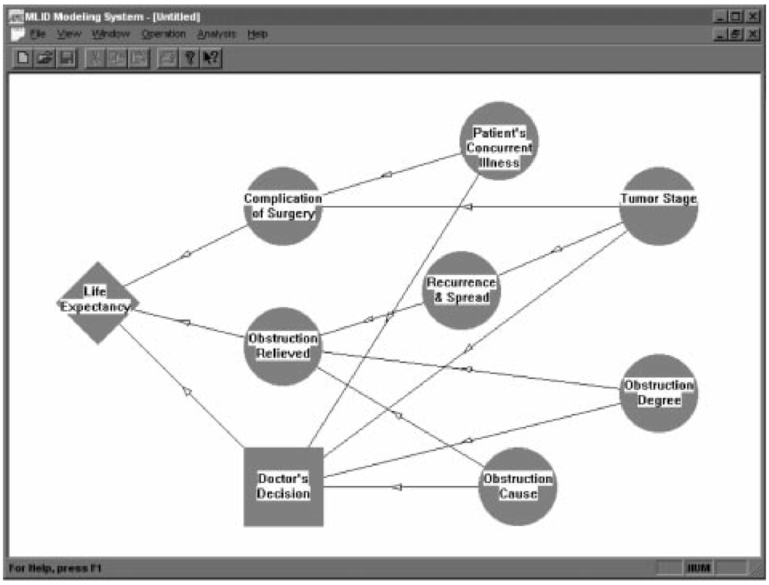

Figure 21 The global influence diagram for a medical decision problem

influence diagrams in knowledge-based decision-model construction. We start off with a small example problem from the medical domain before proceeding to a more complex example problem from the energy planning domain.

6.1 A medical decision problem

The example medical problem we are presenting here differs from the example in the previous section in that the current example now includes information arcs. Figure 21 shows the global influence diagram for the current medical problem (Lin, 1996).

As before, we start out with the top-level influence diagram for the problem as shown in Figure 22. At this highest level of abstraction, the doctor performing the modelling is only concerned with two major uncertainties, namely "complication of surgery" and "obstruction relieved". We apply

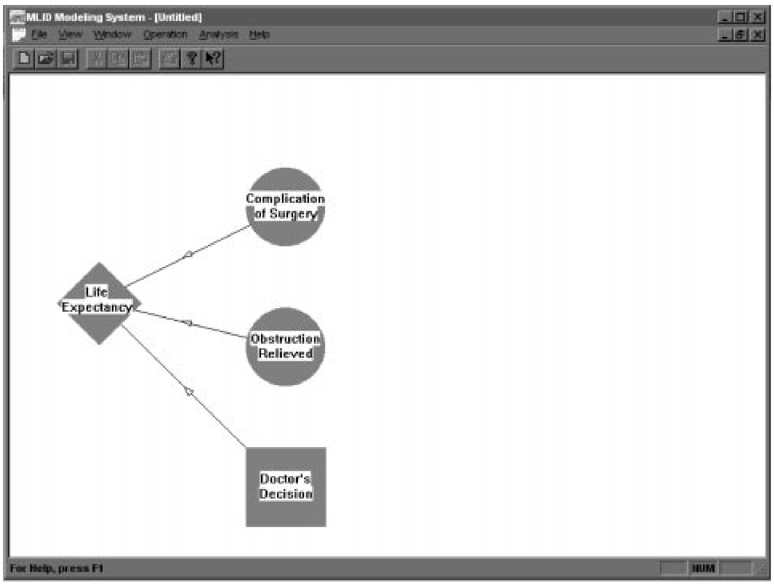

Figure 22 The top-level influence diagram for the medical decision problem

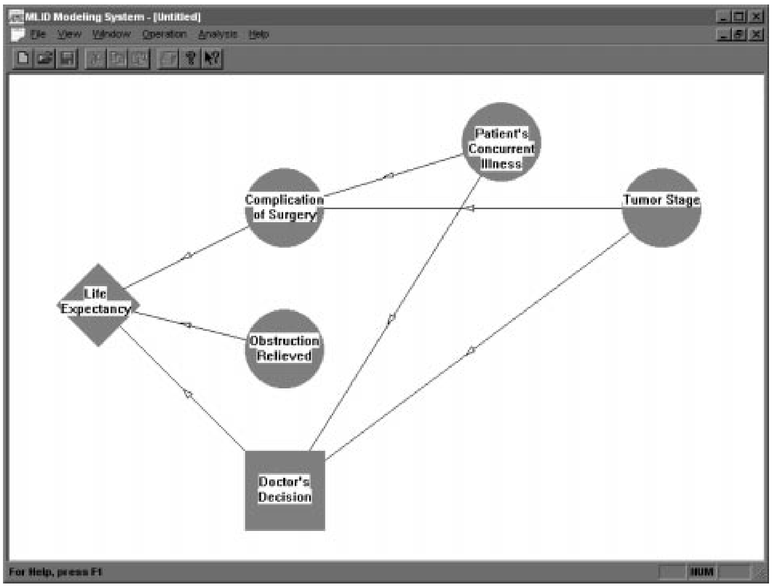

Figure 23 A slightly more detailed-level influence diagram for the medical decision problem

the extend operation twice, first on the ‘‘complication of surgery’’ node and then on the ‘‘obstruction relieved’’ node, to obtain the influence diagrams as shown in Figures 23 and 24 respectively. Finally, by applying the extend operation on the ‘‘recurrence & spread’’ node, the global influence diagram in Figure 21 is obtained.

In this example, information arcs were added, together with the appearance of new nodes in the diagram. For instance, in the global influence diagram, there is an information arc from the chance node ‘‘obstruction cause’’ to the decision node ‘‘doctor's decision’’. During the construction process, when ‘‘obstruction cause’’ was added into the model, we include the required ‘‘no-forget’’ information arcs automatically, thus preserving the state of information availability in the new influence diagram.

This example also illustrates the ability of the system to avoid node duplication. For instance, when the extend operation was performed on the ‘‘complication of surgery’’ node, the ‘‘tumor

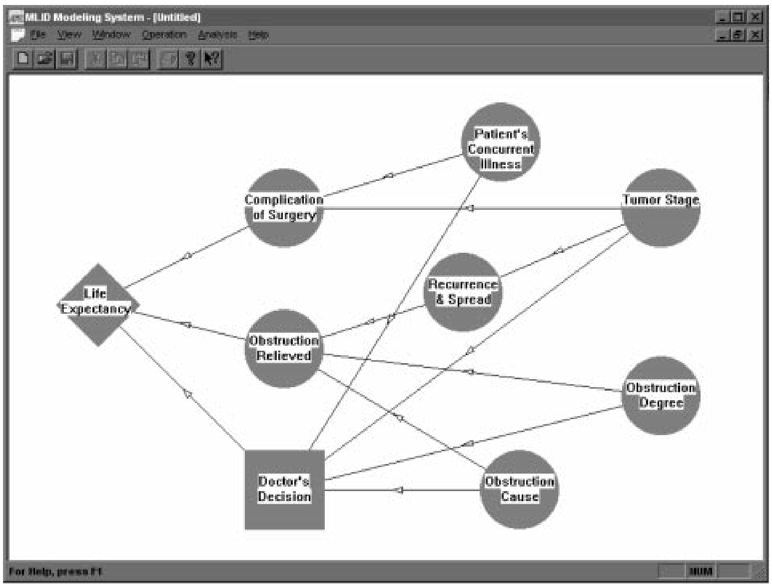

Figure 24 A more detailed-level influence diagram for the medical decision problem

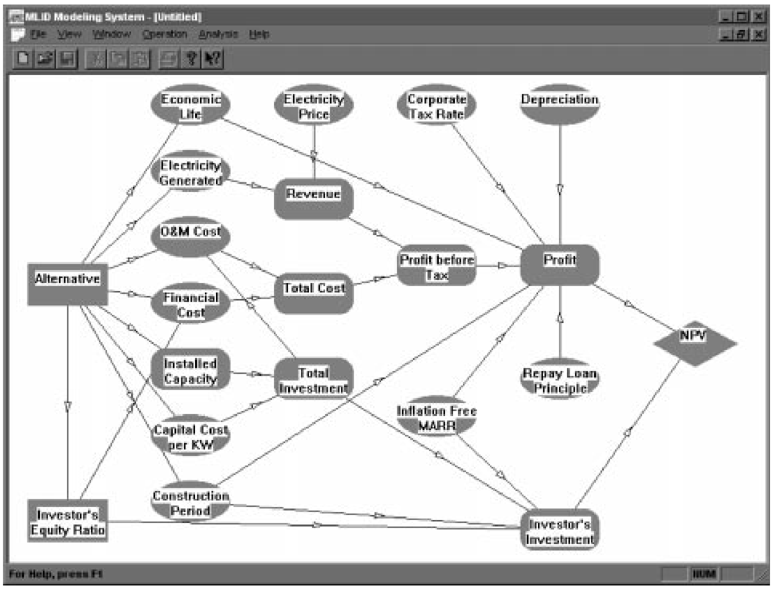

Figure 25 The global influence diagram for an electricity generation plant investment problem

stage'' node was generated. However, when the extend operation was subsequently performed on the ''recurrence & spread'' node, the ''tumor stage'' node was generated again but it was already in the target diagram and would not be duplicated; all that was needed was the addition of the necessary arcs.

6.2 An electricity generation plant investment problem

We shall now show a larger example, which includes the use of deterministic-extend and extend-function operations. The example will involve two decision nodes. This problem is based on the electricity generation plant investment decision problem developed by Huang et al. (1995), whose global influence diagram is shown in Figure 25.

Figure 26 shows the top-level decision model where only a minimal number of major factors are included. By applying the extend-function operation on the deterministic nodes ''profit'' and

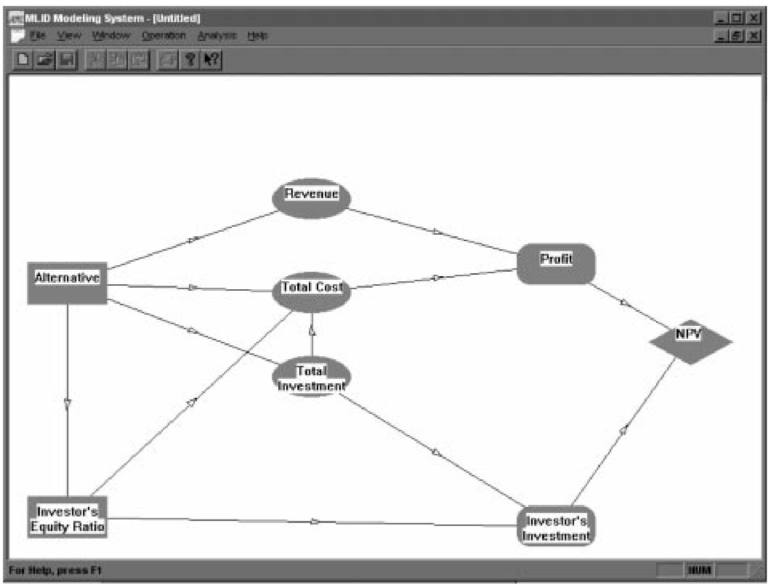

Figure 26 The top-level influence diagram for the energy generation investment problem

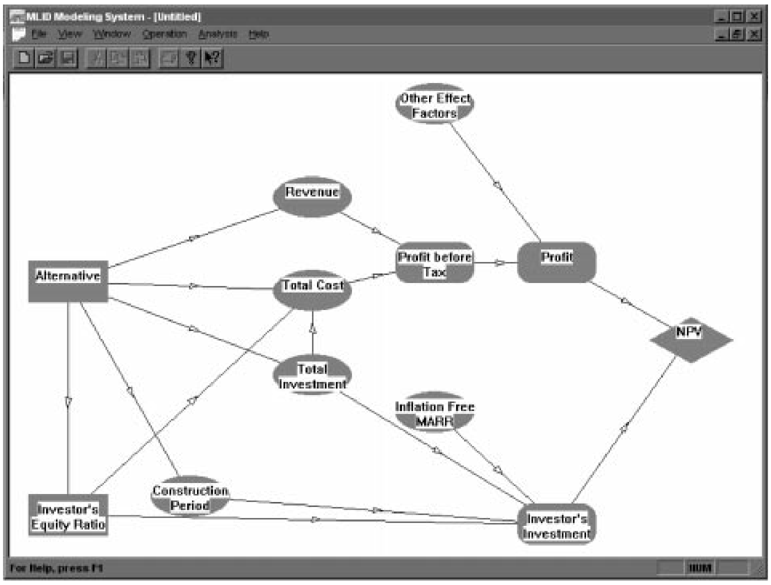

Figure 27 A slightly more detailed-level influence diagram for the energy generation investment problem

"investor's investment", we obtain the influence diagram shown in Figure 27. Next, by applying the deterministic-extend operation on the chance nodes "revenue", "total cost" and "total investment", we obtain the influence diagram shown in Figure 28. Finally, by applying the refine operation on the chance node "other effect factors", we arrive at the global influence diagram as shown in Figure 25.

7 Providing evaluation of model-construction operations

In the proposed model-construction algorithm, the user is free to choose any valid operation on any node in the model being constructed. This has the advantage of providing the user with the greatest flexibility and freedom to go about building the required model. However, for large and complex domains, the number of possible choices may be very large. In such situations, the user could benefit by being informed of the relative merit of each operation that can be chosen. A major benefit of such a scheme is that it will certainly help to reduce the time required to reach the desired model by a

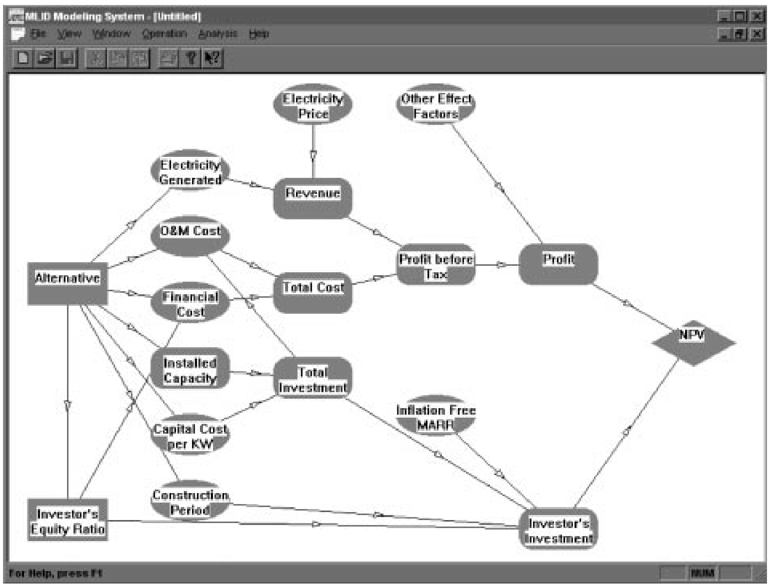

Figure 28 A more detailed-level influence diagram for the energy investment problem

significant amount. It can also be used to explicitly consider the trade-off between the potential benefits of using more detailed or refined concepts and the additional computational costs that may be incurred in the final model-solution process. Such trade-offs during the incremental model construction process have been considered in many other model-construction approaches (Goldman & Breese, 1992; Poh, et al., 1994).

A possible approach is to model a construction guidance system that will direct the user to a set of the most promising expansions or contractions, based on the idea of expected value of refinement (EVR) developed in a related work (Poh & Horvitz, 1993). EVR is based on the idea of explicitly comparing the trade-off between a possible incremental model improvement step and the associated improvement in solution quality, possibly taking into account the computational cost for the model if it is significant. In our system, whenever a node is selected by the user, the system can display a list of the most promising and applicable operations, in order of EVR. The user can then pick one out of the list for immediate execution or may, by default, always choose the operation with the maximum EVR. By using this EVR computation facility, we can provide a very effective system that guides the user to improve the target model in directions that lead to models that generate the highest-quality recommendations. It should be noted that while we may provide guidance on the most promising direction of model improvement in our system, the user is still very much in control of the process. In an interactive system, such EVR-directed guidance can certainly shorten the time needed for model construction, which is often the most significant contributor to the total cost. Finally, it should be noted that the incorporation of EVR into our system provides a promising way to realize a fully automated intelligent model construction for autonomous agents.

8 Summary and conclusion

In this paper, we have introduced and defined the multilevel influence diagram representation for knowledge-based model construction. The formalism includes a series of operations that relate an influence diagram at one level to those at another level. Each influence diagram that can be generated represents a possible target model that a user may construct based on his or her specifications and knowledge and on the situation at hand. We have also demonstrated that the eight operations are sufficient at the structural level to construct any target influence diagram in any practical situation. We have also specified conditions under which the operations are considered sound.

We have developed a simple algorithm to facilitate the interactive construction of decision models in real time. We have implemented the multilevel influence diagram modelling system under the Microsoft Windows environment and have tested it on a number of real-world problems. Our experience has indicated that multilevel influence diagrams representation is a promising way to perform knowledge-based model construction. We have also discussed the possibility of adding a facility that can provide the users with the value of each potential model-construction step so as to guide the user towards the best target decision model. Finally, we are exploring the possibility of applying and extending the system to support the construction of dynamic decision models for medical applications such as the planning of follow-up actions for cancer patients at a local hospital.

References

Breese, JS, 1987. "Knowledge representation and inference in intelligent decision systems", Ph.D. thesis, Dept of EES, Stanford University.

Breese, JS, 1992, "Construction of belief and decision networks" *Computational Intelligence* **8**(4) 624–647.

Breese, JS, Goldman, RP and Wellman, MP, 1994. "Introduction to the special section on knowledge-based construction of probabilistic and decision models" *IEEE Transactions on System, Man and Cybernetics* **24**(11) 1577–1579.

Chang, K and Fung, R, 1990. "Refinement and coarsening of Bayesian networks" *Proceedings of the Sixth Conference on Uncertainty in Artificial Intelligence* pp 475–482.

Chen, XL, 1996. "A normative portfolio decision system" MSc thesis, Dept of Information Systems & Computer Science, National University of Singapore.

Clemen, RT, 1996. *Making Hard Decisions* International Thomson Publishing.

Egar, JW and Musen, MA, 1993. "Graph-grammar assistance for automated generation of influence diagrams" *Proceedings of the Ninth Conference on Uncertainty in Artificial Intelligence* pp 235–241.

Goldman, RP and Breese, JS, 1992. "Integrating model construction and evaluation" *Proceedings of the Eighth Conference on Uncertainty in Artificial Intelligence* pp 104–111.

Goldman, RP and Charniak, E, 1990. "Dynamic construction of belief networks" *Proceedings of the Sixth Conference on Uncertainty in Artificial Intelligence* pp 90–97.

Heckerman, DE, 1991. *Probabilistic Similarity Networks* MIT Press.

Heckerman, DE and Horvitz, EJ, 1991. "Problem formulation as the reduction of a decision model" *Uncertainty in Artificial Intelligence* **6** 159–170.

Holtzman, S, 1989. *Intelligent Decision Systems* Addison-Wesley.

Howard, RA and Matheson, JE (eds), 1981. *Readings on the Principles and Applications of Decision Analysis Volume 2* Strategic Decision Group.

Huang, JP, Poh, KL and Ang, BW, 1995. "Decision analytic model for electricity projects investment evaluation" Working paper, Dept of Industrial & Systems Engineering, National University of Singapore.

Keeney, RL, 1992. *Value-Focused Thinking: A Path to Creative Decisionmaking* Harvard University Press.

Lehner, PE, Elsaesser, C and Musman SA, 1994. "Constructing belief networks to evaluate plans" *Proceedings of the Tenth Conference on Uncertainty in Artificial Intelligence* pp 416–421.

Leong, TY, 1991. "Representation requirements for supporting decision model formulation" *Proceedings of the Seventh Conference on Uncertainty in Artificial Intelligence* pp 212–219.

Lin, Z, 1996. Working research paper. Medical Expert System Research Group, National University of Singapore.

Liu, CL and Wellman, MP, 1995. "On abstraction approximations in Bayesian networks" Working notes of the IJCAI-95 Workshop on Anytime Algorithm and Deliberate Scheduling.

Mahoney, SM and Laskey, KB, 1998. "Constructing situation specific belief networks", *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence* pp 370–378.

Miller, AC, III, Miley, W, Merkhofer, MW and Howard, RA, 1978. "Development of automated decision aids for decision analysis", Research Report, SRI International.

Morgan, MG and Henrion, M, 1990. *Uncertainty: A Guide to Dealing with Uncertainty in Qualitative Risk and Policy Analysis* Cambridge University Press.

Owen, D, 1978. "The concept of influence and its use in structuring complex decision problems", Ph.D. Thesis, Department of Engineering-Economic Systems, Stanford University.

Poh, KL and Fehling, MR, 1993. "Probabilistic conceptual network: a brief representation scheme for utility-based categorization" *Proceedings of the Ninth Conference on Uncertainty in Artificial Intelligence* pp 166–173.

Poh, KL, Fehling, MR and Horvitz, EJ, 1994. "Dynamic construction and refinement of utility-based categorization models" *IEEE Transactions on System, Man and Cybernetics* **24**(11) 1653–1662.

Poh, KL and Horvitz, EJ, 1993. "Reasoning about the value of decision-model refinement: methods and application" *Proceedings of the Ninth Conference on Uncertainty in Artificial Intelligence* pp 174–182.

Provan, GM, 1991. "Dynamic network updating techniques for diagnostic reasoning" *Proceedings of the Seventh Conference on Uncertainty in Artificial Intelligence* pp 279–286.

Provan, GM, 1994. "Tradeoffs in knowledge-based construction of probabilistic models" *IEEE Transactions on System, Man and Cybernetics* **24**(11) 1580–1592.

Shachter, RD, 1988. "Probabilistic inference and influence diagrams" *Operation Research* **36**(4) 589–604.

Spetzler, CS and Stael von Holstein, CAS, 1972. "Probability encoding in decision analysis" *Proceedings of the ORSA-TIMS-AIEE 1972 Joint National Meeting*.

Wallsten, TS and Budescu, DV, 1980. "Encoding subjective probabilities: A psychological and psychometric review" Report to the Strategies and Standards Division of the US Environmental Protection Agency.

Wellman, MP, 1988. "Formulation of tradeoffs in planning under uncertainty" Ph.D. thesis, Dept of EECS, MIT.

Wellman, MP, Breese, JS and Goldman, RP, 1992. "From knowledge bases to decision models" *The Knowledge Engineering Review* **7**(1) 35–53.

Wellman, MP and Liu, CL, 1994. "State-space abstraction for anytime evaluation of probabilistic networks" *Proceedings of the Tenth Conference on Uncertainty in Artificial Intelligence* pp 567–574.

Yuan, ST, 1993. "Knowledge-based decision model construction for hierarchical diagnosis: a preliminary report" *Proceedings of the Ninth Conference on Uncertainty in Artificial Intelligence* pp 274–281.