A methodology for acquiring qualitative knowledge for probabilistic graphical models

Kjærulff, Uffe Bro; Madsen, Anders L.

Published in:
Proceedings of the 10th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems

Publication date:
2004

Document Version
Early version, also known as pre-print

A Methodology for Acquiring Qualitative Knowledge for Probabilistic Graphical Models

Uffe B. Kjærulff & Anders L. Madsen
HUGIN Expert A/S, Aalborg, Denmark
{uk,alm}@hugin.com

Abstract

We present a practical and general methodology that simplifies the task of acquiring and formulating qualitative knowledge for constructing probabilistic graphical models (PGMs). The methodology efficiently captures and communicates expert knowledge, and has significantly eased the model development process for three real-world problems in the domain of robotics.

Keywords: Probabilistic graphical model, Bayesian network, influence diagram, knowledge acquisition

1 Introduction

Probabilistic graphical models (PGMs) is a powerful paradigm for reasoning and decision making under uncertainty [3, 8]. Unfortunately, however, the construction of a PGM can be a labour intensive task with respect to both knowledge acquisition and formulation. This paper presents a practical and general methodology that simplifies the task.

The methodology has been developed as part of the joint European research and development project ADVOCATE II [1], in which PGMs were developed for advanced onboard diagnosis of (semi-)autonomous vehicles. The knowledge acquisition process associated with building these models involved knowledge engineers and domain experts located in four different countries with limited possibilities for face-to-face meetings. Therefore, a knowledge acquisition scheme had to be developed that did not rely on familiarity with the terminology of PGMs.

The scheme is based on building a problem hierarchy for an overall problem. The problems (or causes) of the hierarchy relate to the states of the different parts of a vehicle and its environment. The methodology is described from a practical point of view and is illustrated on one of the three real-world problem domains on which it has been used. Experience has shown that the methodology provides an efficient means of capturing and communicating expert knowledge in the domain of robotics. The use of the methodology has significantly eased the model construction process.

The literature contains a number of methods and ideas to help acquire the necessary information for building PGMs. Exploiting various forms of independence properties of the problem domain, Heckerman's similarity networks [2] support construction of independent Bayesian networks for subsets of a domain. A valid network for the entire domain can then be constructed from the individual networks. The method suggested by Skaanning [9] has some similarities with our method. There is, however, an important distinction: His method is focusing specifically on acquiring cause-effect knowledge for decision-theoretic troubleshooting models under the single-fault assumption, effectively reducing the models to the simple-Bayes type. Möbus & Schröder [7] describe a method for qualitative and quantitative learning of Bayesian networks from knowledge acquired from a domain expert. The qualitative knowledge is acquired via composition of constrained natural language sentences in a linguistic model editor, whereas the quantitative knowledge is compiled from fuzzy, qualitative statements about stochastic relations. In his report on the knowledge acquisition process for a particular medical expert system, Lucas [6] emphasizes the use of specific domain models in guiding the process, and he presents various other techniques that may be helpful in designing a Bayesian network. Lacave and Díez [5] reports on the process of constructing PROSTANET, a Bayesian network for diagnosing prostate cancer. They present hints to facilitate acquisition of probabilities. Acquisition of structure was performed through oral interviews of domain experts.

2 Probabilistic Graphical Models

A PGM is a formal representation of the qualitative and quantitative knowledge about relations among variables of a problem domain.

Our knowledge acquisition scheme focuses on acquiring the qualitative (i.e., structural) knowledge for the two most popular PGMs, namely Bayesian networks and influence diagrams. By "PGM" we shall henceforth refer to a Bayesian network or an influence diagram over discrete variables, where each variable, V, has a finite set of exhaustive and mutually exclusive states, $\mathrm{Sp}(V) = (v_1, \ldots, v_n)$.

An *acyclic, directed graph* (DAG) is a pair $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is a finite set of *vertices* (or *nodes*) and $\mathcal{E}$ is a set of ordered pairs $(U, V) \subseteq \mathcal{V}$ of *directed edges*, where U is a *parent* of V and V a *child* of U. The parents and children of V are denoted $\mathrm{pa}(V)$ and $\mathrm{ch}(V)$, respectively. If $|\mathrm{pa}(V)| = 1$ for each $V \in \mathcal{V} \setminus \{R\}$ and $|\mathrm{pa}(R)| = 0$, then $\mathcal{G}$ is a *tree* and R is its *root*.

A PGM is a triple $(\mathcal{G}, \mathcal{P}, \mathcal{U})$, where $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is a DAG that represents the qualitative knowledge and $(\mathcal{P}, U)$ are sets of conditional probability distributions (CPDs) and utility functions, respectively, that represent the quantitative knowledge. Each vertex $V \in \mathcal{V}$ represents a discrete random (or chance) variable, a discrete decision variable, or a utility function. We shall refer interchangeably to vertices (or nodes) and variables or utility functions. To each chance variable, V, is associated exactly one CPD, $P(V \,|\, \mathrm{pa}(V)) \in \mathcal{P}$, which consists of a set of conditional probability distributions, $P(V \,|\, \mathrm{pa}(V) = \pi)$, where $\mathrm{pa}(V)$ contains no utility nodes and $\pi \in \mathrm{Sp}(\mathrm{pa}(V))$ is a vector of states of $\mathrm{pa}(V)$. To each utility node, V, is associated exactly one real-valued utility function, $U(\mathrm{pa}(V))$, where $\mathrm{pa}(V)$ contains no utility nodes.

An edge $X \longrightarrow Y$ can have different interpretations depending on what X and Y represent. If Y is a chance variable or a utility function, the edge can represent a causal or a functional relationship between X and Y. If Y is a decision variable, the edge represents the fact that the value of X will be known before the decision Y is made.

3 Three Real-World Applications

PGMs have been developed for three problem domains of the ADVOCATE II project: (i) energy problem for an autonomous ground vehicle (AGV), (ii) energy problem for an autonomous under-water vehicle (AUV), and (iii) sonar image assessment for an AUV.

The models will be used as components of intelligent modules of the autonomous vehicles. Based on sensor input, decisions made by the vehicle piloting module, and historical data, the intelligent modules must provide diagnoses and recommend preventive or recovery actions in case of an abnormal situation.

The set of probability-ranked diagnoses associated with each problem must be exhaustive in terms of the possible root causes of the problem. Similarly, the set of expected-utility-ranked preventive and error recovery actions must exhaustively list the possible action options in response to different abnormal mission states.

4 Acquiring Domain Knowledge

Our knowledge acquisition approach relies on a hierarchical decomposition of the overall abstract target problem into a number of less abstract sub-problems, etc, until the overall abstract problem has been decomposed into its possible root causes. The decomposition process is followed by descriptions of relevant diagnostic information and possible recovery actions for each root cause. Given such a problem decomposition and associated diagnostic and error recovery information, knowledge engineers are able to construct a corresponding qualitative PGM representation.

4.1 Cause Hierarchy

Given an abstract problem formulation, a hierarchical decomposition of the problem should be provided as the first step of the knowledge acquisition process. Note that sub-problems of a problem may be considered as causes of the problem. We shall therefore use the terms "problem" and "cause" interchangeably.

A cause hierarchy is a tree, where each node represents a sub-cause of the cause represented by the node one step further towards the root of the tree, which is given by the most abstract problem formulation. The "leaf" nodes of the hierarchy comprise causes for which no breakdown into further sub-causes is deemed necessary in relation to the task at hand (i.e., diagnosis, error recovery, troubleshooting, etc). The leaf nodes represent *root causes* of the overall problem, and qualify as permissible diagnoses of the problem. We shall use $X \preceq X'$ to denote the fact that X is a sub-cause of X'; i.e., X' lies on the path from X to the root of the hierarchy. If X is a sub-cause of X' and $X \neq X'$, X is a proper sub-cause of X', denoted $X \prec X'$.

Definition 1 A tree $\mathcal{T} = (\mathcal{V}, \mathcal{E})$ is a *cause hierarchy* for a problem P if each of the following conditions are fulfilled:

1. P is the root of $\mathcal{T}$; i.e., $\nexists X \in \mathcal{V} : P \prec X$.
2. Each $X \in \mathcal{V}$ is a cause of P; i.e., $X \preceq P$.
3. $(X', X) \in \mathcal{E}$ if and only if $X \prec X'$ and $\nexists X'' \in \mathcal{V} : X \prec X'' \prec X'$.
4. For each $X \in \mathcal{V}$, $\mathrm{ch}(X)$ is an exhaustive set of direct sub-causes of X, and no pair in $\mathrm{ch}(X)$ are mutually exclusive. □

A cause qualifies as a root cause if it provides a satisfactory explanation of the overall problem. Thus, non-leaf nodes can also be root causes. Whether or not a cause is to be considered a root cause may be situation specific. For example, to a car owner "Ignition system problem" may be considered a root cause of a "Car won't start" problem, as it provides him with sufficient information to decide if he should consult a mechanic. His mechanic, however, would not accept this as a root cause, as he would want to diagnose the problem in further depth, allowing e.g. "Ignition cable broken" to be a root cause.

Parsimony is strongly recommended when constructing the cause hierarchy. More precisely, the level of detail of the hierarchy should be just large enough to allow each variable relevant for solving the inference problem at hand to be elicited from the cause hierarchy. For example, in the "Car won't start" problem, including a sub-cause like "Forgot to turn off the light" of the cause "Dead battery" or a sub-cause like "More than 50,000 km since last service" of the cause "Worn spark plugs" might not be relevant, as they provide no extra information relevant for solving the problem or pinpointing the cause of it.

Definition 2 A node X of a cause hierarchy $\mathcal{T}$ for a problem P is a *permissible diagnosis* (or *root cause*) of P if it potentially provides a satisfactory explanation for P and it provides more information relevant for solving P than do $\mathrm{pa}(X)$. A permissible diagnosis is a *possible diagnosis* if and only if it can be distinguished from other permissible diagnoses of P given any diagnostic information. $\mathrm{pd}(X) \subseteq \mathrm{ch}(X)$ is the maximal subset of $\mathrm{ch}(X)$ of permissible diagnoses of P. A cause of P that is not a permissible diagnosis of P is called an *abstract cause.* □

A possible diagnosis can be thought of as a "diagnosable root cause".

4.2 Building the Cause Hierarchy

We now present a method for constructing a cause hierarchy, illustrating it using the AGV energy problem mentioned in Section 3.

The AGV is a simple four-wheel vehicle that carries no active payload systems. The vehicle has two actuators that are responsible for driving it forward and backward, as well as turning it. The actuators are small electrical motors driven by an onboard battery.

There are two different aspects (or sub-causes) of the energy problem: "High energy consumption", indicating that the current level of energy consumption is significantly higher than recommended, and "Low state of charge (SOC)", indicating either an abnormally high level of cumulative energy consumption or a poor state of the battery.

These two aspects relate to, respectively, the present and the cumulative energy consumption. To simplify the analysis, we shall assume that if both situations occur simultaneously, the situation shall be characterized as "Low state of charge", as this is the more serious cause, calling for mission re-planning or mission abortion.

In terms of diagnostic and error recovery behaviour of the intelligent module of the AGV, these two aspects translate into a proactive and a reactive behaviour, respectively. Thus, if the present consumption level is higher than recommended, the AGV system should act proactively to avoid reaching a critical situation later in the mission. On the other hand, if a critical situation has occurred, the AGV system should act reactively, trying to make sure the AGV is able to complete its mission (possibly involving re-planning).

The root of the cause hierarchy, "Energy problem", provides the most abstract problem description (or diagnosis). Following the above discussion, this most abstract problem diagnosis is naturally decomposed into sub-causes "High energy consumption" and "Low state of charge". Neither of these sub-causes are to be considered as root causes, as more detailed diagnoses will be required.

A natural decomposition of "High energy consumption" could be "External problems" and "Internal problems", referring, respectively, to causes in the working environment of the AGV and to causes in the AGV itself. Due to the simplicity of the AGV, the only relevant internal problem is "Actuator problem".

The health state of an actuator can influence its consumption of energy. However, the health state cannot be detected directly, as there are no sensor data available for the actuators. Still, it can be useful to include "Actuator problem" in the model, as we can get indirect indications of actuator problems. For example, if all external problems have been ruled out as causes of "High energy consumption", we get reason to believe that a problem with one of the actuators is the cause of the unexpected high consumption level. Since there is no sensor data available for the actuators, "Actuator problem" is the most specific diagnosis that can be provided for "Internal problems", and is thus a root cause.

This decomposition process continues until we have a complete hierarchical description of the causes of the energy problem. Figure 1 shows the resulting cause hierarchy.

The subset of the permissible diagnoses (root causes) that are possible diagnoses are marked with a "+" in front of them. The remaining permissible diagnoses are marked with a "−" in front of them.

Sometimes one may not want to incorporate a root cause as a possible diagnosis if it is trivially true given previous diagnoses or actions. Such root causes may then be marked with a "−". Note, for example, that the root causes of "High cumulative consumption" as well as the sub-causes of these root causes are all marked with a "−". The reason is that each of them simply represents repetitive diagnoses or decisions of a particular kind. For example, the diagnosis "Large number of obstructing objects" should be immediately obvious from the multiplicity of the "Obstructing object" diagnosis. This information will be available from the decision module that is assumed to request and record diagnoses from the intelligent module on a regular basis.

In summary, the procedure for establishing the cause hierarchy can be outlined as follows:

1. Define the overall problem, P, and the

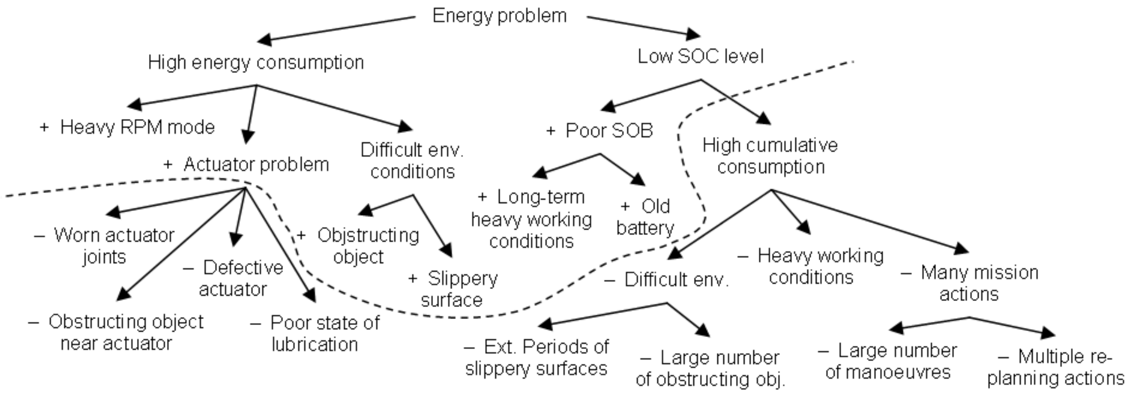

Figure 1: Cause hierarchy for the UAH energy problem.

purpose of the resulting PGM.

2. Let $\mathcal{T}$ be a cause hierarchy with a single node labelled P. Let X refer to this node.
3. Identify the set $\mathrm{ch}(X)$.
4. For each $D \in \mathrm{ch}(X)$ do:
   (a) Add a node labelled D to $\mathcal{T}$ and let X be its parent.
   (b) Let $X := D$ and go to 3.
5. Let X refer to the root node of $\mathcal{T}$.
6. For each $D \in \mathrm{pd}(X)$ do:
   (a) If D can be distinguished from each $D' \in \mathrm{pd}(X) \setminus \{D\}$ given available information, mark D with a "+"; otherwise, mark D with a "–".
7. For each $D \in \mathrm{ch}(X)$ do:
   (a) Let $X := D$ and go to 6.

4.3 Diagnostic Information and Recovery Actions

The cause hierarchy acts as a roadmap for describing the relevant diagnostic information and the possible recovery actions. We divide this information into background information (e.g., the age of a battery) and symptom information provided through sensor readings, inspection, etc., as well as information provided through interaction with the system during the course of diagnosing the system.

For example, in the "Car won't start" example, information of the latter kind could be "Radio is dead" or "Coated spark plugs" provided through interaction with the car. Thus, this kind of information can be characterized as information obtained through deliberate information-gathering actions or test actions.

Thus, the diagnostic information relevant for identifying a cause, C, as the root cause of a problem, P, can be elicited by describing (i) all pieces of background information that potentially provide information relevant for identifying C as the cause of P, (ii) all the symptoms that can be observed if C is the cause of P, and (iii) all investigating actions performed to obtain further information to identify C as the cause of P.

Relevant background information for identifying a cause C is information that has a causal influence on C. Symptom information, on the other hand, is information that can be observed as a consequence of the cause being present. In other words, C has a causal influence on its symptoms.

Although an abstract cause does not provide a satisfactory explanation for a problem, it might still be important to provide relevant diagnostic information for the cause, as this information may have a significant impact on the probabilities of the permissible diagnoses.

By a closer examination, however, it becomes apparent that a cause of a sub-tree of the cause hierarchy that does not contain any possible diagnoses is unlikely to provide relevant diagnostic information or error recovery information. That is, if there are no observable manifestations of the cause strong enough to identify a possible diagnosis for the cause, we probably need not worry about it when eliciting the diagnostic and error recovery information. In particular, none of the causes below the dotted line in Figure 1 contain any possible diagnoses. A cause of a sub-tree of the cause hierarchy that contains one or more possible diagnoses is called an *eligible cause*.

The actions used for eliminating causes of the problem must be represented in the PGM. Such actions are referred to as recovery actions if they recover the system from non-fatal states and as repair actions if they recover from fatal states.

The elicitation of actions is simply a matter of running through the list of possible diagnoses (i.e., those marked with a "+" in our cause hierarchy). That is, we assume that all relevant recovery actions can be identified by listing the possible recovery actions for each possible diagnosis.

For a particular root cause, there might be several actions that can eliminate the cause, and an action can be something that permanently solves the problem or it can be a workaround that can be used as a temporary solution until a proper fix can be made. Information of that kind should also be mentioned.

In ADVOCATE II, the diagnostic and error recovery knowledge were simply stated in tabular form, where there is one row for each eligible cause (i.e., those above the dotted line in Figure 1). The diagnostic and error recovery knowledge elicited for the AGV energy problem appears in Table 1, where both the investigating actions and the recovery actions are listed in the "Actions" column.

5 Constructing the PGM

We now describe how the elicited qualitative knowledge can be translated into a PGM.

First, the variables of the model must be identified. These come from three different sources: (i) Each cause of the cause hierarchy qualifies as a chance variable. (ii) Each source of background information and symptom information, as well as the result of each kind of investigating action qualifies as a chance variable. (iii) Each subset of mutually exclusive recovery actions qualify as a decision variable.

From the information provided in Figure 1 and Table 1 it is relatively straightforward to identify the chance and decision variables, the domains of relevant utility functions, and the causal and functional relations. We shall spare the reader for a complete and tedious description of how each variable and relation are identified, and only give two examples.

Figure 2 shows the structure of the resulting PGM for the AGV energy problem. The ovals, rectangles, and diamonds represent, respectively, random variables, decision/action variables, and utility functions. The black ovals represent the possible diagnoses, the grey ovals with white labels the background information, and the grey ovals with black lables the symptom information and the information acquired by performing investigating actions. The white oval nodes represent auxiliary random variables (i.e., variables never observed and for which their probability distributions are of no immediate interest).

Note that the cause hierarchy of Figure 1 is reflected directly by the diagnosis variables, the symptom variables (except "NSS vehicle stalled"), those of the background-information variables that are parents of diagnosis variables, and the auxiliary variable "SOC level".

To get a feeling for how the structure has been derived from the elicited knowledge, consider the following couple of examples:

- The most abstract problem, "Energy problem", that appears as the root node of the cause hierarchy, is naturally represented as a Boolean variable with two parent variables, namely a variable, SOC level, representing the current state of charge (SOC) of the battery and a variable, Energy consumption, representing the current energy consumption. The variable Energy problem has no child variables, as there are no direct symptoms associated with it (see Table 1).
- From Table 1 it can be concluded that

Cause	Background	Symptoms	Actions	Remarks
Energy problem	SOC level and energy consumption level		Reduce velocity, re-plan mission, abort mission, or emergency stop	Action depends on severity of problem
High energy consumption		High measured RPM	Reduce velocity or re-plan mission	Estimated energy consumption level is provided
Actuator problem		High energy consumption	Abort mission, or emergency stop	Action depends on severity of problem
Heavy RPM mode	Expected RPM and measured RPM		Reduce velocity	
Obstructing object		Vehicle stalled diagnosis from NSS, measured RPM, and energy consumption level	Perform a back/forth manoeuvre	If 3 consecutive back/forth actions haven't solved the problem, the AGV should be liberated
Slippery surface		Energy consumption and measured RPM	Reduce velocity	Slippery surface results in relatively low energy consumption
Unexpected low SOC (state of charge)			See "Energy problem" + perform a SOB test if the acceptable SOC is significantly less than actual SOC	Estimated SOC is provided
Poor SOB (state of battery)	Working age of battery and its long-term working conditions	Negative high derivative of the battery SOC	See "Energy problem" + replace battery	
Old battery	Age of battery and number of recharges	Poor SOB		
Long-term heavy working conditions	History log of AGV	Poor SOB		
High cumulative consumption	Working conditions, environment conditions, # re-planning actions, # manoeuvres	Low SOC level		

Table 1: The diagnostic and error recovery information provided for the AGV energy problem.

there must be a decision variable with mutually exclusive states representing various crucial mission actions: Reduce velocity, Re-plan mission, Abort mission, and Emergency stop. To make the set of decision options exhaustive we need to add one additional option, Continue, representing the decision in the "normal" situation. It appears from Table 1 that the desired decision option depends on SOC level, Poor state of battery, and Actuator problem.

Hopefully, the above examples clearly show that deriving the structure of the model is straightforward given the information provided in Figure 1 and Table 1.

There are some degrees of freedom, however, and some need to refine the information provided. For example, in a discussion following the initial knowledge acquisition process, it appeared that the utility function associated with the Mission action decision variable needed to depend also on the actual velocity of the AGV. That is, if the velocity is already low, the model should not recommend Reduce velocity, but rather Re-plan mission in response to indications of an energy problem. As another example, from Figure 1 the variable Heavy RPM mode appears to be a parent of Energy consumption, whereas in Table 1, Heavy RPM mode appears to be a child of Measured RPM and Expected RPM. Thus, since Measured RPM is a child of Energy consumption, we cannot allow Heavy RPM mode to be a parent of Energy consumption, as this would introduce a directed cycle, which is prohibited in PGMs.

Note that, for clarity of exposition, we have left out information about the order in which the actions are to be made and which information is available before performing each of the actions. This kind of information can have a crucial impact on the results of the inference performed. The information can, however, be acquired very easily.

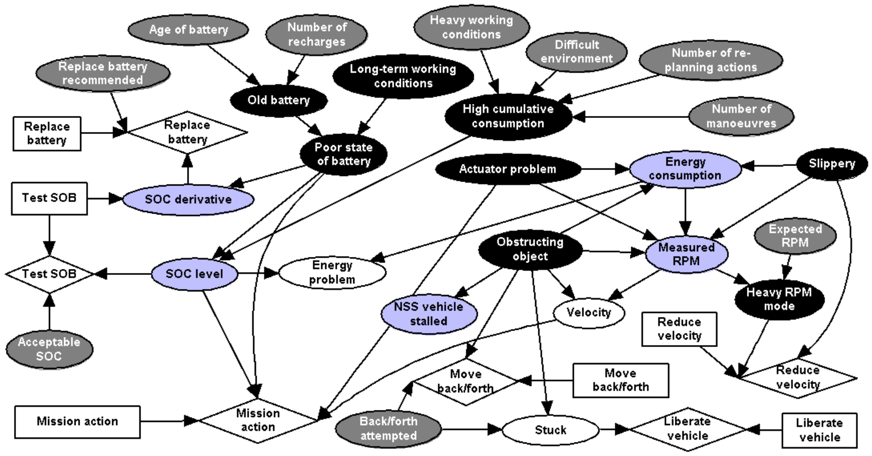

Figure 2: Structure of the probabilistic graphical model of the AGV energy problem.

6 Concluding Remarks

Our knowledge acquisition scheme has been successfully applied for constructing the qualitative parts of three different PGMs within the robotics domain. Although the scheme relies exclusively on the use of a colloquial vocabulary in its knowledge formulation process, experience has shown that knowledge engineers are able to straightforwardly translate the knowledge expressed into a graphical model representation, ready to be populated with probability and utility parameters.

Despite the successful application of the scheme, much work remains to make it a comprehensive knowledge acquisition scheme for constructing PGMs. This work includes things like improved support for specification of diagnostic and error recovery information that is more closely linked to the cause hierarchy, support for specification of quantitative knowledge, support for construction of hierarchical (object-oriented) models [4], etc. We think, however, that this work is an important first step in that direction.

Acknowledgements

We wish to thank our colleagues in the ADVOCATE II project. The project has been supported by European IST programme (grant number IST-2001-34508).

References

[1] ADVOCATE II (2001). The official website, *http://advocate2.e-motive.com/*.

[2] Heckerman, D. (1990). Probalistic Similarity Networks, *Networks*, 20, 607–636.

[3] Jensen, F. V. (2001). *Bayesian Networks and Decision Graphs*, Springer.

[4] Koller, D. and Pfeffer, A. (1997). Object-Oriented Bayesian Networks, in *Proc. 13th UAI*, 302–313, Morgan Kaufmann.

[5] Lacave, C. and Diez, F. J. (2003). Knowledge Acquisition in PROSTANET — A Bayesian Network for Diagnosing Prostate Cancer, in *Proc. 7th Conf. on Knowledge-Based Intelligent Info. & Eng. Systems*, 1345–1350, Springer.

[6] Lucas, P. (1996). Knowledge Acquisition for Decision-theoretic Expert Systems, *AISB Quaterly*, 94, 23–33.

[7] Möbus, C., Schröder, O. (1997). Building Domain Models by Novices in Stochastics: Towards the Probabilistic Semantics of Verbalized Stochastic Relations, in *AI in Education: Knowledge and Media in Learning Systems*, 394–401, IOS Press.

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

[9] Skaanning, C. (2000). A Knowledge Acquisition Tool for Bayesian-Network Troubleshooters, in *Proc. 16th UAI*, 549–557, Morgan Kaufmann.