Applications of Probabilistic Graphical Models to Diagnosis and Control of Autonomous Vehicles

Madsen, Anders L.; Kjærulff, Uffe Bro; Kalwa, Jörg; Perrier, Michel; Sotelo, Miguel Angel

Published in:
The Second Bayesian Modeling Applications Workshop

Publication date:
2004

Document Version
Publisher's PDF, also known as Version of record

Applications of Probabilistic Graphical Models to Diagnosis and Control of Autonomous Vehicles

Anders L. Madsen
Hugin Expert A/S
Denmark
anders@hugin.com

Uffe B. Kjærulff
Aalborg University
Denmark
uk@cs.auc.dk

Jörg Kalwa
ATLAS Elektronik
Germany
kalwa@atlas.de

Michel Perrier
Ifremer
France
Michel.Perrier@ifremer.fr

Miguel Ángel Sotelo
University of Alcalá
Spain
michael@depeca.uah.es

Abstract

We present the main elements of a distributed architecture supporting diagnosis and control of autonomous robots. The purpose of the architecture is to assist the operator or piloting system in managing fault detection, risk assessment, and recovery plans under uncertainty. The architecture is generic, open, and modular consisting of a set of interacting modules including a decision module (DM) and a set of intelligent modules (IMs). The DM communicates with the IMs to request and obtain diagnosis and recovery action proposals based on data obtained from the robot piloting module. The architecture supports the use of multiple artificial intelligence techniques collaborating on the task of handling uncertainty.

In this paper we focus on the application of Bayesian modeling to three problems of diagnosis and control of autonomous robots or vehicles. The paper describes and discusses how we use Limited Memory Influence Diagrams (LIMIDs) to represent and solve complex problems of diagnosis and control of ground and underwater robotic vehicles. In particular, we describe how battery monitoring and control problems related to an underwater and a ground vehicle are solved and how a sonar image quality assessment problem related to an underwater vehicle is solved.

1 Introduction

Within the scope of the European Union project, ADVOCATE, the needs to increase the performances of unmanned underwater vehicles were identified in terms of safety for the system itself as well as for its environment, availability of the system, and efficiency and reliability of the system. The aim of ADVOCATE II is to design and develop an architecture to increase the performance of (semi) autonomous vehicles focusing on Autonomous Underwater Vehicles (AUVs) and Autonomous Ground Vehicles (AGVs). The objectives are to increase the safety for the system itself as well as the environment, to increase automation, and to increase efficiency and reliability of the system. The interest of such a concept from the market point of view has been demonstrated by a market study.

These objectives will be reached by adding intelligence into existing and new control software to diagnose and recover from any dysfunction situation of the system. The architecture is designed with the ability to incorporate and merge different AI techniques. The main objective is to have a better management of uncertainty in robots by the use of intelligent diagnosis and control software, but without too specific non-reusable developments.

Three end-user partners are involved in the ADVOCATE II project: University of Alcalá designs piloting modules for AGVs for surveillance applications, Ifremer designs AUVs for scientific applications, and ATLAS Elektronik designs AUVs and semi-AUVs for industrial applications.

This paper focuses on the application of Bayesian modeling to problems of diagnosis and control of autonomous vehicles. It discusses how we use LIMIDs to represent and solve complex problems of diagnosis and control of AGVs and AUVs. In particular, we describe how battery monitoring and control problems related to an AUV and an AGV are solved and how a sonar image quality assessment problem related to an AUV is solved.

Section 2 introduces the domain of semi-autonomous vehicles focusing on ground and underwater vehicles. The ADVOCATE communication architecture is described in Section 3. Section 4 presents some preliminaries and notation on the LIMID representation used to model the diagnosis and control problems. The knowledge extraction process and the knowledge extraction method developed as part of the ADVOCATE II project is described in Section 5. Section 6 describes the models developed to solve the diagnosis and control problems, and the integration of the models into the vehicles. Finally, Section 7 ends the paper with a

discussion of our work.

2 Problem Domain

The transition of autonomous vehicles from research tools to real applications increases the need for reliable and safe performance of the vehicles. This includes detection, avoidance, and recovery from any dysfunction.

Both the AGVs and the AUVs considered in this paper are supplied with energy by a battery. This poses the problem of monitoring the remaining energy level of the battery and providing diagnosis and recovery actions in order to manage the mission parameters related to the energy consumption and to avoid unnecessary mission aborts. One AUV is equipped with an advanced object detection and avoidance system. This system works well in situations where obstacles can be detected by sonar. Hence, it is important to assess the quality of the sonar image and to suggest recovery actions to improve the sonar image quality or to suggest reductions in speed due to too poor image quality.

2.1 DeepC

ATLAS Elektronik is developing a new type of underwater vehicles operating with autonomous mission durations of up to 60 hours. The vehicle is referred to as DeepC (see Figure 1).

The long mission durations impose the need for advanced AI techniques to detect, avoid, and recover from any dysfunction. All end-users and ATLAS Elektronik in particular were faced with problems, which could not easily be solved by existing systems. The current approach to handle mission faults is to abort the mission. However, a mission abort is very expensive in both time and money.

Figure 1: The DeepC underwater vehicle.

As a fully autonomous system, the DeepC vehicle has to rely on its sensors to survive operationally. The DeepC is equipped with an advanced object detection and avoidance system. The object detection system consists of a mechanically scanning, forward looking sonar and its electronics. This system works well when the sonar image is of sufficient quality. The problem considered is to construct a model for assessing the sonar image quality and for suggesting actions to avoid object collisions.

2.2 VORTEX

Ifremer has developed the remotely operated underwater vehicle VORTEX (see Figure 2), which for our purpose is functionally considered as an AUV, as it allows programming of autonomous complex missions.

Figure 2: The VORTEX underwater vehicle.

The motivation for equipping the vehicle with AI technology is much the same as for the DeepC, including optimization of the mission plan, diagnosis of abnormalities, recovery planning in case of abnormalities, avoidance of mission abortion (which is very expensive), and avoidance of vehicle loss.

2.3 BART

To put the ADVOCATE II concept into practice, the University of Alcalá is deploying a telesurveillance application using the BART AGV (see Figure 3). Two independent actuators powered by an onboard battery drive the vehicle.

Figure 3: The BART autonomous ground vehicle.

In order to increase the probability of mission success in case of energy problems or in case the vehicle gets stalled, the overall mission of the vehicle as well as other navigational issues will be managed using ADVOCATE II technology. The ADVOCATE II system will provide the AGV with intelligent diagnosis capabilities and ability to recommend optimal recovery actions resulting in more reliable and safe operations. The diagnosis and recovery capabilities will be concerned with aspects of navigation, energy system, sensors, actuators, etc.

3 ADVOCATE II Architecture

The ADVOCATE II architecture is a distributed architecture based on a generic communication protocol. The architecture is modular and easy to evolve and adapt to legacy piloting systems. The architecture is designed to allow easy integration of different artificial intelligence techniques into preexisting systems.

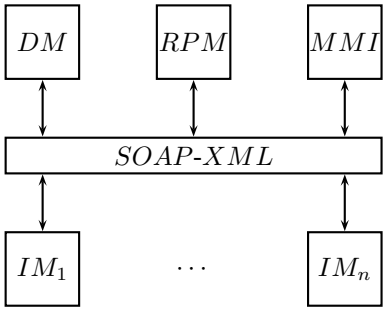

Figure 4: The ADVOCATE II software architecture.

The purpose of the architecture is to assist the operator or piloting system in managing fault detection, risk assessment, and recovery plans under uncertainty. The generic communication protocol is based on SOAP/XML technology implementing HTTP for communication between different types of modules (see Figure 4). The DM communicates with the IMs to request and obtain diagnosis and recovery action proposals based on data obtained from the Robot Piloting Module (RPM).

The architecture supports the use of multiple artificial intelligence techniques collaborating on the task of handling uncertainty. Probabilistic Graphical Models (PGMs), Neuro-Symbolic Systems (NSS), and Fuzzy Logic are being used to solve diagnosis and control problems related to the AGVs and AUVs. This raises the question of how to most efficiently integrate different AI techniques into new and existing systems. We have found that the most efficient way to integrate multiple AI techniques into existing and new systems is through an open and generic architecture with a sophisticated communication interface.

4 Preliminaries

An influence diagram (Howard and Matheson 1981) $\mathcal{N} = (\mathcal{G}, \mathcal{P}, \mathcal{U})$ is a Bayesian network (Pearl 1988; Cowell *et al.* 1999; Jensen 2001; Neapolitan 2003) augmented with decision variables and utility functions. It consists of an acyclic, directed graph $\mathcal{G} = (V, E)$ over vertices V, connected by directed edges $E \subseteq V \times V$, a set of conditional probability distributions $\mathcal{P}$, and a set of utility functions $\mathcal{U}$. The vertices V of $\mathcal{G}$ correspond one-to-one with random variables, decision variables, and utility functions of $\mathcal{N}$.

An influence diagram supports the representation and solution of sequential decision problems under the no-forgetting assumption (i.e., perfect recall of all observations and decisions made in the past that are influential in a given decision situation is assumed). The LIMID (Lauritzen and Nilsson 2001) is an influence diagram relaxing the non-forgetting assumption to a limited memory assumption. Mathematically, a LIMID is a compact representation of a joint expected utility (EU) function:

$$\mathrm{EU}(V) = \prod_{X \in V_C} P(X \,|\, \mathrm{pa}(X)) \sum_{u \in \mathcal{U}} u.$$

where V_C are random variables. To solve a LIMID $\mathcal{N}$ is to determine an optimal strategy $\hat{\Delta}$ for the decision maker to follow. The strategy consists of one decision policy δ_D for each decision variable D in $\mathcal{N}$. A policy δ_D is a mapping from the requisite past of D to the state space $\|D\|$ of D.

An OO LIMID $\mathcal{N} = (\mathcal{G}, \mathcal{P}, \mathcal{U})$ is an extension of the LIMID with support for object-oriented constructions (Koller and Pfeffer 1997). In addition to the elements of a LIMID, the OO LIMID contains instance nodes. An instance node X represents the realization of a LIMID class $\mathcal{N}$ within another LIMID class $\mathcal{M}$ following the object-oriented paradigm. In graphical representations of LIMIDs, decision variables are indicated using box-shaped nodes, random variables using oval-shaped nodes, utility functions using diamond-shaped nodes, and instance nodes using box-shaped nodes with rounded corners. The interface of $\mathcal{N}$ is its input $\mathcal{I}(\mathcal{N})$ and output $\mathcal{O}(\mathcal{N})$ variables (partly gray nodes where input variables are indicated using a dashed black border), see Figure 8 for an example.

5 Knowledge Extraction

Unfortunately, the construction of a PGM can be a labor intensive task with respect to both knowledge acquisition and formulation. LIMIDs are not exceptional in this respect. The knowledge acquisition and formulation process associated with building the three LIMID models involved knowledge engineers and domain experts located in four different countries. The knowledge engineers and domain experts had limited possibilities for face-to-face meetings and the domain experts had limited knowledge of LIMIDs. Therefore, a knowledge acquisition scheme had to be developed that did not rely on familiarity with terminology of PGMs and direct contact with the knowledge engineers.

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.

Figure 5 shows such a cause hierarchy related to the energy problem of the BART AGV. The causes of the hierarchy are grouped into causes that qualify as satisfactory explanations of the overall problem and causes that do not. The first group of causes are referred to as *permissible diagnoses*. The subset of these that can actually be identified based available information are referred to as *possible diagnoses*. Possible diagnoses are marked with a "+" in Figure 5, and permissible diagnoses that are not possible are marked with a "−". Sometimes the knowledge engineer might want to not represent a possible diagnosis as such in

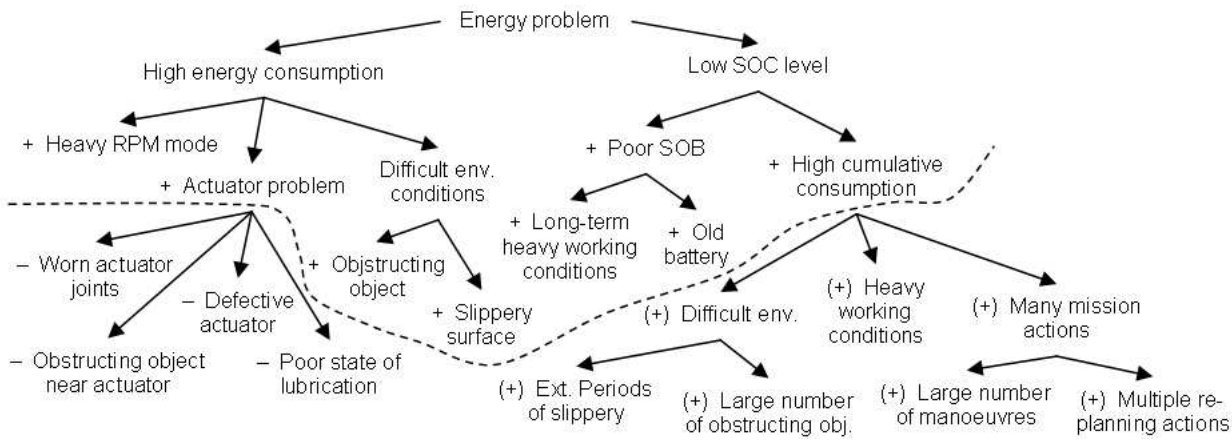

Figure 5: Cause hierarchy for the BART AGV energy problem.

the model if it is trivially true given previous diagnoses or actions. Such causes are marked with a "$(+)$".

The cause hierarchy acts as a road-map for describing the relevant diagnostic information and the possible recovery actions. 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. Thus, if there are no observable manifestations of the cause strong enough to identify a possible diagnosis for the cause, we 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 5 contain any possible diagnoses. The domain expert provides the relevant diagnostic information and the recovery actions in matrix form with one row for each cause "above the dotted line" and one column for each kind of diagnostic information (i.e., background and symptom) and one column for possible recovery actions.

The qualitative knowledge elicited following such a scheme provides a sufficient basis for a knowledge engineer to construct the structure of a PGM, on the basis of which a the quantitative knowledge can then be elicited.

This section is based on (Kjærulff and Madsen 2004).

6 Models

One LIMID model for each of the vehicles has been developed in tight collaboration with the end-user using the knowledge extraction method described in Section 5. For reasons of space limitations, we include only a subset of the cause hierarchies and models developed.

6.1 VORTEX

The purpose of the PGM IM of the VORTEX is to assess the status of the energy consumption of the actuators and the payload systems of the AUV. The payload systems consist of various sensors for scientific investigations. More concretely, the task of the module is to compute the probabilities of the various possible root causes of unexpected high energy consumption and the expected utilities of the various recovery actions given the information available.

There are two different aspects (or sub-causes) of "Energy consumption problem", namely "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 (SOB). These two aspects relate to, respectively, the *present energy consumption* and the *cumulative energy consumption*. The present energy consumption is defined as the average consumption over the last 10 seconds.

To identify the cause of low SOC as a high cumulative energy consumption, the model should either be dynamic, capable of representing phenomena evolving over time, or rely on a measurement of the accumulated total consumption and an indication of the recommended accumulated total consumption at any given point of the mission. We decided to go with the latter approach, as a dynamic model would result in serious computational complexity problems. Also, given that periodic requests are issued from the DM to the VORTEX IM, determining that the accumulated energy consumption is high is a straightforward task that might as well be performed by the DM itself.

Figure 6 shows the resulting LIMID. There are four groups of random variables in Figure 6: Ten diagnosis variables, eleven background information variables, nine symptom variables, and eighteen auxiliary variables. The ten diagnosis variables represent the following distinct root causes of an energy consumption problem of the VORTEX: *Old battery*, *Long-term heavy working conditions*, *Poor SOB*, *Cold battery*, *High cumulative energy consumption*, *Ob-*

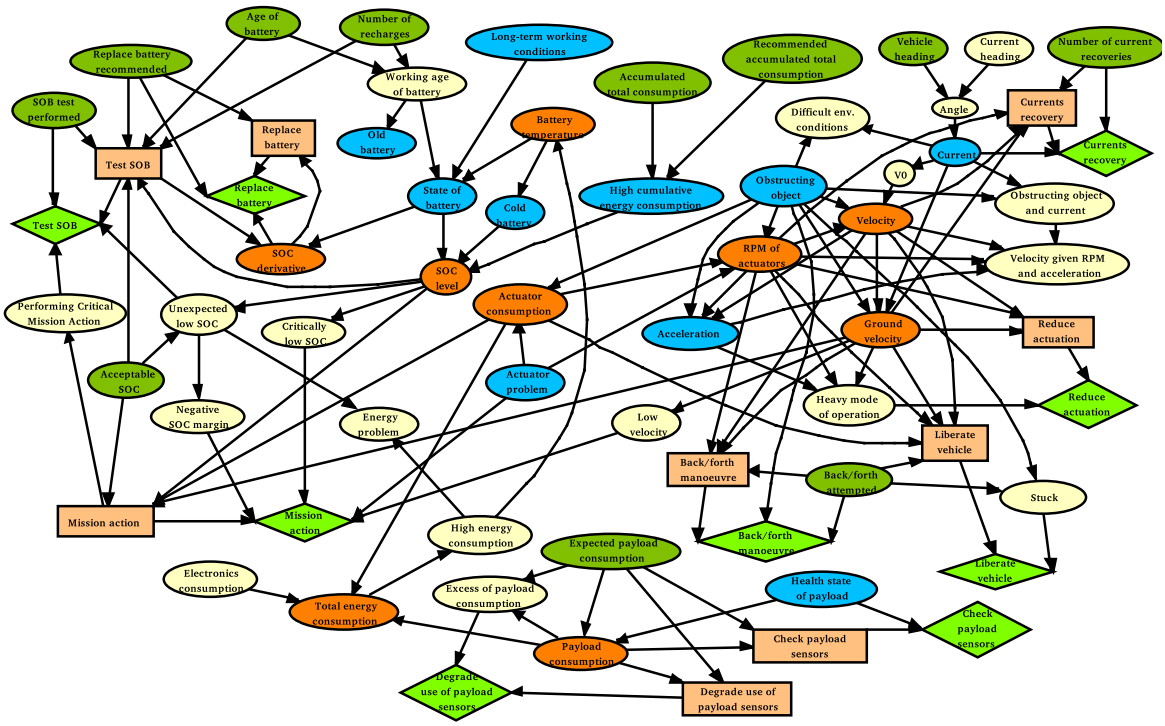

Figure 6: The VORTEX LIMID.

structing object, *Strong currents*, *Fast acceleration*, *Actuator problem*, and *Unhealthy payload*. The posterior probability distributions for these diagnoses are computed on the basis of information provided through the twenty evidence variables (symptom measurements and background information).

The domain experts identified a group of nine different actions that can be performed in response to energy problems: *Mission action* (e.g., "Continue", "Reduce velocity", "Abort mission", etc.), *Test SOB*, *Replace battery*, *Back/forth manoeuvre* (i.e., to escape from an obstructing object), *Check payload sensors*, etc.

Except that *Replace battery* must be preceded by a *Test SOB* action there are no natural orderings among the actions. Also, observations will be provided for all twenty evidence variables (symptom measurements and background information) before any decisions are going to be made. These two facts imply that the model is not naturally represented as an influence diagram. Also, it would make exact inference absolutely intractable.

The LIMID framework therefore offers an ideal representation of this combined diagnosis and decision problem. In fact the size of the junction tree for the network in Figure 6 is only about 30K (measured as the sum of the sizes of the clique tables). We should note, however, that the "limited memory" aspect of the model contributes significantly to this fact, as there are observed variables that belongs to the "relevant past" (Shachter 1999) of some decision variables that do not appear as parents of these variables. For example, according to the model in Figure 6 the observed variables *RPM of actuators* and *Velocity* appear to be relevant for the *Mission action* decision, but there are no information links from these variables to the decision variable, as the *Actuator consumption* and the *Ground velocity* variables are assumed to cater for their influences.

Despite the "limited memory" aspect of the model in Figure 6, preliminary evaluations of the model provided satisfactory results.

6.2 BART

The purpose of the PGM IM of the BART AGV is very similar to that of the VORTEX AUV. The fact that the BART carries no payload systems and the obvious difference that the BART is an AGV and the VORTEX an AUV, give rise to some differences in the two models, but for the most part, the BART model shown in Figure 7 constitutes a subset of the VORTEX model. After some adjustments of the model a preliminary evaluation of the model showed an almost complete agreement between expert diagnoses and recommendations and those provided by the model.

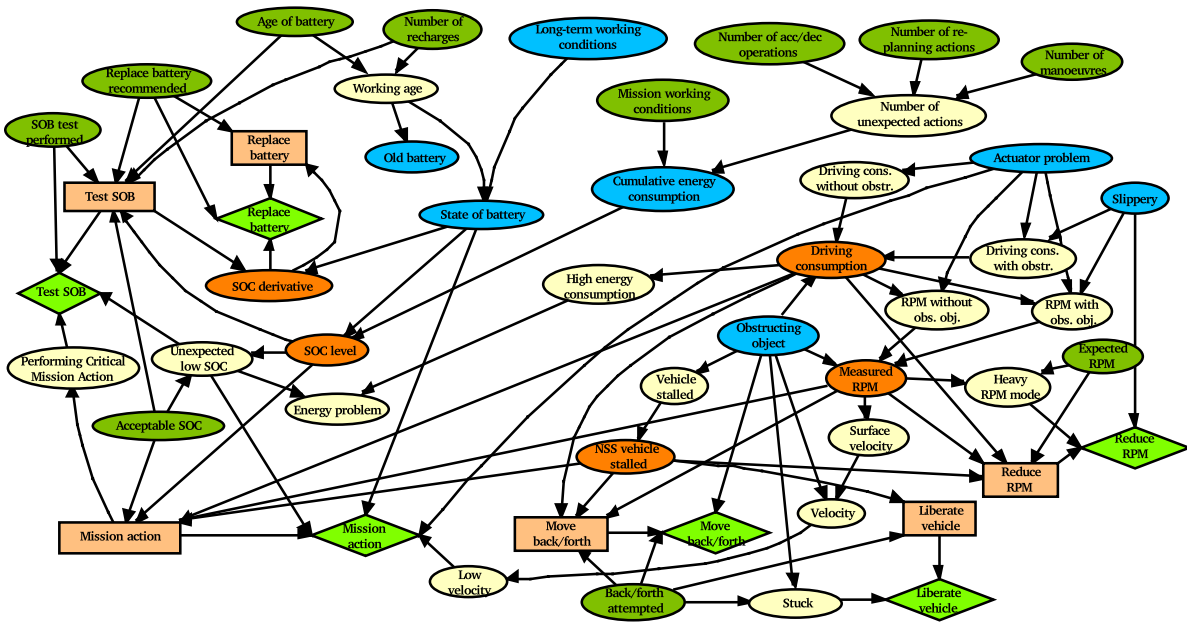

Figure 7: The BART LIMID.

6.3 DeepC

The purpose of the PGM IM of the DeepC is to assess the quality of the sonar image and in the case of bad sonar image quality to suggest appropriate actions to avoid damage to the vehicle or even a lost vehicle.

The assessment of the sonar image quality is based on the computation of three sonar image quality indicators. The quality indicators are determined by the RPM and fed into the model as evidence. The sonar image quality indicators are pixel entropy, pixel mean value, and pixel substance, see (Kalwa and Madsen 2004) for details. From the above description of the problem domain, it is clear that the amount of disturbance in the sonar image and the presence of objects is time dependent.

The main modeling challenges were to capture the dynamics of the process (how the quality indicators relate to the position and behaviour of the vehicle, the noise sources, and the quality of the image), to address the inherit infinite horizon problem, and to maintain a computationally efficient model (small cliques and policies).

We model the problem as a discrete time, finite horizon partially observed Markov decision process. The model is dynamic in the sense that it models the behaviour of the system over (discrete) time. The state of the system at any given point in time is partially observed as sensor readings are available, but not all entities of the problem domain are observed.

The top-level LIMID class $\mathcal{N}$ contains three instantiations

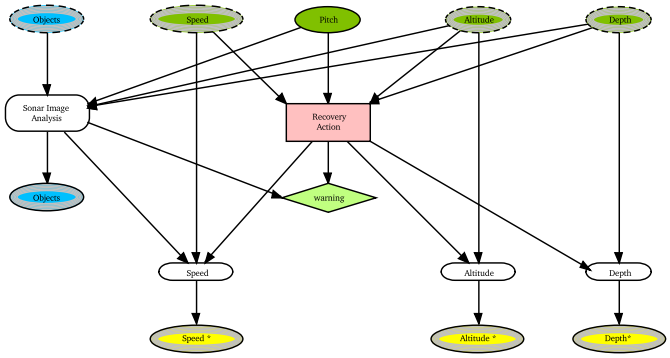

Figure 8: The generic time-slice for sonar image assessment.

$\mathcal{M}_i$, $\mathcal{M}_{i+1}$, and $\mathcal{M}_{i+2}$ of the class $\mathcal{M}$ shown in Figure 8. The input variables are located at the top of the figure, while the output variables are located at the bottom of the figure. In $\mathcal{N}$, the output variables $\mathcal{O}(\mathcal{M}_i)$ of $\mathcal{M}_i$ are connected to the input variables $\mathcal{I}(\mathcal{M}_{i+1})$ of the subsequent time-slice $\mathcal{M}_{i+1}$, and similar for $\mathcal{M}_{i+1}$ and $\mathcal{M}_{i+2}$.

Each instantiation of $\mathcal{M}$ represents the system at a given point in time. The model $\mathcal{N}$ represents the system at three consecutive time steps with an 8 seconds interval. Time is discretized into intervals of 8 seconds, which is equal to the time the image analysis component needs to analyze a single sonar image.

To avoid combinatorial explosion and thereby main-

tain computational efficiency, the model specifies that $Altitude$, $Depth$, $Pitch$, and $Speed$ are observed prior to the decision $Recovery\ Action$, but not the image quality indicators. The values computed for the image quality indicators are inserted as evidence and subsequently policies are recomputed. Hence, the policy for the decision in the next time-slice will only depend on the most recent observations on $Altitude$, $Depth$, $Pitch$, and $Speed$. Since the image quality indicators are observed each time a sonar image is analyzed, we need to resolve the LIMID with the observations on the image quality indicators entered as evidence.

The decision has a potential impact on speed, altitude, and depth of the vehicle. Deciding on a recovery action changing any of these properties will impact the quality of the next sonar image. The probability of a collision is modeled in the class instance $Speed$.

The instance $Sonar\ Image\ Analysis$, which is an instance of the network class shown in Figure 9, models the sonar image assessment process. The three image quality indicators are represented in this class by the variables $Entropy$, $Mean$, and $Substance$. The quality indicators are influenced by the presence of disturbance or objects in the sonar image. Disturbance may be caused by reverberation or noise. The hierarchical construction of the LIMID enforced by the object-oriented paradigm has simplified the knowledge acquisition phase considerably as it is easy to focus on well-defined subparts of the LIMID in isolation. Using class instances, it is a simple task to create and maintain multiple instances of the same LIMID class. Furthermore, it is a simple task to change the class of an instance to another class. This is particularly useful in the knowledge acquisition phase where each LIMID class has been revised and updated multiple times.

Based on first laboratory tests the image quality assessment of the LIMID seems to be of sufficiently accurate and the suggested recovery actions are reasonable. The next step in the validation process is to verify the functionality of the IM in trials performed in a natural sea environment. This will take place before end of 2004.

This section is based on (Kalwa and Madsen 2004).

6.4 Integration and Validation of Modules

Each LIMID model is encapsulated in an IM. The IM takes care of the communication with the DM and RPMs. The module integration is being performed using a special-purpose integration tool. The integration tool has greatly simplified the integration phase as it allows developers to integrate their modules into an architecture consisting of a mix of mock-up modules, MMIs, and real modules. This is very helpful as module developers are located far part in different countries with limited possibilities for face-to-face meetings and in some case even with different working hours.

The validation of each module is equivalent to validation of the knowledge base or model. The validation of a model was performed by a careful investigation of the performance and behavior of the system based on a selected set of test scenarios.

Extensive prototyping has helped to ensure appropriate performance and behavior of each module. The test cases used to validate and measure the performance of a model covers all important, critical situations that possibly can occur in realistic operations. Finally, domain expert(s) and the knowledge engineers have evaluated the model against the test cases iteratively.

7 Discussion

The main objective of the ADVOCATE II project is as mentioned above to develop an architecture to allow the implementation of IMs for AGVs and AUVs, in order to increase their reliability and efficiency.

The performance of the ADVOCATE II architecture is constrained by (soft) real-time requirements. This implies that the performance of the communication protocol and the IMs needs to be very high. This has implied a high focus on computational performance in the model construction.

Not only does the communication architecture enable efficient integration of different AI technologies into new and existing systems, but it also allows various AI techniques to interact (through the DM and RPM though). This option of interactions has been used to dedicate PGMs to certain types of problems and to have variables in a PGM represent the output of an NSS IM (see Figure 7). In addition, different AI techniques may be used to solve the same problem. This will often be the case for mission critical error handling. In our case, this raises the issue of a common scale of measurement of the usefulness of actions. We have chosen to use normalized expected utility as the measurement of usefulness of recovery actions.

Even though the framework of PGMs have been available for more than 15 years, it is our experience that the efficient use of these models in real-world applications still requires a substantial amount of research. It is often a problem of the technology that only few very convincing success stories are available to the public. Often a PGM captures all or almost all the knowledge a company has on a business area. This implies that the company is not interested in sharing this model or even sharing the knowledge that such a model exists. The results of the ADVOCATE II project add to the increasing number of successful applications of PGMs available to the public.

One of our key experiences from research and development

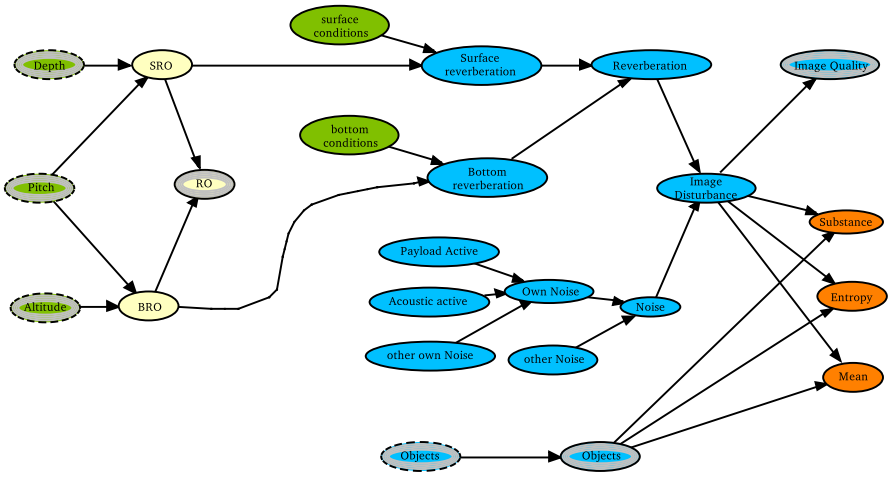

Figure 9: The network class representing the sonar image assessment process.

projects is that even though graphical models are intuitive, they are difficult to build for domain experts. It is often necessary to develop new methodologies or adjust existing methodologies in order to simplify the knowledge acquisition task. We believe the reason is that the flat knowledge representation a PGM offers is too complicated for inexperienced knowledge engineers to use. We have developed a knowledge elicitation and formulation method, which is applicable in general to problems of reasoning and decision making on complex machinery such as AGVs and AUVs.

The LIMID representation (Lauritzen and Nilsson 2001) and the object-oriented knowledge representation paradigm (Koller and Pfeffer 1997) implemented in the Hugin tool (Andersen *et al*. 1989; Jensen *et al*. 2002) have been two major cornerstones of the success of Bayesian modeling in the ADVOCATE II project. Still a lot of work remains to develop the object-oriented framework further.

As the technology of PGMs is often used to assess or solve mission critical problems, it is very important that the system is reliable and robust. Interestingly, although very important, such aspects of the application of PGMs have not yet attracted much attention in the UAI community. For this purpose, we have developed a special purpose tool for network performance assessment.

Acknowledgment

We acknowledge the partners of ADVOCATE II (www.advocate-2.com): University of Alcalá (Spain), Getronics (France), Technical University of Madrid (Spain), ATLAS Elektronik GmbH (Germany), Ifremer (France), Hugin Expert A/S (Denmark), Innova S.p.A (Italy), and e-motive (France). The EU Commission supports the ADVOCATE II project (grant IST-2001-34508).

References

Andersen, S. K., Olesen, K. G., Jensen, F. V., and Jensen, F. (1989). HUGIN — a Shell for Building Bayesian Belief Universes for Expert Systems. In *Proc. of IJCAI'89*, pp. 1080–5.

Cowell, R. G., Dawid, A. P., Lauritzen, S. L., and Spiegelhalter, D. J. (1999). *Probabilistic Networks and Expert Systems*. Springer-Verlag.

Howard, R. A. and Matheson, J. E. (1981). Influence Diagrams. In *The Principles and Applications of Decision Analysis*, chapter 37, pp. 721–62.

Jensen, F., Kjærulff, U. B., Lang, M., and Madsen, A. L. (2002). HUGIN - The Tool for Bayesian Networks and Influence Diagrams. In *Proc. of PGM'02*, pp. 212–21.

Jensen, F. V. (2001). *Bayesian Networks and Decision Graphs*. Springer-Verlag.

Kalwa, J. and Madsen, A. L. (2004). Sonar image quality assessment for an autonomous underwater vehicle. In *Proc. of ISORA'04*. to appear.

Kjærulff, U. B. and Madsen, A. L. (2004). A methodology for acquiring qualitative knowledge for probabilistic graphical models. In *Proc. of IPMU'04*. to appear.

Koller, D. and Pfeffer, A. (1997). Object-oriented Bayesian networks. In *Proc. of UAI'97*, pp. 302–13.

Lauritzen, S. L. and Nilsson, D. (2001). Representing and solving decision problems with limited information. *Management Science*, **47**, 1238–51.

Neapolitan, R. E. (2003). *Learning Bayesian Networks*. Prentice Hall.

Pearl, J. (1988). *Probabilistic Reasoning in Intelligence Systems*, Series in representation and reasoning. Morgan Kaufmann Publishers.

Shachter, R. (1999). Efficient Value of Information Computation. In *Proc. of UAI'99*, pp. 594–601.