Generating Explanations based on Markov Decision Processes

Francisco Elizalde[1], Enrique Sucar[2], Julieta Noguez[3], Alberto Reyes[1]

[1] Instituto de Investigaciones Eléctricas, Reforma 113, Col. Palmira, Cuernavaca, Morelos, 62490, México.
[2] Instituto Nacional de Astrofísica Optica y Electrónica, Luis Enrique Erro No. 1, Tonantzintla, Puebla, 72000, México.
[3] Tec de Monterrey, Campus Cd. de México, Calle del Puente 222, Ejidos de Huipulco, 14380, México, D.F.

Abstract. In this paper we address the problem of explaining the recommendations generated by a Markov decision process (MDP). We propose an automatic explanation generation mechanism that is composed by two main stages. In the first stage, the most *relevant variable* given the current state is obtained, based on a factored representation of the MDP. The relevant variable is defined as the factor that has the greatest impact on the utility given certain state and action, and is a key element in the explanation generation mechanism. In the second stage, based on a general template, an explanation is generated by combing the information obtained from the MDP with domain knowledge represented as a frame system. The state and action given by the MDP, as well as the relevant variable, are used as pointers to the knowledge base to extract the relevant information and fill–in the explanation template. In this way, explanations of the recommendations given by the MDP can be generated on–line and incorporated to an intelligent assistant. We have evaluated this mechanism in an intelligent assistant for power plant operator training. The experimental results show that the automatically generated explanations are similar to those given by a domain expert.

1 Introduction

An important requirement for intelligent trainers is to have an explanation generation mechanism, so that the trainee has a better understanding of the recommended actions and can generalize them to similar situations [1]. This work is motivated by an application for training power plant operators. Under emergency conditions, a power plant operator has to assimilate a great amount of information to promptly analyze the source of the problem and take the corrective actions. Novice operators might not have enough experience to take the best action; and experienced operators might forget how to deal with emergency situations, as these occur sporadically. So in both cases, an intelligent assistant can help to train the operators so they can react appropriately when an emergency situation arises; and in particular it is important that the assistant explains to the user the recommendations generated by a Markov decision process.

Although there has been extensive work on explanation generation for rule-based systems and other representations, there is very little work on explanations using probabilistic representations, in particular for decision–theoretic models such as Markov decision processes (MDPs).

We have developed an automatic generation explanation mechanism for intelligent assistants. For this, we analyzed the explanations given by the domain expert, and designed a general template. This template is composed of three main parts: (i) the recommended action and the relevant variable in the current situation; (ii) a graphical representation of the process highlighting the relevant variable, and (iii) a verbal explanation. To generate the information required to fill the template we combine several knowledge sources. The optimal action is obtained from the MDP that guides the operator in the training session. The relevant variable, which is a key element in the explanation, is obtained from the MDP by analyzing which variable has the highest impact on the utility given the current state. The graphical part is generated from a general block diagram of the process, where the relevant variable is highlighted. The verbal explanation is obtained from a domain knowledge–base (KB) that includes the main components, actions and variables in the process, represented as a frame system. To extract the important information from the KB, we use the information from the MDP (state, action and relevant variables) as pointers to the relevant components, actions and variables; and then follow the links in a frame system to extract other relevant information. The information in the explanation template depends on the level of the operator: novice, intermediate or expert; which is obtained from a simple student model.

Although the explanations are motivated by an application in training power plant operators, the mechanism can be applied in other domains in which a person is being trained or guided by an assistant based on MDPs. For instance, training operators in other industrial environments, training pilots or drivers, etc. This work is one of the first efforts in explaining the actions recommended by an MDP, which main contributions are the selection of the relevant variable, and the combination of the MDP and domain knowledge to generate explanations. We have evaluated the explanations generated in a realistic application in the power plan domain.

2 Related Work

The work on explanations based on probabilistic models can be divided according to the classes of models considered, basically Bayesian networks (BN's) and decision-theoretic models. Two main strategies have been proposed for explanation with BN's. One strategy is based on transforming the network to a qualitative representation, and using this more abstract model to explain the relations between variables and the inference process [2, 3]. The other strategy is based on the graphical representation of the model, using visual attributes (such as colors, line widths, etc.) to explain relations between nodes (variables) as well as the the inference process [4].

Influence diagrams (IDs) extend BNs by incorporating decision nodes and utility nodes. The main objective of these models is to help in the decision making process, by obtaining the decisions that maximize the expected utility. So explanation in this case has to do with understanding why some decision (or sequence of decisions) is optimal given the current evidence. There is very little work on explanations for IDs. Bielza *et al.* [5] propose an explanation method for medical expert systems based on IDs. It is based on reducing the table of optimal decisions obtained from an ID, building a list that clusters sets of variable instances with the same decision. They propose to use this compact representation of the decision table as a form of explanation, showing the variables that have the same value as a rule for certain case. It seems like a very limited form of explanation, difficult to apply to other domains. The explanation facilities for Bayesian networks proposed by Lacave et al. [4] were extended to IDs and integrated in the Elvira software [6]. The extension is based on a transformation of the ID into a BN by using a strategy for the decisions in the model. Lacave *et al.* [6] describe several facilities: incorporating evidence into the model, the conversion of the influence diagram into a decision tree, the possibility of analyzing non-optimal policies imposed by the user, and sensitivity analysis with respect to the parameters.

MDPs can be seen as an extension of decision networks, that consider a series of decisions in time (dynamic decision network). Some factored recommendation systems use algorithms to reduce the size of the state space [7] and perform symbolic manipulations required to group similarly behaving states as a preprocessing step. [8] also consider top-down approaches for choosing which states to split in order to generate improved policies [9]. Recently [10] proposed an approach for the explanation of recommendations based on MDPs. They define a set of preferred scenarios that correspond to set of states with high expected utility, and generate explanations in terms of actions that will produce a preferred scenario based on predefined templates. They demonstrate their approach in the domain of course selection for students, modeled as a finite horizon MDP with three time steps. In contrast, out approach considers an infinite-horizon and incorporates domain knowledge in the explanations generated.

In summary, there is very limited previous work on explanation generation for intelligent assistants systems based on MDPs. In particular, there is no previous work on determining the most relevant variable, which is a key element in the explanation mechanism we propose, and in integrating the information from the MDP and domain knowledge to generate explanations that are understandable for a user with no previous knowledge on probability or decision theory.

3 Factored Markov Decision Processes

A Markov decision process [11] models a sequential decision problem, in which a system evolves in time and is controlled by an agent. The system dynamics is governed by a probabilistic transition function Φ that maps states **S** and actions **A** to new states **S'**. At each time, an agent receives a reward R that depends on

the current state s and the applied action a. Thus, the main problem is to find a control strategy or *policy* π that maximizes the expected reward V over time. For the discounted infinite-horizon case with any given discount factor γ, there is a policy π^* that is optimal regardless of the starting state and that satisfies the *Bellman* equation [12]:

$$V^{\pi}(s) = max_a\{R(s,a) + \gamma \sum_{s' \in \mathbf{S}} P(s'|s,a)V^{\pi}(s')\} \quad (1)$$

Two methods for solving this equation and finding an optimal policy for an MDP are: (a) dynamic programming and (b) linear programming [11].

In a factored MDP [13], the set of states is described via a set of random variables $\mathbf{S} = \{X_1, ..., X_n\}$, where each X_i takes on values in some finite domain $Dom(X_i)$. A state $\mathbf{x}$ defines a value $x_i \in Dom(X_i)$ for each variable X_i. Dynamic Bayesian networks (DBN) are used to describe the transition model concisely, so the post-action nodes (at the time t+1) contain smaller matrices with the probabilities of their values given their parents' values under the effects of an action.

4 Automatic Explanation Generation

4.1 Overview

The explanation generation mechanism is inspired on the explanations provided by a domain expert, and it combines several knowledge sources: (i) the MDP that represents the process and defines the optimal actions, (ii) a domain knowledge base, (iii) a set of templates, and (iv) an operator model.

The explanation system consists of three main stages (see Fig. 1). The relevant variable is the key factor to build an explanation, therefore in the first stage the relevant variable is obtained and additional elements as the current state S_i and the optimal action a^*. In a second stage, according to the operator model (novice, intermediate or advanced), a template is selected. In the last stage, the template is filled–in with information from a domain knowledge–base. Each of these stages is detailed in the following sections.

4.2 Explanation Template

The explanations are based on a predefined structure or *template*. Each template has a predefined structure and is configured to receive additional information from different sources. There are three types of templates according to the user: novice, intermediate and advanced. The main difference is the amount and depth of the information provided. For novice users is more detailed while for advance operators a more concise explanation is given.

The explanation template has three main components (see section 5):

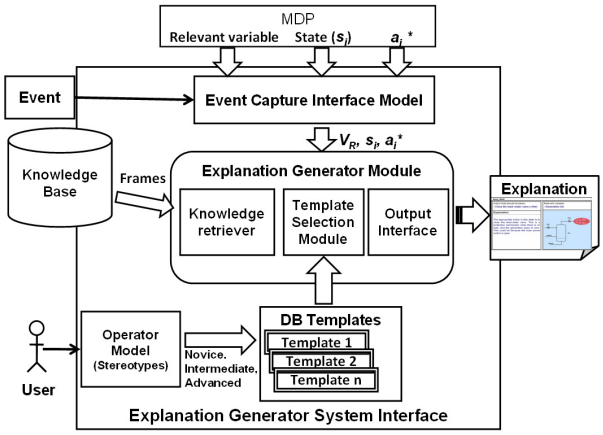

Fig. 1. Block diagram of the explanation generation system. When the trainee makes an error (event), the current state, optimal action and relevant variable are obtained from the MDP. These are used as pointers to obtain the relevant elements from the KB to fill–in an explanation template that is according to the user's model.

1. The optimal action given the current state.
2. A description in natural language of the main reasons for the previous action; which depends on the user level.
3. An schematic diagram of the process, highlighting the relevant variable.

The optimal action is obtained from the MDP given the current state of the plant, and the schematic diagram is previously defined for the process (or relevant subprocess). Thus, the two aspects that require a more complex procedure are the relevant variable and the verbal component, which are presented next.

4.3 Relevant Variable Selection

The strategy for automatic explanation generation considers as a first step, to find the most relevant variable, V_R, for certain state s and action a. All the explanations obtained from the experts are based on a variable which they considers the most important under current situation and according to the optimal policy. Examples of these explanations in the power plant domain are given later. We expect that something similar may happen in other domains, so discovering the relevant variable is an important first step for explanation generation.

Intuitively we can think that the relevant variable is the one with greatest effect on the expected utility, given the current state and the optimal policy. So as an approximation to estimating the impact of each factor X_i in the utility, we estimate how much the utility, V, will change if we vary the value for each variable, compared to the utility of the current state. This is done by maintaining all the other variables, X_j, $j \neq i$, fixed. The process is repeated for all the variables, and the variable with the highest difference in value is selected as the relevant variable.

Let us assume that the process is in state s, then we measure the relevance of a variable X_i for the state s based on utility, denoted by $rel_s^V(X_i)$, as:

$$rel_s^V(X_i) = \max_{s' \in neigh_{X_i}(s)} V(s') - \min_{s' \in neigh_{X_i}(s)} V(s') \tag{2}$$

where $neigh_{X_i}(s)$ is the set of states that take the same the values as s for all other variables X_j, $j \neq i$; and a different value for the variable of interest, X_i. That is, the maximum change in utility when varying the value of X_i, maintaining all the other variables fixed. This expression is evaluated for all the variables, and the one with the highest value is considered the most relevant for state s:

$$X_R^V = argmax_i(rel_s^V(X_i)), \forall(i) \tag{3}$$

4.4 Knowledge Base

To complement the information obtained from the MDP model, additional domain knowledge is required about relevant concepts, components and actions in the process. A *frame* [14] is a data structure with typical knowledge about a particular concept. Frames provide a natural way for representing the relevant elements and their relations, to fill–in the explanations templates.

In the KB, the frames store the basic knowledge about the domain components, variables, actions, and their relationships (see Fig. 2). This representation is an extension of the one proposed by [15]. The KB is conformed by three hierarchies: (i) procedures and actions, (ii) components, and (iii) variables. It also includes relationships between the frames in different hierarchies: which actions affect each component and the variables associated to each component.

Each frame contains general knowledge for each procedure, component or variable; which is relatively easy to obtain from written documentation or do-

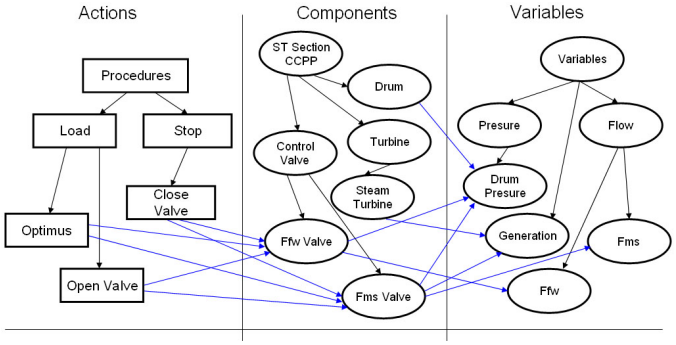

Fig. 2. The KB represented as a frame hierarchy; it is divided in 3 parts: (i) actions, (ii) components, and (iii) variables.

main experts. The advantage of this representation is that the same KB can be used to generate explanations for any procedure related to the specific domain.

4.5 Filling the Template

A template is selected according to the user level. The optimal action and relevant variable are deduced from the MDP model of the process and the policy, and together with the process schematic diagram are inserted in the corresponding template (Complete algorithm is depicted in figure 3). The missing element is the textual explanation of *why* the action should be selected in the current situation. The explanation mechanism must then determine *what* should be included and *how much* detail to give to the user. *What* is determined by the template structure and *how much* by the operator level.

Deduced from the expert explanations, the following elements are defined for the textual explanation structure:

1. Optimal action, including information on what is the purpose of the action obtained from the corresponding frame.
2. Relevant variable, with information of the relevance of this variable and its relation to certain component.
3. Component, including its main characteristics and its relation to the action.

```
ExplanationGeneration (a*, V_R, s, Comp, Exp, UsrLev)
    For each event e
    For each a*(s), V_R(x_i), s ∈ S, Comp, Exp, UsrLev;
    Where: a* = optimal action; V_R = relevant variable; s = current state;
           Comp = component; Exp = explanation; UsrLev = user level;
    IF e=true THEN {
            Get from MDP: a*(s)
            Get from ProbabilisticModel: V_R
            Get from Simulator: s
            Get from OperatorModel: UsrLev
            IF UsrLev = Novice THEN {
               Select Template = novice
               Get Frame(a*, Comp, V_R)
               Fill Template = Frame(a*, Comp, V_R) & s
               Exp = Template }
            ELSE IF UsrLev = Intermediate THEN {
               Select Template = Intermediate
               Get Frame(a*, V_R)
               Fill Template = Frame(a*, V_R) & s
               Exp = Template }
            ELSE IF UsrLev = Advanced THEN {
               Select Template = Advanced
               Get Frame(a*, V_R)
               Fill Template = Frame(a*, V_R)
               Exp = Template }
            }
    e = False;
    Return Exp;
```

Fig. 3. High–level algorithm for explanation generation.

The elements obtained from the MDP, state, action and relevant variables, are used as pointers to the corresponding frames in the KB from where the basic elements of the textual explanation are extracted. For intermediate and novice users the textual explanation is extended by following the links in the KB hierarchies, adding information from frames in the upper part of the hierarchy. For instance, if the important component is the *feed water valve*, it includes additional information from the *control valve* frame for intermediate users, and more general knowledge on *valves* for the novice user. In this way, the amount of detail in the explanation is adapted to the user level.

5 Experimental Results

First we describe the intelligent assistant in which we tested our method for explanation generation, and then the experiments comparing the automatic explanations against those given by a domain expert.

5.1 Intelligent assistant for operator training

We have developed an intelligent assistant for operator training (IAOT) [16]. The input to the IAOT is a policy generated by an MDP, which establishes the sequence of actions that will allow to reach the optimal operation of a steam generator. Operator actions are monitored and discrepancies are detected regarding the operator's expected behavior. The process starts with an initial state of the plant, usually under an abnormal condition; so the operator should return the plant to its optimum operating condition using some controls. If the action performed by the operator deviates from the optimal plan, either in the type of action or its timing, an explanation is generated.

We considered a training scenario based on a simulator of a combined cycle power plant, centered in the drum. Under certain conditions, the drum level becomes unstable and the operator has to return it to a safe state using the control valves. The variables in this domain are: (i) drum pressure (Pd), (ii) main steam flow (Fms), (iii) feed water flow (Ffw), (iv) generation (G), and (v) disturbance (this variable is not relevant for the explanations so is not included in the experiments). There are 5 possible actions: a0–do nothing, a1–increase feed water flow, a2–decrease feed water flow, a3–increase steam flow, and a4–decrease steam flow.

We started by defining a set of explanation units with the aid of a domain expert, to test their impact on operator training. These explanation units are stored in a data base, and the assistant selects the appropriate one to show to the user, according to the current state and optimal action given by the MDP. An example of an explanation unit is given in Figure 4.

To evaluate the effect of the explanations on learning, we performed a controlled experiment with 10 potential users with different levels of experience in

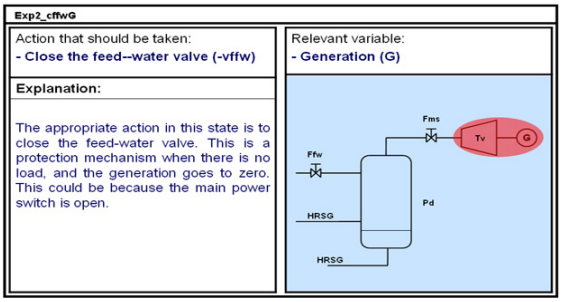

Fig. 4. An example of an explanation unit defined by a domain expert. In this example the relevant variable is *generation*, G, as the absence of generation is the main reason to close the feed–water valve.

power plant operation. An analysis of the results [16] shows a significant difference in favor of the group with explanations. These results give evidence that explanations help in the learning of skills such as those required to operate an industrial plant.

Next we analyze the results of the automatic explanation generation mechanism, in terms of a comparison with the human–generated explanations.

5.2 Results

Relevant Variable Selection.- In a first stage we evaluated the selection of the most relevant variable, and compared these to the ones given by the domain expert. In the power plant domain there are 5 state variables with a total of 384 states. We analyzed a random sample of 30 states, nearly 10% of the total number of states. For the 30 cases we obtained the most relevant variable(s); and compared these with the relevant variables given in the explanation units provided by the expert.

Figure 5 presents the results of 11 of the 30 cases (the rest are omitted due to space limitations). For each case we show: (i) the current state, (ii) the value, (iii) the optimal action, (iv) the variable selected according to the change in utility, including this change, and (v) the relevant variable(s) given by the expert. Note that for some cases the expert gives two relevant variables. The system selects in 100% of the 30 cases one of the most relevant variables according to the expert. In the future we plan to extend the relevant variable selection mechanism so we can obtain more than one relevant variable.

Explanation Generation.- To evaluate the explanations generated by the system, we generated explanations for different situations, and compared this to

Test	selected S Var fms	ffw	d	pd	g	U	Π*	Experimental results Changes in Utility |ΔU|	Relevant Var Selected by Expert
1	0	0	0	0	1	2601.29	a1	g = 2132.44	**g,fms**
2	0	0	1	3	1	2514.00	a2	g =2235.79	**g,fms**
3	1	1	0	4	0	796.95	a2	fms = 2413.53	**fms, g**
4	2	0	0	7	1	3488.60	a4	pd =2762.60	**pd, fms**
5	3	0	0	0	1	3295.55	a3	g =2427.94	**g, pd**
6	3	1	0	2	1	3053.19	a2	g = 2109.73	**g, pd**
7	4	0	1	0	1	2986.18	a1	g = 2257.11	**g, pd**
8	4	1	1	7	0	843.27	a4	pd = 3271.43	**pd, g**
9	5	1	0	1	1	3632.66	a0	fms = 3720.05	**fms**
10	5	1	1	1	1	3287.13	a0	fms = 3642.53	**fms**
...	...	...	...	...	...	...	...	...	**...**
30	5	0	0	5	1	2761.32	a4	fms = 2116.43	**fms**

Fig. 5. Results for the relevant variable selection phase. Is shows for each test case: the state (random variables), the expected value (U), the optimal action (Π^*), the relevant variable with $\mid \Delta U \mid$, and the variable(s) selected by the expert.

the explanations given for the same state by the domain expert. Figure 6 depicts an example of an explanation template generated. It contains on the left side: (i) the optimal action, (ii) why is important to do this action, (iii) which component is related with optimal action, and (iv) a description of the current state. In the right side the relevant variable is highlighted in an schematic diagram of the process.

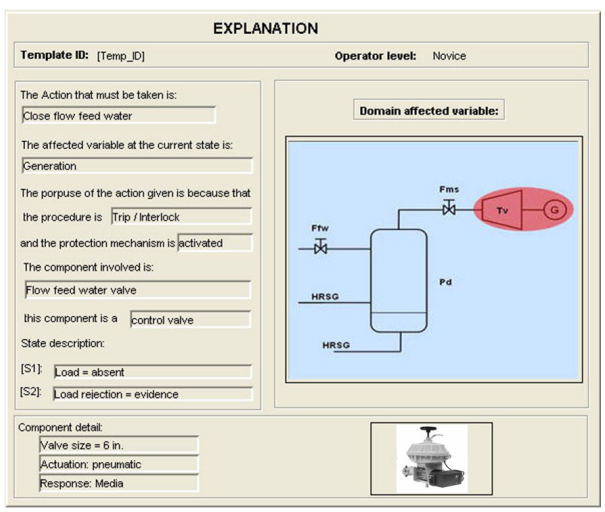

Fig. 6. An example of an explanation template generated.

To evaluate explanations generated by the system, the the methodology proposed by [17] was adapted. The main idea is that an expert panel compares

explanations given by the system to those given by a human, in terms of some evaluation criterium. It considers four aspects: (i) coherence, (ii) contents, (iii) organization, and (iv) precision. This not intend to generate explanations in natural language. Each criterium is evaluated in a quantitative scale: 1=bad, 2=regular, 3=good, 4=excellent. Each expert is presented with two explanations, side by side, for the same case; one given by a human expert, the other generated by the system. It is asked to grade each aspect for the generated explanations taking as reference the expert's explanations. 15 different explanations were considered in a variety of situations for the power plan domain, and asked 6 different external domain experts to evaluate them. Results are summarized in table 7.

Results are between *good* and *excellent* in average for all the aspects and for all the experts. The criterium with highest score is *organization* and the lowest *contents*, although the gap is small among all the aspects. We consider that these are encouraging results, as we are comparing the explanations generated automatically against those given by an experienced domain expert, a very high standard. Given these results, and the previous results on the impact of the explanations on learning [16], a similar impact is expected with automatically generated explanations; however, the plan is to conduct an additional user study in the future.

CRITERIA	CASES	EVALUATION PANEL						SCORE	
		Eval 1	Eval 2	Eval 3	Eval 4	Eval 5	Eval 6	Average	Ranking
COHERENCE	*C1 to C15*	3.67	3.47	2.47	3.80	3.67	3.07	**3.36**	**2º**
CONTENT	*C1 to C15*	2.00	3.40	2.80	3.60	3.87	3.53	**3.20**	**4º**
ORGANIZATION	*C1 to C15*	2.87	2.93	2.93	4.00	3.93	3.67	**3.39**	**1º**
CORRECTNESS	*C1 to C15*	4.00	2.87	2.47	3.60	3.67	2.87	**3.24**	**3º**
								3.30	
Total		**3.13**	**3.17**	**2.67**	**3.75**	**3.78**	**3.28**		

Fig. 7. Results of the evaluation by a panel of experts. The table shows the average score per criteria–evaluator, and the totals for each dimension.

6 Conclusions and future work

We have developed a novel explanation generation mechanism for the actions recommended by an intelligent assistant based on an MDP. For this we developed and algorithm to determine the most relevant variable for certain state–action; and combine this information with domain knowledge to generate explanations based on a general template and adapted according to the user level. We have evaluated the explanation generation system in the power plant domain with good results. We consider that this mechanism can be easily extended to other training and assistant systems based on decision-theoretic models.

Currently the explanations are centered on why the recommended action should be selected, but not on why other actions are not appropriate. Another

limitation is that it considers only one optimal plan, and there could be alternative solutions. In the future we plan to extend the system to consider other types of explanations, and to apply it to other domains.

References

1. Herrmann, J., Kloth, M., Feldkamp, F.: The role of explanation in an intelligent assistant system. In: Artificial Intelligence in Engineering. Volume 12., Elsevier Science Limited (1998) 107–126
2. Druzdzel, M.: Explanation in probabilistic systems: Is it feasible? Will it work? In: Intelligent information systems V, Proceedings of the workshop, Poland (1991) 12–24
3. Renooij, S., van der Gaag, L.: Decision making in qualitative influence diagrams. In: Proceedings of the Eleventh International FLAIRS Conference, Menlo Park, California, AAAI Press (1998) 410–414
4. Lacave, C., Atienza, R., Díez, F.: Graphical explanations in Bayesian networks. In: Lecture Notes in Computer Science. Volume 1933., Springer-Verlag (2000) 122–129
5. Bielza, C., del Pozo, J.F., Lucas, P.: Optimal decision explanation by extracting regularity patterns. In Coenen, F., Preece, A., Macintosh, A., eds.: Research and Development in Intelligent Systems XX, Springer-Verlag (2003) 283–294
6. Lacave, C., Luque, M., Díez, F.J.: Explanation of Bayesian networks and influence diagrams in Elvira. IEEE Transactions on Systems, Man and Cybernetics—Part B: Cybernetics **37** (2007) 952–965
7. Givan, R., Dean, T., Greig, M.: Equivalence notions and model minimization in markov decision processes. Artif. Intell. **147**(1-2) (2003) 163–223
8. Dean, T., Givan, R.: Model minimization in markov decision processes. In AAAI, ed.: In Proceedings AAAI-97, Cambridge, Massachusetts, MIT Press (1997) 106–111
9. Munos, R., Moore, A.: Variable resolution discretization for high-accuracy solutions of optimal control problems. In: IJCAI '99: Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence, San Francisco, CA, USA, Morgan Kaufmann Publishers Inc. (1999) 1348–1355
10. Khan, O.Z., Poupart, P., Black, J.: Explaining recommendations generated by MDPs. In et al., T.R.B., ed.: 3rd International Workshop on Explanation-aware Computing ExaCt 2008, Patras, Greece, Proceedings of the 3rd International ExaCt Workshop (2008)
11. Puterman, M.: Markov Decision Processes: Discrete Stochastic Dynamic Programming. Wiley, New York (1994)
12. Bellman, R.: Dynamic Programming. Princeton U. Press, Princeton, N.J. (1957)
13. Boutilier, C., Dean, T., Hanks, S.: Decision-theoretic planning: structural assumptions and computational leverage. Journal of AI Research **11** (1999) 1–94
14. Minsky, M.: A framework for representing knowledge. In Winston, P., ed.: The Psychology of Computer Vision, McGraw-Hill (1975)
15. Vadillo-Zorita, J., de Ilarraza, A.D., Fernández, I., Gutirrez, J., Elorriaga, J.: Explicaciones en sistemas tutores de entrenamiento: Representacion del dominio y estrategias de explicacion, Pais Vasco, Espa*ñ*a (1994)
16. Anonymous. (2005)
17. Lester, J.C., Porter, B.W.: Developing and empirically evaluating robust explanation generators: the knight experiments. Computational Linguistics **23**(1) (1997) 65–101