Robot Planning under Uncertainty with Unpredictable Events

Stefan J. Witwicki[1], Francisco S. Melo[1], Jesús Capitán Fernández[2], Matthijs T.J. Spaan[3], and José Carlos Castillo Montoya[4]

[1] INESC-ID / Inst. Sup. Técnico, UTL, Portugal, {witwicki,fmelo}@inesc-id.pt
[2] University of Duisburg-Essen, Germany, jesus.capitan@uni-due.de
[3] Delft University of Technology, The Netherlands, m.t.j.spaan@tudelft.nl
[4] ISR, Institute Superio Técnico, UTL, Portugal, jccastillo@isr.ist.utl.pt

Abstract. In planning robot behavior with a Markov decision process (MDP) framework, there is the implicit assumption that the world is predictable. Practitioners must simply take it on good faith the MDP they have constructed is comprehensive and accurate enough to model the exact probabilities with which all events may occur under all circumstances that the robot may encounter. Here, we challenge the conventional assumption of complete predictability, arguing that some events are inherently *unpredictable*. Towards more effectively modeling problems with unpredictable events, we develop a flexible framework that explicitly distinguishes decision factors whose probabilities are not assigned precisely while still representing known probability components using conventional principled MDP transitions. Our approach is also flexible, resulting in a factored model of variable abstraction whose usage for planning results in different levels of approximation. We illustrate the usage of our modeling framework in a robot surveillance domain.

1 Introduction

Modeling an intelligent agent acting in an uncertain environment is challenging. For this purpose, researchers have developed elegant mathematical frameworks, such as the Markov Decision Processes (MDPs), that encode all states of the environment, actions, and transitions, as a dynamical system [1]. However, in order to apply these frameworks to robots, there are inherent obstacles that the practitioner must overcome, as robots interact directly with the real world (instead of with a model of it).

First, it is intractable to model the real world comprehensively or with any extensive level of detail. Instead, the practitioner should choose an appropriate depth of abstraction, such as a coarse topological map instead of a detailed metric map. Second, the practitioner must select which features to include in the environment state, related to the task of the robot. Not only should these features capture the critical events on which robots should base smart decisions, but they should also comprise a system whose dynamics are self-contained. In particular, the probability of a next state must be an ascertainable function of

the previous values of the selected features (and only their latest values, in the case of a Markov model). This means that all modeled events must be strictly predictable (from modeled features) and their probabilities accurately prescribed. Obtaining these probabilities can be cumbersome or impossible in case of robotic scenarios. The amount of real-world trials to obtain reliable statistical models is prohibitive, while models obtained from simulations are likely to be inaccurate.

In this paper, we propose an alternative framework that relaxes the conventional assumption of complete predictability. We take the position that, from an agent's perspective, an event may be inherently unpredictable. This could be because the event's underlying causes are prohibitively complex to model as part of the agent's state, or because circumstances surrounding the event reside in a portion of the environment that the agent cannot sense. Yet another reason to label an event as *unpredictable* could be that it is so rare as to preclude an accurate estimate of transition probabilities (neither through collected data nor through expert knowledge).

We contend that the agent should treat the occurrences of unpredictable events with corresponding features that it explicitly distinguishes from the conventional, predictable features using a factored model. We show that, independently of the complexity required to accurately model the dynamics of unpredictable features, equivalently accurate predictions are obtained by a model that depends only on the history of observable features. In constructing such a model, the agent avoids assigning arbitrary probabilities to the occurrences of unpredictable events. Yet it retains the ability to plan for all possible future paths, while accounting for known probability components associated with predictable feature values.

Our framework has several other advantages over conventional modeling options. First, it is simpler for a practitioner to specify the model, since some of the hard-to-estimate probabilities can be avoided. Second, it avoids the computational complexity of modeling additional features that only enable weak prediction of rare events. Third, our modeling approach naturally circumvents errors associated with probabilities assigned to unpredictable events. Our approach is also flexible in the model that it produces. At one end of the dial, the practitioner can specify a dependence on complete histories of observable features, yielding optimality guarantees but at a computational cost. We also contribute a principled approach on how such dependence can be alleviated by varying the order of the history dependence. We expect such approximation, in practical situations, to strike an effective balance in computational performance and the quality of approximation.

2 Motivating Scenario

As a motivating example, we will use a scenario with surveillance activities. A robot moves within the simplified surveillance environment in Fig. 1, corresponding to a floor of the Institute for Systems and Robotics (ISR) in Lisbon. For robot navigation, the position of the robot is defined as its location in a topological map with 8 possible positions (nodes in Fig. 1). In addition to the robot, there

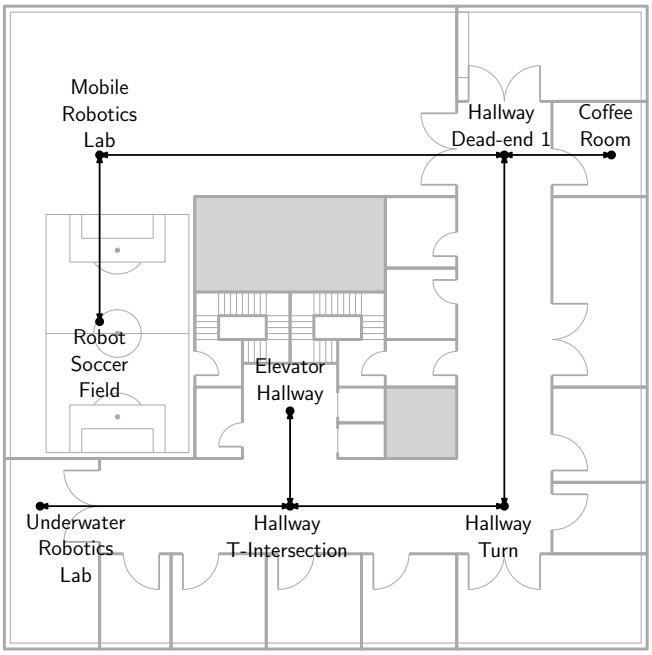


Fig. 1. Outline and topological representation of the ISR scenario.

is a network of video cameras that is able to detect events such as a *fire* in the Coffee Room and *visitors* at the Elevator Hallway who require assistance.

Thanks to its local sensors and a path planner, the robot can move from location to location by selecting high-level navigation actions $\{N, S, E, W\}$ corresponding to the four cardinal directions. Nevertheless, the underlying machinery is not perfect, sometimes resulting in failed navigation actions, which we can reliably predict using a Markovian probabilistic model. For example, taking the action N at the Hallway T-intersection moves the robot successfully to the Elevator Hallway with a particular probability. The robot is in charge of completing several tasks, namely:

Surveillance of the environment. The robot should maintain under close surveillance the Underwater Robotics Lab and the Robot Soccer Field, where valuable items are stored, and the Coffee Room and the Elevator Hallway, to complement the surveillance network in the task of detecting fire and people arriving.

Fire assistance. If a fire is detected, the robot should head to the Coffee Room to assist in putting out the fire.

Assistance to visitors. If a person arrives at the Elevator Hallway and requires assistance, the robot should head to that location to assist the person.

Associated with each of these tasks is a relative priority value; the objective of the robot is to plan its movement so as to balance its expected completion of tasks given these priorities.

3 Background

In this section, we review the conventional factored MDP framework [2] for planning activities. This formalism sets the stage for our approach, but it presents some challenges when modeling events in problems such as our example.

3.1 Conventional (Factored) MDP Model

A Markov Decision Process (MDP) $\mathcal{M} = (\mathcal{X}, \mathcal{A}, \mathsf{P}, r, \gamma)$ can be used to model tasks related to robot planning in a factored manner [2]. The components are the following: (i) the *state space* is $\mathcal{X}$ (in factored models, this is the cartesian product of several feature spaces $\mathcal{X}_p \times \mathcal{X}_s \times \cdots \times \mathcal{X}_f$); (ii) the *action space* is $\mathcal{A}$; (iii) the *transition function* (it can be factored) is $\mathsf{P} : \mathcal{X} \times \mathcal{A} \times \mathcal{X} \mapsto \mathbb{R}$, and it encodes the probabilities of next state $X(t+1)$ as a Markov function of current state $X(t)$ and action $A(t)$; (iv) the *reward function* (it could be factored too) is $r : \mathcal{X} \times \mathcal{A} \mapsto \mathbb{R}$, and it encodes the reward obtained given the current state $X(t)$ and action $A(t)$; (v) the *discount factor* is γ, which weighs future rewards when computing the total expected reward accumulation.

To model the example problem introduced in Section 2, we can represent the state space as a set $\mathcal{X} = \mathcal{X}_p \times \mathcal{X}_s \times \mathcal{X}_f \times \mathcal{X}_w$. The state factor, or feature, $X_p(t) \in \mathcal{X}_p$ encodes the position of the robot at time t (which is any of of the labeled graph nodes in Fig. 1). The remaining features encode the statuses of the robot's completion of its various tasks.

We associate with the *surveillance task* feature $X_s(t)$ that indicates which of the target locations have been visited recently (at time t). This feature takes values in $\mathcal{X}_s = \{0, \ldots, 15\}$, which is the decimal representation of a 4-bit sequence corresponding to 4 flags indicating the target locations that have been visited (see Fig. 2, left). When one of the target locations is visited, the corresponding flag is set to 1. Whenever $X_s(t) = 15$, then $X_s(t+1) = 0$, indicating that the robot should repeat its surveillance of all target sites.

We associate with the *fire assistance task* binary feature $X_f(t)$ that indicates whether a fire is identified as active at time t. It is set to 1 when a fire is detected in the Coffee Room. After the robot visits that room, this feature is reset to 0, indicating that the robot has successfully put out the fire.

And we associate with the *waving visitor assistance task* binary feature $X_w(t)$ that, when set to 1, indicates that the system has detected a person waving in the Elevator Hallway. Only after the robot visits the Elevator Hallway and assists the visitor is the flag reset to 0.

Thanks to the factored structure of the model, the transition probabilities may be encoded potentially more compactly using a 2-stage Dynamic Bayesian Network like the one in Fig. 2 (right). In this case, the set of possible actions is $\mathcal{A} = \{N, S, E, W\}$. The factored structure also allows for a compact representation of the transition probabilities. In fact, the transition probabilities associated with a state-factor X_s, for example, can be represented as a conditional probability table (CPT) that represents the probability distribution

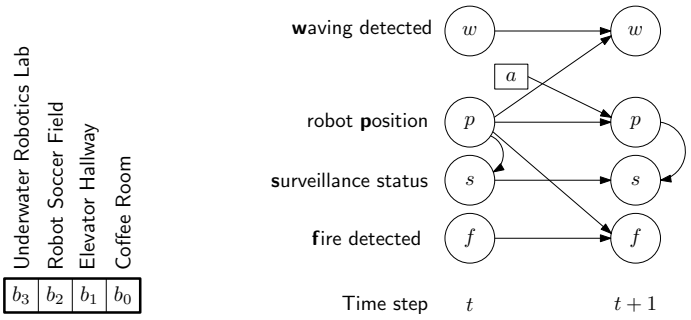


Fig. 2. (Left) A binary representation of state-feature X_s, where each bit indicates whether or not the corresponding location has been visited recently. (Right) A DBN representation of the dependencies in the state-transitions of the MDP.

$\mathbb{P}[X_s(t+1) \mid X_s(t), A(t), X_p(t+1)]$, with one entry for each combination of action and values of $X_s(t)$, $X_p(t+1)$ and $X_s(t+1)$. The transition probabilities associated with each action $a \in \mathcal{A}$ can then easily be obtained from the product of such CPTs.

Similarly, we can also specify a factored reward function that separately represents the priorities of the individual tasks.

$$r(x) = w_s r_s(x) + w_f r_f(x) + w_w r_w(x). \tag{1}$$

Each component r_i encodes the goals of a task, and (1) indicates that the robot should complete all tasks. The weights w_i indicate the relative importance/priority of the different tasks. For concreteness, we define the reward components as

$$\begin{aligned} r_s(x) &= \mathbf{1}_{\{x_s=15\}}(x), \\ r_f(x) &= -\mathbf{1}_{\{x_f=1\}}(x), \\ r_w(x) &= -\mathbf{1}_{\{x_w=1\}}(x), \end{aligned}$$

where the operator $\mathbf{1}$ works as follows: component r_s rewards those states in which the robot recently visited all critical locations (corresponding to $x_s = 15$); component r_f penalizes those states in which a fire is active ($x_f = 1$); and component r_w penalizes those states in which waving has been detected but the robot did not yet respond ($x_w = 1$).

3.2 Planning in the Conventional MDP Model

Planning in a conventional MDP model consists of determining a policy (an action selection rule) that maximizes the total reward accumulated by the agent throughout its lifetime. Formally, this amounts to determining the policy $\pi : \mathcal{X} \mapsto \mathcal{A}$ that maximizes the corresponding *value* for every state $x \in \mathcal{X}$,

$$V^\pi(x) = \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t r(X(t), A(t)) \mid X(0) = x \right],$$

where the expectation is taken with respect to trajectories $\{X(t), t = 0, \ldots\}$ induced by the actions $A(t)$, which in turn are selected according to the rule $A(t) = \pi(X(t))$. The policy with maximal value is known as the *optimal policy*, and its corresponding value function, V^*, the *optimal value function*. The optimal value function V^* is known to verify

$$V^*(x) = \max_{a \in \mathcal{A}} \mathbb{E}_{Y \sim \mathsf{P}(x,a)} \left[r(x,a) + \gamma V^*(Y) \right],$$

and it is possible to define the *action-value function* Q^* as

$$\begin{aligned} Q^*(x,a) &= \mathbb{E}_{Y \sim \mathsf{P}(x,a)} \left[r(x,a) + \gamma V^*(Y) \right] \\ &= \mathbb{E}_{Y \sim \mathsf{P}(x,a)} \left[r(x,a) + \gamma \max_{b \in \mathcal{A}} Q^*(Y,b) \right]. \end{aligned}$$

The recursive relation above can be used to iteratively compute $Q^*(x,a)$ for all $(x,a) \in \mathcal{X} \times \mathcal{A}$, a dynamic programming method known as *value iteration*. From Q^*, the optimal policy can then be trivially computed as

$$\pi^*(x) = \operatorname*{argmax}_{a \in \mathcal{A}} Q^*(x,a).$$

3.3 Modeling Events

Using a factored model such as the one just described, an event may be modeled by simply associating a boolean state feature with the occurrence of the event [3, 4]. In this paper, we restrict consideration to uncontrollable events:

Definition 1. *An* ***uncontrollable event*** *i corresponds to a boolean feature $X_{Ui} \in \{0,1\}$, such that*

- *i is said to occur when X_{Ui}'s value changes from 0 to 1;*
- *Given current state x, the probability of occurrence at time t+1 is independent of the action a taken at time t:*

$$\mathbb{P}[X_{Ui}(t+1)=1 \mid X(t)=x, A(t)=a] = \mathbb{P}[X_{Ui}(t+1)=1 \mid X(t)=x].$$

For instance, in our example problem X_w corresponds to the uncontrollable event that *waving* is detected. As long as we have included features in our current-state representation that together encode sufficient information for predicting the occurrence of the event in the next state, then our Markovian transition model is perfectly suitable.

4 Modeling Unpredictable Events

For some events, however, it may not be possible to accurately prescribe occurrence probabilities. An instance is the *person waiving* event from our running example, which is represented using feature X_w. For illustrative purpose, consider assigning the occurrence probability $\mathbb{P}[X_w(t+1) = 1 \mid X_w(t) = 0]$ as a small

constant p_{waiving}. However, if the transition model is presumed to have been estimated from tracking actual people waving upon arrival at the elevator, there is simply not enough data to accurately predict the such detections from previous detections. In other words, the true transition probabilities of this feature has a high degree of uncertainty. Therefore, the assignment of p_{waiving} is bound to be arbitrary. Our event model is, at best, an approximation.

Definition 2. *An **unpredictable event** is an uncontrollable event whose occurrence probability cannot be accurately estimated as a Markov function of the latest state and action.*

We now describe two solution approaches for modeling such events. The first approach augments the conventional model with additional features that effectively render the event predictable. In the second approach, we devise a model wherein unpredictable events are explicitly treated as special factors whose CPTs are not assigned precisely, and provide a formal method for recasting the problem as a bounded-parameter model.

4.1 Expanded Event Model (EEM)

To help the robot to better predict whether or not a person is waiving, consider additionally modeling the underlying process of how the detection takes place. In general, an unpredictable event could be made predictable if the factored model were to include additional features (with known transitions) sufficient for predicting the event. Figure 3 gives the reader a flavor of what such a model might look like. In particular, we have expanded our original model with additional features that enable a better prediction of whether an actual waving event took place. In particular, we include X_v: whether or not a visitor is actually waiting which is the primary cause of waving detections (and is actually the feature that we would like the robot to respond to). However, there are additional factors. If people from the underwater robotics lab are conversing in the hallway, it is much more likely for the system to pick up their small motions and interpret them as a person waiving. The motion of the cleaning lady sweeping the hallway is also very likely to be confused with a person waving in the hallway. Lastly, the sensitivity of the detection algorithm (in turn dependent on lighting conditions and the time of day) also impacts the likelihoods of false detections (false positives) and missed detections (false negatives).

The challenge with this expanded event model (EEM) is that it makes the robot's decision-making problem more complex. We doubled the number of features in our state representation, thereby increasing our state space by an order of magnitude or more. Furthermore, the robot may not be able to directly observe the added features due to sensory limitations of the system. In this case the problem would become a Partially Observable Markov Decision Process (POMDP) [5], introducing significant additional complexity (besides that caused by the state space expansion).

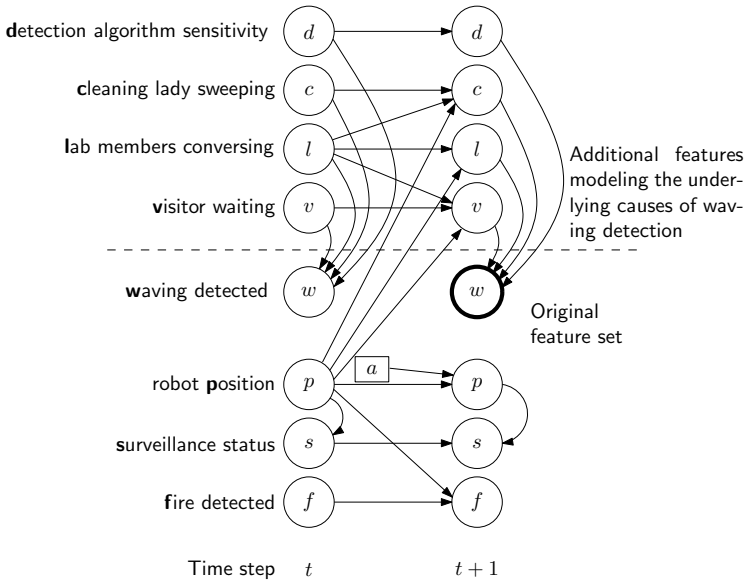


Fig. 3. Expanded factored model for predicting X_w.

4.2 Boundedly-Predictable Event Model (BPEM)

Another challenge with the aforementioned modeling approach is that it assumes that we are able to construct a well-specified Markov model of the underlying causes of all events. These may themselves be difficult to model or predict, requiring additional layers of causes that underly the underlying causes, thereby combinatorially exploding the augmented model. Here we develop a flexible and principled approach for leaving out any or all of these additional factors.

The idea is to treat some features as *external* to the agent. Given a prior distribution over external feature values but a lack of agent observability of these values, these variables can be marginalized out of the EEM. The result is a reduced model that highlights CPT entries (denoting event occurrence probabilities) that are not precise. Instead, these are bounded probability values. Fortunately, the bounds on these parameters may be tightened given knowledge about the dynamics of the process underlying the events.

Theorem 1. *If we choose a set of external features X_E that serve to inform predictions of events encoded by feature set X_U, and all remaining internal features X_I, such that:*

1. *the state is factored into $X = \langle X_E, X_U, X_I \rangle$,*
2. *at each time t, the agent is assumed to directly observe $O(t) = \langle X_U(t), X_I(t) \rangle$ but not the external feature values $X_E(t)$,*
3. *the external features are conditionally independent of the agent's action $\mathbb{P}[X_E(t+1)|X(t), A(t)] = \mathbb{P}[X_E(t+1)|X_E(t), X_U(t), X_I(t)]$, and*
4. *the internal features are conditionally independent of the external features as well as concurrent events $\mathbb{P}[X_I(t+1)|X(t), A(t), X_U(t+1)] = \mathbb{P}[X_I(t+1)|X_U(t), X_I(t), A(t)]$,*

then maintaining an alternate state representation

$$X'(t) = \langle X_U(0 \ldots t), X_I(0 \ldots t) \rangle, \tag{2}$$

is sufficient for predicting the events:

$$\mathbb{P}[X_U(t+1)|A(0 \ldots t), O(0 \ldots t)] = \mathbb{P}[X_U(t+1)|X_U(0 \ldots t), X_I(0 \ldots t)]\,. \tag{3}$$

Proof (Sketch). The equality in Equation 3 is proven in two steps. First, consider that, since all internal feature values $X_I(0 \ldots t)$ are observed, as well as the event feature values $X_U(0 \ldots t)$, and all external feature values unobserved, the left-hand side can be rewritten as follows:

$$\mathbb{P}[X_U(t+1)|A(0 \ldots t), O(0 \ldots t)] = \mathbb{P}[X_U(t+1)|X_U(0 \ldots t), X_I(0 \ldots t), A(0 \ldots t)]\,. \tag{4}$$

All that remains to reduce Equation 4 to the right-hand side of Equation 3 is to prove that $X_U(t+1)$ is conditionally independent of $\{A(0 \ldots t)\}$ given evidence $\{X_I(0 \ldots t), X_U(0 \ldots t)\}$. This holds as consequence of the *d-separation* relationship [6] shown in Fig. 4. □

Corollary 1. *Given properties 1–4 in Theorem 1, alternate state representation* $X'(t) = D(t)$*, where* $D(t) \subseteq \{X_U(0 \ldots t), X_I(0 \ldots t)\}$ d-separates $X_U(t+1)$ *and* $\{A(0 \ldots t), \{X_U(0 \ldots t), X_I(0 \ldots t)\}/D(t)\}$*, is sufficient for predicting the events.*

The implication of the Theorem and Corollary 1 is that we can capture all the relevant effects of unobservable external variables without explicitly modeling the external variables. Whether or not we have the knowhow or capability to model external variables, a model containing only observable event-effectors is sufficient for making predictions. Fig. 5 portrays a reduced model for our running example, where only the robot's position history and the detection history suffice to predict future waving detections *regardless of* how complex the external waving detection process is.

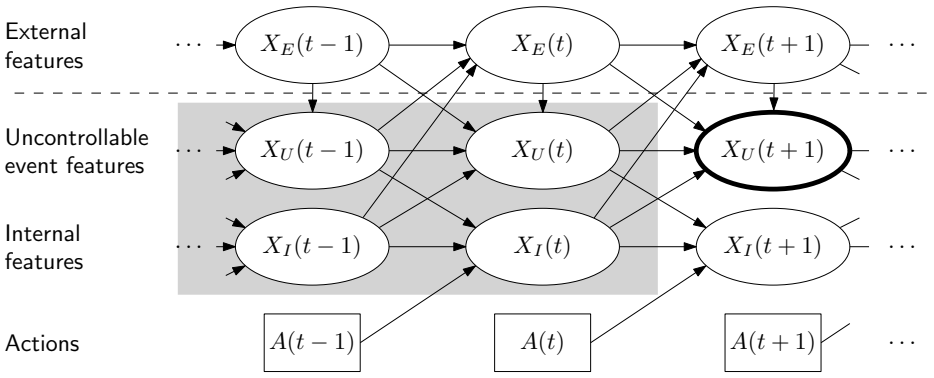


Fig. 4. Illustration of the d-separation of $X_U(t{+}1)$ and $\{A(0 \ldots t)\}$ by the grayed region.

4.2.1 Inferring a reduced model. Given a model of event-underlying external variables, we can compact our model without losing predictive power, by simply marginalizing these variables out. Reducing our model becomes an inference problem that updates the conditional probability tables of the affected event features as the affecting external variables are removed. The equations below describe this inference problem as an iterative process that computes three kinds of terms for each decision stage. The first term, which we refer to as the *joint-external-event* distribution, or $\boldsymbol{J}(t)$ at decision stage t, effectively merges the two variables $X_E(t)$ and $X_U(t)$ into one node. The second term is the *marginal* event distribution, $\boldsymbol{M}(t)$, which is induced by marginalizing over the first term. The third term, which we call the *induced-external* distribution $\boldsymbol{IE}(t)$, is used for computing the joint-external-event distribution of the next decision stage.

For stage $t = 0$, $\boldsymbol{IE}(0) \equiv \mathbb{P}[X_E(0)|X_U(0), X_I(0)] = \dfrac{\mathbb{P}[X_U(0)|X_E(0)]\,\mathbb{P}[X_E(0)]}{\mathbb{P}[X_U(0)]}$

For stages $t \geq 1$:

$$\boldsymbol{J}(t) \equiv \mathbb{P}[X_E(t), X_U(t)|X_U(0\ldots t{-}1), X_I(0\ldots t{-}1)] = \sum_{X_E(t-1)} \Big(\mathbb{P}[X_U(t)|X_E(t), X_E(t{-}1), X_U(t{-}1), X_I(t{-}1)]\; \mathbb{P}[X_E(t)|X_E(t{-}1), X_U(t{-}1), X_I(t{-}1)]\; \boldsymbol{IE}(t{-}1)\Big) \quad (5)$$

$$\boldsymbol{M}(t) \equiv \mathbb{P}[X_U(t)|X_U(0\ldots t{-}1), X_I(0\ldots t{-}1)] = \sum_{X_E(t)} \boldsymbol{J}(t) \quad (6)$$

$$\boldsymbol{IE}(t) \equiv \mathbb{P}[X_E(t)|X_U(0\ldots t), X_I(0\ldots t)] = \frac{\boldsymbol{J}(t)}{\boldsymbol{M}(t)} \quad (7)$$

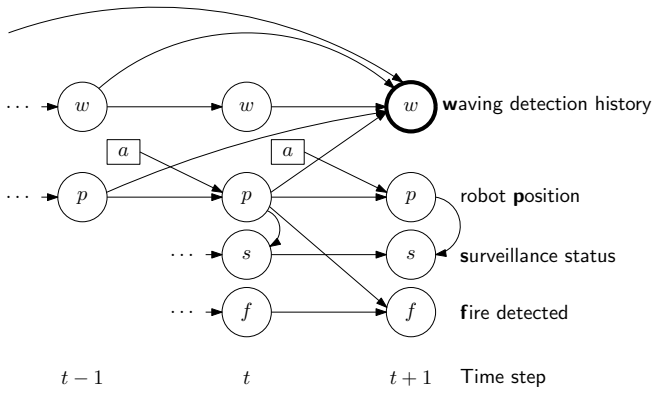


Fig. 5. An equivalent reduction of the model in Figure 3.

Upon augmenting the state with the necessary history ($X'(t) = \langle X_U(0 \dots t), X_I(0 \dots t) \rangle$, in the worst case), Equations 5–7 allow us to complete our specification of the reduced model. The marginal distribution $\boldsymbol{M}(t)$ defines the new CPT of the event features $X_U(t+1)$ in terms of old CPT entries from the EEM. As we reduce the EEM, the CPTs for all other internal state variables remain unaffected, since they are conditionally independent of past external variables given the event features.

Note that we are replacing the external variables with histories of event features, but that we expect that this is a reasonable trade-off in the case that many external variables can be eliminated. Moreover, depending on the dynamics of the problem, histories of events can often be encoded compactly such as by encoding the past times that the robot visited the elevator hallway, or that waving was detected, rather than the whole bit sequence. Note also that in modeling problems with our framework, there is flexibility in terms of which and how many variables we choose to marginalize out. Marginalizing out fewer external variables could, for instance, decrease the history dependence (though if these variables are unobservable, we would have a POMDP instead of a history-augmented MDP).

4.2.2 Dependence of Rewards on External Features. For some problems, it may be desirable to prescribe rewards that depend on external variables from the expanded event model (i.e. X_E is in the scope of the (PO)MDP reward function $r : \mathcal{X} \times \mathcal{A} \mapsto \mathbb{R}$). In particular, in our example problem, it would be more useful to reward the robot based on whether or not a person is waiting for assistance *instead of* on whether or not waving has been detected. Otherwise, a policy that causes the robot to respond to false positives (e.g. waving events in the absence of a person waiting for assistance) would be unfairly credited.

The good news is that, even with reward dependence on external variables in the EEM, we can eliminate these external variables from our model using the above iteration (Equations 5–7). One additional inference step is required for each stage t: computation of an induced reward function $r\prime()$, whose scope only includes X_U, X_I, and A (easily derived as a function of the EEM reward $r()$).

$$
\begin{aligned}
&r\prime(X_U(0 \dots t), X_I(0 \dots t), A(t)) \\
&\quad = \sum_{X_E(t)} \mathbb{P}[X_E(t) | X_U(0 \dots t), X_I(0 \dots t)]\, r(X_E(t), X_U(t), X_I(t), A(t)) \\
&\quad = \sum_{X_E(t)} \underbrace{\boldsymbol{IE}(t)}\, r(X_E(t), X_U(t), X_I(t), A(t)) \qquad (8)
\end{aligned}
$$

4.2.3 Propagating Bounds. If the dynamics of some or all of the external variables are not known, or imprecisely specified, then the model reduction above results in a bounded probability distribution for predicting the events. Here, we assume that the CPT entries involving X_E in the EEM are each represented with a lower and upper bound $[p_{lb}, p_{ub}]$. These bounds should in turn be propagated to the reduced model as the external variables are marginalized out.

We can still apply Equations 5–8, but we should do so twice, so as to compute an upper and lower bound for each entry of the distribution. In particular, to compute lower(upper) bounds, each term indicated with a faint bracket underneath should be substituted with the lower(upper) bound for that term, and each term indicated with a bracket above should be substituted with the upper(lower) bound.[5] Likewise with each entry in the induced reward function $r\prime()$, with special attention paid to negative rewards.

Some bounds may be tighter than others. For instance, although we may not know the probabilities of faultiness in the coffee-machine wiring or of how likely this is to cause a fire, we can more easily collect data about bystander and coffee machine usage and patterns. Moreover, we may be certain that if the coffee machine is not on, then no fire will break out. This knowledge may improve the bounds that are propagated through marginalization, tightening the bounds on various parameters of our reduced model.

The resulting factored model is an instance of a bounded-parameter MDP (BMDP) [8], and so solution methods for BMDPs can be readily applied. In contrast to the general BMDP, our model has the advantage that we have explicitly highlighted which parameters are uncertain. If we were to encode the problem without performing such factorization, and systematically propagating uncertainty, we would be left with the relatively more daunting task of assigning bounds to all parameters for all states in the transition matrix.

4.2.4 Approximating the event distribution. Although we have eliminated the external variables, we are left with a history dependence whose computational consequences may be undesirable. Moreover, if the horizon of the planning problem is infinite, maintaining a dependence on the entire history is untenable. Fortunately, there are principled methods for approximating such distributions finitely [9, 10].

One such solution is to predict the events using a k-order Markov model. This amounts to assuming that the event-generating process is k-order stationary, or to assuming that such a process is a sensible approximation of the true process. Under this assumption, we can infer our *reduced model* using exactly the same inference technique described by Equations 5–8, iterating only up to time step k. In particular, the new CPT for event features $X_U(t)$ is:

$$\begin{aligned} \boldsymbol{M(t)} &\equiv \mathbb{P}[X_U(t)|X_U(t-k\ldots t-1), X_I(t-k\ldots t-1)] \\ &= \mathbb{P}[X_U(k)|X_U(0\ldots k-1), X_I(0\ldots k-1)] \end{aligned} \tag{9}$$

for any given time step $t \geq k$.

This approach flexibly approximates the event prediction model to a desired level of granularity. An appropriate level of k may be selected depending on computational restrictions and on the presumed complexity of the underlying event process. The larger the value of k, the closer the prediction model will be

[5] In the interest of space here, we have presented the most simplistic method for propagating bounds, but there exist more complicated approaches capable of tighter bound propagation (e.g., [7]).

to the underlying process. However, if less is known about the process, a larger k can also lead to looser bounds in the probability parameters of the model. The smaller the value of k the simpler the decision model used to plan. At the extreme, we can model the events as depending on neither history nor on state by approximating the distribution with a single probability denoting the likelihood of the event taking place at any given time step.

5 Results

In this section, we illustrate the application of our modeling approach to the robot surveillance scenario introduced in Section 2. We describe two sets of experiments. In the first set of experiments, we use simulation to compare the performance of our approach with that of a standard MDP model that treats each waving detected as an actual person waving that must be attended. In a second experiment, we deploy and analyze the policy observed in the actual robot in a real experiment.

5.1 Experimental setup

Let is first expand on our earlier description of the robot navigation scenario (Section 2) and the EEM shown in Figure 3. The robot localization module (described in Section 5.4) is sufficiently accurate for variables X_p and X_s to be considered fully observable. We assume that, through the use of specialized fire detection hardware connected to the surveillance network, our system is able to unambiguously perceive the occurrence of a fire X_f. The same cannot be said about detecting a visitor waving X_v; instead, the system perceives X_w, which encodes the result of the waving detection algorithm [11] deployed in the camera network.

The detection sensitivity X_d may in general depend on (known) parameters of the vision algorithm *and* on unpredictable lighting conditions. For simplicity, we specify a boolean domain $\mathcal{X}_d = \{low, high\}$ and assume that, throughout each experiment, X_d remains constant. Further, we conducted our experiments in a closed environment at a time in which cleaning staff was not present on the floor, nor were there people conversing in the hallways. As such, we were able to eliminate variables X_c and X_l from our EEM (Figure 3) because we knew that neither lab members nor cleaning staff could possibly affect waving detections.

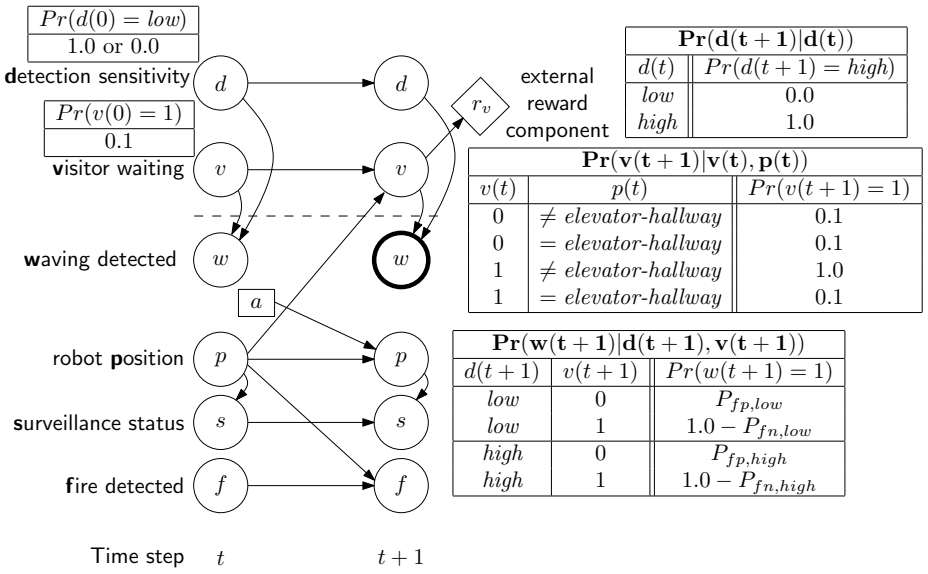
We note that, even under these controlled conditions, the vision algorithms we employ do not behave flawlessly. Sometimes we receive a detection when there is no visitor waving (*false positive*); and sometimes a person waving to the camera does not trigger a detection (*false negative*). The probabilities of false positives and false negatives were the hardest parameters to estimate, varying greatly from run to run. Thus, instead of fixing exact probabilities, we specified the following bounds:

	false positives	false negatives
sensitivity $X_d = low$	$P_{fp,low} \in [0.1, 0.2]$	$P_{fn,low} \in [0.2, 0.5]$
sensitivity $X_d = high$	$P_{fp,high} \in [0.1, 0.3]$	$P_{fn,high} \in [0.2, 0.3]$

Notice that with a higher detection sensitivity, the system tends to produce more false positives, but fewer false negatives.

5.2 Reducing the Event Model

Following the discussion above, we obtain the expanded event model shown in Figure 6, along with the conditional probability tables associated with the event variable $X_w(t)$ and the external variables $\{X_d(t), X_v(t)\}$. We next apply our approach to reduce the model, eliminating all but the observable state variables.



$\mathbf{Pr(d(t+1)|d(t))}$	
$d(t)$	$Pr(d(t+1) = high)$
low	0.0
high	1.0

$\mathbf{Pr(v(t+1)|v(t), p(t))}$		
$v(t)$	$p(t)$	$Pr(v(t+1) = 1)$
0	$\neq$ *elevator-hallway*	0.1
0	$=$ *elevator-hallway*	0.1
1	$\neq$ *elevator-hallway*	1.0
1	$=$ *elevator-hallway*	0.1

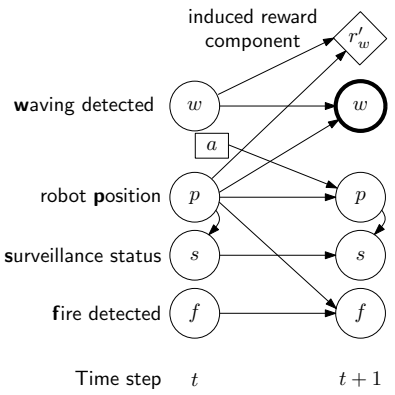
$\mathbf{Pr(w(t+1)|d(t+1), v(t+1))}$		
$d(t+1)$	$v(t+1)$	$Pr(w(t+1) = 1)$
low	0	$P_{fp,low}$
low	1	$1.0 - P_{fn,low}$
high	0	$P_{fp,high}$
high	1	$1.0 - P_{fn,high}$

Fig. 6. Expanded Event Model used in the experiments.

In reducing our model, it turns out that there is additional structure in this problem that we can take advantage of. In particular:

- Although, in principle, detection sensitivity $d(t)$ may vary from time step to time step, we ran our experiment under controlled lighting conditions to ensure a static detection sensitivity throughout.
- Such as was suggested in Corollary 1, the d-separating set (which imposes conditional independence between the agent's actions and the external variables) need not contain all histories of all internal variables. Given the structure of the DBN in Figure 6, surveillance status history and fire detection history are unnecessary.
- The robot can only stop a visitor from waiting for assistance by visiting the elevator hallway. As such, for predicting waving detection $w(t+1)$, the factor $p(t)$ can be treated as either $=$*elevator hallway* or $\neq$*elevator hallway*.

Although in general, the d-separating set is $D(t) = \{w(0 \dots t), p(0 \dots t)\}$, containing the entire history of robot position and waving detections (as depicted in Figure 5), for simplification, we approximated the BPEM as a first-order stationary process using the approach described in Section 4.2.4. This yielded a smaller MDP, simpler inference (Equations 5-8), and the ability to compute infinite-horizon policies due to the stationarity. Because the inference is approximate, the probabilities and rewards returned by the algorithm vary from time step to time step. To improve the scope of our approximate bounds, we executed two iterations of influence and took the wider bounds at stage $t = 2$. The resulting CPT and induced reward function are partially shown in Figure 7.



$\mathbf{r\prime(p(t-1), w(t-1))}$			
$d(*)$	$p(t-1)$	$w(t-1)$	$r\prime$
low	$\neq$ *elevator-hallway*	0	$[-0.318, -0.0873]$
low	$\neq$ *elevator-hallway*	1	$[-0.151, -0.0602]$
low	$=$ *elevator-hallway*	0	$[-1.0, -0.230]$
low	$=$ *elevator-hallway*	1	$[-0.241, -0.0377]$
$\vdots$	$\vdots$	$\vdots$	$\vdots$

$\mathbf{Pr(w(t+1) \mid w(t), p(t))}$			
$d(*)$	$w(t)$	$p(t)$	$Pr(w(t+1)=1)$
low	0	$\neq$ *elevator-hallway*	$[0.133, 0.387]$
low	0	$=$ *elevator-hallway*	$[0.120, 0.302]$
low	1	$\neq$ *elevator-hallway*	$[0.187, 0.930]$
low	1	$=$ *elevator-hallway*	$[0.0753, 0.482]$
$\vdots$	$\vdots$	$\vdots$	$\vdots$

Fig. 7. BPEM used in the experiments.

5.3 Simulation experiments

In this first set of experiments, we considered separately the conditions $X_d = low$ and $X_d = high$. For the purpose of experiments, we considered that fires break out at every time-step with a constant probability $p_{\text{fire}} = 0.1$. Similarly, we assume that a new visitor requests assistance at every time-step with a constant probability $p_{\text{visitor}} = 0.1$.

We used the *interval value iteration* (IVI) algorithm to solve the obtained BMDP [8]. We computed two policies, $\pi_{\max}$ and $\pi_{\min}$. The two policies attain, respectively, the best and worst possible performances, given the uncertainty in the BMDP parameters. For comparison, we also computed the optimal policy for an approximate MDP, π_{app} that ignores partial observability in X_v and simply takes $X_w = X_v$.

We tested the three policies in our navigation scenario, running each policy for a total of 200 independent Monte Carlo trials. In each trial, a (simulated) robot moved around the environment for a total of 100 time-steps while following the prescribed policy and the total discounted reward accumulated was averaged over all runs, as shown in Table 1.

Several interesting observations are in order. First of all, both BMDP policies have a minimal difference in performance, which indicates that, in fact, for the environment parameterization considered, the bounds computed are actually tight. Second, it is interesting to note that the performance of the BMDP policies also does not change significantly as the sensitivity of the waving detection changes. This indicates that, in fact, the BMDP model is able (to some extent) to cope with the partial observability associated with this feature.

Finally, we point out the significant difference in performance between the BMDP policies and the MDP model. This difference can be explained by the fact that the MDP policy will attempt to assist every waving detected. The existence of false positives impacts its ability to tackle the other tasks (surveillance and fire assistance). The BMDP, on the other hand, is more effectively able to handle false positives, attaining a better balance between the different tasks.

5.4 Executing Policies on Real Robots

Finally, we illustrate the execution of one of the policies we computed on a real robot (Pioneer 3-AT) interacting with our 8th floor ISR surveillance environment (photos of which are shown in Figure 8). All the different components in the system are connected through ROS [12]. Image processing algorithms, including people and waving detection [11], are run on live feeds from 12 static cameras. Aside from receiving waving and fire detections from the network, the robot is equipped with a laser and a map of the environment. This allows the robot to localize itself (using AMCL) and navigate (using off-the-shelf path planning and obstacle avoidance) via ROS. The robot also uses an MDP state estimator, that identifies the robot's position among pre-labeled regions in the map, and a policy controller, that maps actions to waypoints, both of which we implemented in ROS.

Figure 9 shows two partial trajectories of the robot during its execution of a policy. The left side (a) and the right side (b) demonstrate two distinct behaviors that we observed: *patrolling* and *responding to a waving event*. For the sake of clarity, the first half of each trajectory is plotted in red and the return portion plotted in blue. Arrows are drawn on the path each time the robot made a decision (corresponding to a new waypoint sent for navigation). In the absence of events (Fig. 9(a)), the robot behaves as expected, going around the floor and visiting all the relevant rooms specified in Section 2. When the robot decides to

Policy	$X_d = low$	$X_d = high$
$\pi_{\max}$	-85.08 ± 17.9	-86.47 ± 17.0
$\pi_{\min}$	-85.46 ± 18.8	-86.62 ± 18.8
π_{app}	-97.76 ± 27.4	-100.85 ± 24.6

Table 1. Average total discounted reward of BMDP and approximate MDP policies.

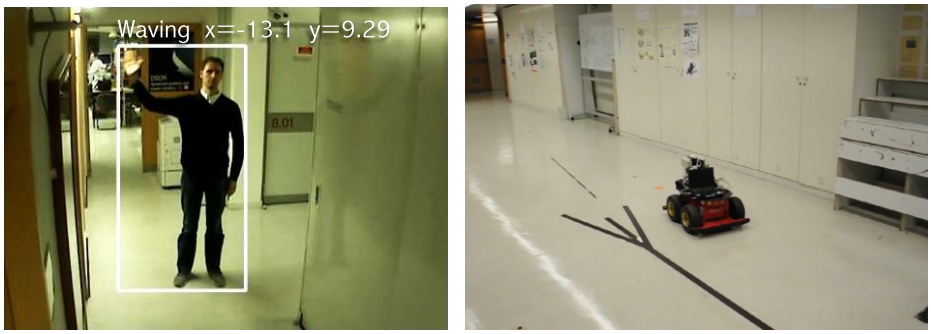


Fig. 8. A robot responding to a waving event in the real surveillance system.

respond to a waving event (Fig. 9(b)), it navigates to the elevator hallway where the waving was detected. Point 1 shows the position of the robot at the time of detection. We observe the robot navigating to the elevator directly, without entering intermediate rooms (such as the coffee room), and meeting the person at point 2.

6 Related Work

The conventional method of reasoning about uncertain events while planning is to include the events as part of the state and model their occurrence as transition probabilities. For instance Becker *et al.* [3] solve Dec-MDPs in which the interaction between the agents only take place at certain events (state transitions). Other work [13] models an MDP that reasons only about these kind of events instead of states. However, these approaches are not appropriate to model events that are external to the system. In these models, events occur continuously, since they model the internal state transitions. Moreover, no actions

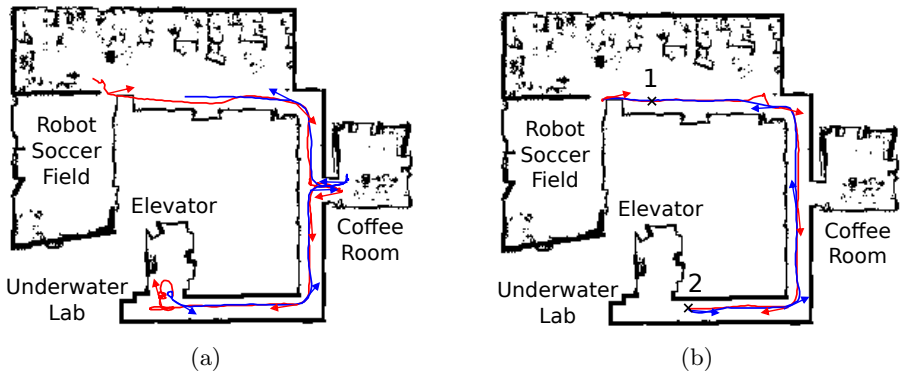


Fig. 9. Robot Patrolling (a) and Event Response (b) behavior exhibited by a policy.

are taken while there are no events, which would be most of the time if the probability of the events is very low. To combine asynchronous events and actions, others have employed Generalized Semi-Markov Decision Processes (GSMDPs) [14, 15]. They consider a continuous time and model a set of external events that can be triggered at any time affecting the system state. The main problem is that a time distribution is assumed for each event, and those are not always easy to find, mainly for rare events.

Others have acknowledged that rare events are difficult to model conventionally. There is extensive work on estimating the probabilities of rare events [16, 17]. In general, the idea is to simulate the system and apply different sampling techniques in order to derive small probabilities for the events. These techniques are used in a wide spectrum of applications, such as biological systems, queue theory, reliability models, etc. Usually the rare events (e.g., a queue overflow or a system failure) have a very low probability but they can occur if the simulations are properly driven. A challenge is that, given the existence of many hidden variables, fine-tuning an arbitrarily complex model is not always possible.

Our approach is complementary to previous work that defined bounded and imprecise parameter models and developed solution algorithms, such as MDPs with imprecise probabilities [18], POMDPs with imprecise probabilities [19], bounded-parameters variations in MDPs [8] and POMDPs [20], as well as Factored MDPs with imprecise probabilities [21]. The modeling framework that we develop describes a principled approach to actually specifying a bounded-parameter model. Thus, our work contributes a useful precursor to, and a formal context for, applying bounded-parameter models.

The idea of reducing a decision model by eliminating unobserved external variables has also been explored in multiagent settings [22, 23], where agents model abstract *influences* from peers rather than the peers' full decision models. Here we show that by modeling environmental influences abstractly, the same principle also facilitates single-agent decision making. Along a similar vein, PSRs [10] also seek to make predictions using a compact representations of histories of observable features.

7 Conclusions and Future Work

In this paper, we have addressed the challenge of modeling unpredictable events for the purposes of intelligent planning and decision-making under uncertainty. In contrast to the majority of work in this area, which assumes a perfectly predictive model or world dynamics, we have acknowledged that an accurate MDP model may be impossible to prescribe. As a precursor to planning in these scenarios, we have developed a framework for accounting for events whose underlying processes are prohibitively complex and/or unknown.

Our main contribution is a principled modeling approach. We have shown that if (potentially large) portions of the underlying event process can be treated as unobservable external variables, those variables do not need to be included in the decision model. In particular, we have proven that an MDP model *without*

the external variables provides the same predictive power as a POMDP model *with* the external variables. And we have formulated how inference techniques can be used to reduce the model through marginalization.

For those situations where there is uncertainty in the underlying process, we have developed a method for systematically propagating the uncertainty to distinguished parameters in our reduced model. And for when knowledge is gained about the underlying process, the same method incorporates that knowledge in the form of tightened probability bounds on event prediction probabilities. The output of our method, the BPEM, is a special type of BMDP, which our principled approach helps the modeling practitioner to specify. Our approach offers the benefit of flexibility in the richness of the BPEM, accommodating different levels of knowledge about event dynamics, and supporting trade-offs in computational complexity and predictive precision.

We have illustrated how our framework can be used to model events in a robot planning problem. In the future, we plan to extend our experimental work to carefully analyze the trade-offs among different modeling choices, as well as to conduct more experiments using the real robot interacting in our surveillance environment.

Acknowledgments

This work was partially supported by project CMU-PT/SIA/0023/2009 under the Carnegie Mellon Portugal Program and its Information and Communications Technologies Institute, and the Portuguese Funda cão para a Ci encia e Tecnologia (INESC-ID multiannual funding) under project PEst-OE/EEI/LA0021/2011. Matthijs Spaan is funded by the FP7 Marie Curie Actions Individual Fellowship #275217 (FP7-PEOPLE-2010-IEF).

References

1. Puterman, M.: Markov Decision Processes: Discrete Stochastic Dynamic Programming. John Wiley & Sons, Inc. (1994)
2. Boutilier, C., Dean, T., Hanks, S.: Decision-theoretic planning: Structural assumptions and computational leverage. Journal of Artificial Intelligence Research (JAIR) **11** (1999) 1–94
3. Becker, R., Zilberstein, S., Lesser, V.: Decentralized markov decision processes with event-driven interactions. In: Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems-Volume 1, IEEE Computer Society (2004) 302–309
4. Goldman, R.P., Musliner, D.J., Boddy, M.S., Durfee, E.H., Wu, J.: "Unrolling" complex task models into MDPs. In: AAAI Spring Symposium: Game Theoretic and Decision Theoretic Agents. (2007) 23–30
5. Kaelbling, L.P., Littman, M.L., Cassandra, A.R.: Planning and acting in partially observable stochastic domains. Artificial Intelligence **101** (1998) 99–134
6. Pearl, J.: Probabilistic reasoning in intelligent systems: networks of plausible inference. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (1988)

7. Bidyuk, B., Dechter, R.: Improving bound propagation. In: Proceedings of the 17th European Conference on Artificial Intelligence (ECAI). (2006) 342–346
8. Givan, R., Leach, S., Dean, T.: Bounded-parameter markov decision processes. Artificial Intelligence **122**(1-2) (2000) 71–109
9. Begleiter, R., El-yaniv, R., Yona, G.: On prediction using variable order markov models. Journal of Artificial Intelligence Research **22** (2004) 385–421
10. Littman, M., Sutton, R., Singh, S.: Predictive representations of state. In: Adv. Neural Information Proc. Systems. (2002) 1555–1561
11. Moreno, P., Bernardino, A., Santos-Victor, J.: Waving detection using the local temporal consistency of flow-based features for real-time applications. In: Proc. of ICIAR2009 - 6th International Conference on Image Analysis and Recognition. (2009)
12. Quigley, M., Conley, K., Gerkey, B.P., Faust, J., Foote, T., Leibs, J., Wheeler, R., Ng, A.Y.: Ros: an open-source robot operating system. In: ICRA Workshop on Open Source Software. (2009)
13. Cao, X., Zhang, J.: Event-based optimization of markov systems. Automatic Control, IEEE Transactions on **53**(4) (2008) 1076–1082
14. Younes, H., Simmons, R.: Solving generalized semi-markov decision processes using continuous phase-type distributions. In: Proceedings of the National Conference on Artificial Intelligence, Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999 (2004) 742–748
15. Rachelson, E., Quesnel, G., Garcia, F., Fabiani, P.: Approximate policy iteration for generalized semi-markov decision processes: an improved algorithm. In: European Workshop on Reinforcement Learning. (2008)
16. Juneja, S., Shahabuddin, P.: Rare-event simulation techniques: An introduction and recent advances. In Henderson, S., Nelson, B., eds.: Handbooks in operations research and management science. Volume 13. Addison Wesley (2006) 291–350
17. Rubino, G., Tuffin, B., eds.: Rare event simulation using Monte Carlo methods. John Wiley & Sons, Ltd (2009)
18. Harmanec, D.: Generalizing Markov decision processes to imprecise probabilities. Journal of Statistical Planning and Inference **105**(1) (2002) 199–213
19. Itoh, H., Nakamura, K.: Partially observable Markov decision processes with imprecise parameters. Artificial Intelligence **171**(8-9) (2007) 453–490
20. Ni, Y., Liu, Z.Q.: Bounded-parameter partially observable Markov decision processes. In: Proc. 18th Int. Conf. Automated Planning and Scheduling. (2008)
21. Delgado, K.V., Sanner, S., de Barros, L.N.: Efficient solutions to factored MDPs with imprecise transition probabilities. Artificial Intelligence **175**(9-10) (2011) 1498–1527
22. Oliehoek, F., Witwicki, S., Kaelbling, L.: Influence-based abstraction for multiagent systems. In: Proceedings of the Twenty-Sixth AAAI Conference (AAAI-2012), Toronto, Canada (2012)
23. Witwicki, S.J., Durfee, E.H.: Influence-based policy abstraction for weakly-coupled Dec-POMDPs. In: Proceedings of the 20th International Conference on Automated Planning and Scheduling (ICAPS-2010), Toronto, Canada (2010) 185–192