Incorporating PGMs into a BDI Architecture

Yingke Chen[1], Jun Hong[1], Weiru Liu[1], Lluís Godo[1,2], Carles Sierra[1,2], and Michael Loughlin[1]

[1] Queen's University Belfast, Belfast, UK
[2] IIIA, CSIC, Bellaterra, Spain

Abstract. In this paper, we present a hybrid BDI-PGM framework, in which PGMs (Probabilistic Graphical Models) are incorporated into a BDI (belief-desire-intention) architecture. This work is motivated by the need to address the scalability and noisy sensing issues in SCADA (Supervisory Control And Data Acquisition) systems. Our approach uses the incorporated PGMs to model the uncertainty reasoning and decision making processes of agents situated in a stochastic environment. In particular, we use Bayesian networks to reason about an agent's beliefs about the environment based on its sensory observations, and select optimal plans according to the utilities of actions defined in influence diagrams. This approach takes the advantage of the scalability of the BDI architecture and the uncertainty reasoning capability of PGMs. We present a prototype of the proposed approach using a transit scenario to validate its effectiveness.

1 Introduction

SCADA (Supervisory Control And Data Acquisition) systems have proved to be a powerful and successful technology in various application domains, including power generation, power transmission, transportation, and military applications [1]. However, the complexity of such systems increases too rapidly to be handled by traditional software engineering approaches. This complexity comes from the large number of subsystems needed to implement business process requirements [2]. Moreover, trained experts who supervise the entire system also face the challenge of dealing with explosively growing amounts of sensory data, especially in emergency situations. Current SCADA systems lack autonomous and intelligent capabilities to meet these pressing requirements and are difficult to scale up to larger and more complex deployments [3,4].

As SCADA systems are situated in dynamic environments, not all sensory data can be deemed completely accurate. Nevertheless, human experts are capable of estimating the state of the world even if the sensory data has inherent errors/noise or uncertainty, is incomplete or in conflict with data acquired from other sources. Experts also have to make decisions based on uncertain and incomplete information. Therefore, an autonomous SCADA system requires an adequate framework to reason about uncertainty and model decision making based on uncertain information.

The belief-desire-intension (BDI) agent architecture is a successful paradigm for modeling rational agents [5]. The BDI agents have been used to develop

SCADA systems by treating system components as autonomous agents, to provide better scalability, autonomy and intelligence [6,7,2]. Each agent's beliefs (knowledge about the environment), desires (goals), and intentions (commitments to act) are explicitly represented. These beliefs, desires, and intentions are also known as the agent's mental states. Using the BDI architecture, one can specify, design and verify different types of agents in different application domains. AgentSpeak [8] is an agent-oriented programming language for specifying agents within the BDI framework.

In this paper, motivated by the need to address the scalability and noisy sensing issues in SCADA systems, our main contribution is the incorporation of two probabilistic graphical models (PGMs) (Bayesian networks (BNs) and influence diagrams (IDs)) into a BDI architecture. Specifically, we introduce an agent's epistemic state [9] in the hybrid BDI-PGM framework, and define them based on PGMs that model the stochastic environment where the agent is situated. Sensory observations in the environment are first fed into PGMs, and the corresponding belief sets are derived after uncertainty propagation in the PGMs. Since it is possible that an agent may be *ignorant* about the environment because of the inherent uncertainty, we take into account the utilities of actions, and maximize the utility in such situation. For example, when a train agent does not know the actual state of a signal, it prefers to *stop* rather than keeping *moving*. We specify utilities of actions in various situations using influence diagrams (IDs), and formulate plan selection as a utility-based decision problem. The hybrid BDI-PGM framework takes the advantage of the scalability of the BDI architecture and increases its uncertainty reasoning capability by incorporating PGMs (BNs and IDs) into it. In addition, a prototype of the proposed approach using for a transit scenario is designed and implemented to validate its effectiveness.

The rest of the paper are organized as follows. Section 2 reviews the related work. Section 3 presents the essentials of the techniques we use, including the AgentSpeak framework, BNs, and IDs. Section 4 discusses how to embed PGMs into a BDI architecture, and propose a utility-driven plan selection approach. In Section 5, we describe the design and implementation of the transit scenario with noisy sensing to validate our approach. Section 6 concludes the paper and describes future work.

2 Related Work

In [2], the advantages of applying MAS technologies to address the challenges of traditional control systems are discussed. The advantages (e.g., scalability, autonomy and intelligence) are illustrated through a case study describing a SCADA system for electricity transportation management. In [6,7] the authors focus on the application of MAS to power engineering, where problems are formalized, technologies are discussed, and several implementation issues are addressed. In [10] it is reported that MAS technologies outperform classical technologies in the protection of power distribution systems.

There have been several approaches to modeling uncertainty in MAS. In [11] the degrees of beliefs that a BDI agent has are quantified using Dempster-Shafer theory. The graded BDI architecture [12] uses uncertain beliefs (as probabilities) and graded preferences (as expected utilities) to rank plans. Bayesian networks have been widely used for modeling uncertain environments [13]. There have been studies about combining BNs with the BDI architecture to handle the uncertainty in beliefs and select appropriate plans in a dynamic environment. In [14] the agent deliberation process is modeled by a BN. Both causality and quantitative relations between beliefs are taken into account, and applicable plans are sorted so that the plan whose context has the highest likelihood to be valid is selected. Its threshold-based plan selection approach is further extended in [15] by adding bias and randomness to all applicable plans. In [16] a tool connecting an MAS development framework (Jason [17]) with a BN constructor is developed. These approaches essentially focus on selecting plans based on the likelihood of their contexts. However, the acquisition of uncertain beliefs has not been addressed. In this paper, we consider a more realistic situation, where an agent makes observations by performing sensing actions. The sensory observations are first fed into PGMs, leading to an agent's epistemic state being revised after uncertainty propagation in the PGMs. Beliefs may be derived from the revised epistemic state and such beliefs are then added to an agent's belief set.

Autonomous agents have to make rational decisions to pursue their goals (i.e., selecting appropriate plans) in a dynamic environment. Markov decision processes (MDP) and partially observable MDPs (POMDPs), as well as their graphical representations, IDs [18], are popular frameworks to model an agent's decision making processes in stochastic environments. In [19] POMDPs and the BDI architecture are compared, and the correspondences between desires and intensions on the one hand, and rewards and policies on the other hand are illustrated. In [20] the relationship between the policies of MDPs and the intentions in the BDI architecture is further discussed. In particular, it shows that intentions in the BDI architecture can be mapped to policies in MDPs. The performance and scalability of (PO)MDPs and the BDI architecture are compared in [21]. In particular, (PO)MDPs have a better performance when the domain size is tractable since the BDI architecture uses a heuristic planning approach. The BDI architecture has better scalability since the state space in (PO)MDPs grows explosively when modeling complex application domains (e.g., SCADA systems). A hybrid BDI-POMDP framework [22] has been proposed for quantitatively analysing the teaming behaviours of agents in an uncertain environment. Different from this approach, our proposed approach embeds PGMs into the BDI architecture to model the uncertainty about the environment and reason about optimal decisions of agents.

3 Preliminaries

In this section, we describe the basics of the techniques we use in the paper.

3.1 AgentSpeak

An AgentSpeak agent $\mathbb{A}$ can be represented as a tuple $\langle \mathsf{BB}, \mathsf{PLib}, \mathsf{E}, \mathsf{A}, \mathsf{I}, \mathcal{S}_\varepsilon, \mathcal{S}_\mathcal{O}, \mathcal{S}_\mathcal{I} \rangle$, where $\mathsf{BB}, \mathsf{PLib}, \mathsf{E}, \mathsf{A}, \mathsf{I}$ are its belief base, plan library, event set, action set and intention stack, respectively. $\mathcal{S}_\varepsilon, \mathcal{S}_\mathcal{O}, \mathcal{S}_\mathcal{I}$ are the selection functions for events, plans, and intentions, respectively. We define beliefs, goals, triggering events, and plans for an AgentSpeak agent following the notation in [8]. We use Φ to denote a finite set of predicate, action and constant symbols, and $\mathcal{V}$ to denote a set of variables. We use $a, b, \ldots$ to denote elements in Φ and $X, Y, \ldots$ to denote elements in $\mathcal{V}$. A term t is a constant symbol in Φ or a variable in $\mathcal{V}$.

Definition 1. *Let b be a n-ary predicate symbol, and $t_1, \ldots, t_n$ be terms (collectively referred as $\boldsymbol{t}$ thereafter), then $b(\boldsymbol{t})$ is a belief atom. Given belief atoms $b(\boldsymbol{t})$ and $c(\boldsymbol{t})$, the $b(\boldsymbol{t})$, $c(\boldsymbol{t})$, $b(\boldsymbol{t}) \wedge c(\boldsymbol{t})$, and $\neg b(\boldsymbol{t})$ are beliefs.*

Definition 2. *Let g be a predicate symbol, and $t_1, \ldots, t_n$ be terms, then $!g(\boldsymbol{t})$ and $?g(\boldsymbol{t})$ are goals. Specifically, $!g(\boldsymbol{t})$ is an achievement goal and $?g(\boldsymbol{t})$ is a test goal.*

Definition 3. *Let $b(\boldsymbol{t})$ be a belief atom, and $!g(\boldsymbol{t})$ and $?g(\boldsymbol{t})$ be goals, then $+b(\boldsymbol{t})$, $-b(\boldsymbol{t})$, $+!g(\boldsymbol{t})$, $-!g(\boldsymbol{t})$, $+?g(\boldsymbol{t})$ and $-?g(\boldsymbol{t})$ are triggering events. Here operators $+$ and $-$ denote addition and deletion of a belief or goal, respectively.*

Definition 4. *Let a be an action symbol, and $t_1, \ldots, t_n$ be terms, then $a(\boldsymbol{t})$ is an action.*

Definition 5. *Let e be a triggering event, $b_1, \ldots, b_m$ be beliefs, $h_1, \ldots, h_n$ be goals or actions, then $e : b_1 \wedge \ldots \wedge b_m \leftarrow h_1; \ldots; h_n$ is a plan. Here, $b_1 \wedge \ldots \wedge b_m$ is referred as the context of the plan.*

The belief base represents an agent's knowledge about the environment in which it is situated. The plan library PLib contains a set of plans for either achieving the agent's goals or responding to changes in the environment. In AgentSpeak, an agent's behaviour is specified by a set of beliefs and plans. Each plan may specify a set of actions to be performed, a set of subgoals to be achieved and a set of conditions under which the plan is applicable. A plan is applicable, when its *triggering event* occurs and its *context* is valid. The applicable plan can be executed by performing actions and achieving subgoals specified in its *body*. The execution of a plan is the agent's response to changes in the environment and the means to achieve its goals.

In a reasoning cycle as shown in Fig. 1, agent $\mathbb{A}$ responds to the triggering event e at the top of its event stack, by selecting and executing a plan p, which is referred to as an intention. A set of plans, whose contexts are valid according to the belief base BB, will be identified from the plan library PLib. Only one plan will be selected and executed. Performing an action may change the environment and consequently change the agent's belief base. The execution of the chosen plan may also add new events to the event stack, which will be handled in future reasoning cycles.

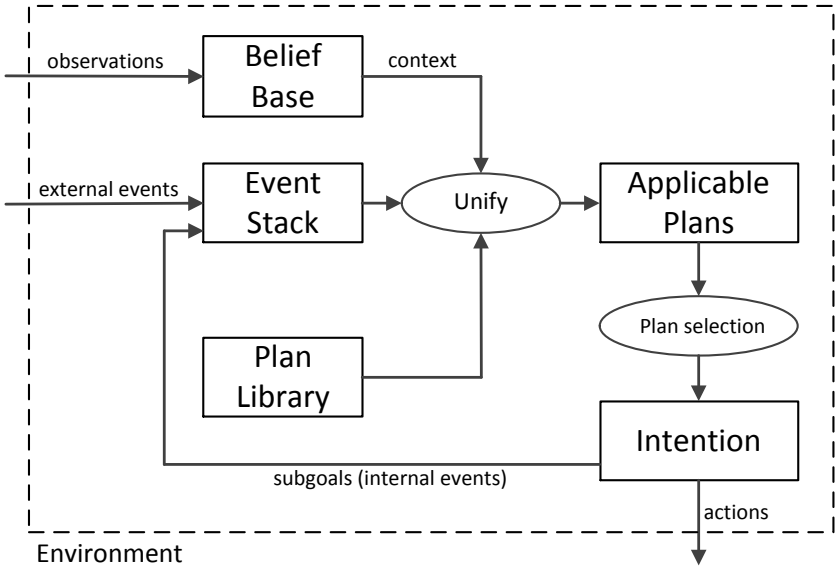


Fig. 1. The reasoning cycle of an AgentSpeak agent

Example 1. Assume that agent $\mathbb{A}$ represents an autonomous train. In train agent $\mathbb{A}$'s belief base BB, some of its belief atoms can be:

- Sig(red): the signal is red;
- Moving: the train is moving;
- Train($\mathbb{A}$, 100): agent $\mathbb{A}$'s own position;
- Train($\mathbb{B}$, 300): another train's position;
- Station(central, 1000): the location of the train station named central.

The train agent can perform the following actions:

- stop: stop from normal speed;
- accel: accelerate from still to normal speed;
- move(X, Y): move from the current position X to the new position Y with normal speed;
- senseSignal: observe the signal within its sensing range;
- openDoor: open doors on the train for passengers to board;
- closeDoor: close doors after boarding;
- senseBoarding: observe whether boarding is completed;

The following plans in PLib specify the train agent $\mathbb{A}$'s behaviours under different contexts. P1 and P2: if a signal is sensed either green or red while the train is moving, it will either keep moving or stop, and keep checking the signal. P3 and P4: while the train is still, it will move when the signal becomes green; otherwise it will stay still and keep sensing.

- (P1)+!drive : Moving $\wedge$ Sig(green) $\leftarrow$ move; senseSignal; !drive;
- (P2)+!drive : Moving $\wedge$ Sig(red) $\leftarrow$ stop; senseSignal; !drive;
- (P3)+!drive : $\neg$ Moving $\wedge$ Sig(red) $\leftarrow$ senseSignal; !drive;
- (P4)+!drive : $\neg$ Moving $\wedge$ Sig(green) $\leftarrow$ move; senseSignal; !drive;

3.2 Probabilistic Graphical Models

Probabilistic graphical models (PGMs) are a combination of graph theory and probability theory by encoding probabilistic dependency relations in the graphical structures. Bayesian networks (BNs) and influence diagrams (IDs) are two popular PGMs. They provide a natural specification language for various problem domains with inherent uncertainty. Over the past decades, a set of efficient algorithms and sophisticated tools have been developed for PGMs [13].

A Bayesian network $\mathcal{B}$ over a set of random variables $\overrightarrow{\mathbf{X}} = \{X_1, \ldots, X_n\}$ is defined by a pair $\mathcal{B}(\overrightarrow{\mathbf{X}}) = \langle G, \Theta \rangle$. G is a directed acyclic graph, in which each node represents a random variable X_j, with edges representing the dependencies between variables. Θ is a set of parameters $\theta_{x_{j,i}|\pi_{j,i}} = P(x_{j,i} \mid \pi_{j,i})$ for each instantiation of $x_{j,i}$ of variable X_j given $\pi_{j,i}$ which is the instantiation of the parents of X_j. These conditional probabilities are used to quantify dependencies between variables. Given a BN $\mathcal{B}$, a joint probability distribution over $\{X_1, \ldots, X_n\}$, $P(X_1, \ldots, X_n)$, is defined.

As an example, the Bayesian network as shown in Fig. 2(a) represents the dependency relation between the observation of a signal and the actual state of the signal. It shows that the observation (O, a binary variable with two states 'oR' (for observed 'red') and 'oG' (for observed 'green')) is determined by the actual state of the signal (S, a binary variable with two states 'red' and 'green'). The conditional probability table in the Fig. 2(b) shows the probability distribution over observations given different states of the signal. Each number in this table represents the conditional probability of seeing an observation given the actual state. For example the probability of observing 'oR' given the signal is 'red' is denoted as $P(\mathsf{oR} \mid \mathsf{red}) = P(O = \mathsf{oR} \mid S = \mathsf{red}) = 0.7$.

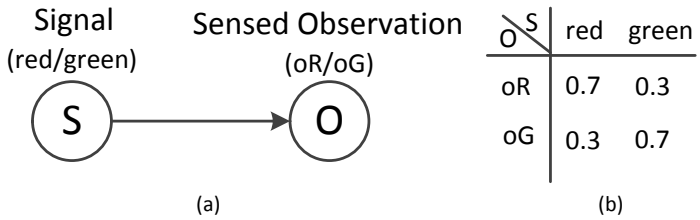
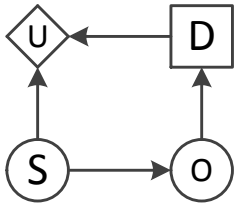


Fig. 2. (a) The Bayesian network for a signal and its observation. (b) The condition probability table for $P(O \mid S)$.

The uncertainty reasoning in BNs is carried out by applying the Bayes' rule. Assuming the initial prior probability distribution over the states of the signal (S node) is uniform, i.e., (0.5, 0.5), the posterior probability distribution can be inferred from its prior probability distribution and the evidence from the observation. For example, the probability of the signal being 'red' given 'red' is observed ('oR') is computed as follows. Note that, BNs also offer updating mechanisms to deal with uncertain inputs. For example, the evidence is given by a distribution over possible observations $P(\mathsf{oR}) = 0.2$ and $P(\mathsf{oG}) = 0.8$ rather than a single observation ('oR').

$$\begin{aligned} P(\mathsf{red} \mid \mathsf{oR}) &= \frac{P(\mathsf{red}, \mathsf{oR})}{P(\mathsf{oR})} = \frac{P(\mathsf{red}, \mathsf{oR})}{P(\mathsf{red}, \mathsf{oR}) + P(\mathsf{green}, \mathsf{oR})} \\ &= \frac{P(\mathsf{oR}|\mathsf{red})P(\mathsf{red})}{P(\mathsf{oR}|\mathsf{red})P(\mathsf{red}) + P(\mathsf{oR}|\mathsf{green})P(\mathsf{green})} \\ &= \frac{0.7 \cdot 0.5}{0.7 \cdot 0.5 + 0.3 \cdot 0.5} = 0.7. \end{aligned}$$

Influence diagrams (also known as decision graphs) are extended from Bayesian networks by introducing actions and their utilities. They can model the problem of making optimal decisions in an uncertain environment given incomplete information. An influence diagram $\mathcal{D}$ is defined by a pair $\mathcal{D}(\overrightarrow{\mathbf{X}}, \overrightarrow{\mathbf{D}}, \overrightarrow{\mathbf{U}}) = \langle G, \Theta \rangle$, which models a set of random variables $\overrightarrow{\mathbf{X}}$, a set of actions $\overrightarrow{\mathbf{D}}$, and the utilities of actions $\overrightarrow{\mathbf{U}}$ in various situations. There are three types of nodes in G: chance nodes (representing the environment), decision nodes (representing actions), and utility nodes (representing the utilities of actions). In Fig. 3(a), D ($\overrightarrow{\mathbf{D}} = \{D\}$) and U ($\overrightarrow{\mathbf{U}} = \{U\}$) are a decision node and a utility node, respectively. The parameter of the U node $\theta_{U|D,S} = U(D, S)$ is set in the utility table as shown in Fig. 3(b). For example, $U(D = \mathsf{stop}, S = \mathsf{red}) = 10$ indicates that 10 reward points can be obtained if the train stops when the signal is red. Thereafter, we will denote $U(D = \mathsf{stop}, S = \mathsf{red})$ as $U(\mathsf{stop}, \mathsf{red})$ for simplicity.



(a)

	red		green	
	stop	move	stop	move
U	10	-100	-1	10

(b)

Fig. 3. (a) The influence diagram for decision making based on information observed from a signal. (b) The utility table for $U(D, S)$.

The optimal decision is selected from a set of possible decisions based on their expected utilities (EU), which takes into account the actions utilities and the probability distribution over the state of the world. For example, assuming the initial prior probability distribution of S is uniform and 'red' is observed ('oR'), $EU(\mathsf{stop})$ is calculated as follows:

$$\begin{aligned} EU(\mathsf{stop}) &= \sum_{s \in S} P(s \mid \mathsf{oR}) U(\mathsf{stop}, s) \\ &= P(\mathsf{red} \mid \mathsf{oR}) U(\mathsf{stop}, \mathsf{red}) + P(\mathsf{green} \mid \mathsf{oR}) U(\mathsf{stop}, \mathsf{green}) \\ &= 0.7 \cdot 10 + 0.3 \cdot -1 = 6.7. \end{aligned}$$

IDs (more specifically, dynamic IDs) are graphical representations of MDPs and POMDPs, all of which can be used to reason about an agent's actions given incomplete information about the environment. Different from (PO)MDPs, the computational complexity of reasoning under uncertainty in PGMs can be reduced by making use of the defined graphical structures, in particular, the conditional dependencies among random variables [18].

4 Incorporating PGMs into a BDI Architecture

In this section, we describe how to incorporate PGMs into a BDI architecture to model the uncertainty about the situated stochastic environment and an agent's decision making process. First, an agent's epistemic states for random variables, which model the uncertainty about the stochastic environment, and the corresponding belief sets of the epistemic state are defined. The possible states of the environment, sensory observations, and their relationships are modeled using PGMs. The uncertainty propagation is carried out by BNs. The belief sets derived from the epistemic states will trigger the selection of relevant plans. When more than one plan is applicable due to uncertainty in an agent's beliefs, we formulate the plan selection as a decision making process, which models utilities of actions in influence diagrams, and propose a utility-driven approach for plan selection.

4.1 Reasoning about Beliefs under Uncertainty

In a foggy day, the signal may not be as clear as in a normal day. We cannot be certain about the colour of the signal based on a sensory observation, e.g., we cannot be sure whether the signal is red ('Sig(red)') when a sensory observation indicates the signal is red ('sensedSignal(red)'). The relationship between the actual state of the signal and a sensory observation can be modelled by a BN as shown in Fig. 2(a). Inferring the actual state (S node) from an observation (O node) is carried out by the standard reasoning procedure in BNs. In other words, an observation is seen as evidence on O node, and the posterior probability distribution over the states of the signal is calculated using the Bayes' rule.

In the reasoning cycle of the original AgentSpeak agent, given an event, usually a set of applicable plans will be identified according to the belief base. As an autonomous and rational agent, in addition to considering uncertainty about the stochastic environment, the agent may also need to take into account the utilities of actions in plans. In the signal example, when the train agent has no clear idea about the actual state of the signal, it will perform according to the utilities of possible actions. The train agent will be rewarded if it stops, or may face a penalty for violating the traffic regulations. In addition to specifying the uncertainty about the state of a variable and its relationship with its sensory observation in a BN, the relevant actions and their corresponding utilities in various situations are specified in an ID as shown in Fig 3. The plan which include actions with the highest utility will be selected and executed.

First, we formally define the epistemic state [9] to link PGMs to the belief base of an AgentSpeak agent.

Definition 6. *Given a set of discrete random variables* $\overrightarrow{\mathbf{X}} = \{X_1, \ldots, X_n\}$*, a set of actions* $\overrightarrow{\mathbf{D}}$*, and the corresponding utilities of actions in various situations* $\overrightarrow{\mathbf{U}}$*, modeled by an ID* $\mathcal{D}(\overrightarrow{\mathbf{X}}, \overrightarrow{\mathbf{D}}, \overrightarrow{\mathbf{U}})$*, an agent's epistemic state about the states of random variable* X_i *is defined as* $\Phi(X_i) = \langle P_{X_i}, \mathcal{D}(\overrightarrow{\mathbf{X}}, \overrightarrow{\mathbf{D}}, \overrightarrow{\mathbf{U}})\rangle$*, where* $P_{X_i} : S_{X_i} \rightarrow [0, 1]$ *is a prior or marginalized probability distribution obtained via* $\mathcal{D}(\overrightarrow{\mathbf{X}}, \overrightarrow{\mathbf{D}}, \overrightarrow{\mathbf{U}})$ *on the state space* S_{X_i} *of variable* X_i*.*

$\mathcal{D}(\overrightarrow{\mathbf{X}}, \overrightarrow{\mathbf{D}}, \overrightarrow{\mathbf{U}})$ will be referred as $\mathcal{D}$ thereafter if the context is unambiguous. Note that, the ID $\mathcal{D}$ included in the definition of the epistemic state can be simplified into a BN when the utilities of actions are not available.

Definition 7. *Let* $\Phi(X)$ *be an epistemic state for a discrete random variable* X *with its state space* $\{x_1, \ldots, x_n\}$*, the belief set of* $\Phi(X)$*, denoted as* $Bel(\Phi(X))$*, is defined as*

$$Bel(\Phi(X)) = \begin{cases} x_i, when \;\; P_X(x_i) \geq \delta_i \\ \mathsf{T}, \qquad ohterwise \end{cases}$$

Here δ is a pre-defined threshold for accepting that x_i represents the real world concerning X and $\delta > 0.5$. Notation T is a special constant representing an agent's *ignorance*, that is, an agent is not certain about the state of variable X.

Definition 8. *Given an epistemic state* $\Phi(X_i) = \langle P_{X_i}, \mathcal{D}(\overrightarrow{\mathbf{X}}, \overrightarrow{\mathbf{D}}, \overrightarrow{\mathbf{U}})\rangle$*, and a new observation on* X_j*, represented by a probability function* P_{X_j}*, the revision of* $\Phi(X_i)$ *by* P_{X_j} *is defined as*

$$\Phi(X_i) \circ P_{X_j} = \langle P'_{X_i}, \mathcal{D}(\overrightarrow{\mathbf{X}}, \overrightarrow{\mathbf{D}}, \overrightarrow{\mathbf{U}})\rangle$$

Here, P'_{X_i} is the posterior probability distribution on state space S_{X_i} after the propagation of an uncertain input modeled by P_{X_j} using $\mathcal{D}(\overrightarrow{\mathbf{X}}, \overrightarrow{\mathbf{D}}, \overrightarrow{\mathbf{U}})$.

Example 2. In the signal example, given the ID $\mathcal{D}(\{S, O\}, \{D\}, \{U\})$ as shown in Fig. 3, we have an epistemic state $\Phi(S) = \langle P_S, \mathcal{D} \rangle$ to represent the probability distribution over the states of the signal. Assuming we initially have the probability distribution $P_S(\mathsf{red}) = 0.15$ and $P_S(\mathsf{green}) = 0.85$, a new sensory observation very likely 'oG' (i.e., $P_O(\mathsf{oR}) = 0.15$ and $P_O(\mathsf{oG}) = 0.85$) will revise the epistemic state to get $\langle P'_S, \mathcal{D} \rangle$ where $P'_S(\mathsf{red}) = 0.09$ and $P'_S(\mathsf{green}) = 0.91$. Given a threshold $\delta = 0.9$, the belief set of the revised epistemic state $\Phi(S) \circ P_O$ is 'green'. Consequently, a belief atom '$\mathsf{Sig(green)}$' will been added into the agent's belief base.

The posterior probability distribution based on an sensory observation, can be used as the prior probability distribution of the next sensory observation. Specifically, in the signal example, the current posterior probability distribution over the states of the signal serves as the prior probability distribution of the next sensory observation.

4.2 Utility-driven Plan Selection

After reasoning about beliefs under uncertainty in the PGMs, beliefs will be added into an agent's belief base and these beliefs are consistent with the original AgentSpeak framework. In the signal example, when either '$\mathsf{Sig(green)}$' or '$\mathsf{Sig(red)}$' is added, the existing plans (P1-P4 in Example 1)) are still applicable.

Note that, when obtaining the belief sets from epistemic states, the pre-defined threshold may not be exceeded. In this case, the belief with the special constant T will be added to the belief base. In the signal example, when the certainty degree on any state of the signal is not high enough, we will have belief '$\mathsf{Sig(T)}$' to represent the agent's *ignorance* about the actual state of the signal. In this case, the agent can either 'stop' or 'move'. Here, we have two additional plans (P5 and P6) in the plan library.

- (P5)+!drive : $\mathsf{Moving} \wedge \mathsf{Sig(T)} \leftarrow \mathsf{move}$; senseSignal; !drive;
- (P6)+!drive : $\mathsf{Moving} \wedge \mathsf{Sig(T)} \leftarrow \mathsf{stop}$; senseSignal; !drive;

According to the ID $\mathcal{D}$ as shown in Fig. 3(a) and the utility table of the utility node as shown in Fig. 3(b), the train agent's plan selection shall also take into account the utilities of actions. After inferring the probability distribution over the states of the signal from sensory observations, the expected utility of each possible action is calculated. In addition to considering the contexts of plans, the expected utilities of actions in the plans will affect the plan selection of the train agent.

Example 3. In the signal example, two plans are applicable (P5 and P6) when the certainty degree on any state of the signal is not high enough, and we want to select a plan which is more reasonable in terms of the utility. First, the probability distribution over states of the signal (e.g., $P_S(\mathsf{red}) = 0.2$ and $P_S(\mathsf{green}) = 0.8$) will be transferred to the ID $\mathcal{D}$. Afterwards, the expected utilities of actions are calculated: $EU(\mathsf{stop}) = 1.2$ and $EU(\mathsf{move}) = -12$. In this

case, since 'stop' is considered as more beneficial than 'move', plan P6 will be selected.

Specifically, when the probability of a particular state s_i is over a threshold, a single plan can still be selected in the first place. In this case, the expected utility is only calculated for the action in this particular plan, and the action will be selected (since its expected utility will be the maximum one). Therefore, the problem of selecting a plan with normal beliefs is a special case of plan selection when a special belief atom represents the agent's *ignorance* about the state of the variable in the environment.

Now, we extend the definition of an AgentSpeak agent as follows.

Definition 9. *An extended AgentSpeak agent* $\mathbb{A}'$ *is defined by a tuple* $\langle \mathsf{BB}, \mathsf{PLib}, \mathsf{E}, \mathsf{A}, I, \mathcal{S}_\varepsilon, \mathcal{S}_\mathcal{O}, \mathcal{S}_\mathcal{I}, \mathsf{EpS} \rangle$*, where* EpS *is a set of epistemic states, with all other items being the same as defined in Section 3.*

It should be noted that we allow an agent to have multiple epistemic states, each for a different variable (or a set of variables). This assumption is reasonable since an agent may have beliefs (which could be uncertain) and knowledge about different parts of the situated environment (we assume these parts are *localized* or *isolated*). For each part, the knowledge, which is represented by a BN or an ID, and its associated beliefs constitute one *localized* or *isolated* epistemic state.

The reasoning cycle of an extended AgentSpeak agent is shown in Fig. 4. Compared to the original reasoning cycle as shown in Fig. 1, the observations containing uncertain information go through epistemic states before affecting the belief base. The PGMs included in the epistemic states are used to reason about uncertain observations and derive belief sets. After applicable plans have been identified, in order to select a unique one to execute, expected unities of possible actions are calculated using the PGM.

5 MAS Transit Scenario

We use a transit scenario to validate the effectiveness of the hybrid BDI-PGM framework. We describe how to use Jason [17] to implement this scenario. Jason is an open-source interpreter for AgentSpeak which implements AgentSpeak's operational semantics and thus provides a platform for developing MAS. We use Hugin [23] to reason over the embedded PGMs. Hugin provides a friendly user interface to create, edit and manipulate BNs and IDs, and it also wraps up all functionalities (e.g., inserting evidence and propagating uncertainty) as an API for further development purposes. The implemented prototype can be considered as a proof of concept of the feasibility of applying advanced MAS technology for developing SCADA systems in the presence of uncertainty about the dynamic environment.

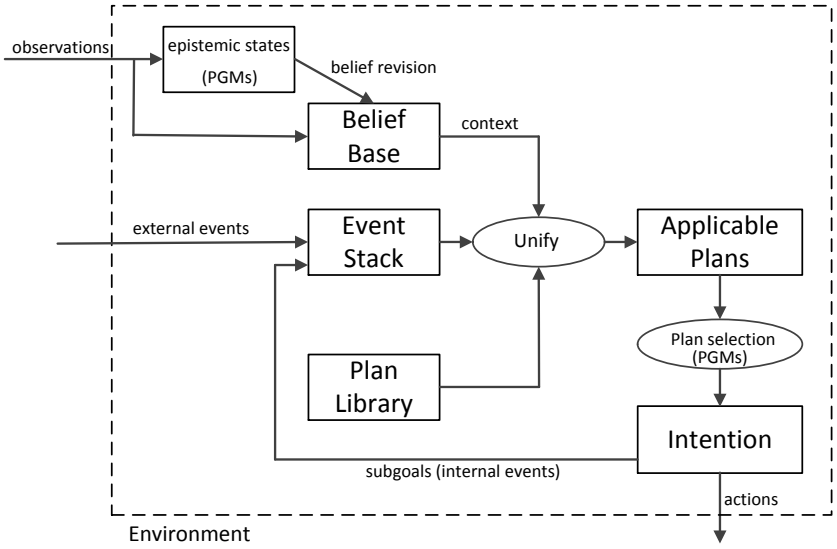


Fig. 4. The reasoning cycle of the hybrid BDI-PGM framework.

5.1 Transit Scenario

There are two transit trains running in the same direction on a one directional rectangle track that links two stations (as shown in Fig. 5). There are two train stations and two signals. Each train starts moving and keeps checking the signal when it is within the sensing range. After a train sees a red signal, it stops and keeps checking the signal. When a trains sees a green signal, it either keeps moving if it is already moving or starts moving if it is still, and it keeps checking the signal.

We assume that the weather is foggy, therefore the signal cannot be sensed with certainty. The rest of this scenario is assumed to be certain (i.e., trains do not fail and the agent always succeed in closing the door).

There is a PGM (specifically, the ID shown in Fig. 3) specified by Hugin, including the graphical structure and the parameters, to represent the knowledge about the noisy sensing situation.

5.2 MAS Implementation

Each agent's belief base contains both static information (i.e., the location of the stations and the signal) and dynamic information (i.e., the locations of both trains). Here is a snapshot of the belief base of the train agent $\mathbb{A}$.

- Train($\mathbb{B}$, 300): train $\mathbb{B}$'s position;
- Signal(1, 500): the signal 1's position;
- Moving: train $\mathbb{A}$ is moving;

– Sig(1, red): the signal 1 is red.

The train agents can perform the actions listed in Example 1. The goal of the entire system is to run trains safely and smoothly. The two train agents require communication with each other about their current positions.

As part of the infrastructure of the transit system, the environment class in Jason is extended and customized to handle the action of each train agent. The environment class revises an agent's belief base as a consequence of an action. The communication between train agents are also treated as actions by the environment class.

The most part of this transit scenario is certain. With respect to sensing the signal, we implemented the approach described in Section 4. Initially, the PGM specified in the Hugin format is included into the train agent's epistemic state, and a uniform prior distribution on the S node is inserted. When the sensory action 'senseSignal' is performed, an sensory observation is first sampled from the probability distribution on the observations given the probability distribution on the states of the signal, following the Monte Carlo principle. Afterwards, the sensory observation is inserted into the O node of the PGM as evidence. After propagating the uncertainty using the Hugin engine, a belief atom is derived from the posterior probability distribution about the signal, and this belief atom will be added into the train agent's belief base. Furthermore, the posterior probability distribution will serve as the prior probability distribution on the state of the signal in the next step if required. As discussed in Section 4, the original AgentSpeak plan selection procedure is still applicable.

When the train agent received the belief 'Sig(T)' for the *ignorance* about the actual state of the signal, it will further consider the utilities of actions. Besides the observation sampling and uncertainty propagation, the expected utility of each action is calculated. In this case, the difference between two applicable plans (P5 and P6) comes from their first actions. Other items in these plans are ignored since they are not relevant to the actions in the PGM. The plan, which has the action with the highest utility, will be selected.

In the customized environment class, there is also a visual panel showing the entire scenario as shown in Fig. 5. All agents (the ones for the trains, the signal, and the train stations) are located on a rectangular track. The belief base and the information from the epistemic state of each train agent are listed in the left bottom. The environment also receives input from the buttons on the right bottom (e.g. switching the signals and informing that the passengers boarding is complete). These buttons simulate the events generated by the agents for signals and train stations. The probability distribution about the signals included in agents' epistemic states are shown near the track.

5.3 Testing Scenarios

Based on the defined actions, a rich class of behaviours of the transit scenario can be specified. For example,

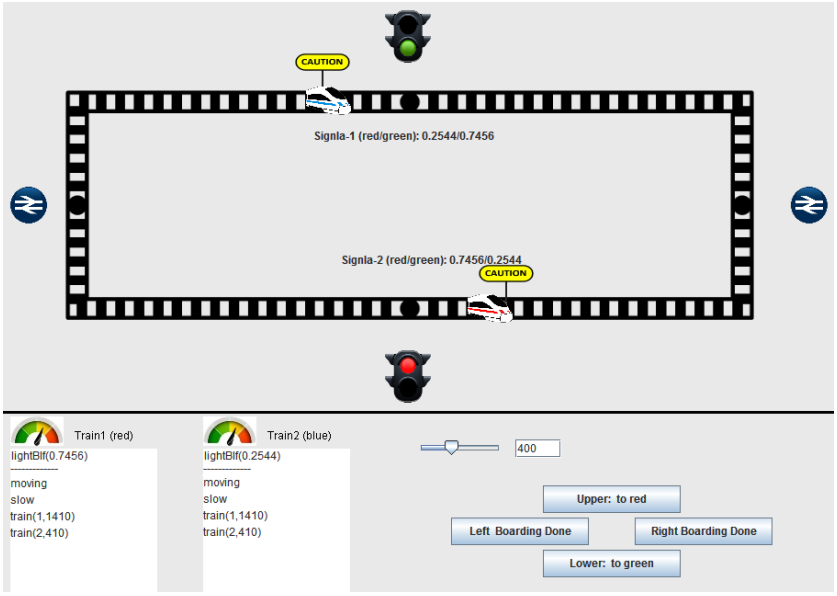


Fig. 5. The transit train scenario.

– When the train is approaching the train station, it stops first and opens its doors for passenger boarding. Afterwards, the train keeps sensing whether the boarding is complete. The train will then close its doors and start moving again.
– The safe distance between trains is always kept.

With respect to the noisy sensing of the signal, we considered the following two situations:

– When one train is entering the sensing range of a signal which is green, it keeps sensing the signal and moving forward at a normal speed. Its epistemic state about the actual state of the signal keeps changing until leaving the sensing range.
– Assume the signal is initially green, and the train proceeds at a normal speed. When the train is approaching the signal, we switch the signal to red. The train's epistemic state and its belief base will change after making several observations. Additionally, since the train is close to the signal, the belief is updated quickly since the noise is small (we implicitly encoded this information into the conditional probability tables in the BNs). A sudden break stops the train.

All these behaviours have been validated by the implemented prototype.

6 Conclusions

In this paper, we addressed the problem of handling uncertainty about a stochastic environment in a BDI architecture by presenting a hybrid BDI-PGM framework. In particular, we used PGMs to model the uncertainty about the environment and their relationships. We further took into account the utilities of actions and specified these utilities in PGMs. The PGMs are included in the agents' epistemic state, and corresponding belief sets can be derived from epistemic states, so that the original AgentSpeak plan selection is still applicable. When an agent has a special belief, e.g., '$\mathsf{Sig(T)}$', representing the *ignorance* about the actual world, we integrate the decision making process into the plan selection procedure. In order to test the proposed hybrid BDI-PGM framework, we designed and implemented a simplified transit scenario. In this prototype, all the required behaviours are validated.

It is possible that in a dynamically changing environment, the pre-specified plans are not complete. As future work, algorithms for automatically building plan libraries for the hybrid BDI-PGM framework are required. In a multiagent setting, uncertainty about the environment can be complicated by behaviours of other agents, which are subject to their relationships. These cooperating/adversary behaviours of agents have not be discussed in this paper, and deserve further investigation. For PGMs, setting reasonable parameters (i.e., the numbers in conditional probability tables and utility tables) is always challenging. There are sophisticated techniques in the literature that can facilitate the parameter learning and the structure discovery for PGMs. Learning from the log data of existing SCADA systems would be beneficial. Furthermore, quantitative comparison study with other uncertain BDI architectures is also necessary.

Acknowledgment

The authors are grateful to Kim Bauters for his helpful comments and anonymous reviewers for their constructive comments and insights.

References

1. S. A. Boyer, *SCADA: Supervisory Control And Data Acquisition*. International Society of Automation, 2009.
2. N. R. Jennings and S. Bussmann, "Agent-based control systems," *IEEE Control Systems Magazine*, vol. 23, pp. 61–74, 2003.
3. I. Guilherme, R. Pedrosanto, A. Teixeira, C. K. Morooka, and C. Sierra, "A multiagent architecture for supervisory and control system," in *CIMCA/IAWTIC/ISE*, pp. 98–103, 2008.
4. P. Valckenaers, J. Sauter, C. Sierra, and J. Rodriguez-Aguilar, "Applications and environments for multi-agent systems," *JAAMAS*, vol. 14, no. 1, pp. 61–85, 2007.
5. A. S. Rao and M. P. Georgeff, "An abstract architecture for rational agents," in *KR*, pp. 439–449, 1992.

6. S. McArthur, E. Davidson, V. Catterson, A. Dimeas, N. Hatziargyriou, F. Ponci, and T. Funabashi, "Multi-agent systems for power engineering applications - part I: Concepts, approaches, and technical challenges," *IEEE Trans. on Power Systems*, vol. 22, no. 4, pp. 1743–1752, 2007.
7. S. McArthur, E. Davidson, V. Catterson, A. Dimeas, N. Hatziargyriou, F. Ponci, and T. Funabashi, "Multi-agent systems for power engineering applications - part II: Technologies, standards, and tools for building multi-agent systems," *IEEE Trans. on Power Systems*, vol. 22, no. 4, pp. 1753–1759, 2007.
8. A. S. Rao, "Agentspeak(L): BDI agents speak out in a logical computable language," in *MAAMAW*, pp. 42–55, 1996.
9. J. Ma and W. Liu, "A framework for managing uncertain inputs: An axiomization of rewarding," *IJAR*, vol. 52, no. 7, pp. 917–934, 2011.
10. I. Baxevanos and D. Labridis, "Implementing multiagent systems technology for power distribution network control and protection management," *IEEE Trans. on Power Delivery*, vol. 22, no. 1, pp. 433–443, 2007.
11. S. Parsons and P. Giorgini, "On using degrees of belief in BDI agents," in *IPMU*, 1998.
12. A. Casali, L. Godo, and C. Sierra, "A graded BDI agent model to represent and reason about preferences," *AIJ*, vol. 175, no. 7-8, pp. 1468 – 1478, 2011.
13. F. V. Jensen and T. D. Nielsen, *Bayesian Network and Decision Graphs*. Springer, 2007.
14. M. S. Fagundes, R. M. Vicari, and H. Coelho, "Deliberation process in a BDI model with Bayesian networks," in *PRIMA*, pp. 207–218, 2007.
15. B. Luz, F. Meneguzzi, and R. Vicari, "Alternatives to threshold-based desire selection in Bayesian BDIagents," in *EMAS*, pp. 208–223, 2013.
16. G. Kieling and R. Vicari, "Insertion of probabilistic knowledge into BDI agents construction modeled in Bayesian networks," in *CISIS*, pp. 115–122, 2011.
17. R. H. Bordini, J. F. Hübner, and M. Wooldridge, *Programming Multi-agent Systems in AgentSpeak using Jason*. Wiley-Interscience, 2007.
18. P. Doshi, Y. Zeng, and Q. Chen, "Graphical models for interactive POMDPs: representations and solutions," *JAAMAS*, vol. 18, no. 3, pp. 376–416, 2009.
19. M. Schut, M. Wooldridge, and S. Parsons, "On partially observable MDPs and BDI models," in *UKMAS*, pp. 243–260, 2002.
20. G. I. Simari and S. Parsons, "On the relationship between MDPs and the BDI architecture," in *AAMAS*, pp. 1041–1048, 2006.
21. G. I. Simari and S. D. Parsons, "On approximating the best decision for an autonomous agent," in *GTDT*, 2004.
22. R. Nair and M. Tambe, "Hybrid BDI-POMDP framework for multiagent teaming," *JAIR*, vol. 23, pp. 367–420, Apr. 2005.
23. S. K. Andersen, K. G. Olesen, F. V. Jensen, and F. Jensen, "HUGIN - a shell for building Bayesian belief universes for expert systems," in *IJCAI*, pp. 1080–1085, 1989.