Reasoning About Intentions in Uncertain Domains

Martijn Schut and Michael Wooldridge and Simon Parsons

Department of Computer Science
University of Liverpool
Liverpool L69 7ZF, U.K.

{m.c.schut, m.j.wooldridge, s.d.parsons}@csc.liv.ac.uk

Abstract. The design of autonomous agents that are situated in real world domains involves dealing with uncertainty in terms of dynamism, observability and non-determinism. These three types of uncertainty, when combined with the real-time requirements of many application domains, imply that an agent must be capable of effectively coordinating its reasoning. As such, situated belief-desire-intention (BDI) agents need an efficient intention reconsideration policy, which defines when computational resources are spent on reasoning, i.e., deliberating over intentions, and when resources are better spent on either object-level reasoning or action. This paper presents an implementation of such a policy by modelling intention reconsideration as a partially observable Markov decision process (POMDP). The motivation for a POMDP implementation of intention reconsideration is that the two processes have similar properties and functions, as we demonstrate in this paper. Our approach achieves better results than existing intention reconsideration frameworks, as is demonstrated empirically in this paper.

1 Introduction

One of the key problems in the design of belief-desire-intention (BDI) agents is the selection of an *intention reconsideration policy* [3, 8]. Such a policy defines the circumstances under which a BDI agent will expend computational resources deliberating over its intentions. Wasted effort — deliberating over intentions unnecessarily — is undesirable, as is not deliberating when such deliberation would have been fruitful. There is currently no consensus on exactly how or when an agent should reconsider its intentions. Current approaches to this problem simply dictate the *commitment level* of the agent, ranging from *cautious* (agents that reconsider their intentions at every possible opportunity) to *bold* (agents that do not reconsider until they have fully executed their current plan). Kinny and Georgeff investigated the effectiveness of these two policies in several types of environments [3]; their analysis has been extended by others [8].

Our objective in this paper is to demonstrate how to model intention reconsideration in belief-desire-intention (BDI) agents by using the theory of Markov decision processes for planning in partially observable stochastic domains. We view an intention reconsideration strategy as a policy in a partially observable Markov decision process (POMDP): solving the POMDP thus means finding an optimal intention reconsideration strategy. We have shown in previous work [8] that an agent's optimal rate of reconsideration depends on the environment's *dynamism* – the rate of change of the environment, *determinism* – the degree of predictability of the system behaviour for identical system inputs, and *observability* – the extent to which the agent has access to the state of the environment. The motivation for using a POMDP approach here is that in the POMDP framework the optimality of a policy is largely based on exactly these three environmental characteristics.

The remainder of this paper is structured as follows. We begin by providing some background information on the BDI framework in which the problem of intention reconsideration arises. In Section 3 we discuss the Markov decision framework upon which our approach builds and present the implementation of intention reconsideration with a POMDP. In Section 4 we empirically evaluate our model in an agent testbed. Finally, in Section 5 we present some conclusions and describe related and future work.

2 Belief-Desire-Intention Agents

The idea of applying the concepts of beliefs, desires and intentions to agents originates in the work of Bratman [2] and Rao and Georgeff [6]. In this paper, we use the conceptual model of BDI agency as developed by Wooldridge and Parsons [10]. The model distinguishes two main data structures in an agent: a *belief* set and an *intention* set[1]. An agent's beliefs represent information that the agent has about its environment, and may be partial or incorrect. Intentions can be seen as states of affairs that an agent has committed to bringing about. We regard an intention as a simple unconditional plan.The behaviour of the agent is generated by four main components: a *next-state* function, which updates the agent's beliefs on the basis of an observation made of the environment; a *deliberation* function, which constructs a set of appropriate intentions on the basis of the agent's current beliefs and intentions; an *action* function, which selects and executes an action that ultimately satisfies one or more of the agent's intentions; and a *meta-level control* function, the sole purpose of which is to decide whether to pass control to the deliberation or action subsystems. On any given control cycle, an agent begins by updating its beliefs through its next-state function, and then, on the basis of its current beliefs, the meta-level control function decides whether to pass control to the deliberation function (in which case the agent expends computational resources by deliberating over its intentions), or else to the action subsystem (in which case the agent acts). As a general rule of thumb, an agent's meta-level control system should pass control

[1] Since desires do not *directly* contribute to our analytical discussion of intention reconsideration, they are left out of the conceptual BDI model in this paper. This decision is clarified in [10].

to the deliberation function when the agent will change intentions as a result; otherwise, the time spent deliberating is wasted. Investigating how this choice is made rationally and efficiently is the main motivation behind the work presented in this paper.

We have to consider that agents do not operate in isolation: they are situated in *environments*; an environment denotes everything that is external to the agent. Let P be a set of *propositions* denoting environment variables. In accordance with similar proposition based vector descriptions of states, we let environment states be built up of such propositions. Then E is a set *environment states* with members $\{e, e', \ldots\}$, and $e = \{p_1, \ldots, p_n\}$, where $p_i \in P$.

The internal state of an agent consists of beliefs and intentions. Let $Bel : E \to [0,1]$, where $\sum_{e \in E} Bel(e) = 1$, denote the agent's *beliefs*: we represent what the agent believes to be true of its environment by defining a probability distribution over the possible environment states. The agent's set of *intentions*, Int, is a subset of the set of environment variables: $Int \subseteq P$. An internal state s is a pair $s = \langle Bel, Int \rangle$, where $Bel : E \to [0,1]$ is a probability function and $Int \subseteq P$ is a set of intentions. Let S be the set of all internal states. For a state $s \in S$, we refer to the beliefs in that state as Bel_s and to the intentions as Int_s. We assume that it is possible to denote values and costs of the outcomes of intentions[2]: an *intention value* $V : Int \to \mathbb{R}$ represents the value of the outcome of an intention; and *intention cost* $C : Int \to \mathbb{R}$ represents the cost of achieving the outcome of an intention. The *net value* $V_{net} : Int \to \mathbb{R}$ represents the net value of the outcome of an intention; $V_{net}(i)$, where $i \in Int$, is typically $V(i) - C(i)$. We can express how "good" it is to be in some state by assigning a numerical value to states, called the *worth* of a state. We denote the worth of a state by a function $W : S \to \mathbb{R}$, and we assume this to be based on the net values of the outcomes of the intentions in a state. Moreover, we assume that one state has an higher worth than an other state if the net values of all its intentions are higher. This means that if $\forall s, s' \in S, \forall i \in Int_s, \forall i' \in Int_{s'}, V_{net}(i) \geq V_{net}(i')$, then $W(s) \geq W(s')$. In the empirical investigation discussed in this paper, we illustrate that a conversion from intention values to state worths is feasible, though we do not explore the issue here[3]. Finally, Ac denotes the set of physical actions the agent is able to perform; with every $\alpha \in Ac$ we identify a set of propositions $P_\alpha \subseteq P$, which includes the propositions that change value when α is executed.

In this conceptual model, the question of intention reconsideration thus basically boils down to the implementation of the meta-level control function. On every given control cycle, the agent must decide whether it acts upon its cur-

[2] We clearly distinguish intentions from their outcome states and we do not give values to intentions themselves, but rather to their outcomes. For example, when an agent *intends* to deliver coffee, an *outcome* of that intention is the state in which coffee has been delivered.

[3] Notice that this problem is the inverse of the utilitarian *lifting problem*: the problem of how to lift utilities over states to desires over sets of states. Discussing the lifting problem, and its inverse, is beyond the scope of this paper, and therefore we direct the interested reader to the work of Lang et al. [4].

rent intentions, or to adopt new intentions and this is decided by the meta-level control function. We continue with discussing how this implementation can be done by using Markov decision processes.

3 Implementing Intention Reconsideration as a POMDP

In this paper, the main point of our formalisation of intention reconsideration is the POMDP implementation of it. The fact that the optimality of a POMDP policy is based on the environment's observability, determinism and dynamism, renders the framework appropriate in the context of intention reconsideration. In this section, we explain what a POMDP is and how to use it for implementing intention reconsideration.

A partially observable Markov Decision Process (POMDP) can be understood as a system that at any point in time can be in any one of a number of distinct states, in which the system's state changes over time resulting from actions, and where the current state of the system cannot be determined with complete certainty [1]. Partially observable MDPs satisfy the Markov assumption so that knowledge of the current state renders information about the past irrelevant to making predictions about the future. In a POMDP, we represent the fact that the knowledge of the agent is not complete by defining a probability distribution over all possible states. An agent then updates this distribution when it observes its environment.

Let a set of states be denoted by S and let this set correspond to the set of the agent's internal states as defined above. This means that a state in the MDP represents an internal state of the agent. We let the set of actions be denoted by A. (We later show that $A \neq Ac$ in our model.) An agent might not have complete knowledge of its environment, and must thus *observe* its surroundings in order to acquire knowledge: let Ω be a finite set of observations that the agent can make of the environment. We introduce an *observation function* $O : S \times A \to \Pi(\Omega)$ that defines a probability distribution over the set of observations; this function represents what observations an agent can make resulting from performing an action $a \in A$ in a state $s \in S$. The agent receives rewards for performing actions in certain states: this is represented by a *reward function* $R : S \times A \to \mathbb{R}$. Finally, a *state transition* function $\tau : S \times A \to \Pi(S)$ defines a probability distribution over states resulting from performing an action in a state – this enables us to model non-deterministic actions.

Having defined these sets, we *solve* a POMDP by computing an *optimal policy*: an assignment of an action to each possible belief state such that the expected sum of rewards gained along the possible trajectories in the POMDP is a maximum. Optimal policies can be computed by applying dynamic programming methods to the POMDP, based on backwards induction; value iteration and policy iteration are the most well known algorithms to solve POMDPs [1]. A major drawback of applying POMDPs is that these kinds of algorithms tend to be highly intractable; we later return to the issue of computational complexity as it relates to our model.

Intention Reconsideration as a POMDP

We regard the BDI as a *domain dependent object level* reasoner, concerned directly with performing the best action for each possible situation; the POMDP framework is then used as a *domain independent meta level reasoning* component, which lets the agent reconsider its intentions effectively. We define a meta level BDI-POMDP as a tuple $\langle S, A, \Omega, O, R, \tau \rangle$. We have explained above that a state $s \in S$ in this model denotes an internal state of the agent, containing a belief part and intention part. As intention reconsideration is mainly concerned with states, actions and rewards, we leave the implementation of observations Ω, the observation function O and the state transition function τ to the designer for now.

Since the POMDP is used to model intention reconsideration, we are merely concerned with two possible meta level actions: the agent either performs an object level action (*act*) or the agent deliberates (*del*). The possible actions $A = \{act, del\}$ correspond to the agent either acting (*act*) or deliberating (*del*). Because the optimality criterion of policies depends on the reward structure of the POMDP, we define the rewards for action *act* and deliberation *del* in state $s \in S$ as follows:

$$R(s,a) = \begin{cases} W(s_{int}) & \text{if} \quad a = act \\ W(s) & \text{if} \quad a = del \end{cases}$$

where $s_{int} \in S$ refers to the state the agent intends to be in while currently being in state s. Imagine a robot that has just picked up an item which has to be delivered at some location. The agent has adopted the intention to deliver the item, i.e., to travel to that location and to drop off the item. The reward for deliberation is the worth of the agent's current state (e.g., 0) whereas the reward for action is the worth of the intented state (e.g., 10) for having delivered the item. The robot consequently acts, which brings it closer to its "correct" intentions. Intentions are correct in case the agent does not waste effort while acting upon them. An agent wastes effort if it is deliberating over its intentions unnecessarily. If an agent does not deliberate when that would have been necessary, the agent has wrong intentions. The reward for acting is thus the worth of the state that the agent intends to reach, whereas the reward for deliberation is the worth of the state as it currently is.

This structure of reward agrees with the intuition that the agent eventually receives a reward if it has correct intentions, it receives no reward if it has wrong intentions, and it receives no *direct* reward for deliberation. With respect to this last intuition, however, we must mention that the "real" reward for deliberation is indirectly defined, by the very nature of POMDPs, as the expected worth of future states in which the agent has correct intentions. As intentions resist reconsideration [2], the agent prefers action over deliberation and the implementation of the reward structure should thus favour action if the rewards are equivalent.

For illustrative purposes, consider the simple deterministic MDP in Figure 1. This Figure shows a 5×1 gridworld, in which an agent can move either right or left or stay at its current location. The agent's current location is indicated with

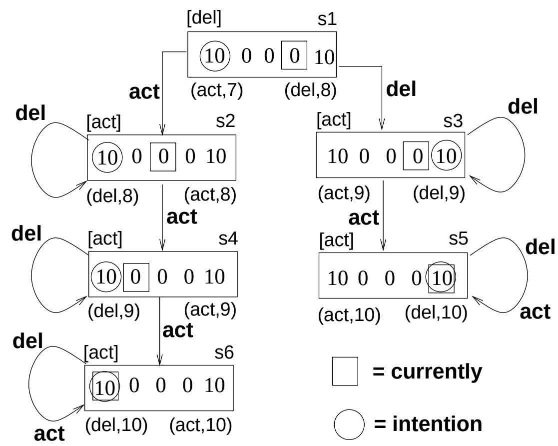


Fig. 1. A 5×1 gridworld example which illustrates the definition of rewards in a BDI-POMDP. Rewards, being either 0 or 10, are indicated per location. With each state we have indicated the expected reward for executing a physical action and for deliberation; the best meta action to execute is indicated in square brackets.

a square and the location it intends to travel to is denoted by a circle. Assume the agent is currently in state s_1: its location is cell 4 and it intends to visit cell 1. Action will get the agent closer to cell 1: it executes a move left action which results in state s_2. Deliberation results in dropping the intention to travel to cell 1, and adopting the intention to travel to cell 5 instead; this results in state s_3. Obviously, deliberation is the best meta action here and the expected rewards for the meta actions in s_1 reflect this: the expected reward for deliberation is higher than the one for action. In all other states, these expected rewards are equivalent, which means that the agent acts in all other states.

Solving a BDI-POMDP means obtaining an optimal intention reconsideration policy: at any possible state the agent might find itself in, this policy tells the agent either to act or to deliberate. The main contribution of our work is that our approach gives a well-founded means of establishing a domain dependent optimal reconsideration strategy. Thus the agent is programmed with a domain independent strategy, which it uses to compute a domain dependent strategy off-line, and then executes it on-line. Until now, empirical research on meta level reasoning aimed at efficient intention reconsideration has, to the best of our knowledge, involved hardwiring agents with domain dependent strategies.

It is important that deciding whether to reconsider intentions or not is computationally cheap compared to the deliberation process itself [10]; otherwise it is just as efficient to deliberate at any possible moment. Using a POMDP to determine the reconsideration policy satisfies this criterion, since it clearly distinguishes between design time computation, i.e., computing the policy, and run time computation, i.e., executing the policy. We recognise that the design time problem of computing a policy is very hard; this problem corresponds with the general problem of solving POMDPs and we do not attempt to solve this problem in this paper. However, the computation that concerns us most is the run time computation, and in our model this merely boils down to looking up the current state and executing the action assigned to that state, i.e., either to act or to deliberate. This is a computationally cheap operation and is therefore suitable for run time execution.

4 Experimental Results

In this section, we apply our model in the TILEWORLD testbed [5], and show that the model yields better results than were obtained in previous investigations of intention reconsideration in this testbed[4].

The TILEWORLD [5] is a grid environment on which there are agents and holes. An agent can move up, down, left, right and diagonally. Holes have to be visited by the agent in order for it to gain rewards. The TILEWORLD starts in some randomly generated world state and changes over time with the appearance and disappearance of holes according to some fixed distributions. An agent moves about the grid one step at a time[5]. The experiments are based on the methodology described in [8]. (We repeated the experiments described in [8] to ensure that our results were consistent; these experiments yielded identical results, which are omitted here for reasons of space.)

The TILEWORLD testbed is easily represented in our model. Let L denote the set of locations, i.e., $L = \{i : 1 \leq i \leq n\}$ represents the mutually disjoint locations, where n denotes the size of the grid. A proposition p_i then denotes the presence ($p_i = 1$) or absence ($p_i = 0$) of a hole at location i. An intention value corresponds to the reward received by the agent for reaching a hole, and an intention cost is the distance between the current location of the agent and the location that the agent intends to reach. An environment state is a pair $\langle\{p_i, \ldots, p_n\}, m\rangle$, where $\{p_i, \ldots, p_n\}$ are the propositions representing the holes in the grid, and $m \in L$ is the current location of the agent.

Combining the $2^n \times n$ possible environment states with n possible intentions means that, adopting explicit state descriptions, the number of states is $2^n \times n^2$, where n denotes the number of locations. Computations on a state space of such size is impractical, even for small n. In order to render the necessary computations feasible, we *abstracted* the TILEWORLD state space. In the TILEWORLD domain, we abstract the state space by letting an environment state e be a pair $\langle p_1, p_2\rangle$, where p_1 refers to the location of the hole which is currently closest to

[4] Whereas until now we have discussed non-deterministic POMDPs, in the experimental section we restrict our attention to deterministic MDPs in order to compare our new results with previous results.

[5] Although it may be argued that the TILEWORLD is simplistic, it is a well-recognised testbed for evaluating situated agents. Because of the dynamic nature of the TILEWORLD, the testbed scales up to difficult and unsolvable problems.

the agent, and p_2 refers to the current location of the agent. Then an agent's internal state is $\langle\langle p_1, p_2\rangle, \{i_1\}\rangle$ where i_1 refers to the hole which the agent intends to visit. This abstraction means that the size of the state space is now reduced to n^3. However, the agent now has to figure out at run time what is the closest hole in order to match its current situation to a state in the TILEWORLD state space. This computation can be done in time $O(n)$, by simply checking whether every cell is occupied by a hole or not. Because the main purpose of this example is merely to illustrate that our model is viable, we are currently not concerned with this increase in run time computation.

In [8], the performance of a range of intention reconsideration policies was investigated in environments of differing structure. Environments were varied by changing the degree of dynamism (γ), observability (referred to by [8] as *accessibility*), and determinism. Dynamism is denoted by an integer in the range 1 to 80, representing the ratio between the world clock rate and the agent clock rate. If $\gamma = 1$, then the world executes one cycle for every cycle executed by the agent and the agent's information is guaranteed to be up to date; if $\gamma > 1$ then the information the agent has about its environment may not necessarily be up to date. (In the experiments in this paper we assume the environment is fully observable and deterministic.) The *planning cost* p was varied, representing the time cost of planning, i.e., the number of time-steps required to form a plan, and took values 0, 1, 2, and 4.

Three dependent variables were measured: effectiveness, commitment, and cost of acting. The *effectiveness* ϵ of an agent is the ratio of the actual score achieved by the agent to the score that could in principle have been achieved. An agent's *commitment* (β) is expressed as how many actions of a plan are executed before the agent replans. The agent's commitment to a plan with length n is $(k-1)/(n-1)$, where k is the number of executed actions. Observe that commitment defines a spectrum from a cautious agent ($\beta = 0$, because $k = 1$) to a bold one ($\beta = 1$, because $k = n$). The *cost of acting* is the total number of actions the agent executes.

Solving the TILEWORLD MDP **off-line** To summarise, the TILEWORLD MDP that we have to solve off-line consists of the following parts. As described above, the state space S contains all possible internal states of the agent. Each state $s \in S$ is a tuple $\langle\langle p_1, p_2\rangle, \{i_1\}\rangle$, where p_1 refers to hole that is currently closest to the agent, p_2 refers to the current location of the agent, and i_1 denotes the hole which the agent intends to visit. The set of actions is $A = \{act, del\}$. (Note that the set of physical actions is $Ac = \{stay, n, ne, e, se, se, sw, w, nw\}$, but that is not of concern to us while specifying the TILEWORLD MDP.) Since we assume full observability, the set of observations is $\Omega = S$. Finally, state transitions are defined as the deterministic outcomes of executing an action $a \in A$. As the agent deliberates in state s resulting in state s' (i.e., $\tau(s, del) = s'$), then $Bel_s = Bel_{s'}$, but possibly $Int_s \neq Int_{s'}$; as the agent acts (i.e., $\tau(s, act) = s''$), then $Int_s = Int_{s''}$, but possibly $Bel_s \neq Bel_{s''}$. Thus deliberation means that the intention part of the agent's internal state possibly changes, and action

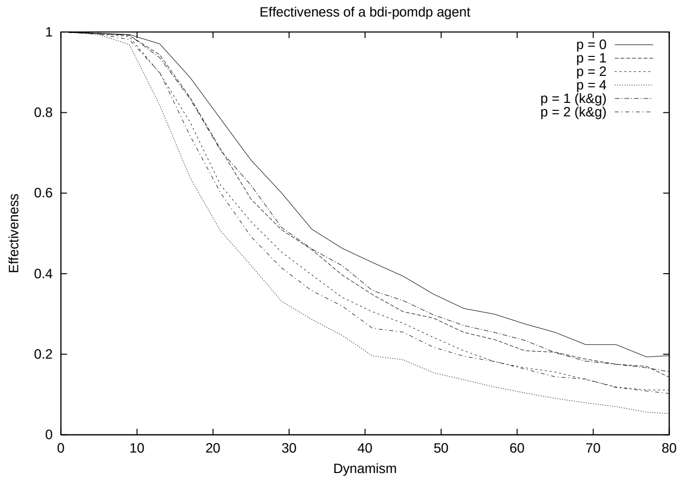


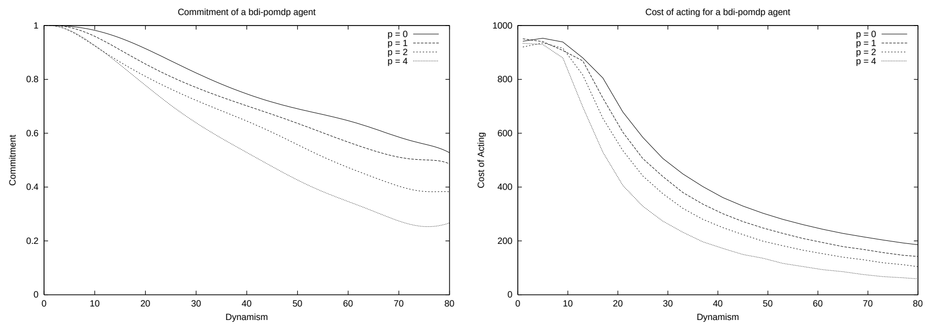
Fig. 2. Overall effectiveness of a BDI-POMDP agent. Effectiveness is measured as the result of a varying degree of dynamism of the world. The four curves show the effectiveness at a planning cost (denoted by p) from 0 to 4. The two other curves show the effectiveness at $p = 1$ and $p = 2$ of Kinny and Georgeff's best reconsideration strategy (from [3]).

means that the belief part of the agent's internal state possibly changes (both ceteris paribus with respect to the other part of the internal state). Although solving MDPs in general is computationally hard, we have shown above that by appropriate abstraction of the TILEWORLD state space, the computations for our TILEWORLD MDP become feasible.

Results The experiments resulted in the graphs shown in Figures 2, 3(A) and 3(B). In every graph, the environment's dynamism and the agent's planning cost p (for values 0, 1, 2 and 4) are varied. In Figure 2, the overall effectiveness of the agent is plotted. In Figure 3(A) we plotted the agent's commitment level[6] and in Figure 3(B) the cost of acting.

Analysis The most important observation we make from these experiments is that the results as presented in Figure 2 are overall better than results as obtained in previous investigations into the effectiveness of reconsideration (as

[6] The collected data was smoothed using a Bezier curve in order to get these commitment graphs, because the commitment data showed heavy variation resulting from the way dynamism is implemented. Dynamism represents the acting ratio between the world and the agent; this ratio oscillates with the random distribution for hole appearances, on which the commitment level depends.



(A) (B)

Fig. 3. (A) Average commitment level for a BDI-POMDP agent. The commitment level is plotted as a function of the dynamism of the world with planning cost (denoted by p) of 0, 1, 2 and 4. (B) Average cost of acting for a BDI-POMDP agent. The cost of acting – the number of time steps that the agent moves – is plotted as a function of the dynamism of the world with planning cost (denoted by p) of 0, 1, 2 and 4.

elaborated below). Our explanation for this observation is that solving the BDI-POMDP for our TILEWORLD domain delivers an optimal domain dependent reconsideration strategy: the optimal BDI-POMDP policy lets the agent deliberate when a hole appears that is closer than the intended hole (but not on the path to the intended hole), and when the intended hole disappears. Kinny and Georgeff [3] concluded that it is best for an agent to reconsider when a closer hole appears or when the intended hole disappears. Besides this observation, we see in Figure 3(A) that our BDI-POMDP agent is able to determine its plan commitment at run time, depending on the state of the environment. This ability contributes to increasing the agent's level of autonomy, since it pushes the choice of commitment level from design time to run time.

Our experimental results confirm the results obtained in previous investigations on selecting an intention reconsideration strategy [3, 8, 9]: the agent's effectiveness and level of commitment both decrease as the dynamism or planning cost increases, and the cost of acting decreases as the dynamism or planning cost increases.

Whereas the focus of previous research was on investigating the effectiveness of *fixed* strategies in different environments, the aim of the investigation in this paper is to illustrate the applicability of our BDI-POMDP model. Kinny and Georgeff [3] have included empirical results for an agent that reconsiders based on the occurrence of certain events in the environment (see [3, p87] Figures 8 and 9 for $p = 2$ and $p = 1$, respectively). Their conclusion from these results was that it is best for an agent to reconsider when the agent observes that either a closer hole appears or the intended hole disappears, as mentioned above. We implemented this strategy for the agent in our testbed and yielded identical results. We observed that an agent using our BDI-POMDP model performs better than the agent using the mentioned fixed strategy with a realistic planning cost ($p \geq 2$). Having compared our results to the results of fixed strategies, we conclude that, as mentioned above, in effect, our agent indeed adopts the strategy that delivers maximum effectiveness.

In the context of *flexible* strategies, we compare our results to the results from [9], where the effectiveness of an alternative flexible strategy, based on discrete deliberation scheduling [7], is explored. The main conclusion we draw from comparing the results from the two strategies is that the empirical outcomes are analogous. Comparing the graphs from Figure 2 to the result graphs from [9], we observe that the agent's effectiveness is generally higher for our BDI-POMDP model; when we compare the graphs from Figure 3(B) to the cost of acting graphs from [9], we see that the cost of acting is lower overall in the discrete deliberation model. However, in our BDI-POMDP model, the level of commitment is more constant, since the BDI-POMDP agent's decision mechanism depends less on predictions of appearances and disappearances of holes.

5 Discussion

In this paper we presented a formalisation of the intention reconsideration process in BDI agents based on the theory of POMDP planning. The motivation for the formalism is that BDI agents in real world application domains have to reconsider their intentions efficiently in order to be as effective as possible. It is important that reconsideration happens autonomously, since an agent's commitment to its tasks changes depending on how its environment changes. The main contribution of our model is that we deliver a meta level and domain independent framework capable of producing optimal reconsideration policies in a variety of domains. The model *applies* POMDP planning to agents; in this paper we do not investigate how intentions can contribute to efficiently solving POMDPs, but regard such an investigation as important further work.

In the work presented, we show that the environmental properties of dynamism, observability and determinism are crucial for an agent's rate of intention reconsideration. Our formalism takes all mentioned environmental properties into account, and they form the basis of the decision mechanism of the BDI agent. A distinctive component in the BDI agent decides whether to reconsider or not, and we use the POMDP framework to determine an optimal reconsideration strategy that is used for implementing this component. We leave open the question whether a similar result can be achieved by the construction of complex sequential and conditional plans, since this defies the very nature of the BDI concept. A BDI agent is concerned with the management of simple plans over time, thus its intelligence is located in its meta-reasoning capabilities and not in its planning capabilities.

We have shown that an agent which is designed according to our formalism, is able to dynamically change its commitment to plans at run time, based on the current state of the environment. (In the experiments that are described in this paper, we assumed the environment to be fully observable and completely accessible, in order to compare our results with previous results.) This agent achieves better performance than existing planning frameworks, in which the level of plan commitment is imposed upon the agent at design time. The BDI-POMDP model has the advantage over the deliberation scheduling model (as used in [9]) that it computes a substantial part of the reconsideration strategy at design time, whereas all computations for deliberation scheduling are at run time. In contrast, the deliberation scheduling model is supposedly more flexible in changing the reconsideration strategy at run time.

References

1. C. Boutilier, T. Dean, and S. Hanks. Decision-theoretic planning: Structural assumptions and computational leverage. *Journal of AI Research*, pages 1–94, 1999.
2. M. E. Bratman, D. J. Israel, and M. E. Pollack. Plans and resource-bounded practical reasoning. *Computational Intelligence*, 4:349–355, 1988.
3. D. Kinny and M. Georgeff. Commitment and effectiveness of situated agents. In *Proceedings of the Twelfth International Joint Conference on Artificial Intelligence (IJCAI-91)*, pages 82–88, Sydney, Australia, 1991.
4. J. Lang, L. van der Torre, and E. Weydert. Utilitarian desires. *Journal of Autonomous Agents and Multi-Agent Systems*, 2001. To appear.
5. M. E. Pollack and M. Ringuette. Introducing the Tileworld: Experimentally evaluating agent architectures. In *Proceedings of the Eighth National Conference on Artificial Intelligence (AAAI-90)*, pages 183–189, Boston, MA, 1990.
6. A. S. Rao and M. P. Georgeff. An abstract architecture for rational agents. In C. Rich, W. Swartout, and B. Nebel, editors, *Proceedings of Knowledge Representation and Reasoning (KR&R-92)*, pages 439–449, 1992.
7. S. Russell and E. Wefald. Principles of metareasoning. *Artificial Intelligence*, 49(1-3):361–395, 1991.
8. M. C. Schut and M. Wooldridge. Intention reconsideration in complex environments. In M. Gini and J. Rosenschein, editors, *Proceedings of the Fourth International Conference on Autonomous Agents (Agents 2000)*, pages 209–216, Barcelona, Spain, 2000.
9. M. C. Schut and M. Wooldridge. Principles of intention reconsideration. In E. Andre and S. Sen, editors, *Proceedings of the Fifth International Conference on Autonomous Agents (Agents 2001)*, Montreal, Canada, 2001.
10. M. Wooldridge and S. D. Parsons. Intention reconsideration reconsidered. In J. P. Müller, M. P. Singh, and A. S. Rao, editors, *Intelligent Agents V (LNAI Volume 1555)*, pages 63–80. Springer-Verlag: Berlin, Germany, 1999.