Intention reconsideration in theory and practice

Simon Parsons[1] and **Ola Pettersson**[2] and **Alessandro Saffiotti**[2] and **Michael Wooldridge**[1]

Abstract. Autonomous agents operating in complex dynamic environments need the ability to integrate robust plan execution with higher level reasoning. This paper describes work to combine low level navigation techniques drawn from mobile robotics with deliberation techniques drawn from intelligent agents. In particular, we discuss the combination of a navigation system based on fuzzy logic with a deliberator based on the belief/desire/intention (BDI) model. We discuss some of the subtleties involved in this integration, and illustrate it with an example.

1 INTRODUCTION

Milou the robot works in a food factory. He has to regularly go and fetch two food samples (potato crisps) from two production lines in two different rooms, A and B, and take them to an electronic tester in the quality control lab. Milou must now plan his next delivery. He decides to get the sample from A first, since room A is closer than B. While going there, however, he finds the main door to that room closed. Milou knows that there is another door that he could use, but he considers the desirability of doing so. The alternative route to A is hard for Milou, since it goes through a long narrow corridor which is usually cluttered with boxes. Besides, doors usually do not stay closed for long. Hence, Milou decides to go to B first, and come back to A later on. He goes to room B, picks up the potato crisps and returns. The door to A is still closed, and this time Milou has no other choice than taking the difficult route. He does so, obtains the desired crisps, and finally goes to the lab and completes his task.

Performing the above task requires the ability to navigate robustly in real-world, unsimplified environments. Milou must be able to reliably find his way, keep track of his own position, avoid any obstacles in the cluttered corridor, and so on. However, this task also requires some higher level capabilities, like reasoning about alternative ways to perform a given task, and reconsidering available options in the face of new events. The development of intelligent mobile robots and their deployment in real-world environments will critically depend on our ability to integrate these two aspects of the autonomous navigation problem.

Today's research on mobile robotics has produced a large number of techniques for robust navigation in real environments in the presence of uncertainty, for example [1, 5, 11]. These techniques typically focus on the navigation problem, and do not involve abstract reasoning processes of the type encountered in the above scenario. On the other hand, research in intelligent agency has resulted in a number of powerful theories for reasoning about actions and plans. Indeed, much of the research activity from the intelligent agent community in the mid-to-late 1980s was focussed around the problem of designing agents that could achieve an effective balance between *deliberation* (the process of deciding what to do) and *means-ends reasoning* (the process of deciding how to do it) [3].

One particularly successful approach that emerged at this time was the belief-desire-intention (BDI) paradigm [3, 7, 13]. The development of the BDI paradigm was to a great extent driven by Bratman's theory of (human) practical reasoning [2], in which *intentions* play a central role. Put crudely, since an agent cannot deliberate indefinitely about what courses of action to pursue, the idea is it should eventually commit to achieving certain states of affairs, and then devote resources to achieving them. These chosen states of affairs are intentions, and once adopted, they play a central role in future practical reasoning [2, 4].

A major issue in the design of agents that are based upon models of intention is that of when to *reconsider* intentions. An agent cannot simply maintain an intention, once adopted, without ever stopping to reconsider it. From time-to-time, it will be necessary to check, (for example), whether the intention has been achieved, or whether it is believed to be no longer achievable [4]. In such situations, it is necessary for an agent to deliberate over its intentions, and, if necessary, to change focus by dropping existing intentions and adopting new ones.

Clearly, an agent's intention reconsideration policy will affect its performance, and the optimal policy for a given agent will be heavily dependent upon its environment. There has been a certain amount of work on this problem in the area of intelligent agents, from both a formal [17] and an experimental [9, 16] perspective. However, most of this work has concentrated on agents in environments which are rather simple when compared to the environment Milou operates in. Indeed, to our knowledge, there has been no work which attempts to investigate intention reconsideration in environments which are both complex and dynamic.

Our research aims to address this deficit, identifying suitable mechanisms and strategies for intention reconsideration which work well when combined with the kind of low-level control mechanisms required by agents which operate in complex dynamic environments. This paper describes one approach which combines a robust navigation system based on fuzzy logic [14, 15] and a BDI system for handling intentions. Before presenting the combination, however, we discuss the problem of intention reconsideration with respect to the formal model developed in [17].

2 THE FORMAL MODEL

Following [17], our agents have two main data structures: a *belief set* and an *intention set*. An agent's beliefs represent information that the agent has about its environment. Let B be the set of all beliefs. For the most part, the contents of B will not be of concern to us here. How-

[1] Department of Computer Science, University of Liverpool, Chadwick Building, Peach Street, L69 7ZF, Liverpool, United Kingdom. http://www.csc.liv.ac.uk

[2] Applied Autonomous Sensor Systems, Department of Technology, Örebro University, S-70182 Örebro, Sweden. http://www.aass.oru.se

ever, it is often useful to suppose that B contains formulae of some logic, so that, for example, it is possible to determine whether two beliefs are mutually consistent or not. An agent's actions at any given moment are guided by its *intention set*, and its intentions may be thought of as states of affairs that the agent has committed to bringing about. These may be structured in some way—for instance in a hierarchy with high level intentions defined as a set of lower level intentions—and may be ordered. Formally, let I be the set of all intentions. Again, we are not concerned here with the contents of I. As with beliefs, however, it is often useful to assume that intentions are expressed in some sort of logical language. An agent's *local state* will then be a pair (b, i), where $b \subseteq B$ is a set of beliefs, and $i \subseteq I$ is a set of intentions. Let $L = \wp(B) \times \wp(I)$ be the set of all internal states of the agent. We use l (with annotations: $l', l_1, \ldots$) to stand for members of L. If $l = (b, i)$, then we denote the belief component of l by b_l, and the intention component by i_l. For the formal model we assume a fixed set of intentions which have been generated from some set of desires in the usual way [3].

Agents do not operate in isolation: they are situated in *environments*; we can think of an agent's environment as being everything external to the agent. We assume that the environment external to the agent may be in any of a set $E = \{e, e', \ldots\}$ of states. For now we assume that an agent knows what state the environment is in, acknowledging that, in future work, we will have to take account of the fact that any agent only has partial knowledge of the environment. Together, an agent and its environment make up a *system*. The *global* state of a system at any time is thus a pair containing the state of the agent and the state of the environment. Formally, let $G = E \times L$ be the set of all such global states. We use g (with annotations: $g, g', \ldots$) to stand for members of G.

Our agents have four main functional components, which together generate their behaviour: a *next-state function*, a *meta-level control function*, a *deliberation function*, and an *action function*. The *next state* function can be thought of as a *belief revision function*. On the basis of the agent's current state and the state of the environment, it determines a new set of beliefs for the agent, which will include any new information that the agent has perceived. An agent's next-state function thus realises whatever *perception* the agent is capable of. Formally, a next-state function is a mapping $\mathcal{N} : E \times \wp(B) \rightarrow \wp(B)$.

The next component in our agent architecture is meta-level control. The idea here is that at any given instant, an agent has two choices available to it. It can either *deliberate* (that is, it can expend computational resources deciding whether to change its focus), or else it can *act* (that is, it can expend resources attempting to actually achieve its current intentions). Note that we assume the only way an agent can modify its intentions is through explicit deliberation. To represent the choices available to an agent, we will assume a set $C = \{d, a\}$, where d denotes deliberation, and a denotes action. The purpose of an agent's *meta-level control function* it to choose between deliberation and action. If it chooses to deliberate, then the agent subsequently deliberates; if its chooses to act, then the agent subsequently acts. Formally, we can represent such strategies as functions $\mathcal{M} : L \rightarrow C$.

The *deliberation* process of an agent is represented by a function that, on the basis of an agent's internal state, determines a new set of intentions. Formally, we can represent this deliberative process via a function $\mathcal{D} : L \rightarrow \wp(I)$. If an agent decides to act, rather than deliberate, then it is acting to achieve its intentions. To do so, it must decide *which* action to perform. The action selection component of an agent is essentially a function that, on the basis of the agent's current state, returns an action, which represents that which the agent has chosen to perform. Let $Ac = \{\alpha, \alpha', \ldots\}$ be the set of actions. Formally, an action selection function is a mapping $\mathcal{A} : L \rightarrow Ac$.

Finally, we define an agent to be a 5-tuple $(\mathcal{M}, \mathcal{D}, \mathcal{A}, \mathcal{N}, l_0)$, where $\mathcal{M}$ is a meta-level control function, $\mathcal{D}$ is a deliberation function, $\mathcal{A}$ is an action selection function, $\mathcal{N}$ is a next-state function, and $l_0 \in L$ is an *initial state*.

3 MILOU IN THEORY

Our intention in introducing this formal model is to shed light upon the problems one faces when attempting to integrate a high-level agent architecture like the BDI model with the concrete requirements of a mobile robot. Consider Milou once again. In the abstract terms used by the BDI model we can consider Milou to have a set of possible intentions:

i_c	test crisps	i_a	take the short route to A
i_f	fetch crisps	$i_{a'}$	take the long route to A
i_t	taste crisps	i_b	go to B
i_l	go to the lab		

These intentions are hierachically structured and ordered, with i_c being composed of i_f followed by i_t, and i_f being composed of i_a or $i_{a'}$ along with i_b and followed by i_l. The lowest level of these intentions have corresponding actions denoted by $\alpha_a, \alpha_{a'}, \alpha_b, \alpha_l$. Milou also has a set of possible beliefs:

b_a	short route to A is viable	b_b	door to B is open
$b_{a'}$	long route to A is viable	b_f	go to A first

Milou starts with the initial state:

$$l_0 = (\{b_a, b_{a'}, b_b, b_f\}, \{i_c\})$$

so he has the intention to carry out his usual task of testing crisps, and believes all is well with the world. Having no possible action, Milou's meta-level control function $\mathcal{M}$ indicates he should deliberate, and he generates a new set of intentions $\{i_c, i_f, i_a\}$—to achieve the intention of testing the crisps he must first fetch them and the first step in this fetching is to go to A by the short route. At this point $\mathcal{M}$ decides to act, calls the action selection function $\mathcal{A}$, and $\mathcal{A}$ selects action α_a. As a result, Milou starts to go to A. Midway through this action, Milou realises this action has failed because the door to A is closed—that is, he revises his beliefs to get $\{\neg b_a, b_{a'}, b_b, \neg b_f\}$, and $\mathcal{M}$ then decides to deliberate. This deliberation generates a new set of intentions $\{i_c, i_f, i_b\}$. $\mathcal{M}$ then chooses to act, $\mathcal{A}$ selects α_b, and Milou starts to execute α_b. When this action is complete, there is, once again, no action to execute, and the meta-level controller once more decides to deliberate.

The reason for stepping through the example like this is to highlight three particular issues that need to be solved in order to use BDI systems, which work at precisely this kind of level of detail, with mobile robots. First, there is the issue of moving from intentions to actions. Although our description is a little abstract, assuming that there is a single action to achieve each intention, it is close to the reality of implemented BDI systems. For instance, PRS [6] works out how to achieve intentions by pulling pre-compiled plans from a plan library. Mobile robots will require rather more sophisticated planners, in particular planners which can plan robustly under the considerable uncertainty that real world mobile robots are subject to. Second, there is the whole issue of when to deliberate as against when to act. Experimental work on the problem [9, 16] has concentrated on the relationship between the speed of change of an environment and the frequency of redeliberation. Our situation is more subtle—because considerable effort can be expended in trying to achieve an intention that is no longer achieveable (like trying to pass through a closed door, outside which Milou will circle forever), it is necessary to be able to detect the failure of a plan *during* execution. Third, there is the need to handle uncertainty in Milou's view of the world. While the formal model assumes boolean beliefs—either Milou believes the door to A is open or he believes it is closed—the reality is more complex. All Milou will have is a degree of belief, based on sensor input, that the door is open or closed. As discussed elsewhere, for example [1, 5, 11], handling this uncertainty requires sophisticated models. To solve these problems we turned to the use of Saffiotti's 'Thinking Cap' [14, 15].

4 FROM THEORY TO PRACTICE

The 'Thinking Cap' (TC)[3] is a system for autonomous robot navigation based on fuzzy logic which has been implemented and validated on several mobile platforms [14, 15]. The main ingredients of the TC are:

- a library of *fuzzy behaviours* for indoor navigation, like obstacle avoidance, wall following, and door crossing;
- a *context-dependent blending* mechanism that combines the recommendations from different behaviours into a tradeoff control;
- a set of *perceptual routines*, including sonar-based feature extraction, and detection of closed doors and blocked corridors;
- an approximate *map* of the environment, together with a positioning mechanism based on natural landmarks;
- a *navigation planner* that generates a behaviour combination strategy, called a B-plan, that achieves the given navigation goal; and
- a *monitor* that reinvokes the planner whenever the current B-plan is no longer adequate for achieving the current goal.

For the purposes of this paper, we regard the TC as a black box that provides a robust navigation service, and that accepts goals of the form '(goto X)'. There are however two characteristics of TC that are important here.

First, navigation goals in TC are fuzzy: in '(goto X)', 'X' is a fuzzy location in the robot's map. (More precisely, a goal is formally defined in the TC framework as a fuzzy set of trajectories.) This means that a goal in TC can be more or less satisfied, as measured by a *degree of satisfaction*, a real number in the interval $[0, 1]$. Typically, this degree depends on the distance between the robot and the desired location, but more complex goals may have more complex degrees of satisfaction.

Second, the 'adequacy' of the current B-plan which is monitored by the TC is in fact a *degree of adequacy*, again measured by a number in $[0, 1]$. This degree of adequacy is the result of the composition of three terms:

1. a degree of 'goodness', that takes into account the prior information available about the environment; for example, a B-plan that includes passing through a long and narrow corridor has a small degree of goodness;
2. a degree of 'competence', that dynamically considers the truth of the preconditions of the B-plan in the current situation; for example, if a door that has to be crossed is found closed this degree drops to 0; and
3. a degree of 'conflict', that measures the conflict between the behaviours which are currently executing in parallel.

[3] `http://www.aass.oru.se/~asaffio/Software/TC/`

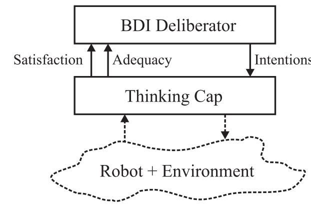

Figure 1. Integration between a BDI deliberator and the Thinking Cap.

Now the BDI model and the Thinking Cap represent two ends of the spectrum as far as the mental abilities of an autonomous robot are concerned. The TC can construct plans to achieve a single high level intention (like 'go to the lab'), but has no grasp of the sequence of high level intentions necessary to carry out the robot's overall goals. In contrast, the BDI model (at least in so far as we have analysed it with respect to intention reconsideration) is only concerned with high level intentions and whether or not they should be reconsidered as its beliefs about the world themselves change. These may be combined as shown in Figure 1.

The BDI deliberator provides the deliberation function $\mathcal{D}$ in the formal model, generating high-level intentions of the type (goto X) and sending them to the TC. (In future versions, intentions may include manipulation or observation activities.) The TC implements the action selection function $\mathcal{A}$, receiving these intentions and considering them as goals. For each goal, it generates a B-plan—each corresponds to an action in the formal model—and starts execution. The two components run as concurrent processes, with control cycles of 2s and 100ms respectively.

The TC also monitors this execution, and switches to a new B-plan if the current one turns out to be inadequate. During execution, the TC recomputes the current degrees of satisfaction and adequacy every control cycle. These degrees are sent back to the BDI deliberator. From the point of view of the deliberator, the degree of satisfaction measures how much the current intention has been achieved, and the degree of adequacy measures how much this intention is considered achievable. This information is thus part of the input to $\mathcal{D}$. In contrast to the standard BDI model, however, this information is not given by binary values, but by continuous measures made possible by the use of fuzzy set theory in the TC. It is these indicators of the state of the world *vis à vis* the current intention which help the deliberator to determine when it is appropriate to reconsider its intentions.

Considering the formal model described above, we should note that, at the moment, the belief set B is partitioned between the BDI interpreter and the TC. In particular, beliefs that are affected by the dynamic nature of the world—in this case b_a, $b_{a'}$ and b_b—are stored in the TC and updated as a result of sensor readings. These beliefs are used to determine the degrees of satisfaction and adequacy. The more static knowledge is kept in the BDI deliberator and updated according to the degrees of satisfaction and adequacy. It is these measures which, in practice, cause Milou to change from believing b_f to $\neg b_f$ when he finds that $\neg b_a$ is true. These measures, therefore, help to relate the sensor-derived beliefs b_a, $b_{a'}$ and b_b which are stored in the TC to the intention determining belief b_f which is stored in the BDI system.

The deliberator also uses these values in two other important ways. First, to decide *when* it is time to deliberate. Two of the possible causes that lead the deliberator to reconsider its intentions are: (i) an increase in the value of satisfaction; and (ii) a drop in the value of adequacy. Second, it uses the values in the deliberation itself as a

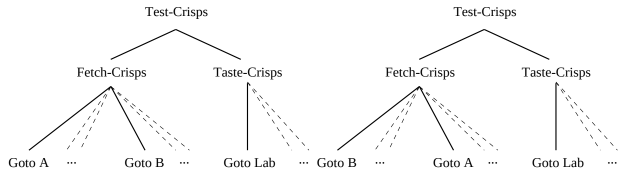

Figure 2. Two intention trees for our example task.

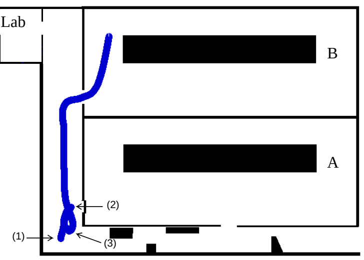

Figure 3. Milou has the intention (goto A), but this turns out to be difficult to achieve, and adopts the new intention (goto B).

means of comparing the available options. If this deliberation results in a new intention being adopted, it is passed to the TC. As we shall see below, considering degrees instead of binary values allows the deliberator to make more informed decisions.

5 EXPERIMENTAL RESULTS

By way of validation of our approach, we report an experiment where we execute the potato crisp scenario in a simulated environment. We have used the Nomadic simulator, which includes simulation of the sonar sensors and some moderate sensor and positioning noise. This experiment is meant to illustrate the concepts and mechanisms involved in our integrated approach to robot deliberation and navigation in a reasonably realistic environment (although it cannot take the place of real experiments on a live robot)[4]. The successive phases of the simulated run are shown in Figures 3, 4, and 5. Figure 6 shows the values of adequacy and of satisfaction of the currently executing intention at each moment of the run.

Initially, the BDI deliberator considers the new task and decides a strategy, represented by the intention tree shown in Figure 2 (left). The details of how this is done are not relevant here (the dots indicate other intentions, like picking up the crisps, which we ignore); it suffices to note that the intentions have a temporal order, which is that of a left to right depth-first traversal of the tree. The deliberator then passes the first intention (goto A) to TC, which generates a suitable B-plan for it. In this case there are two possible B-plans, one for each possible door leading to A, and the TC selects the one with the highest degree of (expected) goodness. Since the TC knows about the low degree of traversability of the lower corridor,[5] the selected B-plan is the one that goes through the main door of A, the one on its left wall. Milou starts executing this B-plan from the lower left corner, as indicated by (1) in Figure 3.

When Milou arrives at this door (2), the sonars detect that the door is closed. Since one of the assumptions in the B-plan is that the door

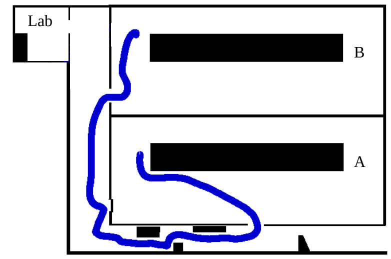

Figure 4. Milou has the intention (goto A) once more.

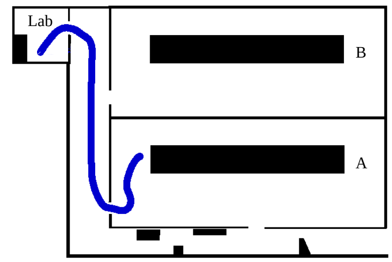

Figure 5. Both previous intentions are fulfilled, and Milou adopts the intention (goto Lab).

must be open, the degree of adequacy of this plan drops to 0 (Figure 6 at about 20 s). The TC notices the problem, generates a new B-plan that goes through the second door, and starts executing it. However, this B-plan has a low degree of goodness because it includes passing through the cluttered corridor. This causes a drop of the adequacy level to a low value of 0.2. The BDI deliberator notices this and reconsiders its options. Since the current intention turns out to be difficult (but not impossible) to achieve, and there is an alternative way to perform the task (Figure 2 right), the deliberator decides to switch to this alternative and to reverse the order of visiting the two production lines. Hence, it sends the new intention (goto B) to the TC (Figure 6 at 30 s). The TC generates a new B-plan for this intention and swaps it in. Poor Milou then stops his journey to the lower corridor (point (3) in Figure 3), turns around, heads to room B, and eventually reaches the collection point in front of conveyer belt B.

The achievement of the intention (goto B) is reflected in the rise of the satisfaction level (Figure 6 at 75 s). This is noticed by the BDI deliberator, which then sends the next intention to the TC: in our case, this is again the intention (goto A). Since the information about closed doors inside the TC is transient, the TC again generates a B-plan for this intention which involves going through the main door. Milou finds his way from room B, but unfortunately he finds that the door is still closed (Figure 4).

As before, the TC generates an alternative B-plan going through the lower corridor and starts to execute it. This produces a drastic drop in the adequacy level, which is noticed by the BDI deliberator (Figure 6 at 160 s). However, this time there is no alternative option, so the deliberator decides to keep with the current intention, even though it is difficult to achieve. The navigation functionalities of the TC allow Milou to safely, if slowly, get around the obstacles, and reach the collection point in front of conveyer belt A.

The first two intentions are now fulfilled, and the BDI deliberator sends the last one (goto Lab) to the TC. Again, the TC tries the main door first. This time we are lucky, since someone has actually opened this door, and Milou eventually finds his way to the lab, thus

[4] We are currently in the process of implementing our integrated system on a Nomad 200.

[5] Currently, this information is stored in the map; in the future, the robot may acquire this knowledge during exploration.

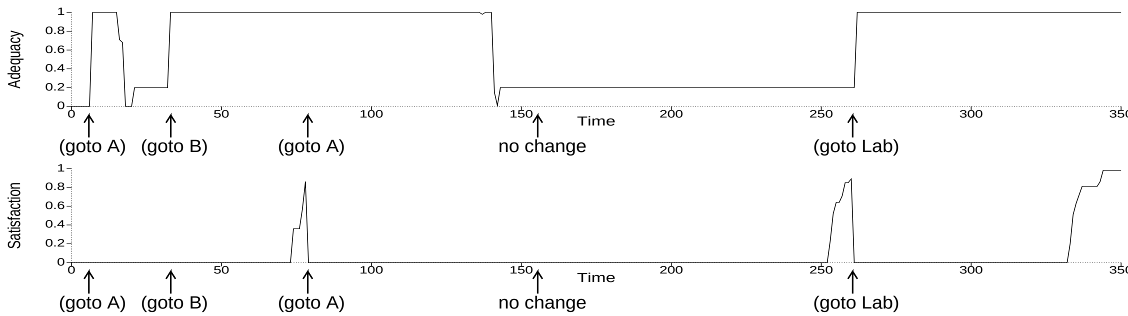

Figure 6. Measures of adequacy (top) and satisfaction (bottom) sent by the TC to the deliberator during the run. The arrows indicate the deliberation points, and the new intentions generated.

completing the mission (Figure 5).

6 DISCUSSION

This paper extends two lines of work; work on intention reconsideration, and work on integrating low-level navigation and higher-level reasoning. It extends existing work on intention reconsideration [9, 16, 17], by considering more complex environments, identifying a whole range of issues which have not been so apparent before. These include the need to build and execute robust plans, the ability to detect the partial failure of those plans, the ability to measure the impossibility of achieving current intentions in the presence of uncertain information, and the ability to use uncertain beliefs about the environment in the deliberation process. Having identified these issues, we have proposed a solution based on the integration of a traditional BDI system with the Thinking Cap software.

There are already a number of proposals which use a BDI approach to integrate low-level navigation and higher-level reasoning. For example, in [8, 10, 12] PRS-like systems are used to arbitrate low-level processes. Our proposal departs from these approaches in the way we partition the responsibilities between the Thinking Cap and the BDI deliberation system. We rely on the underlying navigation abilities of the TC to take care of fuzzy behaviour arbitration and blending in a sophisticated way. And we limit the role of the deliberation system to take care of higher level decisions about which overall navigation goal should be pursued next. This partition allows us to make better use of the respective powers of the TC and of the BDI level. In particular, by passing performance measures from the lower to the upper level we allow the latter to take more abstract, yet still fully informed, decisions.

There are two important ways in which our approach can be developed. First, the information passed by the TC to the BDI level could be much richer, including, for example, the reasons why a B-plan has (partially) failed, the conditions that would increase its level of adequacy, or indications about the existence of alternative B-plans and their degrees of adequacy. This would help the BDI system in its deliberation (for instance in determining whether to drop an intention or try to achieve it later). Second, the choice of the strategy used to decide when the BDI should deliberate and when it should let the TC do its job depends on the characteristics of the environment, and it may itself be the result of another, higher level deliberation. Including this idea in our framework would lead to a ‘tower of meta-controllers’ similar to the one suggested in [17]. Such an approach would allow the robot to dynamically adjust its policy for redeliberation if it finds that the policy is incorrect with respect to its current environment. We are currently working on both these developments.

Acknowledgements: This work benefitted from discussions with Dimiter Driankov and was partially supported by Queen Mary and Westfield College, and the Swedish KK Foundation.

REFERENCES

[1] R. C. Arkin, *Behavior-Based Robotics*, MIT Press, Cambridge, MA, 1998.

[2] M. E. Bratman, *Intentions, Plans, and Practical Reason*, Harvard University Press: Cambridge, MA, 1987.

[3] M. E. Bratman, D. J. Israel, and M. E. Pollack, ‘Plans and resource-bounded practical reasoning’, *Computational Intelligence*, **4**, 349–355, (1988).

[4] P. R. Cohen and H. J. Levesque, ‘Intention is choice with commitment’, *Artificial Intelligence*, **42**, 213–261, (1990).

[5] D. Driankov and A. Saffiotti eds., *Fuzzy logic techniques for autonomous vehicle navigation*, Springer, Berlin, Germany, 2000.

[6] M. P. Georgeff and F. F. Ingrand, ‘Decision-making in an embedded reasoning system’, in *Proceedings of the International Joint Conference on Artificial Intelligence*, pp. 972–978, Detroit, Michigan, (1989).

[7] M. P. Georgeff and A. L. Lansky, ‘Reactive reasoning and planning’, in *Proceedings of the Sixth National Conference on Artificial Intelligence*, pp. 677–682, Seattle, Washington, (1987).

[8] F. F. Ingrand, R. Chatila, R. Alami, and F. Robert, ‘PRS: a high level supervision and control language for autonomous mobile robots’, in *Proceedings of the International Conference on Robotics and Automation*, Minneapolis, MN, (1996).

[9] D. Kinny and M. P. Georgeff, ‘Commitment and effectiveness of situated agents’, in *Proceedings of the International Joint Conference on Artificial Intelligence*, pp. 82–88, Sydney, Australia, (1991).

[10] K. Konolige, K. L. Myers, E. H. Ruspini, and A. Saffiotti, ‘The Saphira architecture: A design for autonomy’, *Journal of Experimental and Theoretical Artificial Intelligence*, **9**, 215–235, (1997).

[11] D. Kortenkamp, P. Bonasso, and R. Murphy eds., *Artificial Intelligence and Mobile Robots*, MIT Press, Cambridge, MA, 1998.

[12] J. Lee, M. J. Huber, E. H. Durfee, and P. G. Kenny, ‘UM-PRS: an implementation of the procedural reasoning system for multirobot applications’, in *Proceedings of the* AIAA/NASA Conf. on Robots in Field, Factory, Service and Space, (1994).

[13] A. S. Rao and M. Georgeff, ‘Decision procedures for BDI logics’, *Journal of Logic and Computation*, **8**(3), 293–344, (1998).

[14] A. Saffiotti, K. Konolige, and E. H. Ruspini, ‘A multivalued-logic approach to integrating planning and control’, *Artificial Intelligence*, **76**, 481–526, (1995).

[15] A. Saffiotti, E. H. Ruspini, and K. Konolige, ‘Blending reactivity and goal-directedness in a fuzzy controller’, in *Proceedings of the 2nd IEEE International Conference on Fuzzy Systems*, pp. 134–139, San Francisco, CA, (1993).

[16] M. Schut and M. Wooldridge, ‘Intention reconsideration in complex environments’, in *Proceedings of International Conference on Autonomous Agents*, (2000 (to appear)).

[17] M. Wooldridge and S. Parsons, ‘Intention reconsideration reconsidered’, in *Intelligent Agents V*, eds., J. P. Müller, M. P. Singh, and A. Rao, Springer-Verlag, Berlin, Germany, (1999).