Planning At The Intention Level

Nick Lacey, Henry Hexmoor, Gordon Beavers
University of Arkansas
313 Engineering Hall
Fayetteville, Arkansas
{nlacey,hexmoor,gordonb}@uark.edu

Abstract

Intentions enable an agent to achieve long-term goals. When an agent wishes to achieve an intention, it uses actions to bring it about. In this paper we investigate the relationship between intentions and actions. We begin by describing previous work involving a responsive agent capable of coherence-based belief revision. We then describe the implementation and testing of a system capable of representing and manipulating high level intentions. The implications of planning at the intention level are then discussed. We conclude that the architecture we put forward facilitates the development of agents which are able to plan pro-actively while still being able to cope with variation in the environment.

Introduction

Part of the power of the BDI *(belief-desire-intention)* approach to intelligent agent design stems from the concept of *Intentions*. Intentions occupy a middle ground between desires and actions which allow the agent to focus on a manageable number of achievable short or long-term goals which will maximise, to the best of the agent's knowledge and abilities, its long-term utility. In other words, intentions allow an agent to be goal-driven, rather than event-driven (Schut & Wooldridge 2001).

However, intentions are not actions. Intentions are what an agent uses to commit to achieving a particular goal. Actions are the means by which an agent brings about its intentions. Depending on the nature of the agent's environment, the agent might have to reconsider its intentions, its actions, or both.

A significant difficulty arises when determining the nature by which intention reconsideration should occur. As Wooldridge points out (Wooldridge 2000), this difficulty presents itself in the form of a dilemma:

- An agent which does not reconsider its intentions with insufficient frequency runs the risk of carrying out actions which have been rendered obsolete by changes in the environment.
- An agent which reconsiders its intentions too frequently runs the risk of spending so much time deliberating about what to do that it is unable to achieve some of its desires.

We suggest an architecture based on *coherence*. The architecture is based on the philosophical concept of coherence as an approach to justification (Audi 1993), and as such is related to the coherence approach to belief revision (Gärdenfors 1990). The architecture allows an agent to plan sequences of actions at the *intention level*, and not concern itself with the complexities of bringing about the intention during the planning stage. When the time does come to consider the actions necessary to bring about an intention, the architecture allows the agent to deal with a certain amount of variation without reconsidering its long term intentions.

Structuring a Knowledge Base Using High-Level Beliefs

The initial design and implementation of this system, described in (Lacey 2000) and (Lacey & Lee 2001), involved a coherence-based architecture in which the agent's low-level perceptions were organised and given meaning using high-level explanations of the agent's environment. The most coherent explanation at any given time was taken to be the correct explanation of the agent's current sensor data.

This architecture was responsive, rather than pro-active. The agent was able to perceive and react to changes in its environment, but was not able to represent or act on long-term plans or goals. Jennings, Sycara and Wooldridge (Jennings, Sycara, & Wooldridge 1998) define a responsive system as one which is able to perceive its environment and respond to changes which occur in it.

As such, responsiveness can be contrasted with pro-activity. A pro-active agent does not merely respond to its environment, but is able to exhibit goal-directed behaviour. As shown in Figure 1, adding intentions to this architecture renders the agent pro-active, as well as responsive.

Consistency is maintained using a system of *constraints*. Constraints are used to alert the agent to the presence of an inconsistency. Their role can be seen as three-fold:

1. Constraints can be viewed as a shortcut, as they allow the agent to detect an inconsistency as early as possible in the knowledge base derivation process.
2. Constraints allow the high-level representation of states of affairs that cannot obtain in the agent's environment. If a particular constraint is violated under the current explana-

---

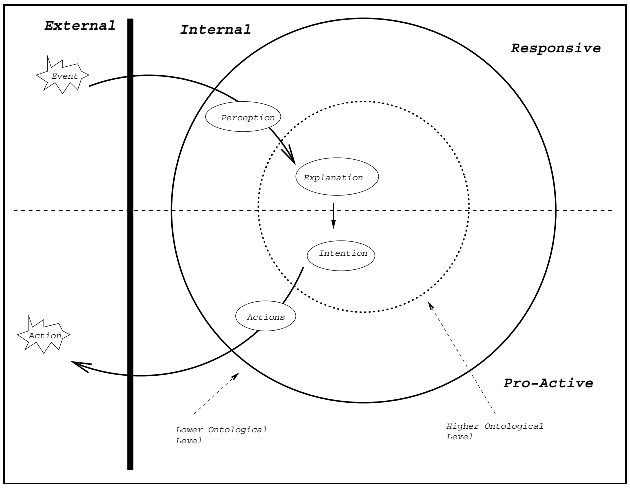

Figure 1: Adding Intentions to the System Yields Pro-Activity

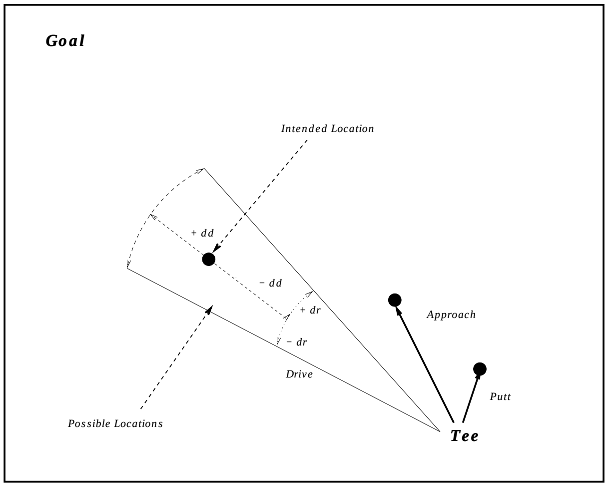

Figure 2: The Different Intentions Available to the Agent

tion, the agent can immediately infer that the explanation is flawed, and thus can invest its energies elsewhere.

3. Due to the high-level nature of constraints, it is possible to associate a particular recovery plan with every constraint. By following the recovery plan, the agent may be able to produce a new explanation which successfully resolves the current problem.

For example, suppose I am sitting at my desk on the third floor of the building, when I hear a knock on the window. The normal explanation which accompanies the sensory input of a knock at the window is that someone is standing on the ground on the other side of the window. However, this explanation would violate several constraints, including my belief that people are not 30ft tall. In order to address the inconsistency, I generate a new set of explanations, the most likely of which is that someone is standing on the ledge outside the window.

Thus, high-level explanations provide methods of combining this low-level data in different ways, depending on which explanation is chosen. If the agent is able to, it will use the default explanation. If this is not possible, alternate explanations are generated and holistically compared.

Implementation and Experiments

In order to test the architecture described in this paper, a system capable of constructing, executing, and manipulating intention trees was implemented in Prolog.

Experimental Domain

The experimental domain that was used was chosen to be as simple as possible while nonetheless requiring that intention trees be constructed and modified as appropriate. The domain that was chosen to be the basis of these experiments was that of a crude discgolf simulation. Discgolf is similar to golf, but players throw plastic discs instead of hitting golf balls. The environment consists of a rectangular area. At the beginning of the simulation, the disc is placed on a tee. The agent must formulate an intention tree which will permit it to throw the disc into the basket using as few throws as possible.

An advantage of this domain is that while the distinction between intentions and actions is clear at the conceptual level, at the implementational level translation from the intention level to the action level is very simple. Intentions concern an attempt to throw a disc to particular location, specified by (x, y) coordinates. The action required to bring about the intention concerns throwing the disc in a particular direction and aiming to cover a particular distance. Deriving the distance and direction values, given the target (x, y) values and the disc's current position, is a matter of simple geometry.

The agent has three types of intention available to it, as summarised in Table 1. The intentions represent different types of throw. Throws which will move the disc further are less accurate, while the more accurate throws do not cover as much distance. This effect was achieved by adding a certain amount of error to the agent's intended throw. This is illustrated in Figure 2. The representation of the possible actual locations of the disc after a throw are based on Shanahan's concept of the circle of uncertainty (Shanahan 1996).

The amount of error added to each throw is controlled by two values.

- dr represents the number of degrees by which the actual direction of the throw may vary from the intended direction.
- dd represents the number of units by which the distance of an actual throw may vary from its intended distance. This figure represents a percentage of the actual intended distance. For example, if the intended distance of a throw is 300 units, and $dd = 5\%$, then the actual distance of the throw will be accurate to within ± 60 units.

The dr and dd values used for the experiments described here are given in Table 1. As these experiments represent a proof of concept rather than an attempt at a realistic simulation, dr and dd were set to the same value in each experiment.

Intention	Max Distance	dr and dd for Experiment		
		1	2	3
Putt	20	0	0.5	1
Approach	200	0	1	2
Drive	300	0	1	5

Table 1: The Intentions Available to the Agent in the Discgolf Simulation

Once the agent has constructed the intention tree, the agent begins to formulate actions which will bring about the intention. It does this using the action template associated with each intention. In this domain, all intentions were associated with the `throw` action. This implements the requirement that intentions should constitute meaningful segments of the agent's plan in their own right, while individual actions need not be meaningful when considered in isolation from their associated intentions.

Thus, intentions take the form:

```
[drive,[x,y]]
```

where x and y are the coordinates of the disc's intended location *after* the intention has been executed. For leaf intentions, this value will be the coordinates of the target, while for intermediary intentions, the intended disc position will be a function of the maximum distance of the intended throw and the disc's position prior to the intention.

Actions, on the other hand, have the following format:

```
[throw,D,B]
```

where D is the distance of throw required and B is the bearing at which the throw should be aimed in order to reach the position specified by the intention. Thus, the distinction in semantic level between intentions and actions is clear: intentions concern x, y coordinates in the environment, while actions concern the strength and direction of a throw.

The constraints associated with the intentions summarised in Table 1 are used to ensure that inaccuracies in throws do not necessarily lead to a change in intention. However, if the error placed on a throw is large enough, intention reconsideration may become necessary. This is achieved by calculating the distance from the disc's current position to the sub-goal specified by the intention. If this distance is greater than the maximum range of the current intention, then the current intention, and all the intentions which follow it, must be reconsidered.

The constraints associated with actions are used to allow the agent to cope with minor variations between the intended and actual course of events which do not require intention reconsideration. If the actual position of the disc prior to the current intention is different from the intended position, but still within the range of the intended throw, then a new action is created. This action will re-use the throw type specified by the intention, but will calculate distance and direction values based on the disc's actual position.

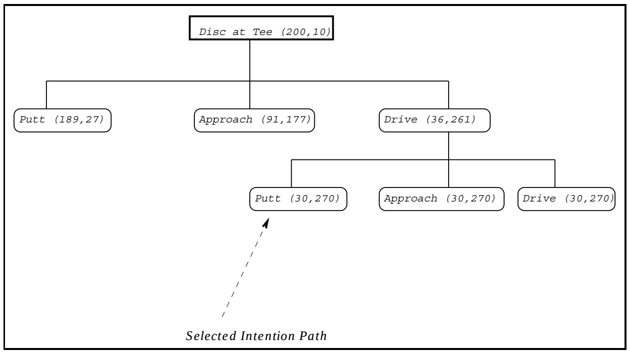

Figure 3: The Intention Tree Constructed In Experiments 1 and 2

Results

The results presented in this section have been rounded to 0 decimal places in the case of (x, y) coordinates, and 1 decimal place in the case of directions.

Experiment 1 did not involve any errors. As such the throws associated with each intention landed with 100% accuracy, meaning there was no need for intention or action reconsideration. The intention tree constructed during Experiment 1 is shown in Figure 3.

With dr and dd set to 0, throws at the action level represented these intentions with 100% accuracy. The intentions and actions used in the experiments are shown in Table 2. Note that both throws used in Experiment 1 share the same direction, as the disc is moving along a perfectly straight line from the tee to the goal.

Exp.	Intention			Action - Throw	
	Type	x	y	Distance	Direction
1	Drive	36	261	300	56.8
	Putt	20	270	11	56.8
2	Drive	36	261	300	56.8
	Putt	20	270	10	43.7
3	Drive	36	261	300	56.8
	Approach	20	270	23	28.8

Table 2: Intentions and Corresponding Actions from Experiments 1,2, and 3

The purpose behind Experiment 2 was to investigate the ability of the architecture to formulate new actions without intention reconsideration. In order to do this a small amount of error was added to each throw, as summarised in Table 1.

Constraints associated with each action are used to ensure that the action originally represented by the intention is still valid. In this case, this was done by checking whether the disc was actually at the location it should be at when the action is undertaken.

The experiment was successful, in that the agent was able to move the disc from the tee to the goal without reconsidering its intentions, despite the fact that the disc never landed exactly where it was intended to. As would be expected, as the error added to the throws was produced randomly, results varied between different runs. The execution of the final throw, which will usually be a putt, was unsuccessful in some cases. In these cases, a new intention had to be created in order to accomplish the goal. In cases where the second throw was successful, the intention tree resulting from Experiment 2 was exactly the same as that resulting from Experiment 1, shown in Figure 3.

Results from a representative run of Experiment 2 are given in Table 2. The first intention and action are carried out as normal. After the first throw, the agent realises that the disc is not at the intended location, namely $(36, 261)$. However, the distance between the disc's actual location and the intended location of intention 2 is such that a putt is still an applicable intention. However, as the disc is not where it should be, the original intention must be brought about using a different action.

In the example shown, the putt required by intention 2 was successful. In cases where this putt was not successful, an additional putt intention was generated, as follows:

```
[putt,[20,270]]
```

The intention is unchanged, as the goal is still that of placing the disc in the target. This intention will be translated into an action, such as:

```
[throw,2,3.4]
```

This process is repeated until the throw is successful. The actual parameters of the throw will clearly vary depending on where the disc lands after each attempt. The agent usually required between 2 and 4 throws to reach the target.

This corresponds to the approach humans take when attempting to bring about a difficult intention. For example, when trying to unlock a door at night, my intention is to place the key in the lock. My first action is to move the key toward the lock, and to attempt to insert it. If my first attempt fails, my intention *remains* that of placing the key in the lock, while at the action level I am moving the key in various directions until I feel that the key has entered the lock.

The purpose of Experiment 3 was to add sufficient error to the throws to cause the agent to reconsider its intentions. Whether or not intention reconsideration was necessary was represented using the following constraint: If the distance between the disc's actual position and its intended position was greater than the maximum distance of the intended throw, then intention reconsideration was necessary.

An intention tree representing a sample run of Experiment 3 is shown in Figure 4. For the sake of simplicity, a run has been selected here in which the agent successfully completed the task in 2 throws. Most of the runs in Experiment 3 required between 3 and 5 throws.

After the first throw, the disc ends up at $(50, 259)$. This is too far for the intended putt, so intentions must be reconsidered. Using the disc's actual position as the starting point, the agent produces a new set of intentions. A putt is indeed too short to reach the target, but an approach throw may be successful. The approach intention is selected, and translated into a distance and direction value, as shown in Table 2.

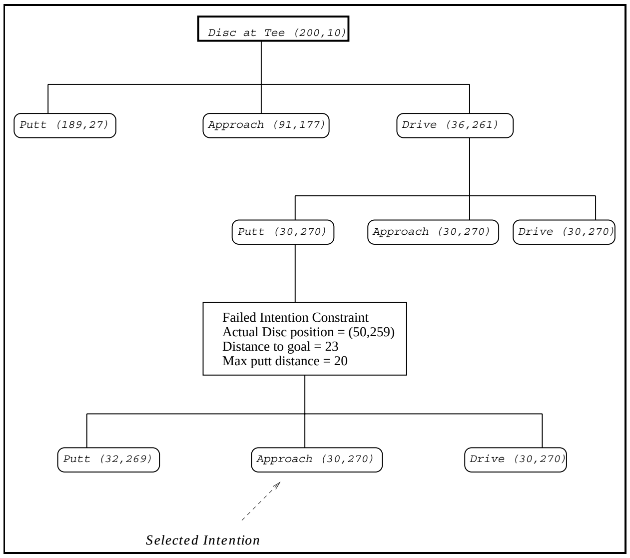

Figure 4: The Intention Tree Constructed In Experiment 3

Despite the relative simplicity of the domain, these experiments show that it is possible for an agent to construct an intention tree in order to bring about a long-term goal. A small amount of variation when executing an intention will not necessarily require intention reconsideration, but the agent will be willing and able to reconsider its intentions on the fly if this becomes unavoidable.

Intentions as Pro-Active Explanations

The architecture proposed in this paper views intentions as the pro-active equivalent of explanations. Competing alternative explanations are used to guide the backward chaining process which eventually yields a maximally coherent interpretation of the agent's environment. Similarly, competing alternative intentions are used to guide the planning process, and will yield a set of intentions which are maximally coherent with the agent's existing beliefs.

Just as dealing with low-level sensor data alone makes perception difficult, so dealing with low-level actions alone makes planning difficult. By considering its actions at the intention level, the agent is better equipped to deal with dependencies and inconsistencies that may exist between its planned actions.

This is not to suggest that the agent will have to generate a competing set of intentions on the basis of every interpretation of the environment. Rather, the default interpretation will always be carried forward from the previous interpretation. If nothing has changed, the agent can go ahead and use its default set of intentions. Even if the interpretation has changed, an agent may still be able to use the default intentions. Whether or not this is possible will depend on whether or not any of the constraints which link the interpretations and intentions have been violated. If they have not, then the default set of intentions is still valid. If constraints have been violated, then the intention reconsideration process must begin.

An intention encapsulates the actions required to bring it about. It is assumed that the actions encapsulated within an intention are internally coherent. This assumption allows us to verify the coherence of the pro-active side of the agent only at the level of the agent's intentions.

Intentions are combined to form a rooted intention tree. The paths from the root intention to any of the leaves represent alternative paths through the intention space.

Intentions Give Meaning to Actions

Intentions combine to form coherent methods of achieving the desired goal. They are then translated into actions which are put into effect to achieve the desired intention. Note that while individual intentions are meaningful, individual actions have little meaning when taken in isolation from their related intentions. This reflects the distinction between human intentions and actions.

For example, consider my intention to type the word "for". At the intention level, I formulate the intention to type the word. At the action level, however, the actions required to carry out this intention are *all* of the form"Move hand horizontally to a certain position, then move fingers vertically."

The point is that very little meaning can be attached to actions in themselves. In order to determine the meanings of actions, we must refer to the intentions which the actions represent an attempt to bring about. This means that when reasoning about actions, we should be reasoning at the *intention* level, rather than the action level.

Grosz, Hunsberger, and Kraus argue that agents operating in a multi-agent environment can hold intentions without knowing *themselves* how to bring the intention about (Grosz, Hunsberger, & Kraus 1999). While this definition of intention may not be completely compatible with ours, we do agree that actions and intentions are distinct, mutually supporting entities. Thus, we are not arguing that intentions should be seen as *replacing* actions, as actions will always be necessary in order to bring about the agent's intentions. Rather, we suggest that agents should construct and manipulate plans at the intention level, rather than the action level.

Intention constraints are used by the system to determine whether variations between intended and actual execution have rendered the current set of intentions obsolete. If they have not, then intention reconsideration is not necessary. If they have, then the agent must reconsider its intentions.

Whether or not the intention constraints succeed, action constraints are applied prior to executing each action. Action constraints are designed to detect cases where the overall intention is still valid, but the original action which was originally associated with that constraint must be varied. This variation is entirely local to the current intention, and does not affect the rest of the system. In effect, this mechanism adds a degree of reactivity to the system.

Further Work

It is possible to identify two major directions in which this research could be extended. Firstly, while the experiments described above provide a useful demonstration of the intention representation and reconsideration mechanism, it would be interesting to implement an agent which performs both coherence-based perception and coherence-based intention reconsideration. It is our belief that such an agent would be well suited to performing flexible goal-driven tasks in a dynamic environment.

Secondly, even using the relatively simply domain of the discgolf simulator, it would be interesting to extend this example to a multi-agent case. The fact that the agent is playing with other agents, either cooperatively or competitively, should produce interesting variations in intention management.

Conclusions

The concept being put forward in this paper is that allowing agents to plan their actions at the intention level allows them to exploit relationships between intentions that do not exist between low-level actions. Furthermore, in the event of a lack of consistency between an existing set of intentions and the agent's revised model of the environment, the high-level explanation based recovery process allows the agent to focus on what it believes to be the cause of the problem it is facing, and reconstruct its intentions accordingly.

References

Audi, R. 1993. *The Structure of Justification.* Cambridge, UK: Cambridge University Press.

Gärdenfors, P. 1990. The dynamics of belief systems: foundations vs. coherence theories. *Revue Internationale de Philosophie* 44:24–46.

Grosz, B.; Hunsberger, L.; and Kraus, S. 1999. Planning and acting together. *AI Magazine* 20(4):23–34.

Jennings, N. R.; Sycara, K.; and Wooldridge, M. 1998. A roadmap of agent research and development. *International Journal of Autonomous Agents and Multi-Agent Systems* 1(1):7–38.

Lacey, N., and Lee, M. 2001. The implications of philosophical foundations for knowledge representation and learning in agents. In Kudenko, D., and Alonso, E., eds., *Proceedings of the AISB 01 Symposium on Adaptive Agents and Multi-Agent Systems*, 13–24. York, UK: The Society for the Study of Artificial Intelligence and the Simulation of Behaviour.

Lacey, N. 2000. *Investigating the Relevance and Application of Epistemological and Metaphysical Theories to Agent Knowledge Bases*. Ph.D. Dissertation, University of Wales, Aberystwyth.

Schut, M., and Wooldridge, M. 2001. Principles of intention reconsideration. In *AGENTS'01*. Montreal, Canada: ACM.

Shanahan, M. 1996. Noise and the common sense informatic situation for a mobile robot. In *AAAI/IAAI, Vol. 2*, 1098–1103.

Wooldridge, M. 2000. *Reasoning About Rational Agents*. Cambridge, MA: MIT Press.