2 The Belief-Desire-Intention Model

In this chapter, I give a detailed introduction to the belief-desire-intention model of agency. As noted in chapter 1, the BDI model combines three distinct components: a *philosophical* component; a *software architecture* component; and a *logical* component. The purpose of this chapter is primarily to introduce the first and second components of the BDI model; the remainder of the book investigates the logical component in detail. I begin by introducing Bratman's theory of human practical reasoning.

2.1 Practical Reasoning

The BDI model has its roots in the philosophical tradition of understanding *practical reasoning* in humans. Put simply, practical reasoning is reasoning directed towards actions — the process of figuring out what to do:

> Practical reasoning is a matter of weighing conflicting considerations for and against competing options, where the relevant considerations are provided by what the agent desires/values/cares about and what the agent believes. [21, p.17]

It is important to distinguish practical reasoning from *theoretical reasoning* [48]. Theoretical reasoning is directed towards beliefs. To use a rather tired example, if I believe that all men are mortal, and I believe that Socrates is a man, then I will usually conclude that Socrates is mortal. The process of concluding that Socrates is mortal is theoretical reasoning, since it affects only my beliefs about the world. The process of deciding to catch a bus instead of a train, however, is practical reasoning, since it is reasoning directed towards action.

Human practical reasoning appears to consist of at least two distinct activities. The first of these involves deciding *what* state of affairs we want to achieve; the second process involves deciding *how* we want to achieve these states of affairs. The former process — deciding what states of affairs to achieve — is known as *deliberation.* The latter process — deciding how to achieve these states of affairs — we call *means-ends reasoning.*

To better understand deliberation and means-ends reasoning, consider the following example. When a person graduates from university with a first degree, he or she is faced with some important choices. Typically, one proceeds in these choices by first deciding what sort of career to follow. For example, one might consider a career as an academic, or a career in industry. The process of deciding which career to aim for is deliberation. Once one has fixed upon a career, there are further choices to be made; in particular, how to bring about this career. Suppose that after deliberation, you choose to pursue a career as an academic. The next step is to decide *how to achieve* this state of affairs. This process is means-ends reasoning. The end result of means-ends reasoning is a *plan* or *recipe* of some kind for achieving the chosen state of affairs. For the career example, a plan might involve first applying to an appropriate university for a Ph.D. place, and so on. After obtaining a plan, an agent will typically then attempt to carry out (or *execute*) the plan, in order to bring about the chosen state of affairs. If all goes well (the plan is sound, and the agent's environment cooperates sufficiently), then after the plan has been executed, the chosen state of affairs will be achieved.

Thus described, practical reasoning seems a straightforward process, and in an ideal world, it would be. But there are several complications. The first is that deliberation and means-ends reasoning are *computational* processes. In all real agents (and in particular, artificial agents), such computational processes will take place under *resource bounds*. By this I mean that an agent will only have a fixed amount of memory and a fixed processor available to carry out its computations. Together, these resource bounds impose a limit on the size of computations that can be carried out in any given amount of time. No real agent will be able to carry out arbitrarily large computations in a finite amount of time. Since almost any real environment will also operate in the presence of *time constraints* of some kind, this means that means-ends reasoning and deliberation must be carried out in a fixed, finite number of processor cycles, with a fixed, finite amount of memory space. From this discussion, we can see that resource bounds have two important implications:

- Computation is a valuable resource for agents situated in real-time environments. The ability to perform well will be determined at least in part by the ability to make efficient use of available computational resources. In other words, an agent must *control* its reasoning effectively if it is to perform well.

- Agents cannot deliberate indefinitely. They must clearly *stop* deliberating at some point, having chosen some state of affairs, and commit to achieving this state of affairs. It may well be that the state of affairs it has fixed upon is not optimal — further deliberation may have led it to fix upon an another state of affairs.

We refer to the states of affairs that an agent has chosen and committed to as its *intentions*. The BDI model of agency is ultimately one that recognizes the primacy of intentions in practical reasoning, and it is therefore worth discussing the roles that they play in more detail.

2.2 Intentions in Practical Reasoning

First, notice that it is possible to distinguish several different types of intention. In ordinary speech, we use the term "intention" to characterize both *actions* and *states of mind*. To adapt an example from Bratman [20, p.1], I might intentionally push someone under a train, and push them with the intention of killing them. Intention is here used to characterize an action — the action of pushing someone under a train. Alternatively, I might have the intention this morning of pushing someone under a train this afternoon. Here, intention is used to characterize my state of mind. In this book, when I talk about intentions, I mean intentions as states of mind. In particular, I mean *future-directed intentions* — intentions that an agent has towards some future state of affairs.

The most obvious role of intentions is that they are *pro-attitudes* [21, p.23]. By this, I mean that they tend to lead to action. Suppose I have an intention to write a book. If I truly have such an intention, then you would expect me to make a *reasonable attempt* to achieve it. This would usually involve, at the very least, me initiating some plan of action that I believed would satisfy the intention. In this sense, intentions tend to play a primary role in the production of action. As time passes, and my intention about the future becomes my intention about the present, then it plays a direct role in the production of action. Of course, having an intention does not necessarily lead to action. For example, I can have an intention now to attend a conference later in the year. I can be utterly sincere in this intention, and yet if I learn of some event that must take precedence over the conference, I may never even get as far as considering travel arrangements.

Bratman notes that intentions play a much stronger role in influencing action than other pro-attitudes, such as mere desires:

My desire to play basketball this afternoon is merely a potential influencer of my conduct this afternoon. It must vie with my other relevant desires [. . .] before it is settled what I will do. In contrast, once I intend to play basketball this afternoon, the matter is settled: I normally need not continue to weigh the pros and cons. When the afternoon arrives, I will normally just proceed to execute my intentions. [21, p.22]

The second main property of intentions is that they *persist*. If I adopt an intention to become an academic, then I should persist with this intention and attempt to achieve it. For if I immediately drop my intentions without devoting any resources to achieving them, then I will not be acting rationally. Indeed, you might be inclined to say that I never really had intentions in the first place.

Of course, I should not persist with my intention for too long — if it becomes clear to me that I will never become an academic, then it is only rational to drop my intention to do so. Similarly, if the reason for having an intention goes away, then it would be rational for me to drop the intention. For example, if I adopted the intention to become an academic because I believed it would be an easy life, but then discover that this is not the case (e.g., I might be expected to actually teach!), then the justification for the intention is no longer present, and I should drop the intention.

If I initially fail to achieve an intention, then you would expect me to *try again* — you would not expect me to simply give up. For example, if my first application for a Ph.D. program is rejected, then you might expect me to apply to alternative universities.

The third main property of intentions is that once I have adopted an intention, the very fact of having this intention will constrain my future practical reasoning. For example, while I hold some particular intention, I will not subsequently entertain options that are *inconsistent* with that intention. Intending to write a book, for example, would preclude the option of partying every night: the two are mutually exclusive. This is in fact a highly desirable property from the point of view of implementing rational agents, because in providing a "filter of admissibility," intentions can be seen to constrain the space of possible intentions that an agent needs to consider.

Finally, intentions are closely related to beliefs about the future. For example, if I intend to become an academic, then I should believe that, assuming some certain background conditions are satisfied, I will indeed become an academic. For if I truly believe that I will never be an academic, it would be nonsensical of me to have an intention to become one. Thus if I intend to become an academic, I should at least believe that there is a good chance I will indeed become one. However, there is what appears at first sight to be a paradox here. While I might believe that I will indeed succeed in achieving my intention, if I am rational, then I must also recognize the possibility that I can *fail* to bring it about — that there is some circumstance under which my intention is not satisfied.

From this discussion, we can identify the following closely related situations:

- Having an intention to bring about φ, while believing that you will not bring about φ is called *intention-belief inconsistency*, and is not rational (see, e.g., [20, pp.37–38]).
- Having an intention to achieve φ without believing that φ will be the case is *intention-belief incompleteness*, and is an acceptable property of rational agents (see, e.g., [20, p.38]).

The distinction between these two cases is known as the *asymmetry thesis* [20, pp.37–41].

Summarizing, we can see that intentions play the following important roles in practical reasoning:

- *Intentions drive means-ends reasoning.*
If I have formed an intention, then I will attempt to achieve the intention, which involves, among other things, deciding *how* to achieve it. Moreover, if one particular course of action fails to achieve an intention, then I will typically attempt others.
- *Intentions persist.*
I will not usually give up on my intentions without good reason — they will persist, typically until I believe I have successfully achieved them, I believe I cannot achieve them, or I believe the reason for the intention is no longer present.
- *Intentions constrain future deliberation.*
I will not entertain options that are inconsistent with my current intentions.
- *Intentions influence beliefs upon which future practical reasoning is based.*
If I adopt an intention, then I can plan for the future on the assumption that I will achieve the intention. For if I intend to achieve some state of affairs while simultaneously believing that I will not achieve it, then I am being irrational.

Notice from this discussion that intentions *interact* with an agent's beliefs and other mental states. For example, having an intention to φ implies that I do not believe φ is impossible, and moreover that I believe given the right circumstances, φ will be achieved. However, satisfactorily capturing the interaction between intention and belief turns out to be surprisingly hard: the way in which intentions interact with other mental states is considered in chapter 5.

```
Algorithm: Agent Control Loop Version 1
1.  while true
2.        observe the world;
3.        update internal world model;
4.        deliberate about what intention to achieve next;
5.        use means-ends reasoning to get a plan for the intention;
6.        execute the plan
7.  end while
```

Figure 2.1
A basic agent control loop.

2.3 Implementing Rational Agents

Based on the discussion above, let us consider how we might go about building a BDI agent. The strategy I use is to introduce progressively more complex agent designs, and for each of these designs, to investigate the type of behavior that such an agent would exhibit, compared to the desired behavior of a rational agent as discussed above. This then motivates the introduction of a refined agent design, and so on.

The first agent design is shown in Figure 2.1. The agent continually executes a cycle of observing the world, deciding what intention to achieve next, determining a plan of some kind to achieve this intention, and then executing this plan.

The first point to note about this loop is that we will not be concerned with stages (2) or (3). Observing the world and updating an internal model of it are important processes, worthy of study in their own right; but they lie outside the scope of this volume. Instead, we are concerned primarily with stages (4) to (6). Readers interested in understanding the processes of observing and updating can find some key references in the "further reading" section at the end of this chapter.

One of the most important issues raised by this simple agent control loop follows from the fact that the deliberation and means-ends reasoning processes are not instantaneous. They have a *time cost* associated with them. Suppose that the agent starts deliberating at time t_0, begins means-ends reasoning at t_1, and subsequently begins executing the newly formed plan at time t_2. The time taken to deliberate is thus

$$t_{deliberate} = t_1 - t_0$$

and the time taken for means-ends reasoning is

$$t_{me} = t_2 - t_1$$

Further suppose that the deliberation process is *optimal* in the sense that if it selects some intention to achieve, then the achievement of this intention at time t_0 would be the best thing for the agent. So at time t_1, the agent has selected an intention to achieve that would have been optimal *if it had been achieved at* t_0. But unless $t_{deliberate}$ is vanishingly small, then the agent runs the risk that the intention selected is no longer optimal by the time the agent has fixed upon it.

Of course, selecting an intention to achieve is only part of the overall practical reasoning problem; the agent still has to determine *how* to achieve the intention, via means-ends reasoning.

As with deliberation, assume that the means-ends reasoning process enjoys optimality in the sense that if it is given an intention to achieve and it starts executing at time t_1, then it will select the course of action to achieve this intention such that this course of action would be optimal if executed at time t_1. Again, means-ends reasoning is not instantaneous: it takes time to find the best course of action to achieve the intention. How long exactly might it take to find the best action? Suppose the agent has n possible actions available to it at any given time. Then there are $n!$ possible sequences of these actions. Since $n!$ has exponential growth, this means that the most obvious technique for finding a course of action (systematically searching through the space of possible action sequences) will be impossible in practice for all but the most trivial n. The problem is exacerbated by the fact that the plan is developed on the basis of an observation made of the environment at time t_0. The environment may be very different by time t_2.

Assuming we have an agent with optimal deliberation and means-ends reasoning components, in the sense described above, it should be clear that this agent will have overall optimal behavior in the following circumstances:

1. when deliberation and means-ends reasoning take a vanishingly small amount of time; or

2. when the world is guaranteed to remain static while the agent is deliberating and performing means-ends reasoning, so that the assumptions upon which the choice of intention to achieve and plan to achieve the intention remain valid until the agent has completed deliberation and means-ends reasoning; or

3. when an intention that is optimal when achieved at time t_0 (the time at which the world is observed) is guaranteed to remain optimal until time t_2 (the

time at which the agent has found a course of action to achieve the intention).

Conditions (2) and (3) will not hold in most realistic environments. As we noted in chapter 1, the kinds of environments in which agents work will tend to be highly dynamic, and such environments will not remain static long enough for an agent to determine from first principles either an optimal intention or, given such an intention, an optimal course of action to achieve it. Similarly, most important tasks that we might assign to agents will be highly time dependent.

Let us now make the agent algorithm presented above slightly more formal. For this, we require some notation. Throughout this chapter, I make one important assumption: that the agent maintains some explicit *representation* of its beliefs, desires, and intentions. However, I will not be concerned with *how* beliefs and the like are represented. One possibility is that they are represented *symbolically*, for example as logical statements *a là* PROLOG facts [34]. Indeed, for most of this book, it is assumed that beliefs take exactly this form. However, the assumption that beliefs, desires, and intentions are symbolically represented is by no means necessary for the remainder of the book. We use B to denote a variable that holds the agent's current beliefs, and let Bel be the set of all such beliefs. Similarly, we use D as a variable for desires, and Des to denote the set of all desires. Finally, the variable I represents the agent's intentions, and Int is the set of all possible intentions.

Next, we need some way of representing an agent's *perception*. Consider a robotic agent, equipped with a video camera, but no other kind of sensor apparatus. Then the video feed delivered to this agent represents the information available to the agent about its environment. Any representation of its environment that the robot creates must ultimately be derived from this video feed. We refer to the video feed as the robot's *perceptual input*. In implemented agent systems, perceptual input is usually packaged into discrete bundles, which are referred to as *percepts*. We use $\rho, \rho', \rho_1, \ldots$ to represent percepts that the agent receives. Let Per be the set of all such percepts.

Finally, we need to say something about *plans*. Plans and intentions are closely related. We frequently use the two terms interchangeably in everyday speech, saying "I plan to..." for "I intend to..."; as Bratman puts it: "Plans are intentions writ large" [20, p.29]. In this book, when we refer to plans, we are referring to plans as *recipes* for achieving intentions. For example, one plan/recipe for achieving my intention of being at the airport might involve catching a taxi; another might involve catching a bus; and so on. Plans and planning are a major research topic in their own right — see [2] for a detailed survey. For our purposes, a simple model of plans will suffice. A plan is viewed as a tuple consisting of:

- a *pre-condition*, which defines the circumstances under which a plan is applicable — for example, one of the pre-conditions of the plan to catch a taxi to the airport is that I have sufficient funds to pay for it;
- a *post-condition*, which defines what states of affairs the plan achieves — for example, post-conditions of my plan to catch a taxi to the airport include me having less money, and me now being located at the airport;
- a *body*, which is the actual "recipe" part of the plan — for our purposes, the body is simply a sequence of actions.

We will use π (with decorations: $\pi', \pi_1, \ldots$) to denote plans, and let *Plan* be the set of all plans (over some set of actions Ac). We will make use of a number of auxiliary definitions for manipulating plans (some of these will not actually be required until later in this chapter):

- if π is a plan, then we write $pre(\pi)$ to denote the pre-condition of π, $post(\pi)$ to denote the post-condition of π, and $body(\pi)$ to denote the body of π;
- if π is plan, then we write $empty(\pi)$ to mean that plan π is the empty sequence (thus $empty(\ldots)$ is a boolean-valued function);
- $execute(\ldots)$ is a procedure that takes as input a single plan and executes it without stopping — executing a plan simply means executing each action in the plan body in turn;
- if π is a plan then by $hd(\pi)$ we mean the plan made up of the first action in the plan body of π; for example, if the body of π is $\alpha_1, \ldots, \alpha_n$, then the body of $hd(\pi)$ contains only the action α_1;
- if π is a plan then by $tail(\pi)$ we mean the plan made up of all but the first action in the plan body of π; for example, if the body of π is $\alpha_1, \alpha_2, \ldots, \alpha_n$, then the body of $tail(\pi)$ contains actions $\alpha_2, \ldots, \alpha_n$;
- if π is a plan, $I \subseteq Int$ is a set of intentions, and $B \subseteq Bel$ is a set of beliefs, then we write $sound(\pi, I, B)$ to mean that π is a sound plan for achieving I given beliefs B. (We will not discuss or attempt to define what makes a plan sound here — the classic paper on the subject is Lifschitz [136].)

We can now define the components of an agent's control loop. An agent's belief update process is formally modeled as a *belief revision function*. Such a belief revision function has the following signature.

$brf : \wp(Bel) \times Per \rightarrow \wp(Bel)$

In other words, on the basis of the current beliefs and current percept, the belief revision function determines a new set of beliefs. (As noted above, in this book we are *not* concerned with how belief revision might work; see [71].)

The agent's deliberation process is given by a function

$deliberate : \wp(Bel) \rightarrow \wp(Int)$

which takes a set of beliefs and returns a set of intentions — those selected by the agent to achieve, on the basis of its beliefs.

An agent's means-ends reasoning is represented by a function

$plan : \wp(Bel) \times \wp(Int) \rightarrow Plan$

which on the basis of an agent's current beliefs and current intentions, determines a plan to achieve the intention. Note that there is nothing in the definition of the *plan*(...) function which requires an agent to engage in *plan generation* — constructing a plan from scratch [2]. In most BDI systems, the *plan*(...) function is implemented by giving the agent a *plan library* [78]. A plan library is a pre-assembled collection of plans, which an agent designer gives to an agent. Finding a plan to achieve an intention then simply involves a single pass through the plan library to find a plan that, when executed, will have the intention as a post-condition, and will be sound given the agent's current beliefs. In implemented BDI agents, pre- and post-conditions are often represented as (lists of) atoms of first-order logic, and beliefs and intentions as ground atoms of first-order logic. Finding a plan to achieve an intention then reduces to finding a plan whose pre-condition unifies with the agent's beliefs, and whose post-condition unifies with the intention.

The agent control loop is now as shown in Figure 2.2. This algorithm highlights some limitations of this simple approach to agent control. In particular, steps (4) to (7) inclusive implicitly assume that the environment has not changed since it was observed at stage (3). Assuming that the time taken to actually execute the plan dominates the time taken to revise beliefs, deliberate, or plan, then the crucial concern is that the environment might change while the plan is being executed. The problem is that the agent remains *committed* to the intention it forms at step (5) until it has executed the plan in step (7). If the environment changes after step (3), then the assumptions upon which this plan

```
Algorithm: Agent Control Loop Version 2
1.  $B := B_0$;      /* $B_0$ are initial beliefs */
2.  while true do
3.      get next percept $\rho$;
4.      $B := brf(B, \rho)$;
5.      $I := deliberate(B)$;
6.      $\pi := plan(B, I)$;
7.      $execute(\pi)$
8.  end while
```

Figure 2.2
A first refinement of the agent control loop.

depends may well be invalid by the time the plan is actually executed.

2.4 The Deliberation Process

So far, we have glossed over the problem of exactly how an agent might go about deliberating. In this section, we consider the process in more detail. It is not hard to see that in real life, deliberation typically begins by trying to understand what the *options* available to you are. Returning to the career choice example introduced above, if you gain a good first degree, then one option is that of becoming an academic; if you fail to obtain a good degree, this option is not available to you. Another option is entering industry. After deciding what the options are, you must *choose between them*, and *commit* to some. These chosen options then become intentions.

From this discussion, we can see that the *deliberate* function as discussed above can be decomposed into two distinct functional components:

- *option generation* — in which the agent generates a set of possible alternatives; and
- *filtering* — in which the agent chooses between competing alternatives, and commits to achieving them.

We represent option generation via a function, *options*, which takes the agent's current beliefs and current intentions, and from them determines a set of options that we will hereafter refer to as *desires*. The intuitive interpretation of a desire is that, in an "ideal world," an agent would like *all* its desires achieved. In any moderately realistic scenario, however, an agent will not be able to

```
Algorithm: Agent Control Loop Version 3

B := B_0;          /* B_0 are initial beliefs */
I := I_0;          /* I_0 are initial intentions */
while true do
    get next percept $\rho$;
    $B := brf(B, \rho)$;
    $D := options(B, I)$;
    $I := filter(B, D, I)$;
    $\pi := plan(B, I)$;
    $execute(\pi)$
end while
```

Figure 2.3
Refining the deliberation process into option generation and filtering.

achieve all its desires. This is because desires are often mutually exclusive. For example, in the OASIS air-traffic control system, which was implemented using a BDI architecture, an agent was tasked with the problem of finding the optimal sequence in which to land aircraft at an airport [138]. The option generation process in OASIS might generate the options of landing two different aircraft on the same runway at the same time. Clearly, such options are mutually exclusive: it would be undesirable to land both aircraft simultaneously.

Formally, the signature of the option generation function *options* is as follows.

$$options : \wp(Bel) \times \wp(Int) \rightarrow \wp(Des)$$

In order to select between competing options, an agent uses a *filter* function. Intuitively, the filter function must simply select the "best" option for the agent to commit to. We represent the filter process through a function *filter*, with a signature as follows.

$$filter : \wp(Bel) \times \wp(Des) \times \wp(Int) \rightarrow \wp(Int)$$

The agent control loop incorporating explicit deliberation and means-ends reasoning is shown in Figure 2.3. Notice that desires are an *input to* the filter process, whereas intentions are an *output from* it.

2.5 Commitment Strategies

When an option successfully passes through the *filter* function and is hence chosen by the agent as an intention, we say that the agent has made a *commitment* to that option. Commitment implies *temporal persistence* — an intention, once adopted, should not immediately evaporate. A critical issue is just *how* committed an agent should be to its intentions. That is, how long should an intention persist? Under what circumstances should an intention vanish?

To motivate the discussion further, consider the following scenario:

Some time in the not-so-distant future, you are having trouble with your new household robot. You say "Willie, bring me a beer." The robot replies "OK boss." Twenty minutes later, you screech "Willie, why didn't you bring me that beer?" It answers "Well, I intended to get you the beer, but I decided to do something else." Miffed, you send the wise guy back to the manufacturer, complaining about a lack of commitment. After retrofitting, Willie is returned, marked "Model C: The Committed Assistant." Again, you ask Willie to bring you a beer. Again, it accedes, replying "Sure thing." Then you ask: "What kind of beer did you buy?" It answers: "Genessee." You say "Never mind." One minute later, Willie trundles over with a Genessee in its gripper. This time, you angrily return Willie for overcommitment. After still more tinkering, the manufacturer sends Willie back, promising no more problems with its commitments. So, being a somewhat trusting customer, you accept the rascal back into your household, but as a test, you ask it to bring you your last beer. Willie again accedes, saying "Yes, Sir." (Its attitude problem seems to have been fixed.) The robot gets the beer and starts towards you. As it approaches, it lifts its arm, wheels around, deliberately smashes the bottle, and trundles off. Back at the plant, when interrogated by customer service as to why it had abandoned its commitments, the robot replies that according to its specifications, it kept its commitments as long as required — commitments must be dropped when fulfilled or impossible to achieve. By smashing the bottle, the commitment became unachievable. [35, pp.213–214]

The mechanism an agent uses to determine when and how to drop intentions is known as a *commitment strategy*. The following three commitment strategies are commonly discussed in the literature of rational agents [187]:

- *Blind commitment*
A blindly committed agent will continue to maintain an intention until it believes the intention has actually been achieved. Blind commitment is also sometimes referred to as *fanatical* commitment.
- *Single-minded commitment*
A single-minded agent will continue to maintain an intention until it believes that either the intention has been achieved, or else that it is no longer possible

```
Algorithm: Agent Control Loop Version 4

$B := B_0$;      /* $B_0$ are initial beliefs */
$I := I_0$;      /* $I_0$ are initial intentions */
while true do
    get next percept $\rho$;
    $B := brf(B, \rho)$;
    $D := options(B, I)$;
    $I := filter(B, D, I)$;
    $\pi := plan(B, I)$;
    while not $empty(\pi)$ do
        $\alpha := hd(\pi)$;
        $execute(\alpha)$;
        $\pi := tail(\pi)$;
        get next percept $\rho$;
        $B := brf(B, \rho)$;
        if not $sound(\pi, I, B)$ then
            $\pi := plan(B, I)$
        end-if
    end-while
end-while
```

Figure 2.4
Introducing reactivity: the agent will engage in means ends reasoning when it believes a plan has (for whatever reason) failed.

to achieve the intention.

- *Open-minded commitment*

An open-minded agent will maintain an intention as long as it is still believed possible.

Note that an agent has commitment both to *ends* (i.e., the state of affairs it wishes to bring about), and *means* (i.e., the mechanism via which the agent wishes to achieve the state of affairs). Currently, our agent control loop is over-committed, both to means and ends. It never stops to reconsider its intentions, and it remains committed to plans until they have been fully executed. We will now see how this basic control loop can be refined in various ways, in order to obtain different types of commitment.

The first modification we make allows the agent to *replan* if ever a plan goes wrong. This reduces the commitment that an agent has towards the means to achieve its intentions. The revised control loop is illustrated in Figure 2.4.

Using this loop, the agent will be committed to a plan to achieve its intentions only while it believes that the plan is sound given its beliefs about

```
Algorithm: Agent Control Loop Version 5

$B := B_0$;       /* $B_0$ are initial beliefs */
$I := I_0$;       /* $I_0$ are initial intentions */
while true do
    get next percept $\rho$;
    $B := brf(B, \rho)$;
    $D := options(B, I)$;
    $I := filter(B, D, I)$;
    $\pi := plan(B, I)$;
    while not ($empty(\pi)$ or $succeeded(I, B)$ or $impossible(I, B)$) do
        $\alpha := hd(\pi)$;
        $execute(\alpha)$;
        $\pi := tail(\pi)$;
        get next percept $\rho$;
        $B := brf(B, \rho)$;
        if not $sound(\pi, I, B)$ then
            $\pi := plan(B, I)$
        end-if
    end-while
end-while
```

Figure 2.5
Dropping intentions that are either impossible or that have succeeded.

the current state of the world. If it ever determines that its plan is no longer appropriate in order to achieve the current intention, then it engages in further means-ends reasoning in order to find an alternative plan. Given that its beliefs are updated every time it executes an action, this implies at least some degree of reactivity.

This version of the control loop is clearly more attuned to the environment than previous versions, but it still remains overcommitted to intentions. This is because although the agent will replan if ever it believes the plan is no longer appropriate, it never stops to consider whether or not its intentions are appropriate. This consideration motivates the next version of the control loop, in which an agent explicitly stops to determine whether or not its intentions have succeeded or whether they are impossible. We write *succeeded*(I, B) to mean that given beliefs B, the intentions I can be regarded as having been satisfied. Similarly, we write *impossible*(I, B) to mean that intentions I are impossible given beliefs B. The revised control loop is then illustrated in Figure 2.5.

It is easy to see that this revised agent control loop implements single-minded commitment. The modifications required to implement open-minded commitment is straightforward, and is left to the reader. It should be stressed that, insofar as we are concerned in this book, there is no one "ideal" commitment strategy. Different circumstances call for different commitment strategies.

2.6 Intention Reconsideration

In our current algorithm, an agent will get to reconsider its intentions once every time around the outer control loop. This implies that it will reconsider its intentions when one of the following three conditions arises:

- it has completely executed a plan to achieve its current intentions; or
- it believes it has achieved its current intentions; or
- it believes its current intentions are no longer possible.

Although this ensures that the agent is neither undercommitted nor overcommitted to its intentions, it is limited in the way that it permits an agent to *reconsider* its intentions. The main problem is that it does not allow an agent to *exploit serendipity.* To see what I mean by this, consider the following scenario.

Sophie is a BDI software agent whose task is to obtain documents on behalf of a user. One day, the user instructs Sophie to obtain a soft copy of the Ph.D. thesis by A. N. Other, and Sophie creates an intention to this effect. Sophie believes that the Ph.D. is resident at A. N. Other's WWW site, and so generates an intention to download it from there. While she is planning how to achieve this intention, however, she is told that a local copy of the thesis exists. It would clearly be more efficient to obtain this version, but since there is nothing wrong with the intention of downloading the Ph.D. remotely (she believes it will succeed), she continues to do so.

In this scenario, Sophie would do better to reconsider her intentions *while she is executing the plan.* A first attempt to modify the agent control loop would involve reconsidering intentions every time the agent executed the inner loop in Figure 2.5: see Figure 2.6. Such an agent is *cautious*, in the sense that it always stops to reconsider its intentions before performing an action.

If option generation and filtering were computationally cheap processes, then this would be an acceptable strategy. Unfortunately, we know that deliberation is not cheap — it takes a considerable amount of time. While the agent is deliberating, the environment in which the agent is working is changing, possibly rendering its newly formed intentions irrelevant.

```
Algorithm: Agent Control Loop Version 6

$B := B_0$;        /* $B_0$ are initial beliefs */
$I := I_0$;        /* $I_0$ are initial intentions */
while true do
    get next percept $\rho$;
    $B := brf(B, \rho)$;
    $D := options(B, I)$;
    $I := filter(B, D, I)$;
    $\pi := plan(B, I)$;
    while not ($empty(\pi)$ or $succeeded(I, B)$ or $impossible(I, B)$) do
        $\alpha := hd(\pi)$;
        $execute(\alpha)$;
        $\pi := tail(\pi)$;
        get next percept $\rho$;
        $B := brf(B, \rho)$;
        $D := options(B, I)$;
        $I := filter(B, D, I)$;
        if not $sound(\pi, I, B)$ then
            $\pi := plan(B, I)$
        end-if
    end-while
end-while
```

Figure 2.6
A cautious agent, which stops to reconsider intentions before performing any action.

We are thus presented with a dilemma:

- an agent that does not stop to reconsider its intentions sufficiently often will continue attempting to achieve its intentions even after it is clear that they cannot be achieved, or that there is no longer any reason for achieving them;
- an agent that *constantly* reconsiders its attentions may spend insufficient time actually working to achieve them, and hence runs the risk of never actually achieving them.

There is clearly a trade-off to be struck between the degree of commitment and reconsideration at work here. To try to capture this trade-off, we can modify the agent to incorporate an explicit *meta-level control* component. The idea is to have a boolean-valued function, *reconsider*, such that $reconsider(I, B)$ evaluates to "true" just in case it is appropriate for the agent with beliefs B and intentions I to reconsider its intentions. The agent control loop incorporating the $reconsider(\ldots)$ function is shown in Figure 2.7.

```
Algorithm: Agent Control Loop Version 7

$B := B_0$;        /* $B_0$ are initial beliefs */
$I := I_0$;        /* $I_0$ are initial intentions */
while true do
    get next percept $\rho$;
    $B := brf(B, \rho)$;
    $D := options(B, I)$;
    $I := filter(B, D, I)$;
    $\pi := plan(B, I)$;
    while not ($empty(\pi)$ or $succeeded(I, B)$ or $impossible(I, B)$) do
        $\alpha := hd(\pi)$;
        $execute(\alpha)$;
        $\pi := tail(\pi)$;
        get next percept $\rho$;
        $B := brf(B, \rho)$;
        if $reconsider(I, B)$ then
            $D := options(B, I)$;
            $I := filter(B, D, I)$;
        end-if
        if not $sound(\pi, I, B)$ then
            $\pi := plan(B, I)$
        end-if
    end-while
end-while
```

Figure 2.7
An agent that attempts to strike a balance between boldness and caution: whether or not the agent chooses to reconsider intentions is determined by the boolean-valued function *reconsider*(...).

In this version of the agent control loop, the burden of deciding when to expend effort by deliberating lies with the function *reconsider*(...). It is interesting to consider the circumstances under which this function can be said to behave *optimally*. Suppose that the agent's deliberation and plan generation functions are in some sense perfect: that deliberation always chooses the "best" intentions (however that is defined for the application at hand), and planning always produces an appropriate plan. Further suppose that time expended always has a cost — the agent does not benefit by doing nothing. Then it is not difficult to see that the function *reconsider*(...) will be behaving optimally if, and only if, whenever it chooses to deliberate, the agent changes intentions [251]. For if the agent chose to deliberate but did not change intentions, then the effort expended on deliberation was wasted. Similarly, if an agent should have changed intentions, but failed to do so, then the effort expended on attempting to achieve its intentions was also wasted.

Table 2.1
Practical reasoning situations (cf. [22, p.353]).

Situation number	Chose to deliberate?	Changed intentions?	Would have changed intentions?	*reconsider*(...) optimal?
1	No	—	No	Yes
2	No	—	Yes	No
3	Yes	No	—	No
4	Yes	Yes	—	Yes

The possible interactions between meta-level control and deliberation are summarized in Table 2.1:

- In situation (1), the agent did not choose to deliberate, and as a consequence, did not choose to change intentions. Moreover, if it *had* chosen to deliberate, it would not have changed intentions. In this situation, the *reconsider*(...) function is behaving optimally.
- In situation (2), the agent did not choose to deliberate, but if it had done so, it *would* have changed intentions. In this situation, the *reconsider*(...) function is not behaving optimally.
- In situation (3), the agent chose to deliberate, but did not change intentions. In this situation, the *reconsider*(...) function is not behaving optimally.
- In situation (4), the agent chose to deliberate, and did change intentions. In this situation, the *reconsider*(...) function is behaving optimally.

Notice that there is an important assumption implicit within this discussion: that the cost of executing the *reconsider*(...) function is *much* less than the cost of the deliberation process itself. Otherwise, the *reconsider*(...) function could simply use the deliberation process as an oracle, running it as a subroutine and choosing to deliberate just in case the deliberation process changed intentions.

The nature of the trade-off was examined by David Kinny and Michael Georgeff in a number of experiments carried out using a BDI agent system [116]. The aims of Kinny and Georgeff's investigation were to:

(1) assess the feasibility of experimentally measuring agent effectiveness in a simulated environment, (2) investigate how commitment to goals contributes to effective agent behavior and (3) compare the properties of different strategies for reacting to change [116, p.82]

In Kinny and Georgeff's experiments, two different types of reconsideration strategy were used: *bold* agents, which never pause to reconsider their intentions before their current plan is fully executed, and *cautious* agents, which stop to reconsider after the execution of every action. These characteristics are defined by a *degree of boldness*, which specifies the maximum number of plan steps the agent executes before reconsidering its intentions. Dynamism in the environment is represented by the *rate of world change*, γ. Put simply, the rate of world change is the ratio of the speed of the agent's control loop to the rate of change of the world. If $\gamma = 1$, then the world will change no more than once for each time the agent can execute its control loop. If $\gamma = 2$, then the world can change twice for each pass through the agent's control loop, and so on. The performance of an agent is measured by the ratio of number of intentions that the agent managed to achieve to the number of intentions that the agent had at any time. Thus if effectiveness is 1, then the agent achieved all its intentions. If effectiveness is 0, then the agent failed to achieve any of its intentions. The main results of Kinny and Georgeff's experiments are shown in Figure 2.8.[1] This graph shows the effectiveness of an agent on the y axis against the dynamism of the world (log scale) on the x axis. The key results of Kinny and Georgeff were as follows.

- If γ is low (i.e., the environment does not change quickly), then bold agents do well compared to cautious ones. This is because cautious ones waste time reconsidering their commitments while bold agents are busy working towards — and achieving — their intentions.
- If γ is high (i.e., the environment changes frequently), then cautious agents tend to outperform bold agents. This is because they are able to recognize when intentions are doomed, and also to take advantage of serendipitous situations and new opportunities when they arise.

The bottom line is that different environment types require different intention reconsideration and commitment strategies. In static environments, agents that are strongly committed to their intentions will perform well. But in dynamic environments, the ability to react to changes by modifying intentions becomes more important, and weakly committed agents will tend to outperform bold agents.

2.7 Mental States and Computer Programs

We have been talking about programs that have a "belief-desire-intention" architecture. This is mentalistic terminology, and it is worth pausing for a

1 I am deeply indebted to Martijn Schut for providing this graph.

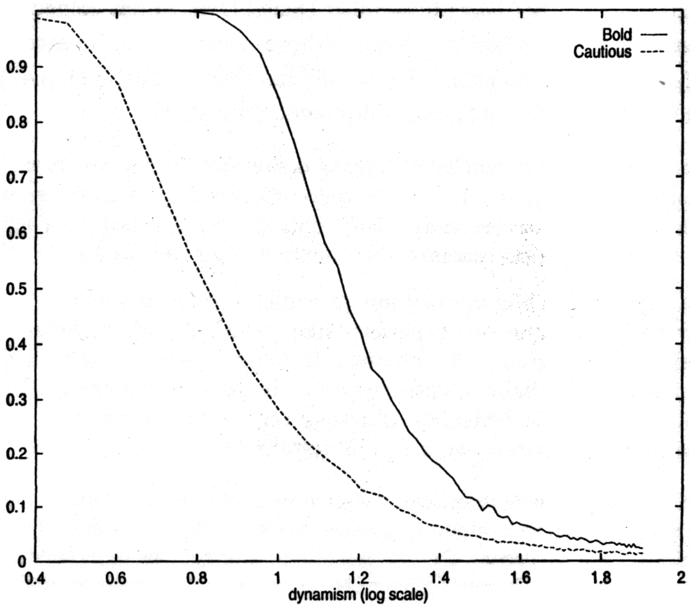

Figure 2.8
Kinny and Georgeff's intention reconsideration experiments.

moment to examine the justification for using such terms. It may seem strange to think of computer programs in terms of mental states such as beliefs, desires, and intentions. Is it either useful or legitimate to use mentalistic terminology to characterize machines? As it turns out, there is an extensive literature on this subject.

When explaining human activity, it is often useful to make statements such as the following:

> Janine took her umbrella because she *believed* it was going to rain.
> Michael worked hard because he *wanted* to finish his book.

These statements make use of a *folk psychology*, by which human behavior is predicted and explained through the attribution of *attitudes*, such as believing and wanting (as in the above examples), hoping, fearing, and so on (see e.g., [226, p.1] for a discussion of folk psychology). This folk psychology is well established: most people reading the above statements would say they found their meaning entirely clear, and would not give them a second glance.

The attitudes employed in such folk psychological descriptions are called the *intentional* notions.[2] The philosopher Daniel Dennett has coined the term *intentional system* to describe entities "whose behavior can be predicted by the method of attributing belief, desires and rational acumen" [43, p.49], [42]. Dennett identifies different "levels" of intentional system:

> A *first-order* intentional system has beliefs and desires (etc.) but no beliefs and desires *about* beliefs and desires. [...] A *second-order* intentional system is more sophisticated; it has beliefs and desires (and no doubt other intentional states) about beliefs and desires (and other intentional states) — both those of others and its own. [43, p.243]

One can carry on this hierarchy of intentionality as far as required.

Now we have been using phrases like belief, desire, intention to talk about computer programs. An obvious question is whether it is legitimate or useful to attribute beliefs, desires, and so on to artificial agents. Isn't this just anthropomorphism? McCarthy, among others, has argued that there are occasions when the *intentional stance* is appropriate:

> To ascribe *beliefs, free will, intentions, consciousness, abilities,* or *wants* to a machine is legitimate when such an ascription expresses the same information about the machine that it expresses about a person. It is useful when the ascription helps us understand the structure of the machine, its past or future behavior, or how to repair or improve it. It is perhaps never logically required even for humans, but expressing reasonably briefly what is actually known about the state of the machine in a particular situation may require mental qualities or qualities isomorphic to them. Theories of belief, knowledge and wanting can be constructed for machines in a simpler setting than for humans, and later applied to humans. Ascription of mental qualities is most straightforward for machines of known structure such as thermostats and computer operating systems, but is most useful when applied to entities whose structure is incompletely known. [148], (quoted in Shoham [205]; underlining is from [205])

What objects can be described by the intentional stance? As it turns out, almost any automaton can. For example, consider a light switch:

> It is perfectly coherent to treat a light switch as a (very cooperative) agent with the capability of transmitting current at will, who invariably transmits current when it believes that we want it transmitted and not otherwise; flicking the switch is simply our way of communicating our desires. [205, p.6]

And yet most adults in the modern world would find such a description absurd — perhaps even infantile. Why is this? The answer seems to be that while the intentional stance description is perfectly consistent with the observed

2 Unfortunately, the word "intention" is used in several different ways in logic and the philosophy of mind. First, there is the BDI-like usage, as in "I intended to kill him." Second, an intentional notion is one of the attitudes, as above. Finally, in logic, the word intension (with an "s") means the internal content of a concept, as opposed to its extension. In what follows, the intended meaning should always be clear from context.

behavior of a light switch, and is internally consistent,

> ...it does not *buy us anything*, since we essentially understand the mechanism sufficiently to have a simpler, mechanistic description of its behavior. [205, p.6]

Put crudely, the more we know about a system, the less we need to rely on animistic, intentional explanations of its behavior.[3] An obvious question is then, if we have alternative, perhaps less contentious ways of explaining systems, why should we bother with the intentional stance? Consider the alternatives available to us. One possibility is to characterize the behavior of a complex system by using the *physical stance* [44, p.36]. The idea of the physical stance is to start with the original configuration of a system, and then use the laws of physics to predict how this system will behave:

> When I predict that a stone released from my hand will fall to the ground, I am using the physical stance. I don't attribute beliefs and desires to the stone; I attribute mass, or weight, to the stone, and rely on the law of gravity to yield my prediction. [44, p.37]

Another alternative is the *design stance*. With the design stance, we use knowledge of what purpose a system is supposed to fulfill in order to predict how it behaves. Dennett gives the example of an alarm clock [44, pp.37–39]. When someone presents us with an alarm clock, we do not need to make use of physical laws in order to understand its behavior. We can simply make use of the fact that all alarm clocks are designed to wake people up if we set them with a time. No understanding of the clock's mechanism is required to justify such an understanding — we know that *all* alarm clocks have this behavior.

However, with very complex systems, even if a complete, accurate picture of the system's architecture and working *is* available, a physical or design stance explanation of its behavior may not be practicable. Consider a computer. Although we might have a complete technical description of a computer available, it is hardly practicable to appeal to such a description when explaining why a menu appears when we click a mouse on an icon. In such situations, it may be more appropriate to adopt an intentional stance description, if that description is consistent, and simpler than the alternatives.

3 Shoham observes that the move from an intentional stance to a technical description of behavior correlates well with Piaget's model of child development, and with the scientific development of humankind generally [205]. Children will use animistic explanations of objects — such as light switches — until they grasp the more abstract technical concepts involved. Similarly, the evolution of science has been marked by a gradual move from theological/animistic explanations to mathematical ones. The author's own experiences of teaching computer programming suggest that, when faced with completely unknown phenomena, it is not only children who adopt animistic explanations. Indeed, it often seems easier to teach some computer concepts by using explanations such as: "the computer doesn't know...," than to try to teach abstract principles first.

Note that the intentional stance is, in computer science terms, nothing more than an *abstraction tool.* It is a convenient shorthand for talking about complex systems, which allows us to succinctly predict and explain their behavior without having to understand how they actually work. Now, much of computer science is concerned with looking for good abstraction mechanisms, since these allow system developers to *manage complexity* with greater ease. The history of programming languages illustrates a steady move away from low-level machine-oriented views of programming towards abstractions that are closer to human experience. Procedural abstraction, abstract data types, and most recently, objects are examples of this progression. So, why not use the intentional stance as an abstraction tool in computing — to explain, understand, and, crucially, *program* complex computer systems?

There are other reasons for believing that an intentional stance will be useful for understanding and reasoning about computer programs [102]. First, and perhaps most importantly, the ability of heterogeneous, self-interested agents to communicate seems to imply the ability to talk about the beliefs, aspirations, and intentions of individual agents. For example, in order to *coordinate* their activities, agents must have information about the intentions of others [105]. This idea is closely related to Newell's *knowledge level* [166]. Later in this book, we will see how mental states such as beliefs, desires, and the like are used to give a semantics to *speech acts* [202, 35]. Second, mentalistic models are a good candidate for representing information about end users. For example, imagine a tutoring system that works with students to teach them JAVA programming. One way to build such a system is to give it a *model* of the user. Beliefs, desires, and intentions seem appropriate for the makeup of such models.

For many researchers in AI, this idea of programming computer systems in terms of "mentalistic" notions such as belief, desire, and intention is the key component of agent-based computing. The idea of programming computer systems in terms of mental states was articulated most clearly by Yoav Shoham in his *agent-oriented programming* (AOP) proposal [206]. BDI systems can be viewed as a kind of AOP.

2.8 Notes and Further Reading

Some reflections on the origins of the BDI model, and on its relationship to other models of agency, may be found in [74]. Belief-desire-intention architectures originated in the work of the Rational Agency project at Stanford Research Institute in the mid-1980s. Key figures were Michael Bratman, Phil Cohen, Michael Georgeff, David Israel, Kurt Konolige, and Martha Pollack. The origins of the model lie in the theory of human practical reasoning developed by the philosopher Michael Bratman [20], which focuses particularly on the role of intentions in practical reasoning. The conceptual framework of the BDI model is described in [22], which also describes a specific BDI agent architecture called IRMA.

The best-known implementation of the BDI model is the PRS system, developed by Georgeff and colleagues [78, 77]. The PRS has been re-implemented several times since the mid-1980s, for example in the Australian AI Institute's DMARS system [46], the University of Michigan's C++ implementation UM-PRS, and a JAVA version called JAM! [100]. JACK is a commercially available programming language, which extends the JAVA language with a number of BDI features [26].

The description of the BDI model given here draws upon [22] and [188], but is not strictly faithful to either. The most obvious difference is that I do not incorporate the notion of the "filter override" mechanism described in [22], and I also assume that plans are linear sequences of actions (which is a fairly "traditional" view of plans), rather than the hierarchically structured collections of goals used by PRS.

Plans are central to the BDI model of agency. An excellent discussion on the BDI model, focusing in particular on the role of plans in practical reasoning, is Martha Pollack's 1991 *Computers and Thought* award lecture, presented at the IJCAI-91 conference in Sydney, Australia, and published as "The Uses of Plans" [180]. Another article, which focuses on the distinction between "plans as recipes" and "plans as mental states" is [179].

It is worth emphasizing that the BDI model is only one solution to the problem of building autonomous rational agents. Many other software architectures for agent systems have been described in the literature [249, 24]. The so-called "layered" architectures are currently very popular [242]; examples include Ferguson's TOURINGMACHINES [57, 58], Müller's INTERRAP [163, 162], and the 3T architecture [17]. I mentioned in passing that belief revision is not a concern of this book: see Gärdenfors [71] for further information.

Finally, although in this book I focus on what might be called the "orthodox" BDI model, the BDI model forms a central component of many other systems, which either draw inspiration from it or implement parts of it. Examples include [61, 111, 158, 26].