Rational Agents and Emotion

Michael Heemskerk

University of Groningen
Groningen, The Netherlands

RMIT University
Melbourne, Australia

SUMMARY

A central issue in the development of agents that have practical reasoning skills is the concept of resource-boundedness. For an agent to reason effectively, he has to use the limited resources that are available to him (e.g. processing time, memory) in an effective way. Research on planning systems such as STRIPS [FN71] and the limitations of these systems have led to the development of the BDI paradigm. According to this approach, an agent can be described as having beliefs, desires and intentions. Roughly speaking, beliefs correspond to the agent's knowledge of the world, desires correspond to the agent's goals, and intentions correspond to the set of goals that an agent has currently adopted.

Adding the 'mental' state intention to agent ontology greatly reduces the computational costs of the agent's planning processes, making it feasible for an agent to perform in real-time. However, in principle the BDI approach cannot guarantee that an agent is responsive to its environment, another important requirement for an agent that functions in a real-world application. Practical systems based on BDI theory such as PRS [GL87] and dMARS [dLW97] have addressed this issue, but not in a conceptually convincing fashion.

In humans, emotions appear to play a key role in ensuring our being responsive to the environment. In this thesis, I explore psychological and computational models of emotion, and incorporate a computational model of control, based on models of emotion, in the BDI architecture. The resulting extended BDI architecture enables agents to adapt the timing of planning processes to the current state of the environment. For example, the resulting agent will interrupt his current planning processes and re-plan his actions when he becomes 'afraid'.

Finally, I have implemented a greatly simplified version of the proposed extended BDI architecture in order to test the effectiveness of the model.

PREFACE

This thesis is the result of my graduation project for a master's degree in Cognitive Science at the University of Groningen. This project was performed at the Artificial Intelligence group at the department of Computer Science at RMIT University in Melbourne, Australia.

I have chosen to do my master's project at RMIT because the Artificial Intelligence group had done a number of interesting projects on computational models of emotion in the past. In my opinion, this topic brings together a number of important aspects of cognitive science, and I believe it is a perfect topic for a master's thesis in cognitive science. On the one hand there is the goal of developing an artificial system that displays coherent behavior, an important goal in artificial intelligence. On the other hand, there is a fascinating phenomenon that seems to play an important role in known intelligent life: emotion. It might well be that computational models of emotion can be used to address some of the current problems of agent architectures.

In this project, I have had two supervisors: Niels Taatgen in the Netherlands and associate professor Lin Padgam at RMIT University. I have started out my project by getting familiar with agent theory, BDI agents in particular, emotion research and existing computational models of emotion. This research led me to believe that concepts from theories of emotion can be used to improve the reasoning processes of existing agent architectures, in particular in balancing planning and responsiveness. In order to test this hypothesis I was to incorporate a shallow implementation of such an emotion system in an existing agent system, the PAC system.

PAC is short for Personality and Cognition and is a system that was developed at RMIT University in order to test the appropriateness of implementing emotions in an agent system. In the course of my project I have worked with PAC extensively and have spent a lot of time fixing some of the problems with the system itself. Partly because of this, I have not been able to test the developed emotional agents in the system.

I have received much-needed financial support for the trip to and stay in Australia from a number of funds: the Marco Polo fund and Groningen University fund from the University of Groningen, and a fund from the Schimmel-Schuurman van Outeren Foundation, for all of which I am very grateful.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

In this introductory chapter I provide some background in agent systems, and BDI systems in particular. Furthermore, I will give a brief overview of theories of emotion, computational models of emotion, and research that has attempted to endow agents with artificial emotions.

1.1 Agents as Intentional systems

Recently, there has been a great deal of interest in artificial agents from the academic and industrial research community. Part of the reason for this interest lays in the fact that it is easy to relate to the level of complexity of an agent. When modeling an environment, in which multiple independent functional units can be distinguished, the agent metaphor provides a convenient and appropriate level of description. Just as there are different departments and divisions in a complex organization, it makes sense to break up a complex piece of software in a number of semi-independent agents, each of which has different abilities and responsibilities.

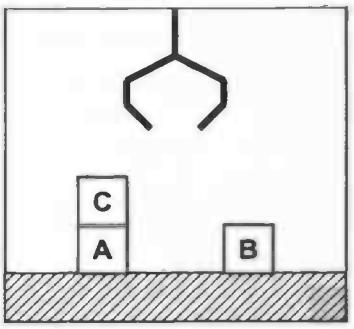
Although the agent metaphor is a very popular one, there is no generally agreed upon definition of the concept of an agent. Still, the general consensus is that at least agents should function continuously and be autonomous to some degree. Opinions differ on what it precisely means for an agent to be autonomous, but in general the term is taken to mean that the agent's activities do not require constant human guidance or intervention [Sho93]. In addition, many applications require agents to be intelligent: to show goal-directed behavior and respond to events in the environment in a timely fashion.

A common approach to the design of agent systems views an agent as an intentional system. An intentional system is a system, which can be described as having beliefs, desires and intentions. For example, a thermostat can be said to 'believe' that the temperature in the room is at a certain level and 'desire' to have the temperature in the room at a different level. Describing a thermostat as an intentional system is not very useful since we already have a clear idea of the inner workings of a thermostat, but it does help us reason about many more complex systems (e.g. other people, robots, computer programs). The idea is that it is not only useful to describe an agent's behavior in these terms, but also to use these concepts in the design of an agent. The *intentional stance* [McC79, Den87], as this approach is called, results in a computational agent having symbolic representations of his beliefs, desires (goals) and intentions.

Beliefs describe the information that is available to an agent about the world and itself. *Desires* are preferred states of the world and roughly correspond to goals. *Intentions* are partial plans of action that the agent has decided to act upon. Since agents are usually designed to perform one or more goals, a mechanism is needed to map an agent's beliefs and desires to actions. In the symbolic AI community a lot of research has been done in the field of planning systems and means-end analysis. It is not surprising that many agent architectures include a reasoning mechanism that is based on this body of research.

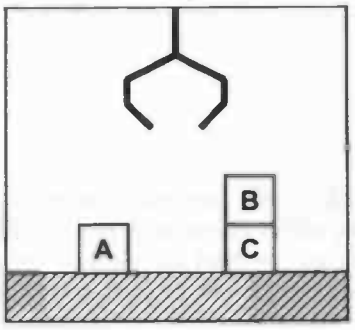
In traditional planning systems, the planning process usually involves a search in state space, using a goal stack. A typical example of this approach is the STRIPS system [FN71], which has representations of the current state, goal states and a set of actions. The system tries to find a sequence of actions that will achieve the goal by simple means-end analysis. As an illustration of STRIPS-like planning systems consider the following classic block-world example. In this example the system has to plan a sequence of actions that will transform the current state of the world into some goal state.

Current state



ontable(A) ∧
on(C, A) ∧ clear(C) ∧
ontable(B) ∧ clear(B) ∧
armempty

Goal state



ontable(A) ∧ clear(A) ∧
on(B, C) ∧ clear(B) ∧
ontable(C) ∧
armempty

Action	Precondition	Postcondition
stack(x, y)	holding(x) ∧ clear(y)	on(x, y) ∧ armempty
unstack(x, y)	on(x, y) ∧ armempty	holding(x) ∧ clear(y)
pickup(x)	clear(x) ∧ ontable(x)	holding(x)
putdown(x)	holding(x)	armempty ∧ ontable(x) ∧ clear(x)

A correct sequence of actions is found by focusing on the goal state and, for each aspect of the goal state that is not yet true, try to find an action that makes it true. If such an action exists but is not applicable, set a sub-goal to make the action applicable. By following this process the following action sequence can be found.

unstack(C, A) → putdown(C) → pickup(B) → stack(B, C)

Planning systems such as STRIPS are not suitable for most real-time applications, that is applications where planning has to be performed in real-time. There are two reasons for this. First, as the number of possible actions increases, the search space increases dramatically. As a result, the time the system spends in the planning processes will often exceed real-time demands. A second problem is the issue of responsiveness. Any system working in real-time has to stay responsive to changes in the environment. However, planning systems such as STRIPS only perform planning for predefined goals and do not include mechanisms for monitoring the environment. In addition, many planning systems assume that the beliefs, on which the planning process operates, remain valid during the planning process. Clearly, in practical applications agents have to function in fast-changing environments and the assumption that an agent's beliefs will not change during the planning process simply will not hold.

These issues have been recognized by Bratman [Bra87], who suggests that, since every real agent is resource-bounded, no agent can continuously decide which goals to pursue. He proposes that an agent should choose a goal, and commit to that goal. Bratman used the term intention to describe such a committed goal. In order to maintain responsiveness, this

commitment should not be unconditional, and suitable conditions for rejecting a committed goal have to be developed. In the next chapter on BDI (beliefs, desires and intentions) agents, I will describe these conditions in more detail.

At the time of Bratman's proposal, the general consensus held that rational behavior could be analyzed in terms of beliefs (knowledge about the world) and desires (goals). His work however, led to the recognition that a third mental state, intention, was needed. The introduction of intention and commitment addresses the issue of responsiveness, but in itself does not restrict the planning process. In order to restrict the planning process, Bratman has proposed to organize intentions into partial plans of action. For instance, the intention of buying a book can be achieved by executing the following plan:

- Go to a bookshop
- Get the book
- Pay for the book

This plan is partial, in that each of the steps needs to be further elaborated. This is done by adopting the intention to, say go to a bookshop, which leads to the execution of another partial plan that specifies which intentions need to be achieved in order to achieve 'go to the bookshop'. In this way a hierarchy of partial plans is executed to achieve a given intention. At the lowest level of this hierarchy, elementary actions are executed.

Following Bratman's proposal of adding a third mental state, intention, to the theory of rational agents, a lot of research has started to investigate what properties intention should satisfy, how intention is related to other mental states like belief and desire and how intentions are formed and retracted. Also, a number of researchers have built agent architectures on the basis of the BDI (belief, desire and intention) theory (e.g. IRMA [BIP88] and PRS [GL87]). The field of research that focuses on the importance of intentions in agent models is better known as the *BDI paradigm* and has become one of the leading frameworks in the deliberative approach to agent systems.

1.2 Alternative approaches

While the introduction of intention and commitment provides some way of dealing with the issue of resource-boundedness within the field of deliberate agent architectures, a group of researchers have proposed radically different approaches to agent systems. According to one of the strongest critics of the deliberative approach, Rodney Brooks, intelligent behavior arises as a result of an agent's interaction with the world. Brooks [Bro91] argues that human level intelligence is too complex and little understood to be correctly decomposed into the right sub-pieces. It would be better to start with very simple level intelligence and incrementally build up the capabilities of the system.

Brooks demonstrated his ideas by building a number of robots based on the subsumption architecture. These robots are very simple in terms of the amount of computation they need to do and do not use explicit abstract reasoning at all. While designing his robots based on this approach, Brooks noticed that for very simple level intelligence explicit representations and models of the world are simply in the way: it is better to use the world as its own model. This observation has led Brooks to formulate the hypothesis that intelligent behavior can arise without the need for a central symbolic representation or explicit abstract reasoning of the kind symbolic AI proposes.

The simplicity of the computational processes in the behavior-based approach, as Brooks' design method has been named, has attracted a lot of researchers to experiment with the approach. It has been found that while the behavior-based approach promises agent systems that can successfully function in real-time environments, the actual design process is extremely difficult. Successful coordination of different competing behaviors involves a long

process of trial and error. Also, it has been argued that the behavior-based approach might well work for low-level behaviors such as obstacle avoidance, but is unsuitable for tasks, which involve multiple complex long-term goals (e.g. [Fer92]).

Seeing that the strong points of the deliberative and the behavior-based architectures complement each other, some researchers have attempted to combine the different approaches in one architecture. These resulting *hybrid architectures* usually consist of a number of different layers, each of which functions at a different level of abstraction (e.g. [Fer92, MPT95]). At the lowest level of abstraction, the reactive layer is capable of reacting to events in the world, without engaging in complex reasoning. An intermediate level abstracts away from raw sensory data and provides a *knowledge level* description of the agent. Typically, this level is capable of performing means-end analysis. Some hybrid architectures add a third layer, which deals with social aspects of the environment: reasoning about and coordination with other agents.

The different layers in a hybrid architecture need to interact in order to produce coherent behavior, but the way in which this is done varies from architecture to architecture. Usually, a potentially large number of ad hoc rules are defined that coordinate the balance between the layers for specific situations or actions. There is a need for a clean, well-motivated model of control in these architectures, if we are to understand how and why these architectures work. It seems that in different situations different layers of the architecture are important: in an emergency, the agent needs to *react* quickly and therefore the reactive layer should play the dominant role. But if all is well, the agent has got more time, and it might pay to do some abstract reasoning in order to find a possibly more appropriate behavior.

In the search for a model of control between the layers of hybrid architectures, interrupting ongoing reasoning and shifting the control to the reactive layer in case of an emergency seems to be a requirement. According to Simon [Sim67] emotions might play this role in humans. His theory implies that there are at least two different mechanisms working at all times: a goal executor that generates and monitors actions, and processes that continuously check for situations that require attentive processing.

1.3 Emotions and agents

Emotions play an important role in our everyday life, and it is hard to imagine how life would be if we did not experience emotions. Still, it has long been thought that emotion is not a prerequisite for the existence of intelligence. Emotions were seen to be exactly what rationality is *not* about. However, recent research in neurobiology has clearly implicated emotion in decision-making, learning and memory processes [Dam94, LeD96].

At a biological level, emotions are complex phenomena that have neurological, hormonal, cognitive and social aspects. When frightened for example, we consciously experience fear. At the same time, the fear emotion is accompanied by a response of the autonomic system that prepares the body for action by increasing heartbeat and respiration. Some of these bodily responses are externally visible, and also serve to communicate the emotion to other people. The autonomic responses depend on subcortical parts of the nervous system, and operate on a subconscious level. In the development and expression of an emotion, several phases can be distinguished:

- The recognition of an important event — seeing your house on fire.
- Subconscious bodily responses to the event — increased heart beat, sweating, etc.
- The conscious experience of the emotion — fear.

Note that the subconscious bodily response precedes the conscious experience of the emotion. The subcortical structures that are involved in the subconscious bodily response (the amygdala, hypothalamus and hippocampus) receive information from both the sensory

systems and the cortex. The sensory information is not very detailed, but provides a rough picture of the external world, enabling a swift response to a potential threat.

There has long been a controversy about the role of the cortex in generating emotions. James and Lange proposed that emotional experience is the direct result of sensory information arriving in the cortex, or to put it as James wrote: *"We feel sorry because we cry, angry because we strike, afraid because we cry and not that we cry, strike or tremble because we are sorry, angry or fearful as the case may be."* The James-Lange hypothesis has since been refined in many ways, most importantly by Stanley Schachter [Sch64] in the 1960s and more recently by Antonio Damasio [Dam94]. In the refined theory, the cortex creates a cognitive response to sensory information that is consistent with the individual's expectations and social context. As such, the experience of emotion is essentially a story that the cortex concocts to explain the bodily reactions.

Initiating swift responses to potential dangers is an important function of our emotions. For instance, when we see a curled shape resembling a snake on the ground ahead of us, we are startled and stop walking. This is clearly functional, since it steers us away from danger, and at the same time forces us to attend the potential threat. It may turn out that the curled shape was just a branch of a tree, and once we recognize it as a branch our heartbeat returns to normal levels, etc. For such an alarm system to work effectively, it has to respond quickly to important events in the environment. It doesn't really matter if the system produces a false alarm from time to time.

Another important function of emotions is their motivating value. Evolution has 'designed' us to enjoy things like eating and sex, because it has turned out to be beneficial for us. Poisonous plants often don't taste very good to us; another very useful property evolution has equipped us with. As such, our emotions often steer us away from potential dangers. Emotions also play an important role in learning. For instance, when you feel embarrassed because you've just said something stupid in public, next time you'll think twice before you open your mouth.

These functions of the emotion system are clearly adaptive, but does it make sense to incorporate a computational model of emotions in an agent architecture? I think it does, because it could improve an agent's responsiveness to the world, especially for agent systems that work in real-world situations. I believe emotions can serve as a basis for a model of control in agent architectures that balances planning and reactivity.

I am not the first to envisage a role for emotions in agent architectures. Recently, a number of researchers from the agent community have tried to incorporate computational models of emotion in their agent architectures in attempts to make their agent systems more adaptive (e.g. [Bat94, FS87, Cn97 and Wri97]). In the remainder of this section I will describe some of this research.

An influential model of emotion appraisal is Ortony, Collins and Clores cognitive model of emotion elicitation [OCC88]. The model describes how an emotional response is constructed by *cognitive analysis* of the features of the environment in relation to an agent's beliefs, goals and standards. Under the model, emotion types are distinguished on the basis of the types of situations, which give rise to them.

Ortony et al. propose three main categories of emotions; emotions arising from an agent's perception of:

- *events*, which are advantageous or disadvantageous with regard to the agent's goals (e.g. joy, distress).
- *actions*, the agent's own actions and those of other agents, which are compared to the agent's standards for behavior (e. g. pride, shame).
- *objects*, towards which the agent has an attitude (e.g. like, dislike).

These global categories can be further divided into a number of distinct emotion types. For instance, examples of emotion types belonging in the category of emotions arising in response to perceived *events* are hope, fear, satisfaction and disappointment.

Although Ortony et al. have not implemented their theory, a number of other researchers have used this model as a basis for the implementation of other computational models of emotion. Two well-known examples are the Oz project at CMU [Bat94] and Elliot's Affective Reasoner [Ell92]. Both of these applications incorporate a model of emotion and emotion processing very similar to the Ortony et al. model. In addition, agents in these applications are able to display their emotional state in their behavior (e.g. by modifying the movements of the agent or by producing simulated facial expressions).

While Ortony et al. focus on the appraisal process, Frijda and Swagerman [FS87] approach the study of emotions from a functional perspective. Central in their research on emotion is the assumption that emotions help people function in an uncertain world. They construct their model by analyzing what properties a subsystem that implements these adaptive functions would need to have. To quote:

> *"The major phenomena are: the existence of the feelings of pleasure and pain, the importance of cognitive or appraisal variables, the presence of innate, preprogrammed behaviors as well as of complex constructed plans for achieving emotion goals, and the occurrence of behavioral interruption, disturbance and impulse-like priority of emotional goals. The system properties underlying these phenomena are facilities for relevance detection of events with regard to the multiple concerns, availability of relevance signals that can be recognized by the action system, and facilities for control precedence, or flexible goal priority ordering and shift."* [FS87], (p. 235).

Concerns play a central role in Frijda's theory of emotion. A concern refers to an agent's preference for certain states of the world and roughly corresponds with what other theories refer to as motives or major goals. Staying alive, being part of a social group and being respected by others are good examples of concerns. These examples also make clear how concerns lead to adaptive behavior: a concern is not a goal, but a goal might be produced when an event is detected that is relevant to a concern. So, being threatened with a gun will threaten the concern of staying alive and may ultimately lead to the adoption of a goal that will lessen the threat (e.g. complying with the person with the gun).

In order to test the theory Swagerman has constructed a fairly simple computer program, ACRES, that satisfies the specifications provided by this model of emotion. The ACRES program successfully shows that concerns may be used to generate goals, but it has failed to point out precisely how these concern-based goals interact with non-emotional goals. In an attempt to overcome this and other shortcomings of ACRES, Frijda and Moffat [FM93] have recently proposed an updated model of emotion that focuses on the interaction of emotion processing and other processes like perception and abstract reasoning. As with the original concern-based theory, they have constructed a broad, but shallow implementation of the updated model. The implementation, *Will*, has been tested in the context of the prisoner's dilemma game and although the environment is quite restricted in its emotional implications, Will shows the major processing features that are proposed by the theory.

Another prominent researcher in the field of emotional agents is Dolores Cañamero. In an implementation of emotional agents in a multi-agent environment, she has shown that emotions may be used to increase the adaptiveness of the agent [Can97]. In this system, emotions arise as a result of the perception of events and patterns of stimulation. Cañamero models emotions at the hormonal level, so whenever the agent experiences an emotion, a distinct set of 'hormones' are released. These hormones in turn influence the levels of several motivators and affect the perception process. In this way emotions bias perception and in effect serve as perceptual filters.

Finally, an important research group in the field of computational models of emotion is the *Cognition and Affect* group at the University of Birmingham, headed by Aaron Sloman. The work of this group pays special attention to the interaction of emotion and other cognitive processes, or more generally the role of emotion in a larger architecture. Sloman et al. distinguish several levels of abstraction in the architecture of human cognition and argue that these different layers in the cognitive information processing apparatus have been formed at different times in our evolutionary history. In addition, Sloman et al. suggest that human mental states and processes depend on the interaction between old and new layers in a biologically plausible control architecture [SL98]. This interaction produces various kinds of internal and external behavior, including internal processes such as motive generation, attention switching, etc.

Sloman proposes that three different layers, each responsible for certain types of emotion, can be distinguished. The oldest layer, the *reactive* layer, consists of 'routine' reactive mechanisms and a global alarm system that produces emotions as a result of rapid, automatic processing (e.g. fear, sexual arousal). A more recent layer, the deliberative layer, is responsible for such processes as planning and decision making and supports cognitively rich emotional states linked to current desires, beliefs and plans (e.g. hope, relief). Finally, the newest layer, the meta-management layer, coordinates planning and attention strategies. This meta-management layer is argued to be responsible for typically human emotional states such as humiliation and guilt.

The remainder of this thesis is organized as follows. Chapter 2 deals with theories and architectures that have been developed in the BDI paradigm. Special attention is paid to the concept of commitment. In chapter 3, I describe a process model of emotion and a model of control for BDI architectures that is based on this model of emotion. Chapter 4 describes a limited implementation of the model presented in chapter 3. A scenario for evaluating the model is presented. Finally, in chapter 5, I present my conclusions and propose future areas of research.

CHAPTER 2

BDI AGENTS

The concept of a BDI agent effectively started with Bratman et al.'s high-level description of an architecture for practical reasoning, IRMA [BIP88]. The architecture was developed to address two competing requirements for agent systems. First, an architecture for a rational agent must allow for means-end analysis, for the weighing of competing desires and for interactions between these two forms of reasoning. Second, the architecture must be able to perform this reasoning in a timely fashion, i.e. address the problem of resource-boundedness.

The basic idea behind the architecture is the observation that a rational agent is committed to doing what he plans. In other words, once an agent has formed the *intention* of achieving a certain goal, he is committed to achieving it. Adopting an intention has two important influences on the process of means-end analysis. First, in IRMA intentions are structured into partial plans. Plans are partial in the sense that they describe an agent's intention, without going into high detail on how to achieve the intention. For instance, an agent may adopt the plan of buying a certain book, without deciding on a particular bookshop or whether to pay with cash or credit card. The means-end reasoning component of the architecture is responsible for the filling in of the means for achieving a certain plan. In this way intentions act as a driving force for means-end analysis.

The second way in which intentions affect planning is in the way they restrict the set of options that the planning process has to consider. Bratman et al. propose that intentions should be consistent, both internally and with the agent's beliefs. So, the intention of paying in cash for a book is inconsistent with the belief of not having enough cash to pay for the book, and having the belief prevents the intention from being adopted. Current intentions act in a similar way as a filter for allowing new intentions to be considered by the planning process.

2.1 Overview of a practical BDI architecture

Bratman's research shows how research on planning systems can be used in agent systems to produce goal-directed behavior without losing reactivity. Much research in the BDI paradigm is aimed at refining mechanisms to maintain the balance between rationality and reactivity. A number of implementations of agent architectures based on the BDI paradigm have been made, of which the best-known are the *Procedural Reasoning System* (PRS) [GL87] and its successor the *distributed Multi-Agent Reasoning System* (dMARS) [dKLW97].

Figure 2-1 gives an overview of an abstract BDI architecture that is largely based on Rao and Georgeff's work on practical BDI architectures [RG95]. Not surprisingly, the architecture comprises three data structures, which represent the agent's *beliefs*, *desires* and *intentions*. In addition to these three mentalistic data structures, the agent architecture also includes an event queue. The BDI interpreter uses the queue and the three mentalistic structures to perform the actual planning. The queue is used as a kind of a blackboard. Both the BDI interpreter and external motor and perception routines put information in the event queue (e.g. new

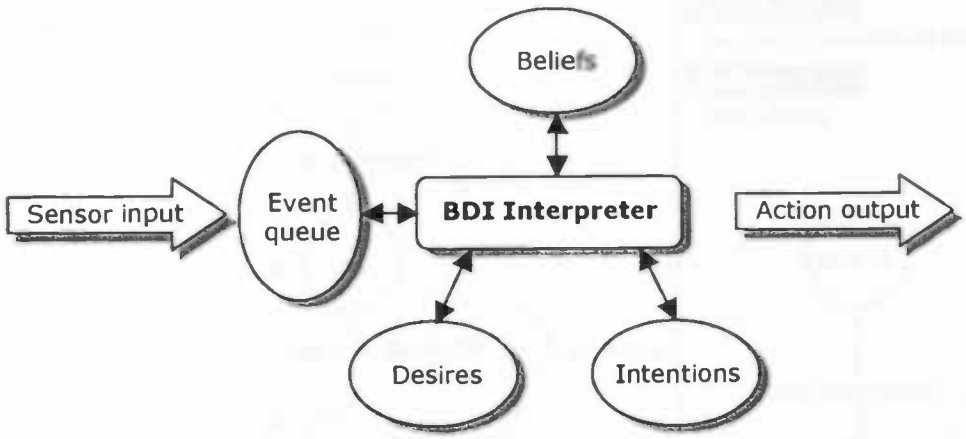


Figure 2-1: The BDI Architecture

(sub)goals, perceived events), which the BDI interpreter uses in the next step in the planning process.

Most work in the BDI paradigm is theoretical in nature, in that it aims to *describe* how an agent represents information about the world, and how an agent's attitudes (e.g. desires/intentions) interact with beliefs to produce adaptive behavior. These so-called agent theories are partly based on research on planning systems that was done in the symbolic AI community over the past decades. An agent's beliefs about the world are usually described in some form of possible worlds semantics. Under these semantics, an agent 'believes' a set of formulas, typically in propositional logic. The agent's knowledge is partial in nature: there are many aspects of the world that are hidden to the agent. The idea is that the agent can still reason about these aspects of the world by considering a number of alternatives. These alternatives can be seen as possible worlds.

For theoretically important reasons, many theories assume that an agent believes all equivalent formulas of his beliefs and also all logical consequences of his beliefs. These properties however are very problematic for practical reasoning systems. Considering the effort it takes a logician to prove that one complex logical formula follows from another complex formula, we cannot expect a real agent system with limited resources to make all these deductions in time to be useful.

Rao and Georgeff [RG95] note that planning by theorem proving is potentially boundless and therefore not very suitable for real agent systems. As a practical solution, Rao and Georgeff propose to represent only the current state of the world, which corresponds to representing only the agent's current beliefs. As a consequence, whenever some logical deduction is needed, the agent needs to adopt an explicit intention to do so.

Desires are like goals, in that they represent the motivational state of the system. However, unlike goals, desires need not be mutually consistent. It is possible for an agent to have both a desire to go to the cinema tonight and a desire to go to a friend's birthday party. While desires are allowed to conflict, they do need to be consistent with the agent's current beliefs about the world. To be more precise, an agent has to believe that a world in which the desire has been satisfied is possible.

Rao and Georgeff, like Bratman, include partial plans in their architecture for representing the means for achieving certain desires. These plans consist of a body, an invocation condition, and a context condition. The body describes the primitive actions and subgoals that have to be achieved for completing the plan. The invocation condition contains the triggering

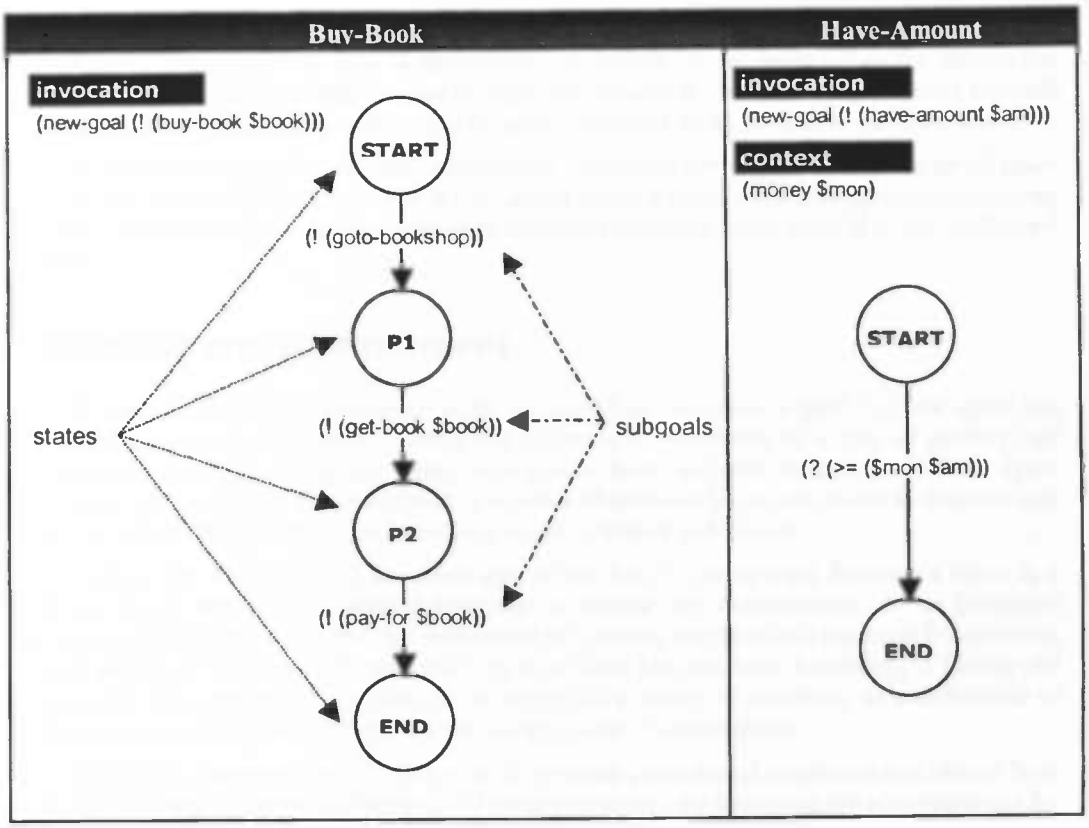


Figure 2-2: Example dMARS plans

event that has to occur before the plan can be considered, and the context condition specifies the situation that must hold for the plan to be applicable. In this practical architecture, desires are modeled as the set of plans that are currently applicable. To illustrate these concepts, I have included two example plans in dMARS syntax in figure 2-2.

dMARS uses a lisp-like syntax in the plans. The buy-book plan in figure 2-2 is only invoked when the invocation condition is true, in this case when the agent has formed a new goal (! (buy-book $book)). The '!'-operator means 'achieve', and '$' denotes a variable. When the plan is activated, the plan starts in the START-state, and will attempt to step through states P1 and P2 to reach the END-state. The plan can only move to another state if the transition link evaluates to true. In the buy-book plan, each transition link causes the agent to formulate a new goal. The transition to the next state can only be made if the new goal is achieved. If no transition can be made from a state other than the END-state, the plan returns 'false'. Suppose for example that the agent succeeds in finding a bookshop and selecting the book in question, but is unable to pay for the book. This will cause the transition from P2 to the END-state to fail and since there are no other transitions leaving state P2, the whole buy-book plan will fail.

The have-amount plan simply checks whether the agent has got enough money to pay for the book. To do so the plan accesses the agent's belief database, using the context condition. In the have-amount plan, the belief database is searched for the logical formula (money *val*). The value of the variable $mon is unified with the *val* value in the belief database. The only transition in the plan checks whether $mon is equal to or larger than $am. If so, the plan returns true.

In most rational agent theories, intentions are required to be consistent with the agent's beliefs, desires and the agent's other intentions. This means that an agent has to believe that a world in which the intention has been realized is possible, and that the agent actually has the

desire to achieve that state of the world. Furthermore, his intentions should be mutually consistent. The reason for this is easily seen: intentions can be regarded as the desires the agent has chosen to try to achieve and as such, they should not be conflicting (turning both left and right at a corner is impossible, and an agent shouldn't try to do both at the same time).

In Rao and Georgeff's practical architecture, intentions are represented by the set of plans the agent is currently pursuing. The set of current plans is hierarchically organized, containing high-level plans and plans that have been adopted to achieve some subgoal of the high-level plans.

2.2 Intention and Commitment

Central in the BDI architecture is the notion of an intention: a goal that the agent has decided to attempt to achieve. Seeing the potential of intentions as a way of guiding and restricting practical reasoning, many researchers have included intentions in their agent theories and architectures. In addition, a number of attempts have been made to describe and define in formal semantics how intentions relate to beliefs and desires.

Cohen and Levesque [CL90] were one of the first to incorporate Bratman's ideas in a formal agent theory. They defined intention as 'choice with commitment'. As in Bratman's proposal [BIP88] an agent that has committed to a certain intention will maintain his intention for a period of time and will repeatedly try to achieve his intention. According to Cohen and Levesque, the concept of commitment is central to a theory of intention, as coordination of future actions would be hopeless without some notion of commitment.

Following Bratman [Bra87] Cohen at al. propose a number of properties that should hold in any reasonable theory of intention. Of these properties, the following are also important for a theory of commitment:

- The agent keeps track of the success of his attempts to achieve his intentions and is disposed to re-plan to achieve the intended effects if earlier attempts fail.
- The agent believes it is possible to achieve his intentions.
- The agent does not believe he will not bring about his intentions.

An important issue in any theory of intention is deciding when to abandon an intention. The above-mentioned properties sketch a rough picture of the criteria Cohen and Levesque suggest for reconsidering current intentions. In their formal agent theory intentions are treated as a special kind of *persistent goal*. A persistent goal is defined as a goal that the agent has decided to pursue and will not be dropped until the agent either believes the goal has been achieved or believes that the goal is unachievable.

Identifying intention with a persistent goal ultimately leads to a fanatically committed agent: an agent that will go to extreme lengths to achieve his goal (e.g. an agent hijacking a bus in order to get to the airport in time to catch a plane). Cohen and Levesque realized this and added a third option for dropping or an intention: an agent should also drop his intentions if he believes that the reasons for adopting the intention no longer hold. To illustrate the Cohen and Levesque theory, consider the following example:

An agent that has heard on the news that it is going to rain today. Now suppose that, as a result of this forecast, the agent adopts the intention of buying an umbrella. According to Cohen and Levesques definition of intention the agent should only drop his intention if he believes he has succeeded in buying an umbrella, believes it is impossible for him to buy an umbrella (e.g. he doesn't have enough money to buy one), or if he believes it is not going to rain after all.

Singh [Sin97] argues that this definition of intention is too strong: in many cases the agent should abandon an intention even though none of Cohen and Levesques conditions for doing

so hold. For instance, an agent that has the intention of starting his own company would like to be able to drop that intention once he realizes that he does not want to invest the required effort in the project (e.g. he has to draw up a business plan in order to get a loan). In this situation it is still possible to reach his goal and h is reason for adopting the intention may still hold (e.g. he doesn't want to work for somebody else), but under Cohen et al.'s model of intention the agent cannot abandon his intention!

2.3 Commitment as Entrenchment

There is a need for additional conditions for dropping an intention in a realistic model of intention. Roughly speaking, the agent should be committed as long as it is beneficial to do so (it has a positive expected utility), and should give up as soon as it is not. Unfortunately, deciding whether a given intention is beneficial or not may not be very straightforward and if the agent has to make these decisions frequently, the reasoning processes involved may very well end up using up the resources intentions should be protecting.

The reason for introducing intention is to avoid having to repeatedly reason about one's actions. Singh goes a step further and defines commitment as *entrenchment*: an agent that is committed to an intention simply does not reconsider his intention until his commitment for that intention runs out. When this happens, the agent can decide if the intention is still worth pursuing (and re-commit to it) or if he should abandon it. Under this model an agent's commitment to an intention is a measure of the amount of time, risk or effort the agent is willing to invest in pursuing that intention. The agent is presumed to be able to make an estimate of the risk or cost of different courses of action prior to committing to one. There will be some computational cost in keeping track of when to reconsider, but this will be relatively small.

Singh admits that an agent that adopts his model of commitment will miss out on some opportunities that he could have noted by rethinking, but he argues that this comes at the advantage of not being swamped by intentions to deliberate on. A significant shortcoming of his argument is that the agent will not only miss out on opportunities, but will fail to notice threats as well, which may prove to be fatal to the agent's goals or ultimately, the agent himself. Even when an agent is committed to an intention, he should not ignore the environment altogether: the agent should drop or suspend his current intention, whenever an event occurs that threatens the agent or the agent's current goals. When this happens the agent can decide which is more important and should be dealt with first: the threatening situation at hand, or the agent's current goal(s).

Summarizing, I propose that a model of commitment should have the following properties:

- Intentions are dropped when they have been achieved, or more precisely when the agent thinks they have been achieved.
- Intentions are dropped when the agent believes that the cost of achieving them is higher than what he is willing to invest in their pursuit.
- Intentions are dropped when the agent believes he cannot possibly achieve them.
- Intentions are dropped or suspended when a more important or urgent situation arises.

I believe the last condition requires a process that continuously and swiftly monitors the environment for important events. Furthermore, I believe this role is fulfilled by emotions in humans and other animals. In the next chapter, I propose a model of control, based on a model of emotion that is set up to satisfy these properties in a fashion that also takes into account the fact that every real agent has limited resources.

CHAPTER 3

AN EMOTION-BASED MODEL OF CONTROL

From the discussion of the BDI architecture in chapter 2 it should be clear that practical BDI systems such as PRS and dMARS are able to display high-level goal-oriented behavior. Reactive behavior can also be generated by these systems. This is realized by defining a number of simple plans for handling emergencies. Both PRS and dMARS support priority schemes, which can be used to ensure that a plan can be adopted even if other plans are already active. In this way, whenever a triggering event for one of these emergency plans occurs, such a plan is activated and executed, in the meanwhile effectively suspending other, often more complex, plans that run at lower priority levels.

3.1 Coordinating behavior with limited resources

Intentions, especially when structured in partial plans, can significantly reduce the computational cost of deliberation. As was described in the previous chapter, adding intention as a functional concept to the agent architecture in principle does not address the issue of responsiveness. An agent that functions in a real environment has to be able to respond quickly to important changes in that environment. In other words, current intentions need to be pushed back or suspended in certain circumstances, in order to deal with more pressing events.

I propose to add a mechanism that monitors the environment for important events and that lets only these events interrupt current reasoning. This results in an architecture that adopts a concept of commitment that is similar to Singh's treatment of commitment as entrenchment: reasoning about which intentions to adopt is only done when an interrupt has been generated, a current intention has been achieved, or it is perceived that it is no longer possible to achieve a certain intention. So, reasoning about which intentions to adopt, and reasoning about how to achieve already adopted intentions is performed in separate, consecutive time intervals.

This approach differs from Singh's in that the level of commitment is not determined at the time an intention is adopted, but instead a mechanism is included hat dynamically monitors the environment for more important events.

Both Bratman [BIP88] and Cohen and Levesque [CL90] have argued that an agent has to monitor the success of his actions in achieving his intentions, and attempt to achieve his intentions in an alternative way if his current actions are not successful. In PRS and dMARS this is implemented by trying all possible ways to achieve a goal, until the goal is achieved or until there are no more alternatives left. I believe that emotions can also be useful in refining this function. If we don't succeed in achieving our goals, we often become frustrated or even desperate. As a result, we decide to persist, try some alternative action, or give up the goal completely, even before we've tried all possible options.

If we are to gain anything from adding a mechanism that monitors the environment for important changes, it needs to do so efficiently. Fast heuristics or spreading activation networks, rather than abstract reasoning, are capable of this kind of fast, efficient processing. A result of relying on heuristics is that the interrupt mechanism will not be perfect. Unnecessary interrupts will be produced from time to time, and at times emergencies will be missed. This latter form of error is potentially more dangerous, and special care should be taken to minimize their occurrence.

In humans, many of the desired functions of an interrupt mechanism can be found in emotional processing. Emotional states like being startled, surprised or being disgusted by some revolting tasting food force us to pay attention to the situation at hand. In addition, when these emotions are very strong we often act quickly by minimizing the amount of planning or rational deliberation: we are not interested in finding the best course of action, but we choose some action that gets us out of trouble.

The above-mentioned emotions can be said to have a global effect. Other emotions have more local effects. For instance, being irritated or frustrated are emotional states that refer to current goals. Presumably, being frustrated at a lack of progress in achieving a certain goal, causes the agent to rethink his commitment to that intention. Commitments to other intentions are not influenced by this frustration.

Most emotion theories emphasize that an agent's emotional state is the result of his subjective *cognitive* appraisal of the current situation. The appraisal process interprets the state of the world, taking into account the agent's current beliefs, desires and intentions. The resulting emotions contain two types of information: *control* information, in the form we have considered above, and *semantic* content. Every emotion type corresponds to a certain type of situation. Anger for instance, corresponds to a situation in which some aspect of the situation is negative for the agent, and some (other) agent is responsible for the negative situation. This semantic information can be used to restrict the set of desires that have to be considered by the planning process.

A number of models have been proposed that describe the control function of emotion, with varying levels of architectural detail (e.g [Sim67, FM93, SC81, OJL85]). In one of the first proposals of an interrupt system for an abstract reasoner [Sim67], Simon lays down similar requirements and functions as we have in this thesis. In an extension of Simon's theory, Sloman [SC81] distinguishes two forms of information processing: a highly parallel, pre-attentive and automatic motive generation process, and attentive, resource-bounded 'motive-management' processes that are largely serial of nature.

The motive generation process produces a number of motivators, which are like emotional states. These motivators have a certain level of insistence for disrupting attentional processing. Sloman argues that attentive processing should be open to interruption by motivators, but at the same time should not be interrupted too often. This balance is dependent on the current state of the world, and Sloman proposes to include a context-sensitive variable threshold interrupt filter in the architecture, in order to manage the interrupts.

I agree with Sloman that not every emotional response should cause the agent to reconsider one or more of his current intentions. Intuitively, an agent should replan his actions only when a significant *change* in the environment has occurred. Since an agent's emotions paint a rough subjective picture of the state of the world, it can be expected that an important change in the world will lead to a significant change in the agent's emotional state. We can use this change in emotional state to trigger a new round of reasoning about intentions.

3.2 A process model of emotion appraisal

Up to this point I have mainly focused on the control function of emotion and have not specified how artificial emotions could be generated. Given the fact that emotions play a role in attention-regulation, it is important that the emotion appraisal process is fast. On the other hand, it has been widely argued that emotions also play a motivational role and as such, they should be as information-rich as possible. Smith et al. [SKS96, SK99] present a process model of emotion appraisal that attempts to satisfy these competing constraints.

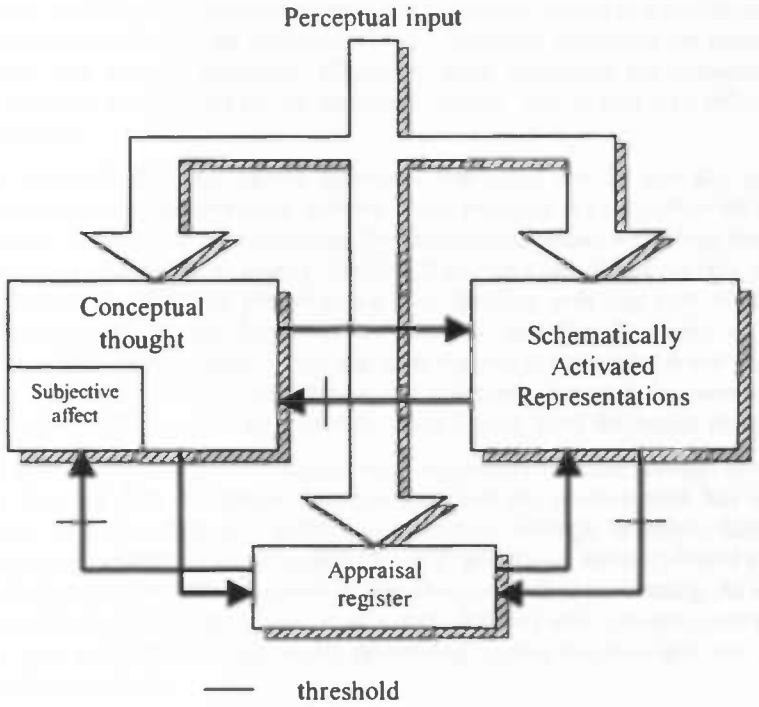


Figure 3-1: Simplified model of emotion appraisal

Figure 3-1 presents a simplified version of the Smith et al. model. The key way in which their model addresses both motivational and attention-regulatory aspects of emotion is in proposing that there are multiple appraisal processes that occur in parallel. One mode of processing is responsible for a rough, and in important ways incomplete, emotion appraisal that is memory-based. Whenever this rough appraisal exceeds some threshold a slower and more deliberate appraisal process is initiated. This mode of processing is more flexible and is able to analyze emotion-eliciting situations more thoroughly.

In order to coordinate the different forms of processing and integrate their respective results, Smith et al. include an *appraisal register* in their model. This register continuously monitors for, and is responsive to, appraisal information from a number of different sources. The person's emotional state is determined on the basis of the appraisal information that has been detected by the appraisal register. The register does not actively perform appraisals in the sense that it is evaluating the person's relationship to the current environment; this is a function that is performed by the two forms of processing. Instead, the function of the register is to integrate appraisal information that is received from various sources and, on the basis of the integrated appraisal information, to initiate processes that generate the person's overall emotion response, including a subjective feeling state, changes in action readiness and a variety of physiological responses.

As can be seen in the figure, the appraisal register receives input from three sources: *direct sensory input*, *schematically activated representations* and *conceptual thought*. It is assumed that some forms of perception have a direct effect on the appraisal register. For instance, the bodily sensation of pain directly leads to a negative emotional appraisal, without the need for any kind of interpretation of the sensation by either automatic or deliberate processing.

Schematic processing is fast, automatic and memory-based, and involves spreading activation and priming. Memories of prior experiences are activated as a result of perceived similarities with the person's current circumstances, or as a result of associations with other, already activated, memories. The similarities can be both perceptual and conceptual in nature. In this way, both highly perceptual cues, such as sounds, colors and smells, and more abstract conceptual ideas can activate memories. If any appraisal meanings are associated with these memories, they too are activated. Whenever these meanings are activated to a sufficient degree, they can be detected by the appraisal register, and in this way influence the person's emotional state.

It is assumed that full-blown appraisal meanings can be quickly activated through automatic processes like spreading activation and priming. As a result, even though schematic processing is a fast mode of processing, the resulting appraisal meanings that are detected by the appraisal register can be highly detailed. They may not fit the current situation too well though, especially when the person is not very familiar with that type of situation. Another important assumption that Smith et al. make is that the threshold at which appraisal information becomes available to the appraisal register is somewhat lower than the threshold, at which activated memories and associated appraisal meanings become available to focal awareness and working memory, and thus to the higher-level deliberate thought processes.

This assumption is important for attention-regulatory reasons: through this assumption it is possible that the first indication a person gets that the environment has changed in some significant way, is a change in subjective emotional feelings. In effect, Smith et al. propose that a change in emotional state, caused by rapid, automatic, memory-based processing should lead to the penetration of this emotional state into conceptual processing. As a result, the agent can then analyze the current situation in a more focused way through conceptual processing. Finally, it is assumed that schematic processing occurs continuously and in parallel with conceptual processing.

Conceptual processing is a relatively slow, controlled and resource-intensive mode of processing. Smith et al. argue that conceptual processing is somewhat more limited in the types of information that are available to it. They propose that whereas schematic processing can operate on any kind of information that can be represented in memory, conceptual processing can only operate on information that has been semantically encoded in some way. Thus, sensations, sounds and images are not readily available to conceptual processing unless they have been associated with some sort of semantic information.

These limitations notwithstanding, conceptual processing is argued to be extremely important in the appraisal process for a number of reasons. First, since schematic processing is largely data-driven, only fairly constant relations in the environment can be reliably detected. By performing attentive conceptual processing, the emotion-eliciting situation can be analyzed more thoroughly, and the resulting reappraisal of the situation can be passed on to the appraisal register. In this way, initial appraisals that have been elicited through schematic processing may be modified to provide a more appropriate evaluation and a more fitting emotional response.

Second, Smith et al. argue that the results of this conceptual appraisal process can be used to fine-tune association strengths in associative memory. In addition, when a situation occurs with which the person is not familiar, the results of conceptual processing may be added to memory to ensure that the next time, schematic processing can provide a fast evaluation.

The Smith et al. model of emotion appraisal shows great promise as a mechanism for balancing reactive and goal-oriented behavior in an adaptive way. The model is essentially a model for emotion appraisal in human beings, but I believe many of the issues that the model addresses are also relevant in agent systems, especially rational agent architectures.

3.3 Extending the BDI architecture

The BDI paradigm is mainly concerned with intentions and the processes of adopting, revising and achieving them. Practical systems that have been developed on the basis of this theoretical research also address the issue of responsiveness: both PRS and dMARS make use of priority schemes to ensure that perceived events are noticed by the BDI interpreter almost instantly. Depending on the relative priority of the newly perceived event, the interpreter handles the event immediately, or deals with other, higher-priority issues first.

In a typical application, incoming external events are produced by a separate perception process, which runs concurrently and independently from the BDI architecture. These external events contain information about a specific change in the state of the world, or the perception of some object. While this is clearly useful information, an agent should not reevaluate his current intentions every time such a change occurs. Especially in applications where the world is complex and the perception system needs to process many external events, there is a need for a mechanism that can determine when it is appropriate to reevaluate the agent's current intentions.

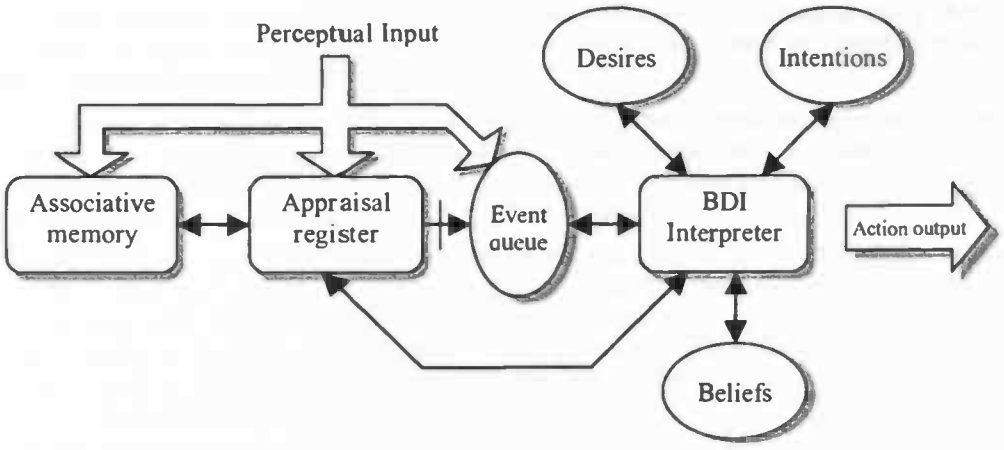


Figure 3-2: Extended BDI Architecture

I have argued above that the mechanism that was presented by Smith et al. satisfies this requirement, and I propose to extend the BDI architecture by adding a mechanism based on Smith et al.'s model to the architecture. The main objective in extending the BDI architecture is to protect ongoing means-ends reasoning by only allowing subjectively important changes in the agent's (internal or external) environment interrupt reasoning. Figure 3-2 presents this extended BDI architecture. The abstract BDI architecture, as was described in section 2.1 can be clearly distinguished in the figure. Two new functional modules have been introduced into the architecture: an associative memory and an appraisal register.

The Smith et al. model consists of three functional units that can also be found in the extended BDI architecture: Smith et al.'s schematically activated representations correspond to the associative memory; the appraisal register remains the same; and the conceptual thought module is represented by the original BDI architecture.

The associative memory performs low-level schematic processing of the form that was described by Smith et al. An associative memory with spreading activation is very suitable for meeting the requirements for fast, context-sensitive appraisal. As a result of spreading activation in the associative memory, appraisal meanings are activated. As elements in associative memory become more activated — as a result of input from perceptual processing, or as a result of current intentions, beliefs or desires — the appraisal meanings that are associated with these elements become more activated as well. Once these meanings have become activated to a sufficient degree, they become available to the appraisal register.

In addition to input from the associative memory, the appraisal register receives input directly from perceptual processes, and from the BDI part of the extended architecture as well: the results of deliberate emotion appraisal from the original BDI architecture. This deliberate appraisal process can best be modeled after the cognitive appraisal theory by Ortony, Collins and Clore [OCC88] that was described in section 1.3.

I have argued that since emotions paint a rough subjective picture of the state of the world, a significant change in emotion signals that the world has changed in some way that is important to the agent. Therefore, I propose that the appraisal register should compare the current emotional state with the agent's emotional state at the last time the agent reasoned about which intentions to adopt.

Whenever the current emotional state differs significantly from that previous emotional state, the register produces an interrupt signal, forcing the BDI part of the architecture to deliberately appraise the situation at hand, and to re-evaluate its current intentions. Depending on the semantic content of the new emotional state resulting from the deliberate appraisal (e.g. fear, anger, joy) this reasoning process involves reconsidering current intentions (e.g. due to frustration or anger), or analyzing the environment thoroughly to find an opportunity or a threatening situation (e.g. due to terror or fear).

The proposed extension to the BDI architecture ensures that current intentions are only re-evaluated when there has been a significant change in the environment, provided that this is correctly reflected by significant change in emotional state. Consider the following example as an illustration of the working of the extended BDI architecture:

Agent X 'lives' in an artificial world, in which his main goal is to collect food parcels. In this world, there are two types of other agents: blue agents that upon encountering agent X, spontaneously offer food to the agent; and red agents that steal food from the agent. As a result of training or design, the perception of red agents causes the associative memory to activate a negative emotional appraisal, while the perception of blue agents activates a positive emotional appraisal. Now consider the following scenario:

Agent X is minding his business, collecting food parcels when he sees a red agent. As a result of this perception, the associative memory outputs a negative appraisal, 'fear', to the appraisal register. As a result of the shift in the appraisal input to the appraisal register, the register interrupts current plans (e.g. collect-food). Agent X now deliberatively appraises the situation, and as a result decides to abort his current plans, and start a flee-plan. As a result of fleeing, the agent moves away from the red agent, until it is no longer in sight. At this point, the associative memory outputs a neutral appraisal, which causes the appraisal register to interrupt the flee-plan and re-evaluate the situation, etc.

CHAPTER 4

A PRELIMINARY IMPLEMENTATION

I believe the extended BDI architecture has a great potential for dealing with complex environments, in which many different aspects of the environment can be important to the agent at a given time. However, this potential has to be demonstrated by a concrete implementation. In this chapter, I describe the first steps toward an implementation of the full model as it was presented in the previous chapter.

4.1 System setup

This first implementation uses PAC, a system that includes facilities for reasoning and a simulated three-dimensional environment, in which agent designs can be implemented and evaluated. The PAC system, short for *Personality and Cognition*, was developed at RMIT University in order to test the appropriateness of implementing emotions in agent systems, with a special emphasis on computer animations [PT97]. It consists of two distinct components: dMARS for the abstract reasoning, and a simulated three-dimensional environment called *AgentWorld*. Figure 4-1 gives an overview of the system.

Like many other agent applications, an agent represented in PAC consists of three functional elements: a *perception* component, a *goal-oriented planning* component, and an *action* component. Both an agent's perception and action components are part of AgentWorld, while dMARS is responsible for goal-oriented planning. The perception component consists of a number of perception routines, which specify which changes in the environment are important enough to be passed on to the goal-oriented planning system. For instance, for an agent in a robotic soccer application, the perception system may contain routines for detecting the position of the ball, other players, the lines of the field and the goals. Perception routines actively monitor aspects of the simulated world by directly accessing the data-structures representing the three dimensional world.

Information is passed from the perception component to the goal-oriented planning component through messages. Such a message typically contains information about a *change* in what the perception routines perceive, since the symbolic representations in the planning system need only be updated when there has been a change.

PAC uses dMARS for the goal-oriented planning. As a result of the reasoning process, the planning system sends messages containing action commands to the communications agent in AgentWorld, which passes the message on to the representation of the agent in AgentWorld. Action command messages specify which action should be executed, and also how the new action should affect already running actions. The action component consists of a number of action routines, each implementing a certain low-level behavior of the agent (e.g. move-to-position, shoot-ball and pass-ball for a robotic soccer player). At this low level of abstraction, it is important not to execute two action routines that are inconsistent (e.g. moving in two different directions at the same time).

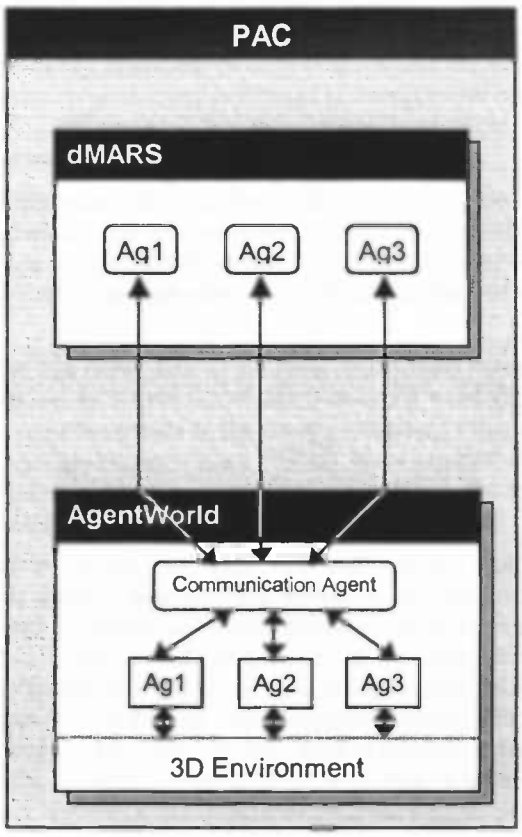


Figure 4-1: Overview of the PAC system

Consistency between action routines is maintained by having each action routine claim the body parts of the agent that are involved in the action. Once a body part has been claimed by an action routine, another action routine that requires the use of the same body part cannot be executed, unless the planning system has specified in the action command that the action should abort any conflicting actions.

Whenever a running action routine finishes, feedback is sent to the agent's dMARS component specifying whether the action succeeded, failed, or was aborted by another action routine. The dMARS component can use this information to search for alternative means for achieving the intention in question, and when there are no more alternatives, drop the intention.

4.2 A shallow implementation

In the presentation of the extended BDI architecture in chapter 3, I have mainly focused on the functional aspects of the model. Creating an implementation of the model introduces a number of additional research questions that need to be addressed before the effectiveness of the model can be demonstrated. For example, what should the associative memory look like? Which learning algorithm should be used? What's the best representation for the agent's emotional state? Addressing all these issues is beyond the scope of a master's thesis and I have chosen to focus on the relationship between the appraisal register and the BDI architecture.

I believe the first logical step toward an implementation of the full model, is to demonstrate that emotion-based interrupts of current intentions are feasible. In order to show this, I have decided to focus on generating interrupts in the appraisal register and dealing with the interrupts in the BDI architecture. For this approach to make sense, the experiment scenario needs to fulfill a number of requirements.

First, the scenario needs to be simple. If the scenario is simple, the associative memory can be replaced by a small number of simple rules that generate rudimentary artificial emotions. In a simple scenario it is easy to identify emotions that should be generated by the associative memory. The appraisal register can generate interrupts on the basis of these emotion-eliciting rules.

Second, there shouldn't be exceptions to the emotion-eliciting rules. With this I mean that there should be no situation in which the emotion-eliciting rules do not apply. If no such exceptions exist, we can trust the results of the emotion-eliciting rules and there is no need for a deliberate emotion appraisal process. Eliot [Ell92] has already demonstrated that such a process if feasible and by not including it, we can focus better on the way an interrupt should influence the BDI architecture.

In the next section, a scenario is described that satisfies both of the requirements mentioned above. Given such a scenario, the appraisal register combines input from the emotion-eliciting rules and generates an overall emotional state. In this first implementation, very rudimentary emotions suffice, so I have chosen to represent only the positive-negative dimension of emotions. The agent's emotional state is represented as a value between –1 and 1. In the previous chapter, I have argued that a significant change in emotional state should lead to an interrupt. Therefore, a record should be kept of the emotional state at the time of the previous interrupt. Finally, a parameter must be set that reflects when a change in emotional state is significant.

When the appraisal register generates an interrupt, the BDI agent should suspend his current intentions, and evaluate the current situation and the emotional state. On the basis of this information, the agent has to decide which plans need to be stopped, and which plans need to be started. To realize this, a number of plans have been written that together make up the control structure of the extended BDI architecture. Figure 4-2 gives a functional overview of the control structure.

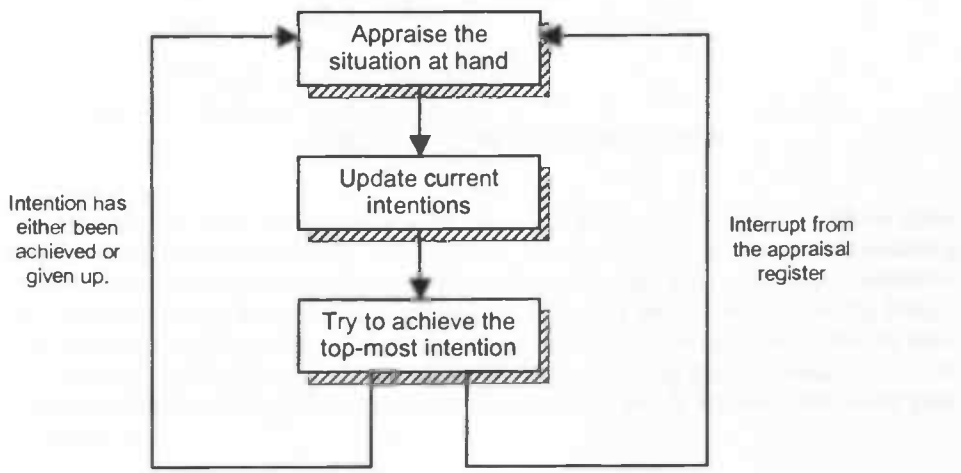


Figure 4-2: The control structure

4.2.1 An introduction to dMARS plans

Before I describe the dMARS plans that implement the control structure in detail, a short introduction to dMARS plans is in place. As was seen in chapter 2, every dMARS plan consists of at least an invocation condition and a body. The invocation condition describes which event has to take place before the plan can be considered. The plan's body describes what has to be done to complete the plan. In addition to these two required elements, a plan can also have a *context condition*, a *maintenance condition* and a *priority level*. The context condition queries the agent's database for some variable at the time the plan is adopted, in order to make the value of the variable available in the plan. The maintenance condition contains a proposition that must remain true while the plan is being executed, otherwise the plan will be aborted. The priority level describes how important the plan is relative to other plans. In dMARS, a plan that has a lower priority value is more important than a plan with a higher priority value.

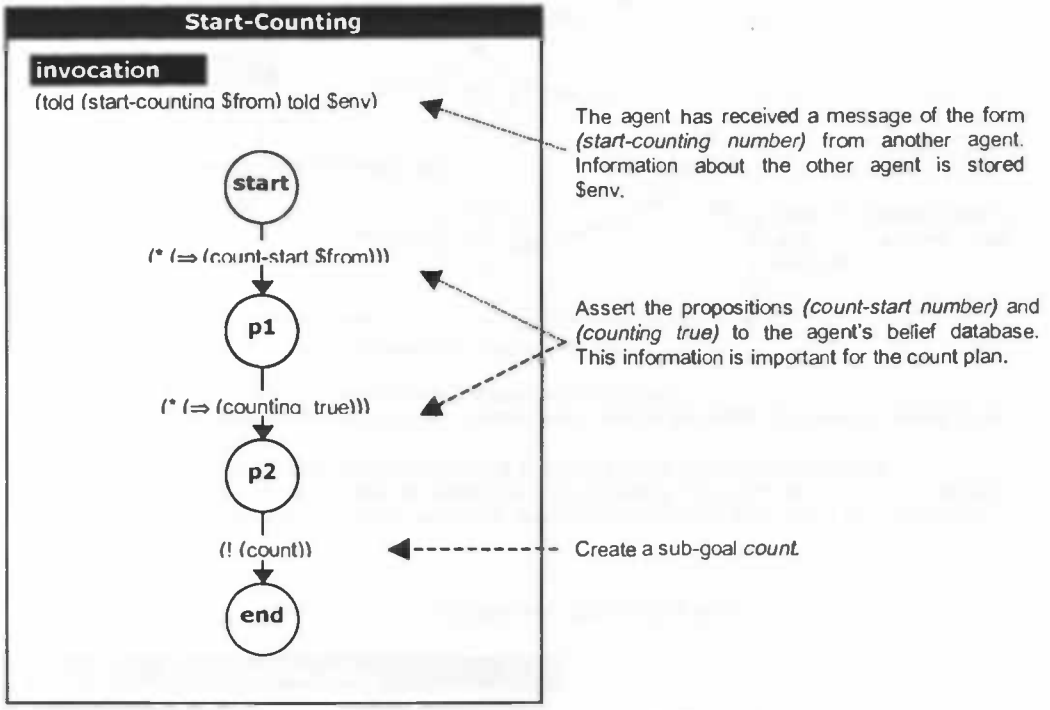
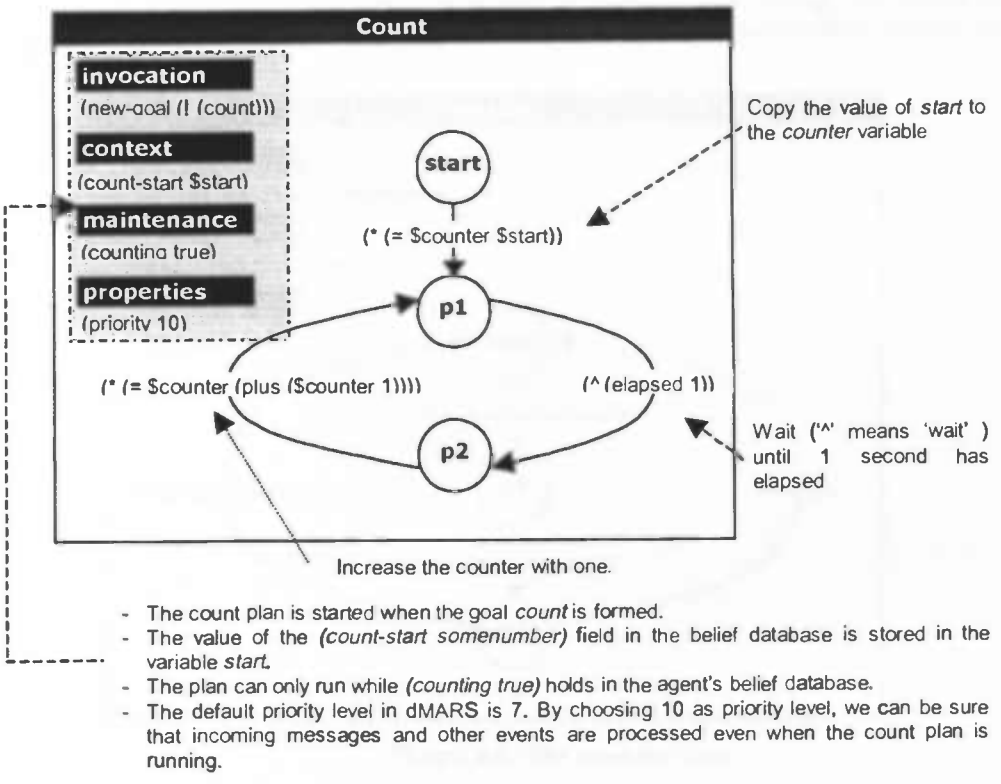


Figure 4-3: The start-counting plan

As an example of these concepts, consider the plans in figures 4-3 to 4-5. These three plans simply make the agent count the seconds from the time that he is told to start counting until the time he is told that he should stop counting. The first plan, *start-counting*, is activated when the agent receives a message from another agent telling him to start counting from a certain number (e.g. *'(start-counting 24)'*). The agent 'remembers' the number it should start counting from and the fact that he has started counting by asserting this information into his belief database. Finally, the agent creates the sub-goal count, which activates the count plan shown in figure 4-4.

The count plan in figure 4-4 reads the value of *(count-start somenumber)* from the belief database and adds 1 to that number every second. Notice that the count plan does not have an end state. The only way to stop this plan is by making the maintenance condition false. This is exactly what the stop-counting plan in figure 4-5 does. The plan is activated when a message (stop-counting) is received from another agent. As a result the agent stores *(counting false)* in the belief database. This in turn makes the count plan's maintenance condition false, which causes the plan to be aborted.



- The count plan is started when the goal *count* is formed.
- The value of the *(count-start somenumber)* field in the belief database is stored in the variable *start*.
- The plan can only run while *(counting true)* holds in the agent's belief database.
- The default priority level in dMARS is 7. By choosing 10 as priority level, we can be sure that incoming messages and other events are processed even when the count plan is running.

Figure 4-4: The count plan

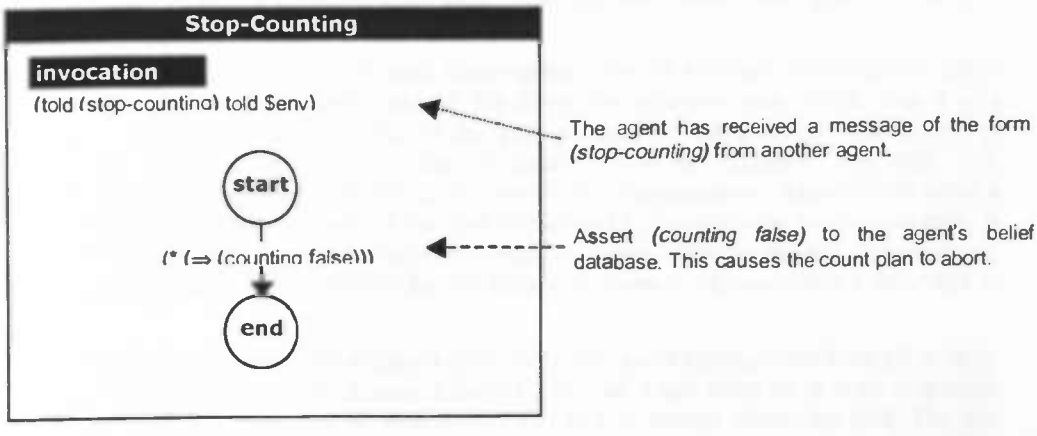


Figure 4-5: The stop-counting plan

4.2.2 The dMARS plans implementing the control structure

Having provided a background in dMARS plans, I can now present the dMARS plans that implement the interrupt-based control structure. The control structure is implemented by the four plans shown in figures 4-6 to 4-9. The main control loop is implemented in the *top-loop* plan in figure 4-6. This plan should be started when the simulation in the PAC system starts, and should run until the simulation ends. In the current implementation, this is achieved by having an init-plan assert *(simulation-running true)* at the simulation startup, and having an end-plan assert *(simulation-running false)* when the simulation ends (neither of these plans are shown here).

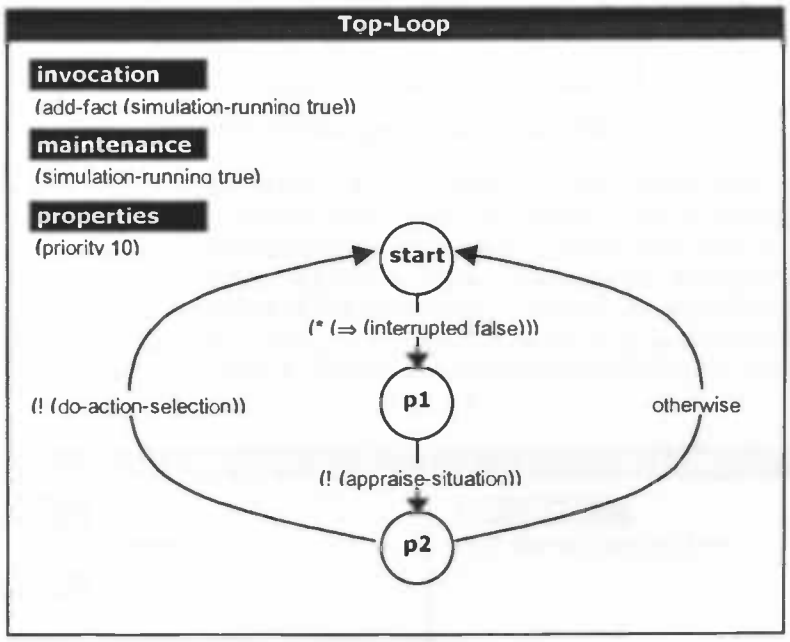


Figure 4-6: The top-loop plan

The *top-loop* plan runs at priority level 10 to ensure that incoming messages from AgentWorld will be processed and handled while the plan is running. All sub-plans inherit the priority level of the parent plan, so in this case the *do-action-selection* and *appraise-situation* plans run at priority level 10.

At the first step in the *top-loop* plan *(interrupted false)* is asserted to the agent's belief database. This proposition is important for the *do-action-selection* plan, which uses it as a maintenance condition. The next step in the plan creates the sub-goal *appraise-situation*, which causes the *appraise-situation* plan in figure 4-7 to be executed. This plan tells AgentWorld to send it information about the world. As a consequence, AgentWorld sends a message containing information about the state of the world. Another plan (not shown here) is responsible for updating the agent's belief database. In future implementations, the *appraise-situation* plan will also be responsible for the deliberate emotion appraisal that I described in chapter 3.

At the next step in the *top-loop* plan's cycle there are two outgoing links from p2 to start. In dMARS, an '*otherwise*' link is only followed when all other links have been attempted unsuccessfully. In this case, the *do-action-selection* link is always attempted first. The *do-action-selection* sub-goal is responsible for deciding on an action and executing it. Regardless of the success or failure of the *do-action-selection* sub-goal a new cycle should be started, hence the *otherwise* link. In this way, the *top-loop* plan stays active until the simulation ends.

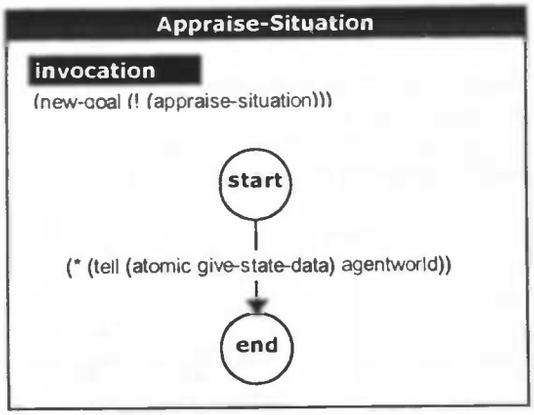


Figure 4-7: The appraise-situation plan

The *do-action-selection* plan in figure 4-8 does little of it's own, except make sure that any action that is being executed is aborted when an interrupt occurs. The plan creates a sub-goal *perform-action,* which is the invocation condition of all individual action plans. So, if an agent can perform three possible actions, each of those three action plans has *(new-goal (! (perform-action)))* as its invocation condition. When an interrupt is received, the interrupt plan in figure 4-9 changes *(interrupt false)* to *(interrupt true)*, and as a result the *do-action-selection* plan is aborted, and all it's sub-plans with it. Control is returned to the *top-loop* plan and a new cycle of *appraise-situation* → *do-action-selection* is started.

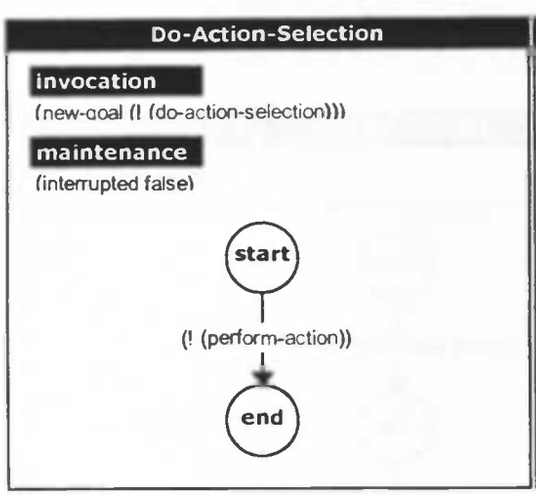


Figure 4-8: The do-action-selection plan

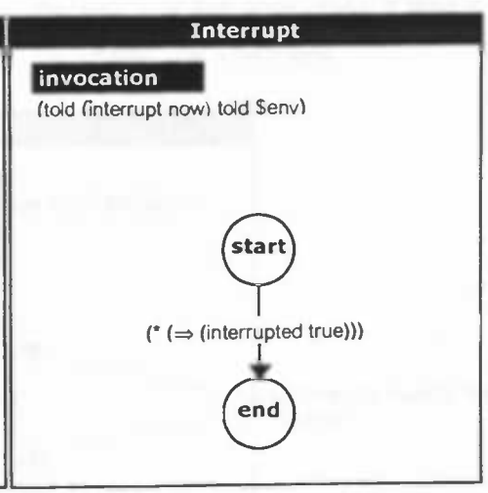


Figure 4-9: The interrupt plan

4.3 The Scenario

As was indicated above, the main goal of this first simulation scenario is to demonstrate that the control structure as described in the previous section works. For this purpose I have chosen to keep the simulation scenario as simple as possible. The agent of interest in this first simulation is Dogbert, a dog. He lives in a world where food parcels appear at random positions, and being a good dog, Dogbert collects as many food parcels as he can. However, in this simulation another dog, Cujo, steals food parcels from Dogbert when he can. Therefore, Dogbert should try to avoid Cujo as much he can.

In this scenario, Dogbert needs only a few perception routines. I have equipped Dogbert with a 'food parcel detector', and a 'Cujo detector': perception routines that sense whether a food parcel and/or Cujo are in sight. Dogbert's emotional state is determined as the weighed sum of two emotion-eliciting rules relating to the proximity of food and Cujo:

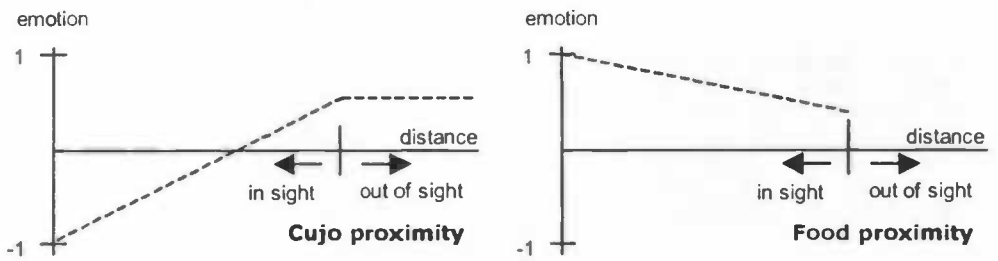


Figure 4-10: The emotion-eliciting functions

Dogbert can perform four actions: *flee-from x, move-to x, pick-up x* and *wander*. Each of these actions is implemented in AgentWorld, as are the emotion-eliciting rules and the appraisal register (i.e. overall emotional state in the current implementation). The following

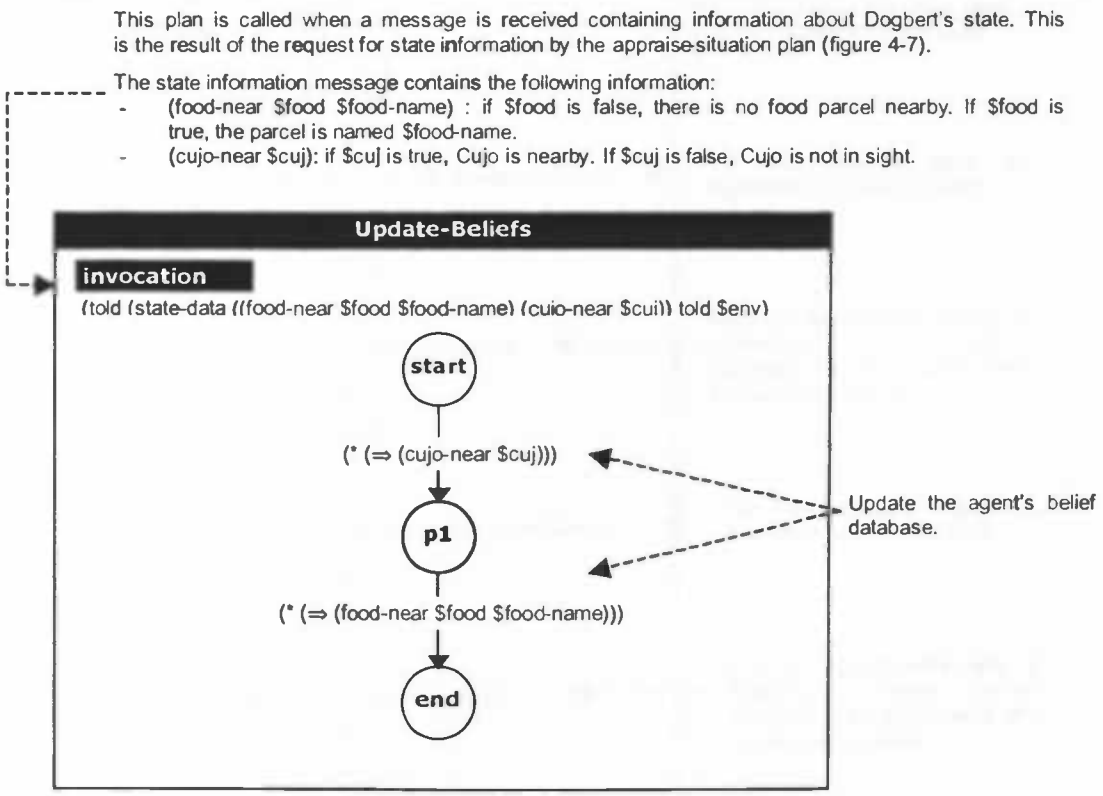
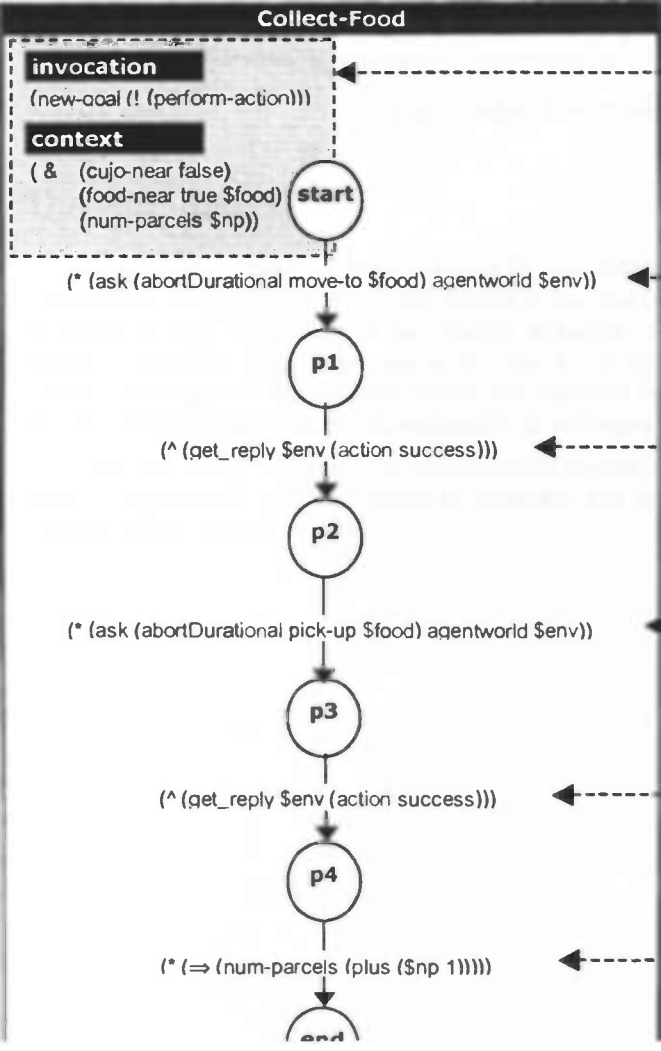


Figure 4-11: The update-beliefs plan

plans were added to the general control plans described in the previous section:


Collect-Food
invocation
(new-goal (! (perform-action)))
context
(& (cujo-near false)
(food-near true $food)
(num-parcels $np))
start
(* (ask (abortDurational move-to $food) agentworld $env))
p1
(^ (get_reply $env (action success)))
p2
(* (ask (abortDurational pick-up $food) agentworld $env))
p3
(^ (get_reply $env (action success)))
p4
(* (⇒ (num-parcels (plus ($np 1)))))

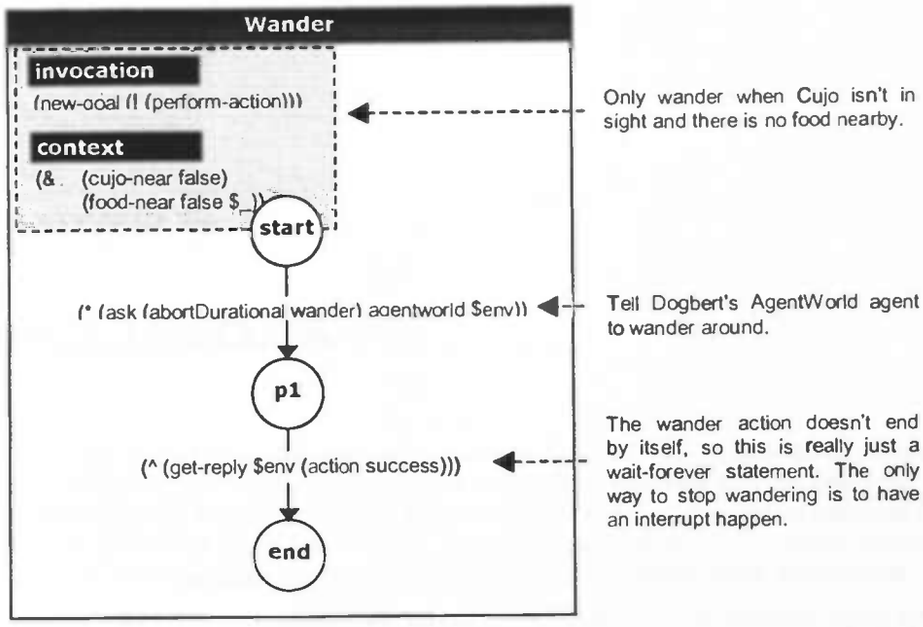


Figure 4-14: The wander plan

4.4 The Simulation

Unfortunately, I have not been able to fully implement all of Dogbert's action routines in AgentWorld, and therefore have been unable to run the simulation scenario described above. However, I have implemented the control structure described in section 4.2.2 and all Dogbert's dMARS plans described in section 4.3. I have tested these plans by manually sending messages to the Dogbert agent, for example telling him to interrupt his current actions, or telling him that he has succeeded in moving to a certain food-parcel.

In this way, I have been able to verify that the control structure that is implemented by the plans in figures 4-6 to 4-9 functions as expected, and that the BDI architecture can use an interrupt-based control structure.

CHAPTER 5

CONCLUSIONS

In this thesis I have presented an extension to the BDI architecture based on psychological and computational models of emotion. I believe that emotions play an important role in coordinating our thought processes, and that some of the characteristics of emotions can also be very useful for agent architectures. My work has focused on the control dimension of emotions: our emotions often force us to attend to certain aspects of the environment.

In chapter 2, I have argued that one of the difficult issues in deliberate agent systems is deciding when to abort an intention. An agent shouldn't give up to soon, nor should he be too fanatic about achieving his goals. Moreover, this region between under-committing and over-committing to a plan seems to depend on the agent's context. In a hostile environment an agent should be more careful and give up sooner than in a friendly environment.

I believe that emotions are very important in deciding when to give up and when to persevere, at least for humans. In chapter 3, I have presented an extension to the BDI architecture that is based on Smith et al.'s model of emotion appraisal ([SK99]). The main feature of the extension of the BDI architecture is that plans are interrupted when there has been a significant change in the agent's emotional state, which should reflect a subjectively significant change in the state of the world.

The first step toward an implementation of the model is to determine how the BDI architecture can be adapted to accommodate interrupts, and to show that, given a plausible emotion-appraisal process, the resulting behavior is sensible. For this purpose, I have designed a simulation scenario and have implemented the interrupt-based control structure in dMARS. I have been able to show that dMARS can accommodate an interrupt-based control structure, but have been unable to run the simulation due to time limitations.

5.1 Topics of Future Research

My focus in this thesis has been on the control function of emotions and on how the BDI architecture can be extended to accommodate an interrupt-based control structure. I have given only relatively high-level descriptions of how the two different types of emotion appraisal processes (automatic and deliberate) can be implemented. There are a lot of open issues that need to be addressed before the model I have presented can be fully implemented.

Aside from these issues, adding a model of emotion to any agent architecture opens up a number of further possibilities. I believe that an emotional state should influence planning processes not only by interrupting current planning, but also by restricting the amount of planning performed. For instance, when you are being attacked, you should minimize planning and act as soon as possible. I believe that emotions are partly responsible for this balance between planning and acting in humans.

Emotions also play a motivational role in humans, and this is another interesting topic that should be explored further for agent systems. Frijda's [FS87, FM93] *concerns*, described in

section 1.3 emphasize the motivational role of emotions, and I think it would be interesting to see how emotions can serve as motivations within the BDI architecture. Also, using emotions as motivations makes the process of designing plans more natural since the plan's motivation corresponds more closely to our own motivations.

Currently, most agent systems do not incorporate learning mechanisms: the plans are hand-made by the programmer. I believe that the valence dimension of emotions is very suitable as a reinforcement value in reinforcement learning. When an agent is faced with a problem that strongly 'distresses' him, a plan that solves the problem could receive a strong positive reinforcement. In this way, the agent could learn from subjectively important situations much easier than not-so important situations.

Finally, another promising field where emotions play an important role is in believable agents: agents that appear to be 'real' creatures. An agent that has an emotional state can also display his emotional state to another agent or a user. Being the social creatures that we are, it is more natural to communicate with such an agent. The potential of emotional agents is enormous. They can be used in a great variety of applications, ranging from computer assisted learning to characters in computer games.

I believe our emotions play much more important role in our everyday life that is often thought, and that we can learn a lot about information processing from the study of emotions. Much research into the nature of emotions is needed, but I believe artificial emotions have a great potential in agent systems.

BIBLIOGRAPHY

[Bat94] — Joseph Bates. The role of emotion in believable agents. Communications of the ACM, Special Issue on Agents, July 1994.

[BIP88] — Michael E. Bratman, David J. Israel, and Martha E. Pollack. Plans and resource-bounded practical reasoning. Computational Intelligence, 4(4):349-355, 1988.

[Bra87] — Michael E. Bratman. Intentions, Plans, and Practical Reason. Harvard University Press, 1987.

[Bro91] — Rodney A. Brooks. Intelligence without representation. Artificial Intelligence, 47:139-159, 1991.

[CL90] — Philip R. Cohen and Hector J. Levesque. Intention is choice with commitment. Artificial Intelligence, 42:213-261, 1990.

[Cn97] — Dolores Cañamero. Modeling motivations and emotion as a basis for intelligent behaviour. In Proceedings of Autonomous Agents '97. ACM, 1997.

[Dam94] — Antonio R. Damasio. Descartes' Error. Emotion, Reason and the Human Brain. New York:Plenum.

[Den87] — Daniel C. Dennett. The Intentional Stance. MIT Press, 1987.

[dKLW97] — Mark d'Inverno, David Kinny, Michael Luck, and Michael Wooldridge. A formal specification of dMARS. Technical Report 72, Australian Artifical Intelligence Institute, 171 Latrobe Street, Melbourne 300, Australia, November 1997.

[Ell92] — Clark D. Elliot. The Affective Reasoner. PhD thesis, Northwestern University, 1992.

[Fer92] — Innes A. Ferguson. Touringmachines: Autonomous agents with attitudes. Technical Report 250, Computer Laboratory, University of Cambridge, April 1992.

[FM93] — Nico H. Frijda and David Moffat. A model of emotion and emotion communication. In Proceedings of RO-MAN'93: 2nd IEEE International Workshop on Robot and Human Communication, pages 29-34, 1993.

[FN71] — R. E. Fikes and N. Nilsson. Strips, a new approach to the application of theorem proving to problem solving. Artificial Intelligence, 5(2):189-208, 1971.

[FS87] — Nico H. Frijda and Jaap Swagerman. Can computers feel? theory and design of an emotional system. Cognition and Emotion, 1(3):235-257, 1987.

[GH98] — Sandra C. Gadanho and John Hallam. Emotion-triggered learning for autonomous robots. In Proceedings of the Fifth International Conference on Simulation of Adaptive Behavior, 1998.

[GL87] — Michael P. Georgeff and A. L. Lansky. Reactive reasoning and planning. In Proceedings of the Sixth National Conference on Artificial Intelligence (AAAI-87), pages 677-682, 1987.

[McC79] — John McCarthy. Ascribing mental qualities to machines. In M. Ringle, editor, Philosophical Perspectives in Artificial Intelligence. Harvester Press, Brighton, UK, 1979.

[MPT95] — J. P. Müller, M. Pischel, and M. Thiel. Modelling reactive behaviour in vertically layered agent architectures. In M. Wooldrige and N. R. Jennings, editors, Intelligent Agents: Theories, architectures, and languages, pages 261-276. Springer-Verlag, Heidelberg, Germany, 1995.

[OCC88] — Andrew Ortony, Gerald L. Clore, and Allen Collins. The Cognitive Structure of Emotions. Cambridge University Press, 1988.

[OJL85] — K. Oatley and P. N. Johnson-Laird. Sketch for a cognitive theory of emotions. Technical report, Cognitive Science, University of Sussex, 1985.

[PT97] — Lin Padgam and Guy Taylor. A system for modeling agents having emotions and personality. In Proceedings of IJCAI'97, 1997.

[RG95] — Anand S. Rao and Michael P. Georgeff. Bdi agents: From theory to practice. In Proceedings of the First International Conference on Multi-Agent Systems (ICMAS-95), June 1995.

[Sch64] — Stanley Schachter. The interaction of cognitive and physiological determinants of emotional state. In L. Berkowitz, editor, Advances in Experimental Social Psychology, Vol. I. New York: Academic Press, pp. 49-80.

[SC81] — A. Sloman and M. Croucher. Why robots will have emotions. In Proceedings of the seventh International Joint Conference on Artificial Intelligence, pages 197-202, 1981.

[Sho93] — Yoav Shoham. Agent-oriented programming. Artificial Intelligence, 60(1):51-92, 1993.

[Sim67] — Herbert A. Simon. Motivational and emotional controls of cognition. Psychological Review, 74:29-39, 1967.

[Sin97] — Munindar P. Singh. Commitments in the architecture of a limited rational agent. In Proceedings of the Workshop on Theoretical and Practical Foundations of Intelligent Agents, pages 72-87, 1997. Invited Paper.

[SK99] — Craig A. Smith and Leslie D. Kirby. Consequences require antecedents: Toward a process model of emotion elicitation. In J. Forgas, editor, The role of affect in social cognition. Cambridge University Press, (forthcoming), 1999.

[SKS96] — C. A. Smith, L. D. Kirby, and H. S. Scott. Toward a process model of appraisal in emotion. In Proceedings of the Ninth Conference of the International Society for Research on Emotions, pages 101-105, 1996.

[SL98] — Aaron Sloman and Brian Logan. Architectures and tools for human-like agents. In Proceedings of the 2nd European Conference on Cognitive Modeling, 1998.

[Wri97] — Ian P. Wright. Emotional Agents. PhD thesis, Faculty of Science, University of Birmingham, February 1997.