Representation and Reasoning for Goals in BDI Agents[*]

John Thangarajah **Lin Padgham** **James Harland**

School of Computer Science and Information Technology, RMIT University
GPO Box 2476V, Melbourne, 3001, AUSTRALIA
{johthan, linpa, jah}@cs.rmit.edu.au

Abstract

A number of agent-oriented programming systems are based on a framework of beliefs, desires and intentions (BDI) and more explicitly on the BDI logic of Rao and Georgeff. In this logic, goals are a consistent set of desires, and this property is fundamental to the semantics of the logic. However, implementations based on this framework typically have no explicit representation of either desires or goals, and consequently no mechanisms for checking consistency. In this paper we address this gap between theory and practice by giving an explicit representation for a simple class of desires. The simplicity of this class makes it both straightforward and efficient to check for consistency. We provide a general framework for conflict resolution based on a preference ordering of sets of goals, and we illustrate how different rules for specifying consistent goal sets (corresponding to different preference orderings) relate to existing commitment strategies. We also report on some implementation experiments which confirm that the cost of consistency maintenance is not significant.

1 Introduction

An increasingly popular programming paradigm is that of *agent-oriented programming*. This paradigm, often described as a natural successor to object-oriented programming [Jennings, 2001], is highly suited for applications which are embedded in complex dynamic environments, and is based on human concepts, such as beliefs, goals and plans. This allows a natural specification of sophisticated software systems in terms that are similar to human understanding (and hence can represent idealised human *actors*), thus permitting programmers to concentrate on the critical properties of the application rather than getting absorbed in the intricate detail of a complicated en vironment. Agent technology has been used in areas such as air traffic control, automated manufacturing, and the space shuttle [Jennings and Wooldridge, 1998].

Whilst there are many possible conceptions of agent-oriented programming, one of the most popular and successful such conceptions is the frame work of Rao and Georgeff [Rao and Georgeff, 1991], in which the notions of *Belief*, *Desire* and *Intention* are central, and hence are often referred to as *BDI* agents. Roughly speaking, beliefs represent the agent's current knowledge about the world, including information about the current state of the environment inferred from perception devices (such as cameras or microphones) and messages from other agents, as well as internal information. Desires represent a state which the agent is trying to achieve, such as the safe landing of all planes currently circling the airport, or the achievement of a certain finnancial return on the stock market. Intentions are the chosen means to achieve the agent's desires, and are generally implemented as plans (which may be thought of as procedures which come with pre-conditions (to determine when a plan is applicable) and post-conditions (to state what is achieved upon the successful completion of the plan)).

Rao and Georgeff gave both a logical system incorporating these concepts [Rao and Georgeff, 1991] (a version of temporal logic extended to include the appropriate notions of belief, desire and intension) and an architecture for the executing of programs following the BDI paradigm [Rao and Georgeff, 1992]. As in general an agent may have multiple desires, an agent can have a number of intentions active at any one time. These intentions may be thought of as running concurrently, with one chosen intention active at any one time. Hence an agent's plans may be thought as executing in a singly-threaded concurrent mode.

Whilst classic planning systems (such as STRIPS) may seem similar, such systems are actually significantly more sophisticated. In particular, a balance must be struck between reactive behaviour (i.e. responding to changes in the environment which may affect the applicability and/or outcomes of a given plan) and proactive behaviour (i.e. persisting with the current plan in order to achieve the desired outcomes). In such systems it is entirely possible that a plan can become inapplicable and doomed to failure during its execution because of changes in the environment. The way that a balance is achieved between these two competing demands is to have a library of pre-compiled plans, and that periodically a check is made on the applicability of the plan (or plans) currently executing. At this point it is possible to switch between concurrently executing plans, or to abandon a current plan in favour of another. However, note that in the latter case the former plan is then abandoned and is not re-tried, and so plans only get one chance to succeed. This may be thought of as demonstrating a certain level of trust in the plan; if it fails, then it is presumably due to environmental changes rather than any inherent problem with the plan itself.

In such an environment, conflicts can arise between goals (i.e. goals can be inconsistent with each other) or between plans (such as two plans needing exclusive access to a particular resource). In general such confliicts are impossible or infeasible to detect in advance, and hence run-time techniques are required in order to detect and resolve such conflicts.

The main technical contribution of this paper is the definition of a number of ways of maintaining goal consistency and the linking of these to commitment styles, along with an empirical evaluation of the cost of these mechanisms. A general frame work is also presented which allo ws for additional commitment styles to be defined.

This paper is organised as follows. In Section 2 we give some background on BDI agents and in Section 3 we

[*]We would to thank Agentis International and the ARC (under grant C49906798, "*Real-time reasoning about interacting activities in agent systems*") for their contribution to this work.

discuss the issue of representing goals (and hence being able to analyse the source of conflicts). Section 4 deals with rules for determining consistent sets of goals and in Section 5 we present some results on the efficiency of the implementation of these rules. Finally in Section 6 we present our conclusions and suggestions for further work.

2 BDI Agent Systems

Whilst systems claiming to be based on agent technology abound[1], there is a growing consensus that an agent system should at least possess the properties below:

- pro-activeness : the agent has an agenda to pursue and will persist in trying to achieve its aims
- reactiveness : the agent will notice and respond to changes in the environmentk
- autonomy : the agent will act without necessarily being instructed to tak e particular steps
- situated : the agent both influences and is influenced by the environment around it

Other possible attributes of agent systems include being *social*, (i.e. teaming up with other agents in order to achieve common goals), *learning* (i.e. taking note of previous actions and adjusting future actions accordingly), and *rationality*, (i.e. working to achieve its aims, and not working against them). The BDI frame work can be thought of as a technical realisation of the desirable attributes of an agent system, and their proactive (or goal-directed) nature is one of their key aspects (especially when compared to more mainstream computing paradigms). However, of the “B”, “D” and “I”, in BDI systems, only the beliefs and the intentions have an explicit representation. Desires (more or less synonymous with goals) if represented at all, have only a transient representation as a type of event. Goals play a central role in some of the properties of rational systems as described by BDI theories. However their lack of explicit representation make it difficult or impossible to realise the theoretical properties within implemented systems.

Goals are essentially a partial state of the world which the agent has decided to attempt to achieve. They are some what different to desires in that desires, following Bratman [Bratman, 1987], are unconstrained, may well conflict with one another and may not even be achievable. Goals on the other hand represent a certain level of commitment; the agent has decided to act in such a way as to attempt to realise its goals.

Rationality seems to require that the set of goals an agent is pursuing be consistent. A rational agent does not simultaneously pursue a goal to spend Christmas on a deserted beach near Broome and to spend Christmas with relatives in Melbourne. Indeed the formal theoretical frame work of Rao and Georgeff which claims to be a basis for implemented BDI systems, requires that goals are consistent for the theory to be valid. However, in order to ascertain if a new potential goal is in conflict with a goal one is already pursuing, it is necessary to have some information about the current goals. This is not available in current BDI systems such as dMars [AAII, 1996], JACK [Busetta et al., 1999, Ltd, 2000], JAM [Huber, 1999], etc. In addition to detection of goal conflict there is a need for some rational policy to determine which goal will be pursued in case of conflict.

Another important issue related to representation of goals is that in many current BDI systems, if it is not possible to immediately form an intention towards a goal i.e. if there is no plan which can be attempted given the current state of the world then the goal is simply dropped. It certainly seems more reasonable that the agent have the ability to ‘remember’ a goal, and to form an intention regarding how to achieve it when the situation is conducive to doing so.

[1]Unfortunately, buzzword overdoses are all too common.

Although goals are an integral part of a large body of theoretical work, there is almost no work that looks in any detail at the representation of goals or at rational policies for maintenance of consistency amongst goals. Hindriks et al.[Hindriks et al., 2001] have presented a programming language called GOAL (for Goal Oriented Agent Language) which does incorporate declarative goals and attempts to bridge the gap between theory and practice. However they allow inconsistent goals and thus do not address issues of rational choice between mutually inconsistent goals. Their goals are more closely related to Bratman’s desires, as influencers rather than controllers of agent action. Bell and Huang [Bell and Huang, 1997] do require that an agent believes each of its goals to be jointly realisable with all other goals considered to be more important. They use preferential entailment to formalise the rational revision of goals and goal hierarchies. Our approach is however more easily integrated into existing implemented systems and is demonstrably efficient, whereas it is questionable that Bell and Huang’s approach could be directly implemented in an efficient manner.

The main technical contribution of this paper is the definition of a number of ways of maintaining goal consistency and the linking of these to commitment styles, along with an empirical evaluation of the cost of these mechanisms. A general frame work is also presented which allows for additional commitment styles to be defined. The broader context of this work is the development of a fine-grained framework for reasoning aboutvarious aspects of agents (including, plans, actions, goals, beliefs, and intentions) which includes a proof-theoretical comonent (and hence specific rules of inference). This, we believe, will provide better support for agent systems than the current BDI frame work.

3 Representation

In order to establish whether two goals are consistent we require some representation of the partial state which will hold when each goal is achieved, and a mechanism to determine whether there are possible worlds where these two partial states co-exist. In general we may want to represent further information about goals (such as beliefs about their possibility and so forth), but that is beyond the scope of this paper.

We choose a simple language for expression of partial state in order to facilitate efficient algorithms for consistency checking. The common use of these systems in real-time environments means that efficiency is vital. Nevertheless we believe that this language is sufficiently expressive for a number of applications.

Our goals states are conjunctions of *basic formulae*, as defined below.

Definition 1 *A* basic formula *is either a propositional letter or its negation (for boolean-valued attributes) or a simple constraint (for numeric-valued attributes).*

A simple constraint *is of the form* **Attribute Relation Number**, *where* **Attribute** *is (in logical terms) a variable,* **Relation** *is one of* $=, <, \leq, >, \geq$, *and* **Number** *is a real number.*

For example, the three formulae

$$buy \wedge (Price \leq 100{,}000)$$

$$buy \wedge \neg buy \wedge (Price \leq 100{,}000)$$

$$buy \wedge (Price \leq 100{,}000) \wedge (Price > 100{,}000)$$

are all goal states (note that the latter two are inconsistent), but the three formulae

$$\forall x\ buy(x) \wedge (Price \geq 100{,}000)$$

$$buy \wedge (Price \geq Reserve)$$

$$buy \vee (Price \geq Reserve)$$

are not goal states.

Intuitively, goal states represent conjunctions of simple state information: a given propositional variable, or a simple statement about a numeric value (i.e. a minimum or maximum). Note that there are no disjunctions present, and that the numeric variables are implicitly existentially quantified, both of which significantly simplify consistency checking. In addition we do not allow constraints of the form $x \neq 2$ for similar reasons, as we shall see below.

Because of the simplicity of this structure, we will often consider goal states as a set of literals, rather than a conjunction per se.

Note that there are only two ways in which goal states G_1 and G_2 can be inconsistent:

1. $p \in G_1$ and $\neg p \in G_2$ for some p, or
2. for some attribute common to G_1 and G_2, the union of the constraints has no solution.

In fact, it is not hard to show the following result.

Proposition 1 *Let G be an inconsistent goal state. Then there exist basic formulae G_1 and G_2 in G such that G_1 and G_2 are inconsistent.*

The proof of this property is straightforward and hence omitted. Note that this property fails in the presence of constraints such as $x \neq 2$. For example, the set of constraints $x \geq 2, x \leq 2, x \neq 2$ is inconsistent (i.e. has no solution); however, every pair of constraints in this set is consistent.

In order to simplify the statement of rules in what follows, we will assume the existence of a predicate Consistent(α, β) which is true precisely when goal states α and β are consistent. If $\neg$ Consistent(α, β) holds, then α and β *conflict*, which we denote by Con(α, β).

We will assume the existence of a priority function *Pr* which indicates the priority of a goal α as a numerical value. Hence goal α is preferable to goal β precisely when Pr$(\alpha) >$ Pr(β). As a first approximation, it seems reasonable to require that if a new goal α is more important than an existing goal β with which it conflicts, then β should be aborted and α pursued. Otherwise, (α is less important or same importance as β), α is not adopted. As we shall see, this is a little too naive in practice, but it is a useful starting point.

We use predicates *Des* and *Goal* in order to state some constraints on the selection of goals. *Des*(ϕ) indicates that ϕ is a desire. *Goal*(ϕ) indicates that ϕ has been adopted as a goal (and hence is required to be consistent with all other adopted goals).

Following Bratman [Bratman, 1987], desires are totally unconstrained, and hence may be inconsistent. In contrast, goals are constrained to be consistent, and in many systems to satisfy other constraints as well (such as being believed possible, and not yet being achieved). In this paper we concentrate on the distinction between desires and goals, and hence on consistency and priorities. In particular, goals may be thought of as desires which have passed through some kind of filter.

Thus we consider a single set of desires, and then we provide rules which constrain the generation of goals from this set of desires.

In a real system, the generation of desires (and hence goals) is dynamic and somewhat unpredictable; changes in the environment, goal priorities and beliefs about the possibility or otherwise of achieving a given goal mean that the set of goals adopted may change over time. Hence the process of generating goals from desires is not simply a matter of selecting a fixed set of goals from a fixed set of desires; the rules for adoption of goals must be capable of taking into account the current context, such as the current set of adopted goals.

Hence we first describe a framework in which such rules about goals can be specified, and then discuss the precise nature of the rules in detail.

A key aspect of this framework is to determine an ordering between goal states, as defined below.

Definition 2 *We define $G_1 \gg G_2$ to hold if one of the following conditions is satisfied:*

1. $\forall G \in G_2\ \exists G' \in G_1$ *such that* $Pr(G') > Pr(G)$
2. $G_1 \supseteq G_2$

We say that goal state G_1 is preferable *to goal state G_2 if $G_1 \gg G_2$.*

Hence a goal state with a single high priority goal is preferable to one with a large number of lower priority goals. Otherwise (i.e. the highest priority goal in each state is at the same level), preference reverts to set inclusion, so that the goal state with the greater number of goals is preferred. This is to capture the idea that we always prefer a maximally consistent goal state over a consistent one.

The next step is to determine what rules should apply to such goal states. The details of the appropriate rules are discussed in the next section. Note that there are (at least) two roles that such rules can play: one as a specification of the adopted goal set, and another as a method of generating such a set. In this paper, we will only pursue the first alternative; we will pursue the second (and related issues) in a subsequent paper.

4 Rules & Commitment

Bratman argues that intentions are attitudes which control, rather than influence behaviour, whereas desires and Beliefs merely influence behaviour. Once a rational agent has adopted an intention, this intention constrains further reasoning and controls agent actions. Intentions imply commitment. Inertia, which according to Bratman is a key component of intention, allows an agent to focus its reasoning and achieve rational behaviour despite computational limitations. The level of inertia associated with intentions has been described by Rao and Georgeff in terms of commitment axioms [Rao and Georgeff, 1991], which determine when an agent is allowed to drop its intentions. Rao and Georgeff define three levels of commitment such that an agent may drop an intention (i) only when it succeeds, (ii) either when it succeeds or it is believed impossible, or (iii) when it succeeds or the Goal no longer exists.

Bratman Identifies two kinds of intention. The first is a top level intention which often arises from a decision to pursue a particular desire. This can be likened to a goal in the BDI theory of Rao and Georgeff. The second is an intention which arises from a step in a partial plan regarding how to achieve a prior intention. Intentions beyond the top level are thus closely related to plans. Typically there are a number of possible plans that could be used to reach a given goal. Once a plan is chosen, the agent commits to a number of intentions, that are the steps in the chosen plan.

We now turn to the issue of describing rules for constructing goals from desires, and hence are similar in spirit to the commitment axioms of Rao and Georgeff. As mentioned in Section 2, in simplest terms, this may be thought of as only dropping one goal in favour of another when the new goal has a higher priority than the original one. However, as we shall see, there are sometimes situations in which we need to take into account the details of the plans involved in achieving two conflicting goals in order to determine the appropriate course of action (and hence consideration of the second kind of intention referred to above).

The basic premise that Goals should not be allowed to conflict is then represented by the rule:

$\mathbf{R_1}$: $Goal(\alpha) \wedge Goal(\beta) \rightarrow \neg\, Con(\alpha, \beta)$

Similarly, it seems reasonable to require that if there are two conflicting desires, only the higher priority one is adopted as a goal, giving:

$\mathbf{R_2}$: $Des(\alpha) \wedge Des(\beta) \wedge Con(\alpha, \beta) \wedge (Pr(\alpha) > Pr(\beta)) \rightarrow Goal(\alpha) \wedge \neg Goal(\beta)$

However, this seemingly sensible rule is too naive, and can lead to inconsistency. For example, the situation where we have three desires α, β and ϕ, where $Con(\alpha, \beta) \wedge Con(\beta, \phi) \wedge \neg Con(\alpha, \phi) \wedge (Pr(\alpha) > Pr(\beta) > Pr(\phi))$. By R_2, comparing α and β we will obtain $Goal(\alpha) \wedge \neg Goal(\beta)$. Using the same rule to compare β and ϕ we obtain $Goal(\beta) \wedge \neg Goal(\phi)$. Combining these we obtain $Goal(\beta) \wedge \neg Goal(\beta)$ which is clearly inconsistent. In addition, if we assume that what we intuitively want is to adopt α (being the highest priority goal) and hence disallow $Goal(\beta)$, then it seems reasonable to take the view that it is unnecessary to disallow $Goal(\phi)$ due to inconsistency with $Goal(\beta)$.

In order to address this we say that desires are *impeded* when they are disallowed as Goals due to a conflict with a higher priority desire (which is not impeded)[2] :

$Conflict(\alpha, \beta) \wedge (Pr(\alpha) > Pr(\beta)) \wedge \neg Imp(\alpha) \rightarrow Imp(\beta)$

We can now rewrite R_2 as:

$\mathbf{R_2}$: $Des(\alpha) \wedge Des(\beta) \wedge \neg Imp(\alpha) \wedge Con(\alpha, \beta) \wedge (Pr(\alpha) > Pr(\beta)) \rightarrow Goal(\alpha) \wedge \neg\, Goal(\beta)$

The third rule which seems reasonable is that if there is no preference ordering between two desires, then we should prefer a goal that is already adopted over one that is not:

$\mathbf{R_3}$: $Des(\alpha) \wedge Goal(\beta) \wedge Con(\alpha, \beta) \wedge (Pr(\alpha) = Pr(\beta)) \rightarrow \neg Goal(\alpha)$

Rules R_1, R_2 and R_3 thus provide a specification of how goals can be derived from desires. However, these rules treat intentions at all levels as equal, and once a plan has been chosen, there is no reconsideration of that plan (or the subgoals that result from steps in the plan) unless a particular subgoal is unable to be adopted or is aborted, due to a conflict. However, in some situations this approach may be too naive. For example, if I have a goal to go to Sydney, and I choose a plan that involves a subgoal of me driving my car to the airport then this would conflict with a goal to lend my car to a friend, which is a subgoal of the top-level goal to be helpful to my friends.

Assuming that the goal of being helpful to my friends is less important than going to Sydney, we would expect to adopt the goal of taking my car to the airport in preference to lending the car to my friend. However, it may be possible to find alternative transport to the airport, such as taking a taxi, which does not conflict with lending my car to my friend. In such cases it may well be rational for an agent to engage in limited reconsideration of its plans, and thus its intentions, due to conflicts with subgoals inherent in another plan. This does not nullify the powerful controlling effect of intentions, as we are not suggesting that the agent reasons about all possibilities, but only about alternative means to the most immediate subgoal.

[2]This is essentially the same as the use of pre-emption in skeptical default inheritance reasoners [Touretzky et al., 1991, Padgham, 1992].

In order to capture this limited reconsideration, which can be seen as a lesser level of commitment than that defined thus far, we define some new predicates and new rules governing the adoptions of goals. Plan(P,G) indicates that there is an applicable plan P to achieve Goal G.[3] Ex(P) indicates that P is a plan which has been chosen and is currently being executed. Step(G,P,S) indicates that S is a step in plan P, which is achieving goal G. A step, like a goal and a desire, is a partial state description. A step, like a desire, is a precursor to a goal. Pref(S_1, S_2) indicates that partial state S_1 is Pref over partial state S_2, where S_1 and S_2 may be associated with desires, goals or steps.

If a step conflicts with some other goal or potential goal, then it seems reasonable to seek an alternative plan, if possible, which avoids any such conflicting step. This preference may be due to the existence of an alternative plan for the less preferred step, or may be due to some other reason such as priority. We assume that preference is only relevant when there is conflict. We incorporate such considerations in the rules below.

In order to simplify the statement of the rules, we define the auxiliary relations *ExStep* and *AltPlan* as follows:

$Plan(P_1, \alpha) \wedge Ex(P_1) \wedge Step(\alpha, P_1, \phi) \rightarrow Goal(\alpha) \wedge ExStep(\alpha, P_1, \phi)$

Thus $ExStep(\alpha, P_1, \phi)$ means that Plan P_1 is currently being executed to achieve goal α and ϕ is a step in this plan. We also define

$(\exists P : Plan(P, \alpha) \wedge (P \neq P_1) \wedge (\forall \gamma \psi : Step(\alpha, P, \gamma) \wedge Goal(\psi) \rightarrow \neg Con(\gamma, \psi))) \rightarrow AltPlan(\alpha, P_1)$

Hence $AltPlan(\alpha, P_1)$ means that there is an alternative plan to P_1 which achieves the same goal as P_1, but where none of the steps conflict with current goals.

Next, we need a rule which captures the fact that if there are two conflicting steps, ϕ and $\phi^{'}$, only one of which (say ϕ) occurs in a plan for which there is a conflict-free alternative, then we will prefer the other step (i.e. $\phi^{'}$).

$\mathbf{R_4}$: $ExStep(\alpha, P_1, \phi) \wedge ExStep(\beta, P_2, \phi^{'}) \wedge Con(\phi, \phi^{'}) \wedge AltPlan(\alpha, P_1) \wedge \neg AltPlan(\beta, P_2) \rightarrow Pref(\phi^{'}, \phi)$

We also need a rule that allows us to reconsider our plan (and thus intentions) when a new desire conflicts with a step in the plan. This is similar to $\mathbf{R_4}$ except that a desire does not have any possible alternative plan:

$\mathbf{R_5}$: $ExStep(\alpha, P_1, \phi) \wedge Des(\phi^{'}) \wedge Con(\phi, \phi^{'}) \wedge AltPlan(\alpha, P_1) \rightarrow Pref(\phi^{'}, \phi)$

We then need to modify R_2 and R_3 so that priority is only considered when the alternative plan discriminator of R_4 and R_5 is not applicable (i.e. both or neither have alternative plans available), and prior commitment (as captured in R_3) is only relevant when neither preference nor alternative plans apply.

Thus R_2 becomes:

$\mathbf{R_2^{'}}$: $Des(\alpha) \wedge Des(\beta) \wedge \neg Imp(\alpha) \wedge Con(\alpha, \beta) \wedge (Pr(\alpha) > Pr(\beta)) \wedge \neg Pref(\alpha, \beta) \wedge \neg Pref(\beta, \alpha) \rightarrow Goal(\alpha) \wedge \neg Goal(\beta)$

And similarly R_3 becomes:

[3]An applicable plan is a plan which is appropriate in the current context, including consideration of both current beliefs and plan failures.

$\mathbf{R_3'}$: $Des(\alpha) \wedge Goal(\beta) \wedge Con(\alpha, \beta) \wedge Pr(\alpha) = Pr(\beta) \wedge \neg Pref(\alpha, \beta) \wedge \neg Pref(\beta, \alpha) \rightarrow \neg Goal(\alpha)$

Finally, we have a rule that states that our goal adoption must be consistent with our preferences:

$\mathbf{R_6}$: $Pref(S_1, S_2) \rightarrow Goal(S_1) \wedge \neg Goal(S_2)$

Note that steps in this sense may be thought of as subgoals, and vice-versa. In other words, R_6 expresses the identification of steps in a plan with subgoals, once the subgoals are sufficiently refined. This provides the linkage between goals and the plans used to achieve them.

Given such rules, it is then straightforward to modify the preference relation given in Section 2 to reflect this finer notion of preference. This is done below.

$G_1 \gg G_2$ if one of the following conditions is satisfied:

1. $\forall G \in G_2 \; \exists G' \in G_1$ such that Pref(G', G)
2. If 1 does not apply, then $\forall G \in G_2 \; \exists G' \in G_1$ such that Pr$(G') >$ Pr(G)
3. $G_1 \supseteq G_2$

It is now possible to discuss varying levels of commitment based on the willingness of the agent to reconsider its intentions. The most committed agent, described by Rao and Georgeff as blindly committed, remains committed to its intentions until they succeed. The next level of commitment (single-minded in Rao and Georgeff's terminology) requires the agent to remain committed to its intentions until they either succeed or fail. Rao and Georgeff then define what they call "open-minded commitment" which requires an agent to be committed to its intentions until they succeed or the goal is dropped. However there is no discussion of when the agent is permitted to drop its goal.

The single-minded commitment can be seen as a special case of open-minded commitment, where goals can be dropped only when they fail. An agent exhibiting this level of commitment will not consider dropping a goal to add a conflicting higher priority goal. It will maintain goal consistency by simply not adding the new higher priority goal. We would argue that this is the minimal baseline for rational agents.

The rule set R_1, R_2 and R_3 can be seen as a further special case of open-minded commitment that allows goals to be dropped if there is a conflicting higher priority goal. We call this *priority-based commitment* and would argue that this (combined with dropping of goals when they have failed) is normally the minimum level one would want for a rational agent if priority information is easily available.

The rule set $R_1, R_2', R_3', R_4, R_5, R_6$ define a less committed agent with a more flexible reasoning strategy regarding level of commitment. An agent using these rules is willing to drop a goal if there is an alternative, conflict-free plan available to achieve the goal which immediately motivated this goal. We call this *alternative-based commitment*.

Clearly there are additional strategies that can be defined which progressively weaken the conditions under which one may drop a goal.

Instead of requiring that there be a conflict-free alternative plan, one may want to require only that there is an alternative plan which conflicts only with lower priority goals than that involved in the current conflict. Or one may wish to use strategies which maximise the number of goals at a given level, thus allowing a goal to be dropped if it conflicts with a larger number of competing Desires or steps than its competitors.

Each new commitment strategy would require a set of rules which provably maintain consistency of the goal set, and which do not allow for removal of goals other than according to the desired strategy. In order to recognise supported models it would be necessary to define the preference ordering which results from each new strategy.

5 Efficiency evaluation

One of the strengths of BDI-style agent systems is that they are suitable for real-time applications. Thus it is important that any additional functionality be evaluated with respect to efficiency and analysed regarding its impact on suitability for interactive and real-time applications.

In order to address this question we took an existing BDI system, JACK[4] and added mechanisms for maintaining a representation of goals and desires and for detecting and resolving conflicts. We implemented conflict resolution according to both priority-based commitment and alternative-based commitment. We then ran a series of experiments to compare the run-time of standard JACK with our augmented version of JACK.

There are a number of factors which will affect the length of an execution cycle in the original JACK system and in our augmented systems. Initial experimentation indicated that the two factors affecting the length of JACK execution cycle were the number of plans matching a particular trigger, and the number of context conditions to be evaluated on each matching plan. The number of context conditions was a much more important factor than the number of triggered plans.

The major factor influencing the length of the execution cycle for conflict detection in our augmented system is the number of potential conflicts — i.e. the number of times the system must determine whether or not a potential conflict concerning some attribute, is a conflict or not. Other factors which also affect the execution cycle are:

1. Depth of the plan hierarchy: the greater the depth, the more sub-goals there will be at any particular time, associated with a particular top level goal. This will require a larger number of consistency checks for each potential new goal.
2. Number of concurrent top-level goals: the more concurrent top-level goals, the more corresponding subgoals and the larger the number of consistency checks required for each potential new goal.
3. Number of literals in the partial state descriptor for each goal or sub-goal: a consistency check is done for each literal on each goal, so this factor increases the number of consistency checks.

In addition the following factor affects the conflict resolution phase for the alternative-based commitment:

4. Number of steps in a plan: each step of each alternative plan must be checked to ensure that it doesn't conflict with existing goals.

For the purpose of this evaluation we took a worst case and a best case approach for each test situation. The best case had no potential conflicts and thus never had to enter the most time consuming part of the conflict detection procedure. The worst case had a potential conflict with every current sub-goal, but no actual conflict was found. This required entering the time consuming constraint solver as many times as there were current goals or sub-goals. This can safely be assumed to be totally unrealistic in practice, but provides an experimental upper bound. The 4 test cases used ranged from a very simple situation with no real concurrency, to a situation with 20 concurrent top-level goals, largish plans, and a hierarchy of depth 5. The results are shown in table 1 and are graphically plotted against times for the JACK execution cycle at similar levels of system complexity in figure 1. Cycle times were calculated using the timing to test one potential conflict in the test situation, multiplied by the number of tests that would be required, resulting from the various parameters.

Test	A	B	C	D	E	PB	AB
1	2	1	1	1	2	0.235 ms	0.271 ms
2	2	3	3	10	2	0.585 ms	0.691 ms
3	2	3	3	20	2	0.945 ms	1.051 ms
4	5	10	5	20	2	1.503 ms	1.861 ms

Table 1: Commitment strategy timings

Legend

A	- Number of context conditions per plan
B	- Number of steps in each plan (i.e. sub goals)
C	- Average depth of each plan set
D	- Number of top level goals handled concurrently
E	- Number of post conditions per goal
PB	- Time taken for the new implementation using the priority-based commitment strategy.
AB	- Time taken for the new implementation using the alternative-based commitment strategy.

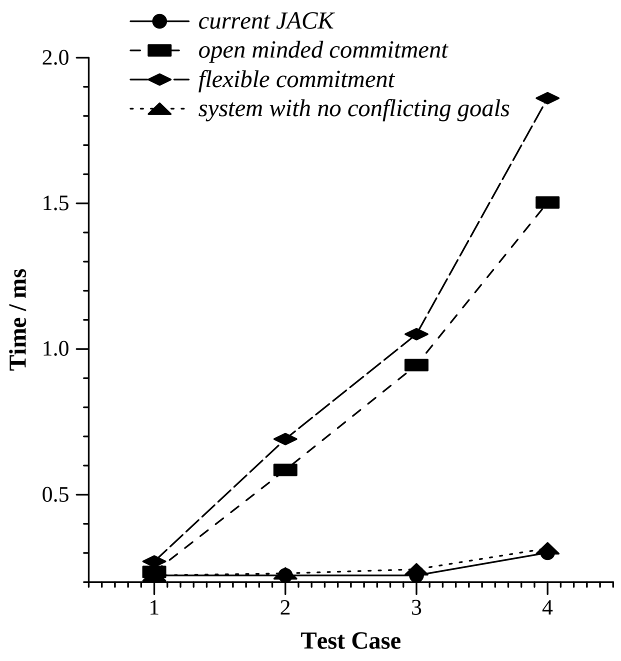

Figure 1: Comparison of Systems

As can be seen, the cost of checking for conflict is negligible if no potential conflicts exist. Predictably, as the number of potential conflicts increase the time does increase steeply. However we note that the level of conflict for which these figures are calculated is an extreme upper bound. Nevertheless, even in the most complex situation, with 20 concurrent top-level goals, large plans, deep nesting, and all subgoals needing to be tested, it is still possible to handle a new goal in under 2 milliseconds. Consequently we feel confident that this addition to BDI agent systems will not be an impediment to their use in real-time domains. For situations where time is critical, the design aspects which can help to ensure efficiency are the size of each plan as measured in number of steps, and the complexity of each goal as measured in the number of literals describing the partial-state to be achieved. Clearly these factors are closely linked, as limited goals will result in shorter plans to achieve them.

[4]Jack is available from Agent Oriented Software, `www.agent-software.com`

6 Conclusions and Further Work

We have seen how an explicit representation of goals can be used to facilitate specification of conflict detection and resolution, thus bringing the theory and implementations of BDI systems closer together. As noted in the previous section, this does not introduce significant overheads.

Naturally the precise form of the rules used will vary with the commitment strategy chosen. However, we believe that the process followed in this paper will be basically the same whatever commitment strategy is used.

There remains the problem of providing formal rules which can be used to *generate* consistent goal sets from desires, as distinct from *specifying* such goal sets. These generative rules exist currently within our implemented system, in the form of Java code. However formalising these and proving equivalence to the specification rules will add another step in the process of closing the gap between theory and practice.

Our broader goal is to develop a framework for rational agents, based on our experiences with the logic of Rao and Georgeff and BDI implementations such as JACK and dMars, in which the theory and implementation are more closely aligned. The explicit representation of goals is thus one step towards such a framework, which will also provide more detailed reasoning about the relationships between beliefs, desires, goals, intentions, actions and plans.

References

[AAII, 1996] AAII (1996). dMARS Technical Overview. *The dMARS V1.6.11 System Overview*.

[Bell and Huang, 1997] Bell, J. and Huang, Z. (1997). Dynamic goal hierarchies. In Cavedon, L., Rao, A., and Wobcke, W., editors, *Intelligent Agent Systems: Theoretical and Practical Issues*, pages 88–103.

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

[Busetta et al., 1999] Busetta, P., Rönnquist, R., Hodgson, A., and Lucas, A. (1999). Jack intelligent agents - components for intelligent agents in java. AgentLink News, Issue 2.

[Hindriks et al., 2001] Hindriks, K., de Boer, F., van der Hoek, W., and Meyer, J.-J. (2001). Agent programming with declarative goals. In *Intelligent Agents VI - Proceedings of the 7th International Workshop on Agent Theories, Architectures, and Languages (ATAL'2000)*, Berlin. Springer Verlag.

[Huber, 1999] Huber, M. J. (1999). Jam: A BDI-theoretic mobile agent architecture. In *Proceedings of the Third International Conference on Autonomous Agents (Agents'99)*, pages 236–243, Seattle, WA.

[Jennings and Wooldridge, 1998] Jennings, N. and Wooldridge, M. (1998). Applications of intelligent agents. In Jennings, N. R. and Wooldridge, M. J., editors, *Agent Technology: Foundations, Applications, and Markets*, chapter 1, pages 3–28. Springer.

[Jennings, 2001] Jennings, N. R. (2001). An agent-based approach for building complex software systems. *Communications of the ACM*, 44(4):35–41.

[Ltd, 2000] Ltd, A. O. S. A. P. (2000). *JACK Intelligent Agents, User Guide*. AOS Pty Ltd, Carlton, Victoria, 3053.

[Padgham, 1992] Padgham, L. (1992). Defeasible inheritance: A lattice based approach. *Computers and Mathematics with Applications*, 23(6-9):527–541. Special Issue on Semantic Nets.

[Rao and Georgeff, 1991] Rao, A. S. and Georgeff, M. P. (1991). Modelling rational agents within a BDI-Architecture. In Fikes, R. and Sandewall, E., editors, *Proceedings of the Second International Conference on Principles of Knowledge Representation and Reasoning, KR '91*, pages 473–484, Cambridge, MA.

[Rao and Georgeff, 1992] Rao, A. S. and Georgeff, M. P. (1992). An abstract architecture for rational agents. In Rich, C., Startout, W., and Nebel, B., editors, *Proceedings of the third International Conference on Principles of Knowledge Representation and Reasoning, KR '92*, pages 439–449, Boston, MA.

[Touretzky et al., 1991] Touretzky, D. S., Thomason, R. H., and Horty, J. F. (1991). A skeptic's menagerie: Conflictors, preemptors, reinstaters, and zombies in nonmonotonic inheritance. In *IJCAI91*, pages 478–483.