Autonomy and Agent Deliberation

Mehdi Dastani, Frank Dignum, and John-Jules Meyer

Utrecht University
Institute of Information and Computing Sciences
{mehdi,dignum,jj}@cs.uu.nl

Abstract. An important aspect of agent autonomy is the decision making capability of the agents. We discuss several issues that agents need to deliberate about in order to decide which action to perform. We assume that there is no unique (rational or universal) deliberation process and that the deliberation process can be specified in various ways. The deliberation process is investigated from two perspectives. From the agent specification point of view the deliberation process can be specified by dynamic properties such as commitment strategies, and from the agent programming point of view the deliberation process should be implemented through the deliberation cycle of the agent, which can be either fixed or determined by a deliberation programming language.

1 Introduction

The autonomy of software agents is directly related to their capacity to make decisions without intervention of the human users. At the lowest level these decisions concern the next action that the agent is going to perform. However, the actions are selected based on many high level decisions such as whether it will stick to its current plan to reach a goal, whether it wants to change its goal, whether to adhere to a norm or not, etc. It seems logical to start with classical decision theory for help in modelling these decisions. On the other hand intelligent agents are modelled using BDI theory. So we will first briefly explore the relation between classical decision theory, a qualitative extension of it called qualitative decision Theory, and the BDI theory. Formal details of this relation are explained in [6].

Classical decision theory [9, 16] assumes that the decision making agent is able to weigh all possible alternative courses of actions before choosing one of them. Given a set of actions and a set of outcomes (states), actions are weighed by two basic attributes that are assumed to be accessible by the decision making agent. The first attribute is the probability distribution that indicates the probability of reaching a state by performing an action. The second attribute is the utility function that indicates the utility of states. Based on these attributes a certain decision rule, such as maximizing expected utility, is used to determine which action should be selected and performed. Although useful in closed systems, the problem with classical decision theory is that in open or complex agent environments, the probability distributions are not fixed or cannot be calculated due to resource or time constraints. Therefore the agents only have partial information and cannot compare the utility of all possible states.

Qualitative decision theory [1, 12] relaxes the assumption of classical decision theory that the probability distribution and the utility function should be provided, and develop representation and reasoning schemes for partial information and generic preferences to represent probabilities of states and generic preferences over those states [7]. Typically qualitative orderings are introduced that represent both the likelihood or normality (probability) and desirability (utility) of states. The maximum expected utility decision rule is replaced by qualitative rules such as Wald's criterion. A rational agent is then assumed to attempt to achieve the best possible world consistent with its likely (default) knowledge. Qualitative decision theory provides the state to be reached without indicating which actions to perform. The decision making agent is then assumed to be a planning agent who can generate actions to reach the given state.

A criticism to both classical and qualitative decision theory is that they are inadequate for real time applications that decide actions in highly dynamic environments. In such applications, the decision making agent, which is assumed to be capable of performing a set of actions, should select and execute actions at each moment in time. As the highly dynamic environment changes either during the selection or the execution of actions, the decision making agent needs to deliberate to either reconsider the previous decisions, also called intentions, or continue the commitment to those decisions (intentions). This deliberation process leads to either a reconsideration of the intentions or the continuation of the commitment to the intentions. This decision aspect is argued by Bratman [2] to be crucial to realize stable behavior of decision making agents with bounded resources. This approach has led to what is called BDI theory [14].

In BDI theory, the (partial) information on the state of the environment is reduced to dichotomous values (0-1); the propositions are believed or not. This abstraction of the (partial) information about the state of the environment is called the beliefs of the decision making agent. Through this abstraction, the information on likelihood ordering among states is lost. Similarly, the (partial) information about the objectives of the decision making agent is reduced to dichotomous values (0-1); the propositions are desired or not. This abstraction of the objectives of the decision making agent is called agent desires; the information on desirability ordering among states is lost. Finally, the information about the selected goal states is represented by dichotomous values (0-1); an agent can intend to reach a state or not. Note that the BDI systems can be extended with either qualitative orderings for likelihood and desirability of beliefs [3], or even with quantitative probabilities and utilities [13]. A decision-theoretic view of BDI theory extends the dichotomous abstraction of qualitative decision theory with the additional deliberative component, i.e. the intentions, to stabilize decision making behavior of the agent.

Unlike decision theories that propose (quantitative or qualitative) decision rules to determine which actions to select and perform, BDI theory assumes that the agent decides about which actions to perform next by reasoning on a plan to reach the selected goal state. Some constraints are proposed to identify the possible goal states and plans. Examples of static properties are realism, weak realism, and strong realism, which state in various degree that (only) states of which the agent believes that they are possible can be selected as goal states. Examples of dynamic properties are various kinds of commitment strategies that determine when the commitments to previous decisions can be dropped, i.e. when intentions can be reconsidered [13].

Whereas decision theory tries to generate the best agent decision at any point in time, the BDI theory tries to limit agent choices to the correct ones according to the constraints that are given beforehand. In order to create agents that are autonomous they have to be able to make decisions on different levels while adhering to the BDI theory. In order for autonomous agents to be programmable they probably need decision theory (which gives some procedural handles). Ideally an agent is autonomous if it would arrive at the best possible decision using some (parts of) decision theory while keeping within the constraints given by the BDI theory.

The rest of this paper is organized as follows. In the next section, various deliberation issues are explained and it is explained how they affect the decision to select actions. In section 3 it is argued that the decision-making ability of agents should be programmable in order to create agent autonomy at different levels. We therefore introduce a language to program possible deliberation processes. In section 4 we consider the specification of some static and dynamic properties of agent behavior, known from BDI literature, and indicate how these properties can be realized by programming certain deliberation processes. Finally, in section 5 we conclude the paper.

2 Deliberation Issues

Classical and qualitative decision theories propose decision rules since they have orderings on possible states. However, BDI theory cannot propose a decision rule since there is no ordering given on possible states. Note that the suggested constraints by the BDI theory cannot be considered as the counterpart of decision rules since they do not specify one single goal state that, when planned, generates a sequence of actions. For example, the realism constraint states that a goal state can be selected if it is believed to be achievable. However, given the beliefs and goals of an agent, many states can be believed as being achievable such that it should still be specified which state should be specified as the goal state.

In this section, we consider the decision making ability of agents from the implementation point of view, i.e. how the decision making ability of agents can be implemented. For the classical decision theory, the implementation of decision rules is straight-forward since it is an algorithmic function defined on the probability distribution and the utility function. Also, for the qualitative version of decision theory the decision rules can be easily implemented based on the likelihood and the preference function. Our problem is that we cannot assume an agent to have such orderings on possible states and thus have these functions available. However, starting from the higher level declarative goals plus the axioms of BDI theory we end up with a set of possible goal states rather than a sequence of actions. We should then select one goal state and propose a planning mechanism to determine the sequence of actions to reach the chosen goal state. Therefore, the implementation of the decision making ability of agents based on the BDI theory is much more complex since no ordering is provided. Even in the extended BDI framework KARO [11] which also contains capabilities, actions and results of actions the decisions on which goals and intentions are selected are deliberately left open to allow for a variety of agent types.

We argue that these choices should be made explicit by the so-called deliberation process rather than fixing them in some internal structure of the agent. In particular, the deliberation process is meant to construct and substitute the ordering information, which was lost through the dichotomous abstraction of decision theory, in a dynamic way and based on many (deliberation) issues. Because in the context of BDI agents we may assume that all relevant aspects of the environment of the agent are modelled as mental attitudes of the agent, we may assume that agents determine the course of actions based on these mental attitudes. The mental attitudes contain at least things such as beliefs and desires (or potential goals), capabilities such as actions and plans, and (defeasible) reasoning rules that can be used to reason about the mental attitudes. When other entities or mental attitudes play a role in agent decision, the corresponding deliberation aspects should be taken into consideration. For example, if we want to consider communication or sensing to play a role in agent decisions, then we should take into account the deliberation issues such as when to perform communication (receive or send) or sense actions and when to revise agent beliefs based on the acquired information. See also [10] for a good description of all possible decision points involving external influences of the agent. Given the above mentioned decision related entities, the deliberation process can be considered as consisting of reasoning about mental attitudes, selecting goals, planning goals, selecting and executing plans.

First of all an agent has to make choices about how to reason about its mental attitudes at each moment in time. For example, an agent can reason about its goals or the agent can reason about its goals only when they are not reachable using any possible plan. Some more moderate alternatives are also possible. E.g. the agent can create a plan for a goal and perform the plan. If this leads to a stage where the plan cannot be performed any further, then the agent can start reasoning about the plan and revise the plan if necessary. If the goal can still not be reached, then the agent can revise the goal. So, this leads to a strategy where one plan is tried completely and if it fails the goal is revised or even abandoned.

The deliberation process should also control the relation between plans and goals. For example, the deliberation process should control whether a goal still exists during the execution of the plan to reach that goal. If the corresponding goal of a plan is reached (or dropped), the deliberation process can avoid or allow continuing with the plan. If it is decided to halt the execution of a plan, then the deliberation process may decide to perform a kind of ”garbage collection” and remove a left-over plan for the goal that no longer exists. If this would not be done the left-over plan would become active again as soon as the goal would be established at any later time. This type of deliberation choices can be exploited to distinguish and implement the so-called maintenance from achievement goals.

Another issue that should be considered by the deliberation process is related to multiple (parallel) goals and/or plans. First, it should be decided whether only one or more plans can be derived for the same goal at any time. If we allow only one current plan for each goal, the plans will all be for different goals. In this case it has to be determined whether the plans will be executed interleaved or consecutively. Interleaving might be beneficial, but can also lead to resource contention between plans in a way that no plan executes successfully anymore. E.g. a robot needs to go to two different rooms that lay in opposite directions. If it has a plan to arrive in each room and interleaves those two plans it will keep oscillating around its starting position indefinitely. Many of the existing work on concurrent planning can, however, be applied straight away in this setting to avoid most problems in this area.

3 A Programming Language for the Deliberation Process

In this section, we assume that autonomous agents have mental attitudes such as beliefs, goals, intentions and plans, as well as a reasoning capability to reason and revise their mental attitudes. From the programming point of view, mental attitudes are implemented as data structures called belief base, goal base, intention base, and plan base. These bases are initially implemented by expressions of specific languages and can change during the deliberation process. An example of languages to implement various mental attitudes is 3APL [8]. The reasoning capability of agents is implemented by means of various rule bases. In particular, we assume a goal rule base, an intention rule base, and a plan rule base. These rule bases contain rules that can revise or reconsider goal, intentions, and plans, respectively. In addition, we assume a planning rule base consisting of rules that can be used to plan intentions, replan an existing plan, or backtrack from an existing plan. Finally, we assume proof engines that can check if a certain formula is derivable from the belief base, goal base, and intention base, respectively. So, we make an explicit distinction between proof engines that provide logical consequence relations for the different modalities and rule bases which regulate the updates of the different modalities.

The programming language to implement the deliberation process is considered as a meta-langauge that can express the selection and application mechanisms and the order in which decisions should be taken [5]. The selection mechanism involves decisions such as which goals or plans should be selected and executed, which rules should be selected and applied, or which mental bases should be updated by a formula. The order of decisions determine whether goals should be updated before getting executed or should the plans be executed before selecting a new goal. The meta-language is imperative and set-based. It is imperative because it is used to program the flow of control and it is set-based because it is used to select goals and rules from the set of goals and rules.

The proposed deliberation language consists of programming constructs to program directly the selection of goals, plans, and reasoning rules, to execute goals and plans, to apply rules, and to update a mental base. These programming constructs can be conditionalized on agent beliefs in order to contextualize the deliberation process. For example, a robot that has a set of goals including a transport goal (to transport boxes from position A to position B) and a goal to clean the space it operates may come in an emergency situation where it should only select and execute the transport goal.

3.1 Deliberation Terms

In order to implement the selection of goals, intentions, and plans, to apply reasoning rules to them, to execute a plan, or to update a mental base, we introduce terms that denote belief, goal, and intention formulas as well as terms that denote reasoning rules and plans. Here we use sorted terms because we need terms to denote different entities such as sets of goals, sets of intentions, sets of plans, individual goals, individual intentions, individual plans, sets of rules, individual rules, numbers, etc. These terms can be used as arguments of programming constructs that can be used to implement the deliberation cycle.

In the following we use Σ, Γ, Δ, and Π, to denote the belief base, the goal base, the intention base, and the plan base, respectively. The first three bases consists of formulae that characterize states, while the plan base consists of a set of plans (e.g. action expressions). Moreover, we use Λ_{sgrr} to denote the goal reasoning rule base consisting of rules that can be used to revise the goal base (e.g. if agent i wants x, but x is not achievable, then i wants y), Λ_{sirr} to denote the intention reasoning rule base, and Λ_{sprr} to denote the plan reasoning rule base. Generally, these (practical) reasoning rules consist of a head and a body (a goal and an intention formulae, or a plan expression), and a condition which is a belief formula. Finally, we use Ω to denote the plan rule base which can be use to plan goals, i.e. to relate plans to goals. The plan base consists of rules that have a goal formula as the head, a plan expression as the body, and a belief formula as the condition.

Definition 1. *Let S be a set of sorts including sorts for sets of beliefs (sb), sets of goal formulas (sg), sets of intention formulas (si), sets of plan expressions (sp), sets of planning rules (spr), sets of goal revision rules (sgrr), sets of intention reconsideration rules (sirr), sets of plan revision rules (sprr), individual belief formulas (ib), individual goal formulas (ig), individual intention formulas (ii), individual plan expression (ip), individual planning rule (ipr), individual goal revision rules (igrr), individual intention reconsideration rules (iirr), individual plan revision rules (iprr), numbers (N), and booleans (B).*
Let also Var_s be a set of countably infinite variables for sort s, and F be a set of sorted functions. The deliberation sorted terms T_s are then defined as follows:

- $\Sigma \in T_{sb}, \Gamma \in T_{sg}$, $\Delta \in T_{si}$, $\Pi \in T_{sp}$, $\Lambda_x \in T_x$ *for* $x \in \{spr, sgrr, sirr, sprr\}$.
- $Var_s \subseteq T_s$ *for all sort* $s \in S$
- *if* $s_1, \ldots, s_n \in S, t_i \in T_{s_i}$ *and* $f : s_1 \times \ldots \times s_n \rightarrow s \in F$, *then* $f(t_1, \ldots, t_n) \in T_s$

Some typical functions that are defined on goals are $max : sg \rightarrow sg$ that selects the subset of maximal preferred goals from the set of goals[1], $empty : sg \rightarrow B$ that determines whether the set of goals is empty, $head : ixrr \rightarrow ix$ and $body : ixrr \rightarrow ix$ for $x \in \{g, i, p\}$[2], $sel\_trans : sg \rightarrow ig$ that selects an individual goal (in this case the transport goal) from a set of goals, or $gain : ig \rightarrow N$ that determines the utility (a number) of an individual goal. Note that nested applications of functions can be used as well. For example, $gain(max(\Gamma))$ indicates the utility of the maximal preferred goal. Similar functions can be defined for practical reasoning rules and plans.

3.2 Deliberation Formulas

Given the terms denoting belief, goal, and plan formulas as well as reasoning rules, the set of formulas of the deliberation language can be defined. Moreover, we have assumed three proof engines that verify if a certain formula is derivable from the belief base, goal base, or intention base. In order to express that a certain formula is derivable from belief, goal, and intention bases, we introduce the *provable* predicate.

Definition 2. *Let* $s \in S$ *be a sort,* $t_s, t'_s \in T_s$ *be terms of sort* s*,* $x \in \{b, g, i\}$*, and provable be a predicate. We define the set of deliberation formulas* DF *as follows:*

- $t_s = t'_s, t_s \geq t'_s, provable(t_{sx}, t_{ix}) \in DF$
- *if* $\phi, \psi \in DF$*, then* $\neg\phi, \phi \wedge \psi \in DF$

The deliberation formula $provable(t_{sx}, t_{ix})$ for $x \in \{b, g, i\}$ expresses that the belief, goal, or intention formula denoted by the terms t_{ix} are derivable from the belief base, the goal base, and the intention base, denoted by t_{sx}, respectively. For example, let Σ be the term that denotes the belief base $\{\phi, \phi \rightarrow \psi\}$ and α be a term denoting the formula ψ, i.e. $|[\alpha]| = \psi$. Then, the deliberation formula $provable(\Sigma, \alpha)$ is a valid formula. Note that the predicate *provable* is only defined over deliberation terms, not over the formulas of DF itself! One important difference is that the deliberation terms do not contain negation.

3.3 Deliberation Statements

Having deliberation terms and formulas, the programming constructs or statements of the deliberation language to program the deliberation cycle of cognitive agents can now be defined.

[1] For this function an ordering among goals is assumed.

[2] A reasoning rule is assumed to be defined in terms of a head and a body.

Definition 3. *Let $s \in S, t_s \in T_s$ and $V_s \in Var_s$. The set of basic statements of the deliberation language is defined as follows:*

- $V_s := t_s$
- $selgoal(t_{sg}, f_c, V_{ig})$
  $selint(t_{si}, f_c, V_{ii})$
  $selplan(t_{sp}, f_c, V_{ip})$
  $selrule(t_{sxrr}, t_{sx}, V_{ixrr})$ *for* $x \in \{g, i, p\}$
- $update(t_{sx}, t_{iy})$ *for* $x, y \in \{b, g, i\}$
  $reviseplan(t_{ip}, t'_{ip})$
- $plan(t_{ii}, t_N)$
  $replan(t_{ip}, t_{spr}, f_c, t_N)$
  $btplan(t_{ip}, t_{spr}, t_N)$
  $explan(t_{ip})$

The set of deliberation statements is defined as follows:

- *Basic statements are deliberation statements*
- *If $\phi \in DF$ is a deliberation formula, and α and β are deliberation statements, then the following are deliberation statements:*
  α ; β ,
  `IF` ϕ `THEN` α `ELSE` β ,
  `WHILE` ϕ `DO` α

The first statement $V_s := t_s$ is designed to assign a sorted term t_s to a variable V_s of the same sort. The following statements are all selecting some item from a particular set of those items. The statement $selgoal(t_{sg}, f_c, V_{ig})$ selects an individual goal from the set of goals denoted by the term t_{sg}. The term denoting the selected individual goal is assigned to variable V_{ig}. The function f_c maps goals to boolean values indicating whether the goal formula satisfies the criterium c. The statement $selint(t_{si}, f_c, V_{ii})$ selects an individual intention from the set of intentions denoted by the term t_{si}. The term denoting the selected individual intention is assigned to variable V_{ii}. The function f_c indicates whether the intention satisfies the criterium c. The statement $selplan(t_{sp}, f_c, V_{ip})$ selects an individual plan from the set of plans denoted by the term t_{sp}. The term denoting the selected plan should satisfy criterion f_c and is assigned to the variable V_{ip}. The statement $selrule(t_{sxrr}, t_{sx}, V_{ixrr})$ selects a rule from the set of (goal, intention, or plan) reasoning rules denoted by the terms t_{sxrr} and assigns the term that denotes the rule to the variable V_{ii}. The selected rule should be applicable to a formula from the set denoted by the term t_{sx}.

The criterium c used in the selection functions can be used to define a preference ordering between the goals, intentions, plans and rules. So, in fact this is the place where a relation with qualitative decision theory can be made. The same argument can be made for the other selection functions. The main advantage over the classical decision theoretic approach is that the deliberation uses several independent preference orderings over different concepts. The combination of all these orderings leads to a decision on which action will be performed next. However, unlike decision theory where all factors have to be combined into one function that determines the best action, we explicitly program this combination. Besides this advantage of having all factors explicitly available (and thus easily adjustable) the combination of these factors into a decision can be made situation dependent. therefore allowing for an adjustable preference ordering of the agent.

The statement $update(t_{sx}, t_{iy})$ updates a mental base (belief, goal, or intention base) denoted by the term t_{sx} with the formula denoted by the term t_{iy}. The statement $reviseplan(t_{ip}, t'_{ip})$ removes the plan that is denoted by the term t_{ip} from the plan base, and adds the plan that is denoted by the term t'_{ip} to it.

The final set of basic statements are all related to updating the plans of the agent in some way. The statement $plan(t_{ii}, t_n)$ generates a plan expression with maximum length t_n to achieve intention t_{ii}. The generated plan expression is assigned to the plan base of the agent. The statement $replan(t_{ip}, t_{spr}, f_c, t_N)$ uses the set of planning rules t_{spr} and generates a new plan expression to replace the plan expression t_{ip}. The new plan expression satisfies the criteria f_c and has maximum length t_N. The statement $btplan(t_{ip}, t_{spr}, t_N)$ does the same as replan except that *btplan* uses an order among planning rules and generates the next plan expression according to that order. Finally, $explan(t_{ip})$ executes the individual plan expression denoted by the term t_{ip}. We assume that the execution of a plan has some external effects. The internal mental effects should be realized using the update statement explicitly.

In this paper, we do not consider the semantics of this deliberation language since we are only interested in the implementation of the deliberation process and how autonomous agent properties related to the agent's decision making ability can be implemented. The semantics for this language is an extension of the semantics of the meta-language already presented in [5].

3.4 Examples of Deliberation processes

A program that implements a deliberation cycle may be a simple fixed loop consisting of the following steps:

1- Select a goal
2- Update the intention base with the selected goal
3- Select an intention
4- Generate plans to achieve the intention
5- Select a plan to execute
6- Execute the plan
7- Select a goal reasoning rule
8- Update the goal base with the body of the selected rule
9- Select an intention reasoning rule
10- Update the intention base with the body of the selected rule
11- Select a plan reasoning rule
12- Update the plan base with the body of the selected rule

This loop illustrates very clearly that the deliberation process consists of two parts. One part is the selection of goals, intentions, and plans finished with the execution of plans (steps 1 to 6). The second part deals with the reconsideration of goals, intentions, and plans (steps 7 to 12). The first part is closely related to planning. According to this example of the deliberation process reasoning rules are not used to update the goal base, the intention base, or the plan base before planning or executing them. This means that we only generate a plan, which is immediately executed. Only after this execution, reasoning rules are selected and mental bases are updated. Let Γ denote the goal base, Δ the intention base, Π the plan base, and Λ_g the goal rule base, Λ_i the intention rule base, and Λ_p the plan rule base. The above deliberation cycle can be implemented using the deliberation language as follows:

```
WHILE ($\neg empty(\Gamma)$) DO
BEGIN
   selgoal($\Gamma, f_c, \gamma_{ig}$);
   update($\Delta, \gamma_{ig}$);
   selint($\Delta, f'_c, \delta_{ii}$);
   plan($\delta_{ii}, t_N$);
   selplan($\Pi, f''_c, \pi_{ip}$);
   explan($\pi_{ip}$);
   selrule($\Lambda_g, \Gamma, \lambda_g$);
   update($\Gamma, body(\lambda_g)$);
   selrule($\Lambda_i, \Delta, \lambda_i$);
   update($\Delta, body(\lambda_i)$);
   selrule($\Lambda_p, \Pi, \lambda_p$);
   reviseplan($head(\lambda_p), body(\lambda_p)$);
END
```

Using the deliberation language, various types of domain dependent deliberation processes can be specified as well. For example, let Π be the term that denotes the plan base, Ω be the set of planning rules relating goals with plans (different rules than plan reasoning rules), *trans* be the criterium to select a plan that is suitable to perform the transport task, $cost(\pi)$ be the term that determines how expensive is the plan that is assigned to plan π, and $gain(\pi)$) be the term that determines the utility of plan π. A part of a deliberation cycle can be specified by the following expressions of the deliberation language.

```
selplan($\Pi$, $trans$, $\pi$)
WHILE (cost($\pi$) > gain($\pi$)) DO
BEGIN
   btplan($\pi, \Omega$, $max\_length$);
END
IF gain($\pi$) > cost($\pi$) THEN explan($\pi$)
```

This part of a deliberation cycle initially selects a plan π to perform the transport task. While the cost of this plan is higher than the utility of the transport task, it attempts to find a new plan by backtracking in the space of possible plans. When it finds a cost effective plan it will execute the plan.

4 Deliberation Properties

The properties of agent decision behavior are determined by the way the deliberation process is implemented. In this section, we consider these properties from the programming point of view and examine how to implement a deliberation process that satisfies certain properties, i.e. how to implement a certain specification of agent decision behavior. In order to answer this question, we consider static and dynamic constraints on the agent decision behavior formulated in the BDI formalism [4, 15]. In the BDI formalism, the operators B, G, and I are used to express agents' beliefs, goals, and intentions, respectively. Moreover, temporal operators from CTL* (Computational Tree Logic) are used to express the dynamics of agents' mental attitudes in terms of operators A to express 'for all possible futures', E 'for some possible futures', U 'until', X 'next', F 'sometimes in future', and G to express 'always in future' [15].

Due to space limitations we will not consider a complete set of constraints but rather exemplify the implementation issues using two of the most commonly used constraints. Those are the static property called realism and the dynamic property called open-minded commitment strategy. Using the BDI formalism, these two properties are expressed by the following axioms:

$$G_i(\varphi) \rightarrow B_i(\varphi)$$
$$I_i(AF\varphi) \rightarrow A(I_i(AF\varphi) \bigcup B_i(\varphi) \vee \neg G_i(EF\varphi))$$

The first (realism) axiom characterizes the compatibility between agents beliefs and goals and states that the goal of agents should be compatible with their beliefs. For example, if the agent i wants p to be true in all possible futures, written as $G_i(AGp)$, then agent i should believe it is possible that p is true always and for all possible futures. If agent i does not have this belief, then he wants something which he does not believe is achievable, which is clearly an unrealistic behavior. In order to implement a deliberation cycle such that agents' decision making behavior satisfies this property, we should guarantee that the selection of a goal and committing to it as an intention can only take place when the goal is derivable from agent's belief.

We assume that the initial agent's belief and goal bases are compatible and illustrate how this compatibility relation can be maintained through the deliberation process. In the previous section, the statement $update(t_{sg}, t_{ig})$ is proposed which, when executed, updates the set of goals denoted by t_{sg} (goal base) with the individual goal denoted by t_{ig}. For example, let the term t_{sg} denote the goal base $\{AFp, \neg EGq\}$. Then, the execution of $update(t_{sg}, EGq)$ modifies the goal base resulting in the new goal base $\{AFp, EGq\}$. In order to implement an agent in such a way that its decision making behavior satisfies the realism axiom, we should guarantee that every occurrence of the statement $update(t_{sg}, t_{ig})$ is conditionalized with corresponding beliefs. Let Σ be the belief base of the agent and Γ be the goal base of the agent. Then, every occurrence of the statement $update(\Gamma, \alpha)$ should be replaced with the following conditional statement:

$$\texttt{IF } provable(\Sigma, \alpha) \texttt{ THEN } update(\Gamma, \alpha)$$

The second (open-minded commitment strategy) axiom characterizes the decision making behavior of agents regarding their commitments to their intentions and the conditions under which the intentions can be dropped. In particular, it states that agents remain committed to their future directed intentions (i.e. intentions of the form $AF\phi$) until they believe their intentions are reached or they do not desire to reach them in the future anymore. Following the programming constructs a future directed intention can be dropped by the instruction $update(t_{si}, \neg t_{ii})$ where t_{ii} is a term denoting a formula of the form $AF\phi$. Note that updating the intention base by the negation of a formula is equivalent with dropping the intention expressed by the formula. In order to program the open-minded commitment strategy, we should guarantee that every occurrence of $update(t_{si}, \neg t_{ii})$, where t_{ii} is a term denoting a future directed formula, is conditionalized with corresponding beliefs and goals. Let Σ, Γ, and Δ be the belief base, the goal base, and the intention base of the agent. Then, every occurrence of the statement $update(\Delta, \neg AF\alpha)$ should be replaced with the following conditional statement:

$$\texttt{IF } provable(\Sigma, \alpha) \ \vee \ \neg provable(\Gamma, EF\alpha) \texttt{ THEN } update(\Delta, \neg AF\alpha)$$

The above examples show us a kind of general strategy on implementing the BDI constraints in the deliberation cycle. Each update of a goal or intention is made conditional upon some beliefs or intentions. One could, of course, integrate these conditions in the semantics of the update function itself. However, this would mean that the agent has a fixed commitment strategy ingrained in it. By incorporating it as conditions in the deliberation program we can still (dynamically) change the agent.

5 Conclusion

In this paper we have shown how some characteristic constraints that determine the autonomy of the agent can be programmed on the deliberation level of the agent. In our opinion this shows that autonomy is closely related to the deliberation level and its properties of the agent. By programming this level explicitly it becomes clear that many choices concerning the autonomy are still open (not restricted by the BDI constraints). Most of the choices have to do with the order in which different mental attitudes are updated and with which frequency they are updated.

Although we did not mention this point above it has also shown us the difficulty of implementing some of the constraints. The second constraint we implemented suggests that an intention can be dropped whenever the agent does not have the goal that φ will be true in some possible future anymore. Although we just stated that this can be tested by using the *provable* predicate, it will be very difficult in practice to prove $EF\varphi$ for any formula φ. This is due to the fact that the future is in principle infinite and it is impossible to check whether in one possible future at some point (possibly infinitely far in the future) φ can be true. Once the *provable* predicate returns false, the condition for dropping the intention becomes true. Thus in practice there will hardly ever be a restriction on dropping an intention.

Besides this computational criticism on the BDI constraints one might also wonder whether they specify exactly the right intuition. Especially the commitment strategies quantify over possible futures. It is very hard to check all possible futures, especially in systems that contain multiple agents and where the environment is uncertain. In future research we hope to explore some more realistic commitment specifications which tie the commitment to an intention to the possible futures that the agent has an influence on itself (through its own plans). Also, we may consider commitment strategies in terms of bounded temporal operators such as bounded eventually. These restrictions of the general commitment strategy as given in BDI theory also render these strategies computationally feasible. I.e. they restrict the possible future traces to a finite number of traces of finite length.

References

1. C. Boutilier. Towards a logic for qualitative decision theory. In *Proceedings of the Fourth International Conference on Knowledge Representation and Reasoning (KR'94)*, pages 75–86. Morgan Kaufmann, 1994.
2. M. Bratman. *Intention, plans, and practical reason.* Harvard University Press, Cambridge Mass, 1987.
3. J. Broersen, M. Dastani, Z. Huang, and L. van der Torre. Trust and commitment in dynamic logic. In *Proceedings of The First Eurasian Conference on Advances in Information and Communication Technology (EurAsia ICT 2002)*, volume 2510 of *LNCS*, pages 677–684. Springer, 2002.
4. P. Cohen and H. Levesque. Intention is choice with commitment. *Artificial Intelligence Journal*, 42(3):213–261, 1990.
5. M. Dastani, F. de Boer, F. Dignum, and J.-J. Meyer. Programming agent deliberation: An approach illustrated using the 3apl language. In *Proceedings of The Second Conference on Autonomous Agents and Multi-agent Systems (AAMAS'03)*, Melbourne, Australia, 2003.
6. M. Dastani, Z. Huang, J. Hulstijn, and L. van der Torre. BDI and QDT. In *Proceedings of the Workshop on Game Theoretic and Decision Theoretic Agents (GTDT2001)*, Stanford, 2001.
7. J. Doyle and R. Thomason. Background to qualitative decision theory. *AI magazine*, 20:2:55–68, 1999.
8. K. V. Hindriks, F. S. D. Boer, W. V. der Hoek, and J.-J. C. Meyer. Agent programming in 3apl. *Autonomous Agents and Multi-Agent Systems*, 2(4):357–401, 1999.

9. R. C. Jeffrey. *The Logic of Decision.* McGraw-Hill, New York., 1965.
10. D. Kinny. *Fundamentals of Agent Computation: Theory and Semantics.* Australia, 2001.
11. J.-J. Meyer, W. van der Hoek, and B. van Linder. A logical approach to the dynamics of commitments. *Artificial Intelligence*, 113(1-2):1–41, 1999.
12. J. Pearl. From conditional ought to qualitative decision theory. In *Proceedings of the Ninth Conference on Uncertainty in Artificial Intelligence (UAI'93)*, pages 12–20. John Wiley and Sons, 1993.
13. A. Rao and M. Georgeff. Deliberation and its role in the formation of intentions. In *Proceedings of the Seventh Conference on Uncertainty in Artificial Intelligence (UAI-91)*, pages 300–307, San Mateo, CA, 1991. Morgan Kaufmann Publishers.
14. A. Rao and M. Georgeff. Modeling rational agents within a bdi architecture. In *Proceedings of Second International Conference on Knowledge Representation and Reasoning (KR'91)*, pages 473–484. Morgan Kaufmann, 1991.
15. A. Rao and M. Georgeff. Decision procedures for BDI logics. *Journal of Logic and Computation*, 8:293–342, 1998.
16. L. Savage. *The foundations of statistics.* Wiley, New York, 1954.