THE BDI MODEL OF AGENCY AND BDI LOGICS

Technical report written by:

Camilo THORNE

L.O.A. - C.N.R.

Trento, May 2005.

Contents

Philosophical Prolegomena

The *BDI* or *Belief-Desire-Intention Model of Agency* is concerned with formally modelling practical reason – i.e. with formalizing psychological and philosophical (mainly coming from the philosophy of mind and of action) accounts and explanations of agency, action, intention, belief, will, deliberation, means-end reasoning, etc. Practical reason is embodied in agents (say, humans) capable of pursuing and thus commiting to a feasible goal (a particular action) through careful planning of the means, of the preliminary conditions and actions conducing to that goal. A brief overview of these notions will help us to better grasp them. A more thorough discussion can be found in Searle (cf. [14]) and, overall, Bratman (cf.[2]).

Beliefs, Desires, Intentions

Desires, *intentions* and *beliefs* are said to be intentional mental states (as opposed to, say, pain or pleasure). The latter are assumed to describe our perceiving the reality – through sense data. They encompass our (common-sense or subtle and theoretical) knowledge about the world (be it external or internal). Together they form what is called a *conceptual scheme*. They are subject to revision, meaning that they can change (can be rejected or added) The former, that is, desires and intentions, can be seen as creatures of more or less the same kind, albeit with some subtle differences. For desires consist in our willing a certain state of affairs (or possible world) to obtain, while intention is more concerned with our committing ourselves to obtain this state of affairs otherwise called *goal*.

Deliberation

By *deliberation* we understand what the literature calls the *practical syllogism* – our inferring an intention from a a set of beliefs and desires. That is, the choice of a feasible desire. A *decision* consists in the last step of this inference by which we choose one among a myriad of desires and potential intentions and is thus a notion closely knit to that of intention. An *action* is (intuitevely) defined, if at all, as the execution, effect or consequence of a decision (what is executed). It

can be intentional, if immediate consequence of an intention

Planning

By *planning* we understand the drawing of a certain sequence of decisions (i.e. means-end reasoning) to attain a goal, for that may not be simple affair and may require a big number of decisions and actions to be performed. We can also say that intentions are partial-plans, for they determine the decisions that compose them.

Practical Reason

Hence, *practical reason* or *will* consists in the combination of deliberating and planning. We can now state the important facts that hold (according to *BDI* theory) in the realm of practical reason:

- Desires and beliefs range over states of affairs, while intentions range over actions and by extension, plans.
- Intentions are persistent, whereas desires can be dropped at any time.
- Intentions need not be holded forever.
- Intentions drive means-end reasoning.
- Beliefs constrain desires.
- Intentions constrain future deliberation and planning.
- Intentions influence beliefs upon which future practical reason is based.
- Intentions imply a degree of commitment to a goal.
- Intentions, beliefs and desires are required to be consistent. Beliefs are required to be consistent with other beliefs. Intentions with goals and beliefs and goals (analogously) with beliefs and intentions. We can thus say (so to speak) that the former are strongly consistent while the latter are weakly consistent [1]. This condition is assumed to imply that of rationality.
- Intentions, beliefs and desires need not be complete or, to put it simply, all-encompassing [2].
- Beliefs are subject to revision.
- Intentions and hence plans can be reconsidered.

[1] Hence, formally, any theory representing them through modal or non modal formulae has to be consistent.

[2] Again, the theory need not be complete nor decidable.

Agents

We are now capable of saying what is an *agent* in the most general case. By it we understand an entity (a moral or a legal person, a computer program) that is capable of reacting to a certain environment through its performing a certain number of actions over which it can exert some kind of control. Or equivalently, that has some kind of will – i.e. a practical reason, and thus mental states, as well as the capability both of deliberating and planning. We say further that an agent is *rational* if his actions, decisions, plans and intentions are consistent or coherent with his beliefs and desires (meaning by that, not contradictory)[3]. There are three main types of agents, following their *commitment strategies*, namely:

- *Blindly-minded agents* are agents that are blindy over-commited to their basic beliefs and intentions or desires, which they never put in question nor revise. They can be seen as bold, or better, as fanatical agents. They follow a single, identical, plan under all situations, under all states of affairs (even though the surrounding world may change).

- *Single-minded agents* are agents whose (derived) intentions may change due to belief revision. They are thus cautiously commited to their intentions. They can be seen as cautious agents. They are able to modify a plan if needed.

- *Open-minded agents* are agents that revise their beliefs and that change both their desires and derived intentions accordingly. They are thus undercommited to their intentions. These are over-cautious and hesitating agents. They can modify a plan or just build one anew.

Historical Remarks

Historically, this theory was first conceived by Aristotle in his *Ethica Nichomachea* and *De Anima*. According to him, practical reason is analog to theoretical reason. Practical reason (*tò logistikón*) is structured as follows: Sensation (*aísthesis*) gives rise (through imagination and memory) to desire (*órexis*) – as concepts or ideas alike by the understanding. Desires are then moulded into assertive action (into intention) through decision (*prohaíresis*) – like judgements do with concepts. Finally deliberation (*boūlēsis*) enhances decision-making by applying beliefs or knowledge to bare desire and to intention by means of a practical syllogism (*syllogismós pragmatikós*) between beliefs, desires and intentions – like theoretical syllogisms. Finally, desires are irrational (*alogikós*) and therefore shared with animals while decision and deliberation is rational (*logikós*)

[3]This is really a necessary but not sufficient condition of rationality, but we prefer to follow the philosophical tradition on this point.

[4]. Little is said though about planning or means-end reasoning. Kant's essay *Metaphysical Foundations of the Morals* and Hume's *Treatise on Human Nature* provide further refinements to Aristotle's doctrine. A more recent development can be found in Davidson's *Essays on Actions and Events*.

A Useful Scheme

The following table summarizes the relationships that exist between the terms used, on the one hand, on the philosophical accounts of agency, and the terms found, on the other, in the description of real-world implementations and in their formal specification. These analogies are important in that they help us to understand which are the concepts that the formal operators and actual software architectures capture and will be henceforth very useful. Note that goals are not *sensu stricto* equivalent to desires, because goals depend on desires and overall on intentions – the latter are necessary but not sufficient conditions of the former [5].

Philosophy	Belief	Desire	Intention	Agent
Theory	Belief	Goal	Intention	Agent
Practice	KB	Event	Running Plan	Embedded System

[4]It goes by itself that this characterization of practical reason is somewhat outdated, for animals are also capable of deliberating and of deciding up to some point – rationality is, again, quite a difficult concept to grasp.

[5]One may want, by way of a whim, to become a car racer, but that does not imply our having that as a goal, since whims usually wane away quickly!

Chapter 1

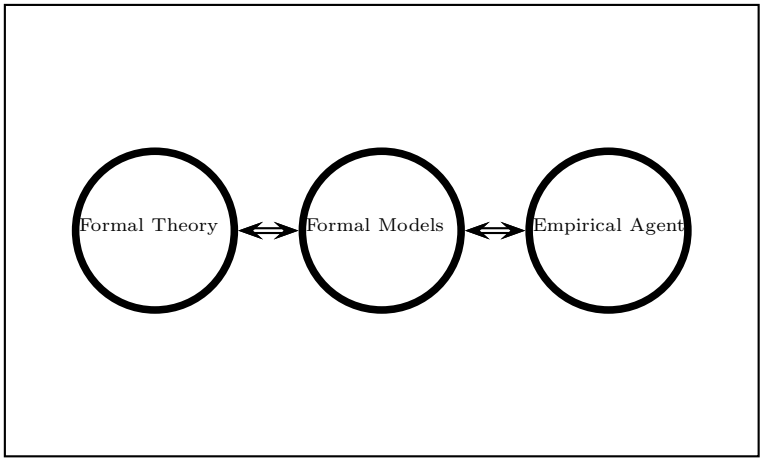
Rao and Georgeff's BDI Logics

We develop here in detail Rao and Georgeff's version of BDI logics as defined in [12], and which provide the common basis upon which all the contemporary accounts (mainly focused in MAS) are built. We will concentrate on their syntax and their basic semantics.

BDI logic are a family of temporal/doxastic/epistemic logics, i.e. a kind of modal logics. Temporal logics were developed mainly as a formal validation tool for dynamic systems. That is, as an *a posteriori* means to model (to formally represent) a dynamical system (intuitevely a software platform or environment whose state may change or evolve through time) in order to prove that it veryfies some critical properties (like mutual exclusion and the queueing of parallell, concurrent, processes consuming the same ressource). This technique is called model-checking, for it builds a model M from both the formal description or representation of the system, a logical temporal theory, say, Γ, and the temporal statement A that encodes the property, and then checks if $M \models \Gamma$ implies $M \models A$ and thus if $\Gamma \models A$. In the present case, it proves to be quite a fine tool to model in the above sense a software agent and by extension, to capture our intuition of (and thus formalize) our philosophical and psychological accounts of agency. It can be thought as a simplified version of *LORA* (*Logic for Rational Agents*) in which we do not quantify any more over agents. It models therefore decision-making in individual agents and drops thier being able to collaborate in view of a common goal. And as an extension of *CTL* (*Computational Tree Logic*), since worlds are trees to which epistemic and doxastic operators of intention, belief and goal have been added, giving way to three distinct accesibility relations. As in the case of propositional alethic modal logic we deal with a whole family of logics, capturing different properties of the accessibility relations through the addition of axioms. We assume that the base system is similar to the KD45 system, which is assumed to best capture the notions of belief and knowledge –

i.e. accessibility should be serial (D), transitive (4) and euclidian (5)[1] For more details, we send the reader to Rao and Georgeff's report (cf. [12]).

The idea that this theory follows can be summarized by a little diagram. The logics serve to characterize kripke-style models that in thier turn are assumed to capture (formally) the concepts and notions discussed in the first section. This has to be bore in mind throughout the whole chapter, however complex the corresponding intuitive notion may be:



1.1 Syntax

As already said, BDI comprises both the usual CTL modalities plus epistemic and doxastic ones. Moreover, The set of formulae is divided in two, those of state formulae and those of path formulae – i.e. of formulae capable of being true at solely one instant and formulae capable of being true along a whole (possibly infinite) sequence of instants.

Definition 1.1.1 *The set F of the formulae of BDI logic is defined by the grammar:*

- $< sort >::= O|E.$

[1] The due axioms are thus:
(K) $\Box(A \rightarrow B) \rightarrow (\Box A \rightarrow \Box B)$,
(D) $\Box A \rightarrow \Diamond A$,
(4) $\Box A \rightarrow \Box\Box A$ and
(5) $\Diamond A \rightarrow \Box\Diamond A$.

- $< object - var >::= x_1^O|...|x_n^O,\ n \geq 0.$
- $< event - var >::= x_1^E|...|x_m^E,\ m \geq 0.$
- $< pred >::= P_1|...|P_k,\ k \geq 0.$
- $< variable >::=< object - var > | < event - var >.$
- $< atom >::=< pred > (< var >, ..., < var >).$
- $< state-form >::= succeeded(< event-var >)|failed(< event-var >)$
  $| < atom > |\neg < F_S > | < state - form > \vee < state - form >$
  $|\exists x^{<sort>} < state - form > |\mathbf{Bel} < state - form >$
  $|\mathbf{Go} < state-form > |\mathbf{In} < state-form > |optional < path-form > .$
- $< path - form >::=< state - form > |\neg < path - form >$
  $| < path - form > \vee < F_P > | < path - form > \mathbf{U} < path - form >$
  $|\Diamond < path - form > |\bigcirc < path - form > .$
- $< form >::=< state - form > | < path - form >.$

Notice that the language is two-sorted. Variables of the form x^O (i.e. of type O) range over a set of objects, while those of the form x^E range over a set of events. To simplify things a little, the former will be called *variables* and the latter *event variables*. No constants or function symbols are introduced to simplify the semantics. The remaining functors are introduced as follows:

Definition 1.1.2 *We put:*

- $A \rightarrow B =_{df} \neg A \vee B.$
- $A \wedge B =_{df} \neg(\neg A \vee \neg B).$
- $\forall x^S A =_{df} \neg\exists x^S \neg A.$
- $inevitable A =_{df} \neg optional \neg A.$
- $\Box A =_{df} \neg\Diamond\neg A.$
- $done(x^E) =_{df} succeded(x^E) \vee failed(x^E).$
- $succeeds(x^E) =_{df} inevitable(\bigcirc succeded(x^E)).$
- $fails(x^E) =_{df} inevitable(\bigcirc failed(x^E)).$
- $does(x^E) =_{df} inevitable(\bigcirc done(x^E)).$
- $\bot =_{df} A \wedge \neg A.$
- $\top =_{df} \neg\bot.$

Intuitively, modal operators like $\bigcirc$ are read as 'next', $\Box$ as 'always', **U** as 'until', **Bel** as 'believes', **In** as 'intends' and **Go** as 'has as a goal' or 'desires'.

Remark 1.1.1 *Any formula containing an ocurrence of the inevitable modality and an action predicate (or likewise β of inevitable) will be called an* A-formula *(likewise, a* E-formula*). A-formulas or E-formulas will be called* O-formulas *and written α, β, γ, etc. (lower-case Greek letters). The 'does' predicate is an expedient to speak about actions, in which events obtain.*

1.2 Semantics

The main idea governing *BDI* logics model theory is to conceive worlds as complex structures, i.e. time trees, rather than simple states. This formalizes reasonably well planning and decision-making. For planning assumes a span of possibilities to assess. Each time point or instant constitutes a different state of affairs that can cause branching sequences of effects and that presupposes a single sequence of causes. Hence time, relatively to an instant t_0, is linear regarding the past and branching regarding the future. This will allow us in turn to quantify over both their paths and their nodes. Before saying what is a model we need to define what is a possible world.

Definition 1.2.1 *Let T be a set ot time points, $\prec$ be a total, transitive and backward-linear relation over T, called the* branching time *relation and E be a set of events. A* world *is a structure $w = (T_w, \prec_w, S_w, F_w)$ where:*

- $T_w \subseteq T$.
- $\prec_w = \prec \restriction T_w$.
- $S_w : T_w \times T_w \to E$.
- $F_w : T_w \times T_w \to E$.
- *S_w, F_w are injective and such that $Dom(S_w) \cap Dom(F_w) = \emptyset$.*

An order can be defined between worlds, namely:

Definition 1.2.2 *Let T be a set ot time points, $\prec$ be a total, transitive and backward-linear relation over T, called the* branching time *relation and E be a set of events. Let w, w' be two worlds. Then $w \sqsubseteq w'$ iff*

- $T_w \subseteq T_{w'}$.
- $\prec_w \subseteq \prec_{w'}$.
- $S_{w'} = S_w \restriction T_{w'}$.
- $F_{w'} = F_w \restriction T_{w'}$.

Now we can define models:

Definition 1.2.3 *A* model *is a structure $M = (D_O, D_E, T, W; \prec, I, B, G; \Phi)$ where:*

- D_O *is a non-empty set called* domain of objects.
- D_E *is a non-empty set called* domain of events.
- T *is a non-empty set of* time points.
- $\prec \subseteq T \times T$ *is the branching time relation.*
- W *is a non empty set of worlds over* T.
- $I \subseteq W \times T \times W$ *an* intention accessibility *relation.*
- $B \subseteq W \times T \times W$ *a* belief accessibility *relation.*
- $G \subseteq W \times T \times W$ *a* goal accessibility *relation*
- $\Phi : \mathcal{S}_{\mathcal{R}} \times W \times T \rightarrow \bigcup_{i \in \mathbb{N}} \wp(D_O^i)$ *is an* interpretation *function for predicate symbols.*
- *We denote by* C *their class.*

The following two properties will prove quite useful:

Definition 1.2.4 *Let* M *be a model. An ordered couple* $(w, t) \in W \times T$ *is called a* situation *and will be henceforth written* w_t.

Definition 1.2.5 *Let* M *be a model. A* fullpath *is a possibly infinite sequence of situations, noted* $< w_{t_1}, w_{t_2}, ... >$ *such that for all* $i \geq 0$, $(t_i, t_{i+1}) \in \prec_w$.

Remark 1.2.1 *We note that:*

- *Worlds are trees, i.e. non-directed, connected and acyclic graphs. In other words, the (binary) relation* $\prec$ *defines a tree structure over the set of time points.*
- *We shall note* B_t^w *the set of belief-accessible worlds, i.e.* $B_t^w = \{w' \in W | (w, t, w') \in B\}$. *And similarly for* I *and* G.

The following fact is also relevant:

Fact 1.2.1 *Let Path(w) denote the set of all paths of a world w, for any given world w in a model M. Then, for any two worlds w and w' in a model M we have that* $w \sqsubseteq w'$ *iff* $Path(w) \subseteq Path(w')$.

Given a model M, the accessiblity relations can satisfy a certain number of properties. They can, on the one hand, satisfy the properties that any binary relation can verify, like seriality, transitivity, symmetry, etc. And on the other hand, they can hold inclusion relations between them, since we are in a multi-modal framework. Normal inclusion is just set-theoretical inclusion. But structural inclusion holds between the paths of the worlds, that are time trees, and is somehow relativized, moreover, to a given time point:

Definition 1.2.6 *Let M be a model and let R, $R' \in \{I, B, G\}$. Then: $R \subseteq_{struct} R'$ iff for any $w, w' \in W$ and any $t \in T_w$, $w' \in R'^w_t$ implies that there is some $w'' \in R^w_t$ such that $w'' \sqsubseteq w'$.*

This latter semantic property is known usually as *strong realism*, the former (usual inclusion) being known as *realism*.

Now, to build a semantics the symbols, beginning with the variables, must be mapped to models:

Definition 1.2.7 *An* assignation *is a function $v : V \to D_O \cup D_E$ such that:*

$$v(x^S) = \begin{cases} d \in D_O \text{ if } S = O \\ e \in D_E \text{ otherwise.} \end{cases}$$

Given v an assignation function, we shall note as usual v^* the assignation that coincides with v but over the (first-order) variable x of type O. And analogously for those of type E (i.e. event variables). Note that the language does not contain any constant symbol. Next, in order to define truth and validity we need a satisfaction relation between models and formulae. It will be defined relatively to an assigation v and a situation w_t or a fullpath $< w_{t_1}, w_{t_2}, ... >$ and written $\models^v_{w_t}$ or $\models^v_{<w_{t_0}, w_{t_1}, ...>}$.

Definition 1.2.8 *The* satisfaction *relation is defined by induction on $\mathcal{F}$ as follows – first on path formulas and then on state formulas:*

- $M \models^v_{w_t} P(x_1, ..., x_n)$ *iff* $(v(x^O_1), ...v(x^O_n)) \in \Phi(P, w, t)$.
- $M \models^v_{w_t} \neg A$ *iff* $M \not\models^v_{w_t} A$.
- $M \models^v_{w_t} A \vee B$ *iff* $M \models^v_{w_t} A$ *or* $M \models^v_{w_t} B$.
- $M \models^v_{w_t} \exists x^O A$ *iff* $M \models^{v^\star}_{w_t} A$ *for some* $d \in D_O$.
- $M \models^v_{w_t} \exists x^E A$ *iff* $M \models^{v^\star}_{w_t} A$ *for some* $e \in D_E$
- $M \models^v_{<w_{t_0}, w_{t_1}, ...>} A$ *iff* $M \models^v_{w_{t_0}} A$.
- $M \models^v_{<w_{t_0}, w_{t_1}, ...>} \bigcirc A$ *iff* $M \models^v_{<w_{t_1}, ...,>} A$
- $M \models^v_{<w_{t_0}, w_{t_1}, ...>} \Diamond A$ *iff for some* $i \geq 0$ *such that* $M \models^v_{<w_{t_i}, ...,>} A$.
- $M \models^v_{<w_{t_0}, w_{t_1}, ...>} A \mathbf{U} B$ *iff either of these conditions hold:*
  1. *For some* $i \geq 0$ *such that* $M \models^v_{<w_{t_i}, ...>} B$ *and for all* $0 \leq j < i, M \models^v_{<w_{t_j}, ...,>} A$.
  2. *For any* $j \geq 0, M \models^v_{<w_{t_j}, ...>} A$.
- $M \models^v_{w_{t_0}} optional A$ *iff for some fullpath* $< w_{t_0}, w_{t_1}, ... >$, $M \models^v_{<w_{t_0}, w_{t_1}, ...>} A$.

- $M \models^v_{w_t}$ *succeded*(x^E) *iff for some time point* t', $S_w(t', t) = v(x^E)$.
- $M \models^v_{w_t}$ $failed(x^E)$ *iff for some time point* t', $F_w(t', t) = v(x^E)$.
- $M \models^v_{w_t}$ **Bel**A *iff for any* $w' \in \mathcal{B}^w_t$, $M \models^v_{w'_t} A$.
- $M \models^v_{w_t}$ **In**A *iff for any* $w' \in \mathcal{I}^w_t$, $M \models^v_{w'_t} A$.
- $M \models^v_{w_t}$ **Go**A *iff for any* $w' \in \mathcal{G}^w_t$, $M \models^v_{w'_t} A$.

The usual definitions of *scope* of a quantifier or a connective hold. The same thing applies to substitutions, sub-formulae, free and bound variables. A formula is thus said to be a *sentence* exactly when it contains no free variables. Truth and validity are then defined as usual.

Definition 1.2.9 *Let* A *be a sentence, then:*

- A *is said to be* true in a time point t *relatively to a a world* w *and a model* M *iff for any assignation* v *we have that* $M \models^v_{w_t} A$. *In which case we write* $M \models_{w_t} A$. *We can also say, alternatively, that* A *is* true in a situation w_t *relatively to a model* M, *and then extend this notion to fullpaths.*
- A *is said to be* true in a world w *relatively to a model* M *iff for any time point* t *in* w *we have that* $M \models^v_{w_t} A$. *In which case we write* $M \models_w A$.
- A *is said to be* true in model M *iff* A *is true in any world* w *relatively to* M. *In which case we write* $M \models A$. *We then extend this notion to set of sentences.*
- A *is said to be* valid in a class $C' \subseteq C$ *of models iff* A *is true in any model of the class. In which case we write* $C' \models A$.
- A *is said to be* valid *iff* A *is valid in any in any model class. In which case we write* $\models A$.

Which allows us to prove immediately for instance that:

Proposition 1.2.1 $\models \top \mathbf{U} A \leftrightarrow \Diamond A$.

Definition 1.2.10 *Let* Γ *be a set of sentences and* A *a sentence. We say that* A *is a* consequence *of Gamma and write* $\Gamma \models A$ *iff for any class* $C' \subseteq C$ *we have that* $C' \models \Gamma$ *implies that* $C' \models A$.

Thanks to this, we can prove, for example, the semantic counterpart of the *modus ponens* rule:

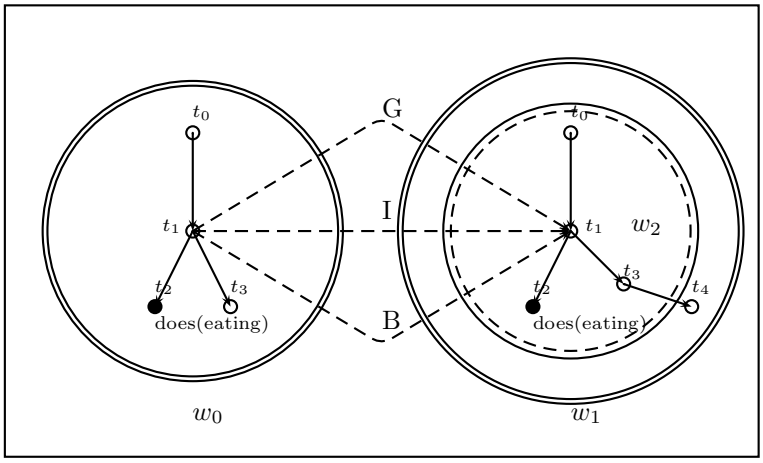
Proposition 1.2.2 $\Gamma \models A \rightarrow B$ *and* $\Gamma \models A$ *implies that* $\Gamma \models B$.

1.3 An Example

In this example we will build a model M_1 satisfying the formula:

$$\mathbf{In}(optional(\Diamond(does(eating)))) \rightarrow \mathbf{Go}(optional(\Diamond(does(eating))).$$

And exhibiting the strong realism property – i.e. structural inclusion of accessibility relations relatively to a situation:



M_1

Clearly

$$M_1 \models^{v_0}_{w_0 t_1} \mathbf{In}(optional(\Diamond(does(eating)))$$

and

$$M_1 \models^{v_0}_{w_0 t_1} \mathbf{Go}(optional(\Diamond(does(eating))).$$

Note by the way that:

- $w_2 \sqsubseteq w_1$.
- $w_0 \neq w_1$.

- $t_0 \prec t_1 \prec t_2 \prec t_3 \prec t_4$.

That is, that the intention modality is so to speak constraining future desires (goals) and beliefs.

1.4 Some Correspondence Theory

Correspondence theory treats about the relationships that exist between model classes and modal formulae. Indeed, modal formulae can capture the properties of the accesiblity relation associated to its main modality – i.e they prove to be equivalent to a condition on the models, that can be dubbed its forml meaning. By adding them as (proper) axioms to a modal logical system we can thus constrain the class of models associated to it by some suitable semantics. That is, define a subclass of models, namely that in which they are valid. The following table summarizes the equivalences (their proofs can be found in [17] and [10]) for the belief, desire and intention modalities. We leave aside the greatest part of the temporal modalities (to which no accessibility relation is associated) exeept *inevitably*, for α in G-B and I-G stands for an O-formula, true along the sub-trees (or paths) that have a given time point as root:

Name	Modal Formula Scheme	Condition on Model
BK	$\mathbf{Bel}(A \rightarrow B) \rightarrow (\mathbf{Bel}A \rightarrow \mathbf{Bel}B)$	B non empty.
BD	$\mathbf{Bel}A \rightarrow \neg\mathbf{Bel}\neg A$	B is serial.
B4	$\mathbf{Bel}A \rightarrow \mathbf{BelBel}A$	B is transitive.
B5	$\neg\mathbf{Bel}\neg A \rightarrow \mathbf{Bel}\neg\mathbf{Bel}\neg A$	B is euclidian.
IK	$\mathbf{In}(A \rightarrow B) \rightarrow (\mathbf{In}A \rightarrow \mathbf{In}B)$	I non empty.
ID	$\mathbf{In}A \rightarrow \neg\mathbf{In}\neg A$	I is serial.
GK	$\mathbf{Go}(A \rightarrow B) \rightarrow (\mathbf{Go}A \rightarrow \mathbf{Go}B)$	G non empty.
GD	$\mathbf{Go}A \rightarrow \neg\mathbf{Go}\neg A$.	G is serial.
G-B	$\mathbf{Go}A \rightarrow \mathbf{Bel}A$	$G \subseteq B$.
I-G	$\mathbf{In}A \rightarrow \mathbf{Go}A$	$I \subseteq G$.
G-B*	$\mathbf{Go}\alpha \rightarrow \mathbf{Bel}\alpha$	$G \subseteq_{struct} B$.
I-G*	$\mathbf{In}\alpha \rightarrow \mathbf{Go}\alpha$	$I \subseteq_{struct} G$.

We will give a proof of the last equivalence in order to better display the idea behind. To begin, we need a useful lemma whose proof is given by Wooldridge (cf. [17]):

Lemma 1.4.1 *Let M be a model, w_t a situation and α an A-formula. Then, if $M \models^v_{w_t} \alpha$ and for any world w', $w' \sqsubseteq w$, $M \models_{w'_t} \alpha$.*

Theorem 1.4.1 *Let M be a model. Then $M \models \mathbf{In}\alpha \rightarrow \mathbf{Go}\alpha$ iff in every M we have that $I \subseteq_{struct} G$.*

Proof The right-left sense is the easiest. The left-right sense is more subtle.

(⇐) Suppose that there is a model M_0 satifying strong realism and such that for some world w_0 and time point t_0 in M_0 and some assignation v_0 we have that:

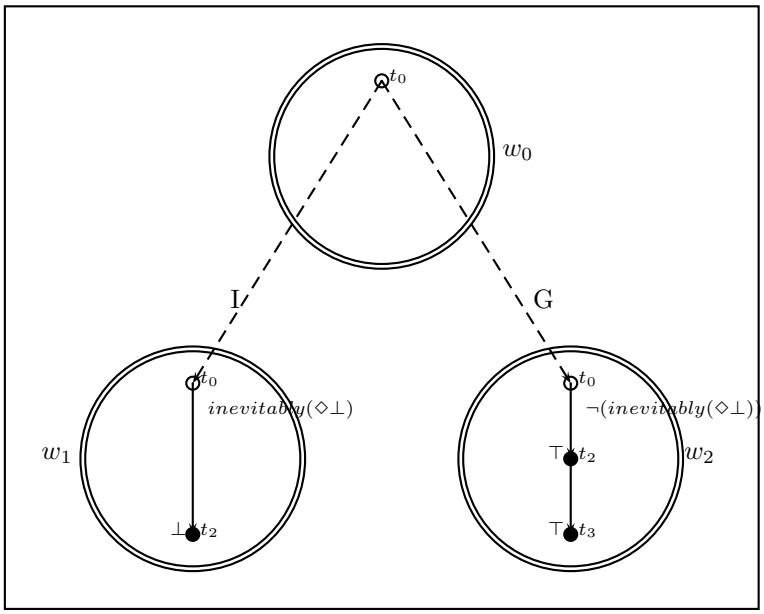
$$(*) M_0 \models^{v_0}_{w_{0_{t_0}}} \mathbf{In}\alpha$$

and

$$(**) M_0 \not\models^{v_0}_{w_{0_{t_0}}} \mathbf{Go}\alpha.$$

Now, $(**)$ implies that there is some $w_1 \in G^{w_0}_{t_0}$ such that $\not\models^{v_0}_{w_{1_{t_0}}} \alpha$. Which, since M_0 satisfies strong realism, entails that that there is some $w_2 \in I^{w_0}_{t_0}$ such that $w_2 \sqsubseteq w_1$. But, if $w_2 \in I^{w_0}_{t_0}$, then, by $(**)$ it follows that $\models^{v_0}_{w_{2_{t_0}}} \alpha$. Hence, by Lemma 1.4.1, $\models^{v_0}_{w_{1_{t_0}}} \alpha$. Contradiction. □

(⇒) We prove it by contraposition. We will build a model M_0 (and an assignation v_0) that does not satisfy strong realism and that will prove to be a counter-model of some instance of $\mathbf{In}\alpha \rightarrow \mathbf{Go}\alpha$. Lets put $\alpha = \mathit{inevitably}(\Diamond\bot)$.



We can clearly see that

$$M_0 \models^{v_0}_{w_{0_{t_1}}} \mathbf{In}\alpha$$

and

$$M_0 \not\models^{v_0}_{w_{0_{t_1}}} \mathbf{Go}\alpha$$

Which closes the proof. □

1.5 Axioms and Proof Theory

As a formal system (Hilbert-type), *BDI* logics, as any other family of first-order multi-modal logics, contains all first-order axioms (and theorems) and is closed by the rules of *modus ponens* and necessitation. Modalities are distributive over implication. Moreover, the deduction theorem holds for them. A proof of a formula A from a set of hypotheses Γ is the usual finite sequence of formulae where each term is either in Γ, an axiom, or the result of applying a deduction rule to former terms, and where A is the last term. The deducibility relation $\vdash$ is defined in terms of it. Theorems are formulae whose set of hypotheses is empty. We will not insist on these notions, but take them for granted. For more details, see Padmadahban (cf. [8]).

1.6 Soundness, Completeness, General Remarks

As *BDI* logics are mainly a modelling tool of agency, they do not need to be complete. They are however presumably sound [2]. Axiom schemata can presumably define different subclasses of models. They are, moreover, undecidable, as any other first-order logic. We will not insist either in developing a more detailed account, because that will require to go beyond this simple presentation. See [8] for results that hold only with respect to Kripke structures and that require our restricting ourselves to a language without temporal operators. We shall only say that any of the usual modal (alethic) axiom schemata has its belief, intention or desire counterpart, written, say, GT (i.e. GT = $\mathbf{Go}A \rightarrow A$).Modalities can be further *de dicto* or *de re* modalities.

BDI logics provide a logical specification (in the sense that computer scientists and engineers give to this term) of a software agent architecture. The need for a temporal logic comes from the fact that this system is assumed to be dynamic in nature. See Wooldridge (cf. [16]) for further insight.

[2] As is usual in modal logic, completeness and soundness is relative to some class $C' \subseteq C$.

Chapter 2

BDI Systems – An Informal Overview

BDI axiom schemata can be given too an intuitive meaning – for they indeed intend to capture formally (though not exacly), to model (in a broad sense) the concepts developed in the prolegomena about desires, intentions, goals, beliefs, actions and agency.

Goals are understood to be desires and no distinction is made between belief and knowledge (i.e. justified true belief, and a properly speaking different epistemic modality) and envelops further all kinds of knowledge whether theoretical or perceptual. We divide the axioms into three distinct groups:

2.1 Basic Axioms – BA system

Name	Modal Formula Scheme	Intuitive property
BK	$\mathbf{Bel}(A \to B) \to (\mathbf{Bel}A \to \mathbf{Bel}B)$	Belief implication closure
BD	$\mathbf{Bel}A \to \neg\mathbf{Bel}\neg A$	Belief consistency
B4	$\mathbf{Bel}A \to \mathbf{BelBel}A$	Belief positive introspection
B5	$\neg\mathbf{Bel}\neg A \to \mathbf{Bel}\neg\mathbf{Bel}\neg A$	Belief negative introspection
IK	$\mathbf{In}(A \to B) \to (\mathbf{In}A \to \mathbf{In}B)$	Intention implication closure
ID	$\mathbf{In}A \to \neg\mathbf{In}\neg A$	Intention consistency
GK	$\mathbf{Go}(A \to B) \to (\mathbf{Go}A \to \mathbf{Go}B)$	Goal implication closure
GD	$\mathbf{Go}A \to \neg\mathbf{Go}\neg A.$	Goal consistency
G-B*	$\mathbf{Go}\alpha \to \mathbf{Bel}\alpha$	Desire-belief compatibility
I-G*	$\mathbf{In}\alpha \to \mathbf{Go}\alpha$	Intention-desire compatibility

The compatiblity axioms deserve further commentary because they are involved in capturing the constrainment of future beliefs and desires by (present) intentions. This feature is modelled by structural inclusion (cf. the table summarizing *BDI* correspondence theory) – by the fact that intention accessible worlds contain both desire and belief accessible worlds, time being defined by

the $\prec$ order relationship over instants. The model from example in the preceding chapter illustrates this property in detail.

2.2 Intention Axioms – IA system

Name	Axiom Scheme	Intuitive property
A11	$\mathbf{In}A \rightarrow \mathbf{Bel}(\mathbf{In}A)$	Intention introspection
A12	$\mathbf{Go}A \rightarrow \mathbf{Bel}(\mathbf{Go}A)$	Goal introspection
A13	$\mathbf{In}A \rightarrow \mathbf{Go}(\mathbf{In}A)$	Desires about intentions
A14	$\forall x^E(\mathbf{In}(does(x^E)) \rightarrow does(x^E)$	Intentions leading to actions
A15	$\forall x^E(done(x^E)) \rightarrow \mathbf{Bel}(done(x^E))$	Awareness of primitive events
A16	$\mathbf{In}A \rightarrow inevitable(\Diamond(\neg\mathbf{In}A))$	No infinite deferral property

2.3 Commitment Axioms

Name	Axiom Scheme	Intuitive property
C1	$\mathbf{In}(inevitable(\Diamond A)) \rightarrow inevitable(\mathbf{In}(inevitable(\Diamond A))\mathbf{U}A)$	Blind-mindedness property
C2	$\mathbf{In}(inevitable(\Diamond A)) \rightarrow inevitable(\mathbf{In}(inevitable(\Diamond A))\ \mathbf{U}(A \vee \neg\mathbf{Bel}(optional(\Diamond A)))$	Single-mindedness property
C3	$\mathbf{In}(inevitable(\Diamond A)) \rightarrow inevitable(\mathbf{In}(inevitable(\Diamond A))\ \mathbf{U}(A \vee \neg\mathbf{Go}(optional(\Diamond A)))$	Open-mindedness property

2.4 The Systems

These are three:

1. $BDI_1 = BA + IA$+C1.
2. $BDI_2 = BA + IA$+C2.
3. $BDI_3 = BA + IA$+C3.

The intuitive idea behind is that $BA+IA$ formalize the concept or notion of deliberation, and by the same token the relationships that mental states (belief, desire, intention) bear upon each others. C1, C2, C2 model or describe planning, or at least the degree of commitment necessary for means-end reasoning.

2.5 Some Properties

The following follows immediately from the axioms:

Proposition 2.5.1 $\vdash \mathbf{In}A \rightarrow \mathbf{Bel}A$

The following follows immediately from Proposition 2.3 and the axioms:

Proposition 2.5.2 *We have:*

- $\vdash (\mathbf{Go}A \wedge \mathbf{In}(A \rightarrow B)) \rightarrow \mathbf{Go}B$.
- $\vdash (\mathbf{Bel}A \wedge \mathbf{In}(A \rightarrow B)) \rightarrow \mathbf{Bel}B$.
- $\vdash (\mathbf{Bel}A \wedge \mathbf{Go}(A \rightarrow B)) \rightarrow \mathbf{Bel}B$.

2.6 Brief Comparison with other Accounts and Systems

2.6.1 Padmanabhan

Consider the next table proposed by Padmanabhan in [10]. E2 is an important property, that captures Bratman's assymetry thesis, that is, the fact that one may intend something without being aware of it:

Name	Modal Formula Scheme	Intuitive property
E1	$\neg(\mathbf{In}A \wedge \mathbf{Bel}\neg A)$	Intention-belief consistency
E2	$\mathbf{In}A \wedge \neg\mathbf{Bel}A$	Intention-belief incompleteness
E3	$\mathbf{Bel}A \wedge \neg\mathbf{Go}A$	Transference property
E4	$\mathbf{In}A \wedge \mathbf{Bel}(A \rightarrow B)$ $\wedge\neg\mathbf{In}A$	Side effects property

Now, while E1 and E3 can be added to $BA + IA$, the addition of either E2 or E4 leads to a contradiction. For instance:

Proposition 2.6.1 $BA + IA + E2 \vdash \bot$.

Proof

1. $\mathbf{In}A \rightarrow \mathbf{Bel}A$ – Proposition 2.3
2. $\neg(\mathbf{Go}A \wedge \neg\mathbf{Bel}A)$ – 1,PL
3. $\mathbf{In}A \wedge \neg\mathbf{Bel}A$ – E2
4. $\bot$ – 2,3,PL $\Box$

But this will mean that the system is inconsistent, something we do not want – it would turn our system trivial and, furthermore, reduce its models to an empty class, in other words, a useless modelling tool. So, basic BDI logics don't admit the assymmetry thesis.

Conclusions

1. We ignore if full BDI logics are sound or complete, although we assume that they are sound. They are, anyway, undecidable.
2. Different axioms convey different properties of agents.
3. Many axioms are counter-intuitive. For instance, axiom $I - G*$ is supposed to capture (through strong realism) future constrainment of goals and beliefs (regarding actions) by present intentions. But this implies too that any action that is desired or intended is believed, while a real-world agent can perform many actions mindlessly (even though they depend on some broader intention).
4. Agents suffer from logical omniscience (due to the closure of their beliefs under implication) and from a far too great measure of coherence.
5. The theory has proven uselful as an specification tool of real-world agents, like the OASIS system (an experimental agent for air traffic, cf. [13]).
6. Not all of Bratman's *provisi* and thesis hold – like the assymmetry thesis.
7. BDI logics model only deliberation and not means-end reasoning. This is somehow a handicap, because planning is an essential part of practical reason. Woolridge in [17] remedies to this by introducing action modalities and action operators which lets us define conditional and iterative control structures.

Annex. Software Agents

2.7 General Description

Sotware agents are embedded reactive systems. They run on a previously or outwardly defined dynamic environment (as, say, a process) whose state can change through time deterministically or non-deterministically modyfing it – i.e. they consume environment variables and their output consists in updating or modifying them. Their aim is to execute autonomously a certain number of tasks of a system. For example, in a client/server architecture (i.e. a network) they may take care of the handling of the clients' requests or routing, or, again, of the servers' task schedules.

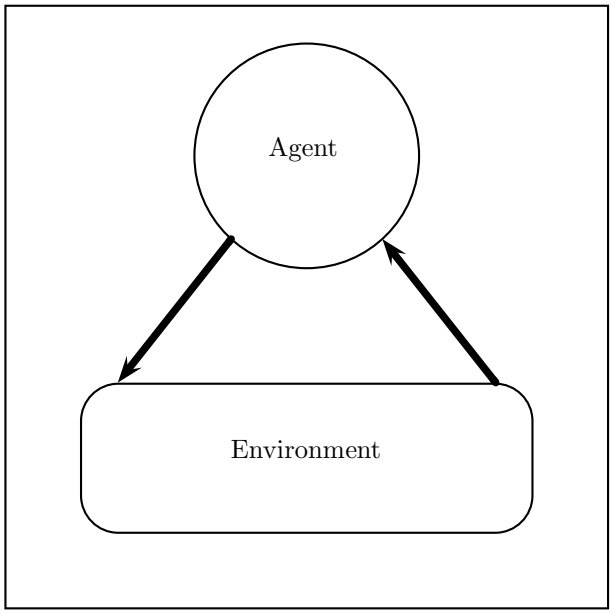
What is interesting about them is their providing us of a very simple instance of agency. In it, the choice of an agent will depend on the environement in which they will run. A blindly-commited agent will be more appropriate for, say, a static environment. A single-minded agent will be better for a dynamic (and overall deterministic) environment disposing of good memory and computation resources – since deliberation and belief revision functions are quite greedy on that account. While an open-minded agent can be better for a dynamic environment with limited resources and whose states change far too often.

Moreover, while belief can be seen as the agent's declarative knowledge (and may involve a call to a knowledge base), the loops constitute the agent's procedural knowledge. This agents can have internal states stocked in local (in the agent's) variables.

The agent specified below will thus capture (albeit differently) the same kind of concept to which BDI logics point. This section follows closely Wooldridge's accounts in [17] and [16].

2.8 Scheme of an Agent

An agent dynamically interacts with a global environment through a constant feedback:



2.9 Formal Definition

We begin by giving a formal definition of basic sotfware agents as processes running in a system or environement. This in turns take us to define agents having internal states – like BDI agents. This section is mainly based on [16].

Definition 2.9.1 *Let* $S = \{s_i | i \in \mathbb{N}\}$ *be a non empty set of* states *and* $A = \{\alpha_i | i \in \mathbb{N}\}$ *be a non empty set of* actions. *Then:*

- $R \subseteq \wp(\mathbb{N} \rightarrow S \times A)$ *is a set of alternating sequences of actions and states called* runs.
- $R^A \subset R$ *is the set of runs that end with an action.*
- $R^S \subset R$ *is the set of runs that end with a state.*

Remark 2.9.1 *Runs are usually written as follows, for* $n \geq 0$*:*

$$s_0 \overset{\alpha_0}{\rightarrow} s_1 \overset{\alpha_1}{\rightarrow} \ldots \overset{\alpha_{n-1}}{\rightarrow} s_n.$$

Definition 2.9.2 *An* environement *is a structure* $En = (S, s_0, \tau)$ *where:*

- S *is a set of states.*

- $s_0 \in S$ *is the* initial state.
- $\tau : R^A \rightarrow \wp(S)$ *is a* state transformer *function.*
- *We denote by EN their class.*

We can now come to agents and systems (i.e. embedding systems – for agents can be seen as systems too, although in a broader sense).

Definition 2.9.3 *An* agent *is a function* $Ag : R^S \rightarrow A$ *that maps runs ending with a state to actions. We denote by AG their class.*

The nature or kind of an agent will therefore vary according to the actual definition of the function that characterizes it, which may include new parameters. Agents in the general sense are also called *standard agents* and their class can be denoted AG_s. Agents together with enviroments define a system:

Definition 2.9.4 *A* system *is a structure* $Sy = (Ag, En)$ *where:*

- *Ag is an agent.*
- *En is an environment.*
- *we denote by SY their class.*

The run of an agent is now defined in terms of the environment whose states it modifies (and implicetely, relatively to a given system):

Definition 2.9.5 *Let* $En = (S, s_0, \tau)$ *be a environement,* $\rho =< s_0, \alpha_0, s_1, \alpha_1, ... >$ *a run.* ρ *is said to be is the* run of an agent *Ag in the environement En iff*

- e_0 *is En's initial state.*
- $\alpha_0 = Ag(e_0)$.
- *For any* $i \geq 0$*;*

  $$s_i \in \tau(< s_0, \alpha_0, ..., \alpha_{i-1} >)$$

  and

  $$\alpha_i = Ag(< s_0, \alpha_0, ..., s_i >).$$

- *We denote their set by* $R(Ag, En)$.
- ρ *is further said to be a* terminated *run if* $\tau(\rho) = \emptyset$.

Furthermore, an immediate equivalence relation can be defined over agents:

Definition 2.9.6 *Let Ag and Ag' be agents. Ag is said to be* behaviourally equivalent *to Ag', in symbols* $Ag \equiv_{beh} Ag'$*, iff for any environment En,*

$$R(Ag, En) = R(Ag', En).$$

As already said, different kinds of agents are obtained through further refining the definition of the function they compute. *BDI* agents are, for instance, state-based agents, that is, agents having a set I of internal states, that change according to environmental input.

Definition 2.9.7 *Let S be a set of states, A a set of actions, I a set of internal states and P a set of percepts. Let $see : S \rightarrow P$, $next : I \times P \rightarrow I$ and $action : I \rightarrow A$ be three functions. A* state-based agent *is then a function $Ag_{st} : S \rightarrow A$ such that $Ag_{st}(s) = action(next(\iota, see(s)))$, for any $i \in I$. We denote by AG_{st} their class.*

Remark 2.9.2 *We can still speak of runs of agents in an environment En and thus of $R(Ag_{st}, En)$ by simply stating that, for any $i \geq 0$:*

$$\alpha_i = Ag_{st}(s_i)$$

instead of the above definition for (standard) agents.

Behavioral equivalence lets us define an order among agent classes:

Definition 2.9.8 *Let AG and AG' be two classes of agents. Then AG is said to be* as expressive as *AG', in symbols $AG \preceq AG'$, iff for any $Ag \in AG$ there is $Ag' \in AG'$ such that $Ag \equiv_{beh} Ag'$.*

Which leads in its turn leads to the following important result:

Theorem 2.9.1 *$AG_{st} \preceq AG_s$.*

(**Proof**) Let En be an environment. We have to show that for any state-based agent Ag_{st} there is a standard agent Ag_s, such that, for any terminated run ρ:

$$(*)\rho \in R(Ag_{st}, En) \iff \rho \in R(Ag_s, En)$$

by induction on the maximum index n of runs (runs being of length $2n$). Knowing that $\tau(\rho) = \emptyset$, and that hence $\rho \in R^A$.

- Let Ag_{st} be a state-based agent where *action*, *next* and *see* are as above (and therefore injective). We define a function $\sigma : R^S \rightarrow S$ by putting:

  $$< s_0, \alpha_0, ..., \alpha_{m-1}, s_m > \mapsto s_m.$$

  Thus σ is a function that projects the last term of a state-finishing run. Let $\iota_0, ..., \iota_i$ $(i \geq 0)$ and $p_0, .., p_j$ $(j \geq 0)$ be two enumerations of, respectively, I and P. We then define $n : P \rightarrow I$ by putting:

  $$n(p_j) = \begin{cases} next(\iota_i, p_j) \text{ if } i = j \\ \iota_j \text{ otherwise.} \end{cases}$$

  Hence:

$$Ag_s = action \circ n \circ see \circ \sigma.$$

For indeed this is an standard agent, since $Ag_s : R^S \to A$. Ag_s has been constructed by dropping, so to speak, the internal states together with the percepts. Behavioural equivalence is then established by induction on n:

- $n = 0$. Then $\rho = \epsilon$ – the empty sequence. It is then clear that

$$\rho \notin R(Ag_{st}, En) \iff \rho \notin R(Ag_s, En)$$

for no empty runs are allowed. The property is hence true. □

- $n = k + 1$. By induction hypothesis, $(*)$ is true up to k. Consider then $\rho =< s_0, \alpha_0, ..., s_{k+1}, \alpha_{k+1} >$ such that $\rho \in R(Ag_{st}, En)$. Now, as En is fixed, the following two conditions are true for both Ag_{st} and Ag_s:

  – s_0 is En's initial state and
  – $s_{k+2} \in \tau(\rho)$.

  This means that we only need to prove that

$$(**) Ag_{st}(s_{k+1}) = Ag_s(< s_0, ..., s_{k+1} >)$$

to have the equivalence – these two being nothing but subsequences of ρ. So, assume that $Ag_{st}(s_{k+1}) = \alpha_{k+1}$. By definition of Ag_s, $Ag_s(< s_0, ..., s_{k+1} >) = \alpha_{k+1}$, which finishes the proof. □

2.10 Specification

We will provide now a very general specification (in pseudo-code) of such an agent below (in fact, of a cautious agent), which contains two imbricated (eventually unbounded) loops. We start by giving the signature of the main functions involved. The two following subsections are based in [17].

2.10.1 Signature of the Procedures

We start by giving the signature of the main functions involved. The functions are grouped in two collections. The first contains the procedures belonging to the deliberation loop and the second those groups those from the means-end sub-routine. B is a set of beliefs, P of percepts, I of intentions and PL of plans. While $Bool$ is a set of boolean values – i.e. $\{\mathbf{true}, \mathbf{false}\}$.

1. **The main loop: the deliberation procedures**. They involve basic decision-making.

   - $brf : \wp(B) \times P \to \wp(B)$ is belief revision function that updates the agent's set of beliefs according to new environmental input (by way of percepts).

- $options : \wp(B) \times \wp(I) \rightarrow \wp(D)$ is a function that properly generates the set of desires that take part in deliberation.
- $filter : \wp(B) \times \wp(D) \times \wp(I) \rightarrow \wp(I)$ is a decision-making function.

2. **The sub-routine: the means-end procedures**. They involve generating a plan and executing it while updating the ρ variable (and thus, of the agent's local environement) and hence veryfing (and eventually modifying) the generated plan's suitability all the way through.

   - $plan : \wp(B) \times \wp(I) \rightarrow PL$ is a function that builds a plan (a sequence of actions to be executed) from the (updated) beliefs and final intentions attained while deliberating – i.e. the agent's choice takes him into an action course. The plan's last action should cause the global environment's modification (i.e. the goal) conveyed by the filtered intentions.
   - $empty : P \rightarrow Bool$ is a boolean valued function that tests if the plan has been completed and the goal attained.
   - $succeded : \wp(I) \times \wp(B) \rightarrow Bool$ is a function that tests whether the goal has been attained or not.
   - $impossible : \wp(I) \times \wp(B) \rightarrow Bool$ is a function that tests whether the goal can be attained or not. If it cannot, the means-end loop is stopped (and we return to the main deliberation loop).
   - $reconsider : \wp(I) \times \wp(B) \rightarrow Bool$ is a function that test if it would be wise or not to update beliefs and conditions throughout the executions of the plan.
   - $sound : P \times \wp(I) \times \wp(B) \rightarrow Bool$ tests if the plan is still sound relatively to the updated beliefs and desires.

2.10.2 Pseudo-code Specification

The software agent is specifid as follows. Note that ρ is a variable that stocks an input (a percept) from the outer (global) environement which is subject to change and that π stands for plan, a finite sequence (or list) of actions to perform, i.e. $\pi =< \alpha_1, ..., \alpha_k >$. Inputs (percepts) come from the global environment and outputs (actions) are sent to it, thereby modifying it.

```
GENERICAL-AGENT ()
1   B ← {β_1, ..., β_n}; /*initial beliefs–knowledge base*/
2   I ← {ι_1, ..., ι_m}; /*initial intentions–desires*/
3   while true do /*deliberation loop*/
4     get ρ ; /*input*/
5     B ← brf(B, ρ);
6     D ← options(B, I);
7     I ← filter(B, D, I); /*derived intentions*/
8     π ← plan(B, I);
```

```
9     while not [ or empty(π) /*means-end loop*/
10                or succeded(I, B)
11                or impossible(I, B) ] do
12      α ← hd(π);
13      return α; /*output*/
14      π ← tl(π);
15      get ρ; /*update of ρ*/
16      B ← brf(B, ρ);
17      if reconsider(I, B) then /*caution condition*/
18        D ← options(B, I)
19        I ← filter(B, D, I)
20      endif
21      if not sound(π, I, B) then /*plan update*/
22        π ← plan(B, I);
23      endif
24    endwhile
25  endwhile
```

Bibliography

[1] Amal El Fallah-Seghrouchni Alejandro Guerra-Hernandez and Henry Soldano. Learning in bdi multi-agent systems. http://centria.di.fct.unl.pt/ jleite/climalV/12.pdf, 2004.

[2] Michael Bratman. *Intentions, Plans and Practical Reason*. Harvard U. Press, 1987.

[3] Stuart Chalmers and Peter M.D.Gray. Bdi agents and constraint logic. *Artificial Intelligence and Simulation of Behaviour*, (1), 2001.

[4] Philip Cohen and Hector Levesque. Intention is choice with commitment. *Artificial Intelligence*, (42), 1990.

[5] Roberta Ferraio and Alessandro Oltramari. Towards a computational ontology of mind. In *Formal Ontology in Information Systems'*, 2004.

[6] Alejandro Guerra Hernandez. *Apprentissage d'agents rationnels BDI dans un univers multi-agents*. PhD thesis, Université de Paris 13, 2003.

[7] James Harland and Michael Winikoff. Agents via mixed-mode computation in linear logic. http://citeseer.ist.psu.edu/harland01agents.html, 2003.

[8] Andreas Herzig and Dominique Longin. Beliefs, goals and intentions. http://www.irit.fr/ Adreas.Herzig, 2003.

[9] Lin Padgham John Thangarajah and James Harland. Representation and reasoning for goals in bdi agents. In *Proceedings of the 25th Australasian Conference on Computer Science*, pages 259–265. Australian Computer Society, 2002.

[10] Vinet Chand Padmanabhan Nair. *On Extending BDI Logics*. PhD thesis, Faculty of Enginneering and Information Technology, Griffith University, Queensland, 2003.

[11] Willem Visser Rafael H. Bordini, Michael Fisher and Michael Wooldridge. Verifiable multi-agent programs. http://www.cs.uu.nl/ProMAS/2003/papers/paper1.pdf, 2003.

[12] Anand Rao and Michael Georgeff. Modelling rational agents within a bdi-architecture. http://citeseer.ist.psu.edu/122564.html, 1991.

[13] Anand S. Rao. Agentspeak(l): Bdi agents speak out in a logical computable language. In *Seventh European Workshop on Modelling Autonomous Agents in a Multi-Agent World*, 1996. http://citeseer.ist.psu.edu/rao96agentspeakl.html.

[14] John Searle. *Intentionality*. Cambridge U. Press, 1999.

[15] Guido Governatori Vineet Padmanabhan and Abdul Sattar. Actions made explicit in bdi. In *Advances in Artificial Intelligence*, pages 390–401. Springer-Verlag, 2001.

[16] Michael Wooldridge. *An Introduction to Multi-Agent Systems*. Wiley and Sons, 2002.

[17] Michael Woolridge. *Reasoning about Rational Agents*. The MIT Press, 2000.