Journal of Artificial Intelligence Research 1 (2004) 1-40 Submitted 1/04; published 5/04

# Beliefs, Desires and Intentions in Agent Systems: A survey of the BDI Agent Model

**Felipe Rech Meneguzzi** FELIPE@CPTS.PUCRS.BR
**Avelino Francisco Zorzo** ZORZO@INF.PUCRS.BR
**Michael da Costa Móra** MICHAEL@INF.PUCRS.BR
**Lúcia Maria Martins Giraffa** GIRAFFA@INF.PUCRS.BR
*6681, Ipiranga Avenue,*
*Porto Alegre, RS 90619-900 Brazil*

## Abstract

The BDI model was initially proposed as a philosophical model for human practical reasoning and has evolved into a solid model of computational agents through a series of theoretical and practical publications. These works can be used to track this evolution. In this article we provide an incremental description of the BDI agent model from its philosophical description through its Computer Science formalization up to the implementation of some of its proposed architectures. At the end of this article we point out possible future works that can be proposed to further improve the BDI model.

## 1. Introduction

An ever increasing number of systems have been modelled in terms of autonomous agents, approaching the development of complex computational systems to the development of multi-agent systems [1, 2]. This approach is the result of a better comprehension of system behavior when it is described as interactions between autonomous entities [3]. Although the use of agents as an abstraction mechanism has increased, various works point out that there are still fundamental questions regarding the construction of agent-oriented systems that remain unanswered [1, 3]. In particular, questions regarding multi-agent systems development methodologies. Besides, the answer to another basic question in the agent systems community has not reached a consensus, which is an exact definition to what an agent is. Although it is possible that the question will never have a definite answer, an increasing number of authors [1, 2] have been using the following definition [4]:

**Definition 1 (Agent)** *An agent is an encapsulated computer system that is situated in some environment and that is capable of flexible, autonomous action in that environment in order to meet its design objectives.*

A few points about this definition should be better explained [1]:

- Agents are clearly identifiable problem solving entities with well-defined interfaces and limits;
- Agents are situated in a particular environment, they receive input related to the state of its environment and act upon this environment through actuators;

- Agents are designed to fulfill a specific purpose in the sense that they have particular objectives to reach;
- Agents are autonomous in the sense that they have control over its internal state and over its behavior;
- Agents are capable of exhibiting a flexible problem solving behavior when pursuing its intended objectives. They must be reactive in the sense that they can respond to changes in their environment in a temporally acceptable manner and pro-active in the sense that they must act expecting to attain future objectives.

Definition 1 characterizes agents described throughout this work. Agent research can be divided into two major branches: *macro-level* research, which is concerned with an agent society, and *micro-level* research, which is concerned with the internal works of an agent [5]. Generally, *macro-level* research abstracts the inner working of an agent considering it a black-box [1, 3, 6, 7, 8, 9, 10]. Similarly, *micro-level* research generally considers agents isolated from the agent-society [11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22].

Throughout the evolution of agent-based systems one of the main agent models to have come into being is the one based on the interaction of Beliefs Desires and Intentions (BDI). The BDI model is, perhaps, one of the best known and studied models of deliberative agents [15], having been based on a number of theoretical studies [23, 24, 25] and resulting in a series of practical applications [11, 14, 19, 26] that proved its applicability in solving problems such as control of a space shuttle [27]. The evolution of the BDI model of agents can be tracked throughout these works, where each one of them has contributed to reduce the distance between Bratman's original philosophical model [23] and a concrete practical reasoning agent model. This article aims at tracking the origins and evolution of the BDI agent-model, describing the most significant theories and applications that came to exist throughout its evolution.

This work starts out by summarizing the origins of the BDI model (Section 2). We then describe the main theoretical works that have been used to underpin BDI agent research (Section 3), and some of the main architectures designed and used to implement concrete BDI agents (Section 4) providing a brief comparison and discussion regarding these architectures. At the end of this article we try to point directions to future work regarding the BDI agent model.

## 2. The BDI Model

This section describes the origins of the BDI model from its inspiring theory proposed by Donald Davidson and Michael Bratman's introduction of the Intentions as a mental state. At the end of this section we describe its initial computer theoretical definitions.

### 2.1 Machines, Intentions and Mental States

The use of mental states to describe computer systems can be traced back to the works of Dennet [28, 29]. These works describe *intentional systems* as entities whose conduct can be foreseen through the attribution of Beliefs, Desires and Rational acumen [28]. In these works, the author divides those systems in levels where a first order intentional system is a system where entities have Beliefs and Desires, but do not have them regarding other Beliefs and Desires. A second order intentional system would be able to have Beliefs and Desires regarding Beliefs and Desires, and that hierarchy can be expanded *ad infinitum*.

### 2.1.1 Describing machines as mental-states

According to McCarthy [30], the motivation for assigning mental states to systems can be analyzed regarding its utility and legitimacy.

McCarthy [30] states that it is legitimate to assign mental states to machines when such assignment allows the same information regarding the machine to be understood as it would be for a person. This assignment is useful when it makes the machine's structure, its past and future behavior easier to understand, and it allows for the machine to be easier to be fixed and improved. This assignment is possibly not even necessary to describe human beings, but the use of some kind of mental quality allows the description of a given machine particular state in a more succinct way. In his work McCarthy also exemplifies the assignment of mental states to systems such as thermostats, multi-processed operating systems, and even to systems designed for reasoning, stating that assigning mental states to simple machines is easier, but it is more useful to assign mental states in order to describe systems whose internal structure is not completely known.

The use of mental states to describe behavior received a great deal of criticism from neuroscience researchers. Such researchers claim that, eventually, it will be possible to describe mental processes in a precise way through a complete neuroscientific theory [31]. Even so, we believe that mental abstractions remain useful for two basic reasons:

- We cannot guarantee that a complete materialistic theory will be created in a short period of time;

- Even if such theory is created in a possibly short time span, the human cognitive ability is and will probably remain limited. Therefore an abstraction mechanism that allows us to handle the complexity in both natural and artificial systems is still desirable.

### 2.1.2 Types of Mental States

A number of classifications for mental states were created as a result of its use in practical reasoning studies and in systems description. In [31], the classification of three different authors regarding mental states are described, namely the ones by Searle [32], Shoham & Cousins [33] and Kiss [34]. These classifications are summarized in Table 1, ordered by increasing approximation to Bratman's model of practical reasoning described in Section 2.2.2.

## 2.2 Mental-States and Practical Reasoning

This section will detail the theories regarding the use of mental states to describe behavior, initially, describing Davidson's theory of Beliefs and Desires, and then describing the consolidated Belief, Desires and Intentions theory created by Bratman.

Table 1: Mental state classification

| Author | Groups | State Name |
|---|---|---|
| Searle | Information | Beliefs, Knowledge |
| | Pro-Active | Desires, Preferences, Intentions, Obligations |
| Shoam & Cousins | Information | Beliefs, Knowledge |
| | Motivational | Desires, Preferences, Intentions, Plans |
| | Social | Permissions, Obligations |
| Kiss | Information | Beliefs, Knowledge |
| | Conative | Commitments, Intentions, Plans |
| | Affective | Desires, Preferences, Obligations |

### 2.2.1 Davidson's Belief-Desire theory

Although the theories of other philosophers have some common points, Donald Davidson is recognized as the author whose theory would inspire Bratman's Beliefs, Desires and Intentions model. In the article "Actions Reasons and Causes" [35, 23], Davidson presents the fundamentals of his practical reasoning theory. Such article defines intentional actions, which are actions that can be explained through the reasons an agent has to execute them. These reasons can be traced back to pairs of beliefs and desires. Therefore, Davidson assumes that it is possible to make an all-out evaluation of the beliefs and desires to generate every action. This all-out evaluation is based on a maximum utility function that, taking the agent's beliefs into account, determines at every moment which is the most efficient action to be taken in such a way as to satisfy his desires. A basic outline of this functioning can be seen in Figure 1.

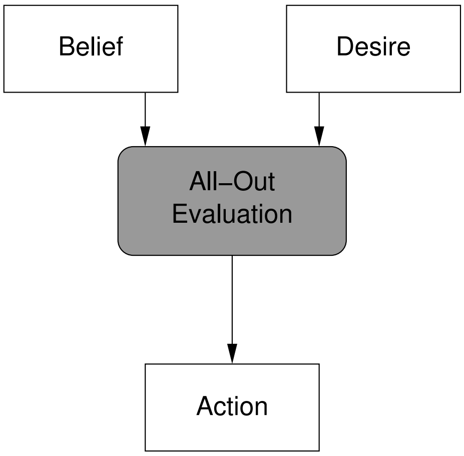

Figure 1: Practical reasoning according to Davidson.

Some problems regarding Davidson's theory were pointed out by Bratman [23, 36] that motivates an expansion of Davidson's model, these problems will be summarized next.

**Problem 1 (Buridan's Problem)** *If a reasoning donkey (*i.e. *with the ability to always take the best decision to deal with a situation) is placed in a position that is equidistant to two equal piles of hay, he will starve because he will not be able to decide which is the best action to be taken*

The first problem observed regarding Davidson's all-out evaluation function, is called Buridan's problem[1] [23, 36]. Considering that all his options are identical, he will not be able to determine which one is the best. This problem illustrates the fact that Davidson's thought and action theory would have serious trouble when facing situations with equally desirable options, situations which human beings are notoriously able to resolve.

**Problem 2 (Partial Reasoning Problem)** *Consider that Agent Smith has the desire to buy a specific computer game. Agent Smith has also the belief that this can be accomplished by going to store $X$, but Agent Smith does not know whether store $X$ actually has a copy of the game in stock or not.*

There must also be a way to generate partial mental states (desires/intentions) in order to advance towards the accomplishment of objectives whose theoretical viability is not completely determined. If the action necessary to accomplish that desire depends solely on the evaluation of Agent Smith's beliefs and desires, it is not clear whether Agent Smith has to go to the store to check the existence of the game or not. In order to accomplish that, an intermediate mental state is necessary, one that allows him to start moving towards the satisfaction of his desire. On the other hand, this state must allow its revision throughout its execution in such a way as to reflect the completion of data as incomplete information is being filled out.

**Problem 3 (No Time Consideration Problem)** *If the time Agent Smith takes to consider all his options is not negligible in comparison to the time it takes for the world to change, then, when the agent finishes reasoning, its decisions might be outdated.*

When performing an all-out evaluation of the beliefs and desires Davidson's model assumes that the time taken to perform such an operation is negligible compared to the time taken for the world-state to change. The main problem regarding the Beliefs-Desires model is that it does not consider the time interval between an all-out beliefs and desires evaluation and an action. This problem becomes clearer when we consider agents with limited resources, such as human beings and computer programs, in a dynamic world. In such a world, the time spent in the deliberation process must be the least possible, given that the current world-state may change during the deliberation process, otherwise the decisions taken during this process could no longer be valid at the end of the process. Although Davidson's reasoning model is somewhat interesting from a theoretical point of view, it fails in some aspects, especially when it comes to describing agent operation in the real world.

[1] Because it is attributed to the philosopher Jean Buridan, even though such problem has never been found within his writings[23].

### 2.2.2 The importance of intentions

Davidson's theory has a few limitations, even though it is an interesting starting point for a complete practical reasoning theory. These limitations are related to the fact that this theory only uses beliefs and desires to resolve conflicting courses of action [37]. Nonetheless it would be possible for deliberation to take place with desires and intentions alone [36], though this deliberation would be extremely limited especially due to constraints in time and resources, which most real world agents have. Deliberation in this case would be flawed in various ways because there are no intermediate states between decision and action, and the agent would suffer from a problem in which the agent would constantly reconsider its course of action possibly leading it to never carry out its desires far enough to satisfy them, given that each action is the result of an all-out beliefs and desires evaluation [19]. On the other hand, if the agent spends some time planning a consistent course of action leading to a given goal and commits itself to carrying out this plan, the agent would only re-consider its actions when something has been achieved or has gone wrong.

Some authors state that planning is fundamental to intelligence, and therefore two abilities are central to planning agents: to act intentionally and to form and execute plans [36]. The first ability is a requirement for the second because an active effort by the agent is necessary in plan-creation. This effort appropriately describes the way our minds work, given that we constantly think about our own acts and intentions, in the past and in the future. It is this reasoning mechanism that allows us to plan complex courses of action *a priori*, overcoming our limitations regarding, reasoning time among other mental resources.

Therefore, Bratman proposes in [36] the use of intentions as a distinct mental state that has the same importance as desires and intentions in the deliberative process. This new model is outlined in Figure 2.

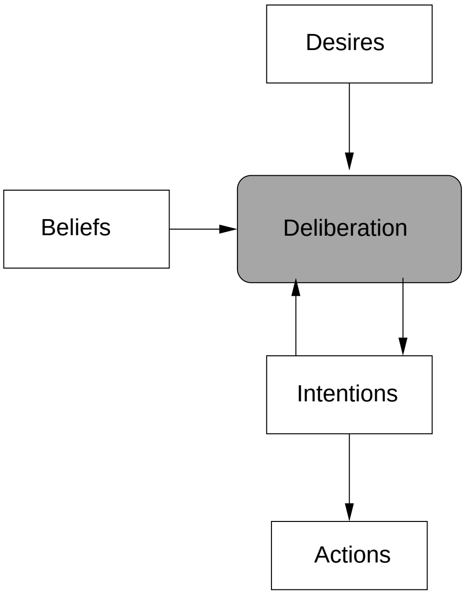

Figure 2: Practical Reasoning according to Bratman.

The role of intentions is not only to complement Davidson's beliefs-desires model as another element that associates rationality to the agent's acts, but to be a component that provides the agent with the ability to structure its planning, ultimately taking part in the agent's means-end reasoning process. The planning to which Bratman refers to is not necessarily the same as the one that occurs in STRIPS-like [38] planning systems, but a higher level planning in which intentions guide planning *per se* (*i.e.* STRIPS-like planning). Intentions, differ from desires as they are conduct controlling pro-attitudes while desires have just the potential to influence conduct. It is important to point out that sequences of intentions are actually partial plans that will be refined and modified as needed. These partial plans allow the agent to start moving towards objectives whose viability is not yet fully perceived. Another important outline that must be made at this point is that the partial plans we refer to here are not the same as those found the AI planning scope. In Bratman's work, partial plans are plans whose steps are possibly incomplete, which will be filled out at the moment of their execution.

Bratman defines three fundamental features of intentions in the reasoning process. Intentions have a more direct effect in the agent's actions, that is, they are conduct controlling. Besides, they have what the author calls inertia, which means that, although intentions are revocable, they resist reconsideration. And, finally, they serve as input to new intentions.

In Bratman's view, intentions not only connect desires and intentions, they also serve as input to the creation of new intentions. When used to this end, they restrain which states will be considered in future reasoning, *i.e.* they establish relevance patterns to the options that will be considered for future deliberation. In this way they create an admissibility filter to the possible solutions for a given problem. Therefore, this pre-deliberation that narrows down the solution search-space provides the way to overcome the time and resource limitations mentioned before. Ultimately these features make intentions a more appropriate solution than a simple maximum utility function. Besides, this admissibility filter solves appropriately the Buridan Problem (Problem 1) by providing, once a course of action has been set, a reasonable justification to why consider certain action threads and not others.

The use of intentions as mental states allows coordination, both internally and externally. Intentions have two important characteristics for coordination: one static and the other conduct controlling. These two characteristics support coordination by allowing an agent to expect that, when the time comes, an agent will at least try to do what it has intended to do. Furthermore, the fact that the agent has thought beforehand how to accomplish what he intended to do, and defined the preliminary steps, will ensure that the agent will be in a better position to perform an action and be successful at it. Therefore, internal coordination is enabled as a result of the agent fixating intentions and thus being able to plan further in the future in the expectation that these intentions will be accomplished. Externally, once the remaining agents are informed of the intentions of the first, they will also be able to plan ahead in the future based on the expectation that the first agent will accomplish its intentions. Bratman's theory is questioned by the defenders of the beliefs-desires approach who tend to doubt the validity of the intentions stating that intentions are:

- Metaphysically objectionable, since they suggest action at a distance;
- Rationally objectionable, if they were completely irrevocable;

- Simple time wasters, if intentions were easily revocable.

These criticisms are easily dealt with if we consider what was just explained in this section. The first criticism is resolved as we observe that intentions are a type of high-level planning and commitment. The solution to the second criticism is made clear by the fact that, although intentions resist reconsideration, they are revocable in certain situations. The third criticism is related to the second, and represents an inherent problem to the BDI model, which is the balance between intention stability and reconsideration. This problem was identified during the creation of the IRMA architecture [11]. An agent that reconsiders its intentions too often is called *cautious*, and an agent that holds on too much to its intentions is called *bold*. Anand Rao have studied solutions to this problem and some of them are described by Wooldridge in [39].

The possible intention feedback (*i.e.* intentions may generate other intentions) also creates difficulties. The main one is called *problem of the package deal* [37]. This problem consists of initially picking an intention to satisfy a given goal, and that intention as it is used to generate other intentions might generate a secondary intention to do something that does no represent the agent's desires. That problem can be seen in Figure 3.

**Problem 4 (Problem of the Package Deal)** *If Agent Smith has the desire to obtain a Masters degree with a very good work (m in Figure 3), he will, as a consequence, have the intention to work hard in this work (t in Figure 3). This intention to work hard may result in the intention of not sleeping a few nights in order for him to increase his working time ($\neg s$ in Figure 3), even though his desires state that he likes to sleep well (s in Figure 3).*

In that case, the intention to work hard has generated an intention that conflicts with his desires because it "came with the package". In this case the intention of not sleeping some days is not to be used as input to other intentions, because it does not represent Agent Smith's desires. Knowing how to separate intentions that might generate new intentions from those that might not is one of the difficulties inherent to the problem.

Hence, Bratman's theory provides an alternate solution not only to the problems that troubled Davidson's model, but also presents a model where it is possible that our intentions and ultimately our actions might be motivated not only by belief-desire pairs. As an example of this possibility we have the problem of the package deal, which allows the creation of intentions that, although potentially undesirable, are necessary for the satisfaction of other intentions or desires considered more important. Moreover, intentions, in theory, can also be generated with no direct relation to the desires, which allows the agent to use opportunities, which is an important characteristic of intelligent behavior [39].

### 2.3 Main components of the BDI Model

The philosophical underpinnings presented in the previous sections met the necessities of the agent theorists that were seeking a simple and reasonably complete theory to describe the practical reasoning process. One of the main advantages of this model is that it allows a system description in an anthropomorphic way [40]. Even though this model describes human behavior in a reductionist way, a simplified approach can be useful for two reasons: it is computationally more efficient and it serves as the basis for an expanded theory. The

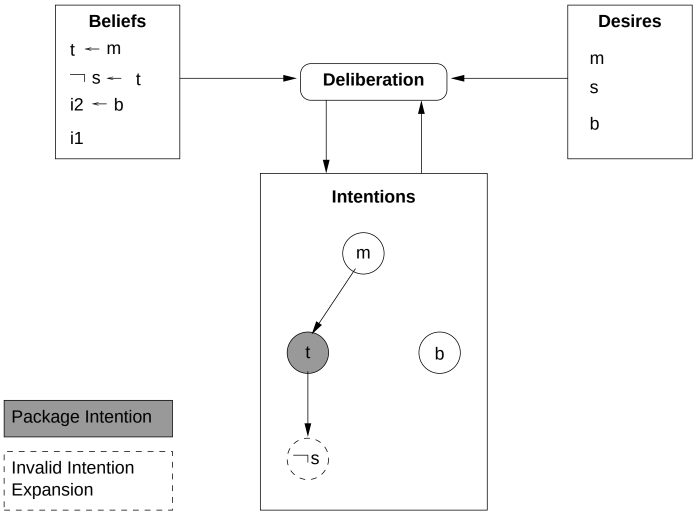

Figure 3: Problem of the package deal, $i1$ and $i2$ represent irrelevant beliefs.

first reason becomes clearer when we analyze the fact that various logic-based formalisms created to describe behavior in dynamic worlds like Modal Logics are not usually processed in bounded time [41]. The second reason is that, similarly to natural sciences such as physics, highly simplified theories can be initially used to describe the real world and, based on these theories, to develop more complete ones [24].

Fundamentally the BDI model is the computational transposition of the reasoning model proposed by Bratman [23, 37, 36]. Such model is based on the primary role that beliefs, desires and intentions as fundamental and necessary mental states used to describe practical reasoning. The use of these mental states is not without criticism, having at least two diametrically opposed groups of critics [19]. The first one(decision theorists) states that beliefs, desires and intentions are too many components. The second one (sociologists) believes that the tree components are to few.

Although the BDI model originated from a common theory, there are various interpretations for the model, in such a way that no single architecture or formal model currently represents the model in its totality. The operation of the BDI model can be defined as the specification of the rational balance between beliefs, objectives, plans, intentions, commitment and actions [24]. If we consider mental states other than beliefs, desires and intentions as just auxiliary abstractions to the model, this definition can be reduced as the balance between these three states. Hence, it is necessary to define a high-level description of each one of the model components.

**Beliefs** represent the agent expectancy regarding the current world state and the possibility that a given course of actions lead to another given world state [42]. Such state is modelled through the possible worlds semantics, although not every BDI agent model uses such a complex semantics. Furthermore beliefs may be represented in a simpler way in a program [15], for example as simple propositions or program variables[43, 31].

**Desires** are an abstraction that specifies preferences in world states or courses of action [42]. A common point in various works such as [37, 24, 31, 44] is that desires are not necessarily consistent. For example, an agent may have the desire to meet the enemy to attack him, and at the same time run away from him in order to survive, what will determine which one of the desires will be adopted will be the world state in which the agent is at, in this case, the agent's health, or similar concept. Desires are, in the same way as beliefs, easily represented in computer programs. Desires may represent simply target world states, in a way very similar to planning systems [15]. Using desires to describe systems turns the system objective-oriented. This is advantageous because conventional programs execute procedures with no sense of purpose. On the other hand objective oriented systems, knowing beforehand what was to be accomplished through the execution of its procedures allows it to detect that something failed, because it did not reach its final objective at the end of the procedure, and try again.

**Intentions** are used to overcome the problem that most natural and artificial agents have limited resources, therefore they are used by the agent to focus the deliberative process in order to choose a consistent desire subset, that is, one or a small set of desires to which the agent is committed to. The process of selecting the desires for commitment is called *formation of intentions* by Müller [42]. The function of the intentions is generally not restricted to a set of commitments. Bratman [37] advocates that intentions are part of a means-end reasoning mechanism. To this end, intentions serve as partial plans, in such a way as to allow the adaptation of the plan should the world change in a subtle way during plan execution, instead of performing a complete re-planning.

## 3. Computational BDI Theories

This section summarizes the main theories that have been used to underpin computational models of BDI agents, namely Cohen and Levesque's theory of intentionality [24], Rao and Georgeff's BDI Computation Tree Logics (CTL)[25], Singh [45] and Wooldridge [39]. These theories are presented in an evolutionary sequence, so as to provide a clear perspective of the direction to which BDI theories have been progressing, and to better explore the shortcomings that are yet to be dealt with.

### 3.1 Cohen and Levesque's theory of intention

Bratman defined philosophical concepts to explain why humans behave efficiently in the real world by establishing three mental states, informally defining the relationship among them and enumerating a number of characteristics these states must have. These concepts were then used by computer science researchers trying to emulate this process the other way around, that is, to define processes around Beliefs, Desires and Intentions that would allow a reasoning agent to operate in the real world in a satisfactory way. In order for Bratman's model of reasoning to be positively used in Computer Science, a formal definition of how these philosophical concepts work was created by Cohen and Levesque [24]. The main objective of Cohen and Levesque's work was to formally define intentions that met a set of functional roles as defined by Bratman and also to tackle some of their possible side-effects foreseen by Bratman [23, 36]. The authors also lay the foundation of a theory of agent interaction based on speech acts.

### 3.1.1 Fundamentals

The authors base their theory of rational action on a form of modal logic using a possible worlds semantics [46] with four primary modal operators, connectives from classical logic ($\neg$, $\wedge$, $\vee$, $\rightarrow$), temporal modalities ($\Diamond$ and $\Box$) and the existential quantifier ($\exists$). The primary modal operators are described below:

- **BEL** is used to represent Beliefs, where ($BEL\ Agt\ P$) means that formula $P$ follows from agent $Agt$ beliefs;
- **GOAL** is used to represent goals, which are used to represent intentions, where ($GOAL\ Agt\ P$) means that formula $P$ follows from agent $Agt$ goals. Or, considering the possible worlds semantics this means that formula $P$ is true in all the worlds accessible from the current world that are compatible with the agent's goals;
- **HAPPENS** is used in conjunction with **DONE** to reason about time, where ($HAPPENS\ ActExpression$) means that $ActExpression$[2] happens next;
- **DONE** complements **HAPPENS** in reasoning about time, where ($DONE\ ActExpression$) means that $ActExpression$ has just happened.

Along with the basic operators, the notion of time is represented by numerals that represent the time at which a certain formula is true to the agent, so that proposition ($At\ Smith\ Brazil$) $\wedge$ 3 means that agent $Smith$ being in $Brazil$ is only true at time 3. Another characteristic of the reasoning about time in this theory is that there are no simultaneous primitive events in it. Actions are either primitive events, represented by an action variable ($a, b, \ldots$), or an action expression, which can be one of the following:

- An action variable;
- The sequential action, $ActExpression; ActExpression$, meaning that the first action expression will be true and the second action expression $HAPPENS$ next;
- The nondeterministic choice action, $ActExpression|ActExpression$, meaning that either one of the first or the second action expression will be true (or $HAPPENS$) next;
- The test action, $P?$, which is used to constrain the reasoning of the agent to the possible worlds where the formula $P$ is true;
- The iterative action, $ActExpression*$, which means that $ActExpression$ happens multiple times as if multiple occurrences of the sequential action operator ";" are used, $ActExpression$ will occur at least once.

These action constructs are used to reason about a variety of real world phenomena regarding the agent's acts and events happening outside the sphere of influence of the agent itself.

[2] By $ActExpression$ we mean any action expression.

### 3.1.2 Beliefs

Agent beliefs characterize what the agent implicitly believes [24], this means that should everything the agent believes in be true, the agent's beliefs would characterize the whole state of the world at that moment. Another characteristic of the implicit nature of beliefs in this theory is that the agent believes in all the possible ramifications of his beliefs, with no need to explicitly reason in order for him to infer consequences. This definition of beliefs is used by the authors to define the notions of knowledge and competence represented by the **KNOW** and **COMPETENT** modal operators. An agent $A$ is said to know a formula $P$ (represented as $(KNOW\ A\ P)$), if the agent believes $P$ to be true and $P$ is, in fact, true. Also, an agent $A$ is said to be competent in some formula $P$ (represented as $(COMPETENT\ A\ P)$), if whenever the agent believes $P$ to be true, $P$ is actually true.

### 3.1.3 Goals

The authors build Bratman's notion of goals and intentions by refining a basic concept of goal. The basic concept of goal is defined in a similar fashion as beliefs in the sense that the **GOAL** operator defines what is implicit in the agents goals. Similarly to the beliefs, an agent has as goals all the consequences of its explicit goals. Considering the implicit nature of both **BEL** and **GOAL**, and the fact that the agent chooses the world he is in, then, when an agent believes some property $P$ to be true, he also has chosen $P$ to be true.

Using this definition of goal, the authors refine the notion of goal by defining an achievement goal $(A - GOAL\ A\ P)$ to mean that $P$ is currently false, but it will eventually be true. The achievement goal captures one important notion of desire defined by Bratman [23, 36] that states that an agent cannot desire something that he already knows is true or that will never be true. The authors then proceed by defining a notion of goal that implies commitment, which is called the persistent goal, where $(P - GOAL\ A\ P)$ means that agent $A$ has $P$ as a persistent goal until it either fulfills it or believes $P$ to be impossible. An important aspect of the definition of persistent goal is that it is closed only under logical equivalence, which will be important when the authors tackle the problem of the package deal (Problem 4).

### 3.1.4 Intentions

After formalizing various notions of goal and analyzing its logical consequences, the authors proceed to defining two types intentions:

- $(INTEND_1\ A\ P)$ means that agent $A$ is committed, through a persistent goal, to believing he is about to achieve $P$, and then achieving it;
- $(INTEND_2\ A\ P)$ means that agent $A$ is committed, again through a persistent goal, to believing he will do something (he might not already know what to do aside from the first step) that will bring about $P$ as a consequence.

These two definitions of intention capture the dual purpose of intentions envisioned by Bratman. $INTEND_2$ represents intentions as components of high-level plans, which are used for the agent to start acting towards a given goal even when the exact means to achieving that goal are not fully known. $INTEND_1$ represent the lower level plan components, which are closer to concrete actions.

Up to the basic definition of intentions, goals and intentions were defined in a way characterized as fanatical by the authors, that is, the agent will want to achieve something until he has done it or believes it is impossible for that to ever be true. By relativizing goals to a condition, the authors specify a criterium other than impossibility for an agent to drop a goal. At the end of their article, Cohen and Levesque [24] define intentions in a way that captures most, if not all of the properties envisioned by Bratman, but they do not specify the process through which an agent decides which desires he chooses to pursue, and how does he deal with conflicting desires.

## 3.2 Rao and Georgeff's BDI Computation Tree Logics

Various authors have proposed a number of logic systems aiming at enabling the construction of rational agents, though most of these systems only captured a subset of the rationality properties required of an agent, thus lacking a complete axiomatization. Among those, the logic systems defined to describe the operation of BDI agents also shared some of the those shortcomings. In order to tackle that lack of a sound and complete axiomatization of BDI logics, Rao and Georgeff [16, 18] have defined a family of logics to describe BDI agents as well as decision procedures to those logics. These logics are called by the authors *Computation Tree Logics* and the most important instances are $BDI_{CTL}$ and $BDI_{CTL*}$ and its variations. This set of logics have been widely used in BDI research [39] having been dubbed *BDI logics.*

### 3.2.1 From Branching Time Structures to Possible Worlds

The logic systems thus defined were intended to relate to classical methods of rationality definition such as decision theory. Therefore, the authors have created a mapping from Emerson's Branching Temporal Logic [47] and Decision Trees into a model representing beliefs, desires and intentions as accessibility relations into a set of possible worlds.

Informally, Decision Trees are composed of nodes representing world-states, branches in the tree represent alternative execution paths. State transitions are either actions taken by the system or a primitive event occurring in the environment, or both. Nodes resulting from actions are called *choice* nodes and nodes resulting from events are called chance *nodes.* The leaves in the tree are called terminal nodes. Chance nodes may be labelled with real-valued probabilities. A payoff function assigns real-valued payoffs to terminal nodes. A deliberation function chooses the best payoff path from the root to a terminal node.

The relation between Decision Trees and BDI logics is defined using chance nodes as reference, the Decision Tree is split into various trees, that no longer have chance transitions, each one representing a possible world, where the probability of the agent being in a given world is the probability on the chance transition that generated that tree. These trees are then separated into a belief accessibility relation using the probabilities in the initial tree and a desire accessibility relation using the payoffs in the initial tree. The paths generated by the deliberation function are said to be the intention accessibility relation.

### 3.2.2 Syntax and Semantics

Considering that BDI Computation Tree Logics are extensions to Emerson's Branching Temporal Logic [47] they include a set of primitive propositions $\Phi \neq \varnothing$ and a set of basic connectives and operators from which others can be defined. The propositional connectives used in this logic are $\vee$ and $\neg$, from which $\wedge$, $\supset$ and $\equiv$ are defined. Linear Temporal Operators used in this logic are **X** (next), **U** (until) and **F** (sometime in the future or eventually, similar to the $\Diamond$ operator). From these temporal operators **G** (all times in the future or always, equivalent to the $\Box$ operator) and **B** (before) are defined. A path quantifier used in this logic is **E** (some path in the future or optionally), from which **A** (all paths in the future or inevitably) is defined. Finally, the logic is extended with modal operators **BEL** (agent believes), **DES** (agent desires) and **INTEND** (agent intends).

Well-formed formulas in this logic are defined using the atomic propositions and the connectives and operators defined above. These formulas are divided into two types: *state formulas*, which express truth in a particular state or world, and *path formulas*, which express truth in a particular world or along a path in trees of possible worlds. A set of rules for the definition of the class of formulas valid in the language are defined below:

**S1** each atomic proposition $\phi$ is a state formula;

**S2** if $\phi$ and $\psi$ are state formulas then so are $\neg\phi$ and $\phi \wedge \psi$;

**S3** if $\phi$ is a path formula then $\mathbf{A}\phi$ and $\mathbf{E}\phi$ are state formulas;

**S4** if $\phi$ is a state formula then $\mathbf{BEL}(\phi)$, $\mathbf{DES}(\phi)$ and $\mathbf{INTEND}(\phi)$ are state formulas;

**P0** if $\phi$ and $\psi$ are state formulas then $\mathbf{X}\psi$ and $\phi\mathbf{U}\psi$ are path formulas;

**P1** each state formula is also a path formula;

**P2** if $\phi$ and $\psi$ are path formulas then so are $\neg\phi$ and $\phi \wedge \psi$;

**P3** if $\phi$ and $\psi$ are path formulas then so are $\mathbf{X}\phi$ and $\phi\mathbf{U}\psi$;

The class of valid formulas for $BDI_{CTL}$ is defined using rules S1-S4 and P0, thereby limiting the composition of path formulas to assertions regarding state formulas (in particular, using only the **X** and **U** operators). The class of valid formulas for $BDI_{CTL*}$ is defined using rules S1-S4 and P1-P3.

Truth valuation in this logic is given by a possible worlds semantics where each possible world is a tree structure with an infinite linear past and a branching future. The branching future represents the possible courses of events the agent can choose in a particular world. The belief accessibility relation mentioned before maps a possible world at a state to other possible worlds, the desire and intention accessibility relations do a similar mapping as well. The authors also provide a number of decision procedures for the verification of validity and satisfiability of formulas in this logic. These procedures are based upon the proof of the existence of a *small model property* for a given formula.

### 3.2.3 BDI Modal Properties

Along with the semantic definitions, this logic framework provides a series of possible axiomatizations so that various types of BDI logics can be defined. Along with the axioms inherited from Emerson's logic, a number of modal axioms applied to the BDI modalities can be included in the logic system resulting in different properties for the associated logic system. The minimal set of modal axioms in the logic is the application of the K-axiom to each of the modalities, which would result in the following axioms:

**B-K** $BEL(\phi) \wedge BEL(\phi \supset \psi) \supset BEL(\psi)$;

**D-K** $DES(\phi) \wedge DES(\phi \supset \psi) \supset DES(\psi)$;

**I-K** $INTEND(\phi) \wedge INTEND(\phi \supset \psi) \supset INTEND(\psi)$;

Besides these axioms, the authors have experimented with the inclusion of the D-Axiom, 4-Axiom, 5-Axiom and the necessitation rule, resulting in various of the BDI properties defined by Cohen and Levesque [48].

### 3.2.4 Extensions to $BDI_{CTL}$

Aside from reasoning in a single-agent environment, the BDI logics described in Section 3.2 was also augmented to cope with the presence of multiple agents in a theoretical framework that was used to support the implementation of the Cooperating Systems (COSY) architecture [49]. In such work, the author defines a theory of joint intentions by which an agent is able to delegate achievement of a task and resulting in commitment among the involved agents. This theory modeled communication using speech acts and used them as a first-order component with regards to the traditional BDI mental states.

The $BDI_{CTL}$ logic framework was later extended into a new one called $\mathcal{LORA}$ [39]. This framework introduces a number of changes and extensions, though some features of $BDI_{CTL}$ are kept. Regarding the representation of time, $\mathcal{LORA}$ uses a similar notion of discrete time unbounded and branching in the future, and linear in the past, though the past is now bounded.

Regarding expressivity, $\mathcal{LORA}$ extends the set of time operators with the $\mathbf{W}$ operator, where $\phi\mathbf{W}\psi$ means $\phi$ unless $\psi$. It also adds the notion of actions and action execution in a fashion similar to Cohen and Levesque's logic system. In particular it uses the constructs for sequential action, $(a; a')$, non-deterministic choice action $(a|a')$, iterative action $(a*)$ and test action $(a?)$, whose semantics are the same as described in Section 3.1.

## 4. BDI Architectures

As mentioned in the previous sections, Bratman's theory of practical reasoning led to the BDI agent model. This model was used to create a series of works that intended to implement such theory as practical computer systems. This section describes two of the most notorious such works, one of which is a proposal by Bratman himself of how a BDI agent should be implemented, the other is one that was used as a reference in the development of various other BDI agent models and theories, especially regarding the use of modal-logics to describe system behavior. Finally we describe a third practical BDI model that uses a different approach than the previous two by using Extended Logic Programming to implement non-monotonic reasoning in an agent.

### 4.1 IRMA: The embryonic BDI architecture

The Intelligent Resource-bounded Machine Architecture (IRMA), was defined by Michael Bratman, Martha Pollack and David Israel [11] to prove Bratman's practical reasoning model applicability. IRMA's objective is to provide reasoning for an agent taking into account its limited resources. Another aspect of IRMA is that is it was, along with PRS, one of the first to incorporate intentions as a primary mental state, playing an important part in the means-end reasoning of the deliberative process. The role of intentions consists of providing a way to keep track of partial progress in the agents activities and taking this partial progress as an input to future deliberation.

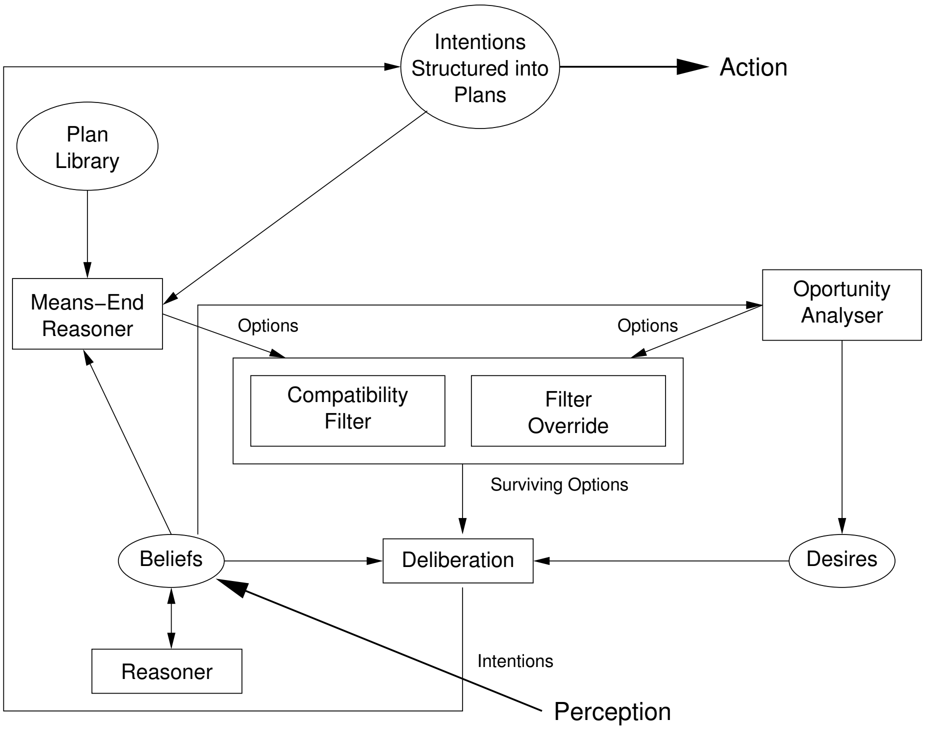

Figure 4: IRMA architecture internal structure [11].

Figure 4 represents IRMA's architecture and contains two basic types of entities: processes (denoted by rectangles) and storage entities (denoted by ellipses). IRMA Agent's intentions are structured into high-level plans in which actual plans (or plans as recipes) are stored in a plan library. The `Opportunity Analyzer` reacts to changes in the environment creating action options based on events that were not predicted by conventional planning, which is performed by the `Means-End Reasoner`. The `Means-End Reasoner` has as its most obvious inputs the `Agent's Beliefs` and the plans stored in the `Plan-Library`. Also, according to Bratman's practical-reasoning model, the role of intentions in means-end reasoning is to narrow down the search space for its problems. The `Means-End Reasoner` and the `Opportunity Analyzer` come up with options for the filtering process, represented by the Compatibility Filter and by the Filter Override to consider. The `Compatibility Filter` checks if the generated options are consistent with the intentions currently adopted, and the surviving options are passed to the `Deliberation` process, which weighs the new options among themselves and incorporates them into the agent's plans. The `Filter Override` was included in the filtering process due to the possible limited agent knowledge, which creates situations in which the consideration of certain options would be interesting despite an indication by the agent's beliefs that these options are inconsistent. Therefore, even when a given option is eliminated by the `Compatibility Filter` it is possible that it might trigger a rule in the `Filter Override` that makes it a surviving option.

Despite IRMA being a highly abstract architecture, it was useful to find out some of the problems that other BDI architectures would have to deal with, such as the need for procedures to [11]:

- Propose new options when changes in the environment are detected;
- Evaluate conflicting options;
- Override the compatibility filter.

Some of the concepts outlined in IRMA's original work were put to a test in the Tileworld system [26], whose main objective was to be an agent architecture testbed for meta-level reasoning strategies. Essentially, the Tileworld system consists of a simulated robot agent and a simulated environment, which is dynamic and unpredictable. Both environment and agent were designed to be highly parameterized so that various situations in which an agent might find itself could be tested, and the behavior of these agent/environment matchings could be evaluated. The main components under scrutiny in the Tileworld agent were the Filtering Mechanism, comprised of the Compatibility Filter and the Filter Override (Figure 4), which is responsible for deciding whether a change in the environment should cause a reconsideration of the current agent's intentions, and the deliberation process. Various Deliberation Strategies of increasing complexity were tested against different environment set ups, which varied in several dimensions, in order to test its suitability to various environmental conditions. The experiments conducted over the Tileworld testbed lead to the conclusion that a filtering mechanism that allows only clear opportunities to be passed down to a deliberation process gets more desirable as the environment gets more dynamic. Using Bratman's notion of Cautiousness and Boldness [11], a Bold agent tends to perform better than a Cautious one in a dynamic environment [50]. Although these conclusions conform to the hypotheses outlined in previous works [11], the authors are careful not to assert their generality [50] regarding real-world applications, given that the environment where the agent was embedded was highly controlled.

### 4.2 PRS: Planning in a procedural way

The Procedural Reasoning System (PRS) [13] was created aiming at a BDI architecture that could be used in real-world applications. It was also meant to support both goal-directed and reactive reasoning. The system was first used in the implementation of a task control system for a NASA spacecraft simulator. This section will describe PRS architecture's components and then the processes that operate in these components through PRS's interpreter.

### 4.2.1 PRS Components

A PRS agent or module consists of a database containing the current system's beliefs about the world, a set of current objectives, a procedure or plan library whose components describe action and test sequences intended to achieve the proposed goals or to react to specific situations [51]. Also, PRS includes an intention structure that consists of a set of plans chosen at run time to be executed, and an interpreter that works as an inference mechanism. This inference mechanism manipulates these components and selects an adequate plan based on the system's beliefs and goals, putting this plan in the intention structure and executing it. PRS is organized as a series of architecture components whose control is delegated to an interpreter, as can be seen in Figure 5.

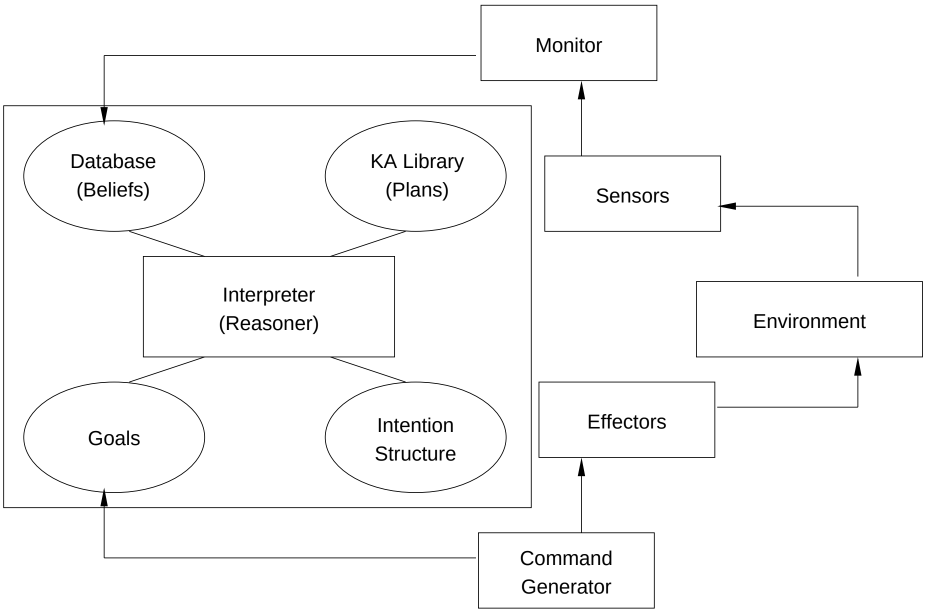

Figure 5: PRS architecture structure.

The first PRS implementation was made using the LISP language in a processor specific for this language. Therefore, a PRS agent description is made through a LISP-based language. Hence, throughout this section PRS component's syntactic forms will be described using the same syntax used by the authors in their original works [14].

**System Database (Beliefs).** The System Database can be viewed as the representation of the system's beliefs regarding the environment, represented as first-order predicates [14]. These beliefs may initially regard constant properties regarding the world and application domain of a PRS system, though they can evolve throughout system execution to include the agent's observations about the world, or even conclusions derived by the system using the knowledge contained within the database, which can change through time. The system database is also the way through which the agent receives information regarding the world [52]; in PRS it is assumed that there is an automatic process that includes new facts coming from the environment. Besides describing the world, the agent's beliefs may refer to the agent's structure itself and its internal states, including beliefs, goals and intentions. This kind of belief is called *meta-level* expression.

**Goals (Desires).** PRS agent goals are expressed as conditions over some interval of time [14], which in PRS is the equivalent of a sequence of world-states, described as temporal operations over state descriptions (see world-states and temporal operations [17]). Goals describe tasks and desired behaviors, which can be expressed in PRS logic in the following goal types [27, 51]:

- Reach a given condition `(!  C)`;
- Test a given condition `(?  C)`;
- Wait until a given condition is true `(^C)`;
- Maintain a condition true `(# C)`;
- Assert a condition as being true `(→ C)`;
- Retract a condition `( > C)`;
- Conclude that a given condition is true `(=> C)`.

Goals are divided into two types: *intrinsic* and *operational.* Intrinsic goals come from the process of executing a plan (or KA, see below) in the intention structure (see Section 4.2.2), that is, they are not intermediate goals derived from a main goal. Operational goals, on the other hand, represent intermediate steps in the process of fulfilling an intrinsic goal. As with beliefs, PRS allows goals to be defined as *meta-level*, thus allowing the specification of goals regarding the internal system behavior.

**Knowledge Area (Plans).** The knowledge of how to achieve a given objective in PRS is described by declarative procedure specifications called Knowledge Areas (KAs) [14, 27]. These specifications use the notion of procedural knowledge [12]. KAs are represented by a *body* and an activation condition. These two components specify a sequence of steps to achieving a given objective in a given situation. A KA body can be viewed as a plan or a plan schema. Such component is represented by a directed graph with a start node and one or more finish nodes. The graph's arcs are labelled with sub-goals to be achieved throughout plan execution. The execution of a KA is said to be successful when the arcs that connect a start goal to a finish goal are reached, and, through this path, all the specified sub-goals are satisfied. That means that a graph path in a KA is actually multi-dimensional, given that satisfying sub-goals may require other KAs to be executed. It is possible that some KAs do not have a body, in which case they are called primitive KAs, because they have some kind of primitive action tied to them that is directly executable by the system. The authors also note that the KA graph construction formalism allows the use of various constructs that allow the control of the execution flow, such as conditional branches, iterations and recursions.

The invocation condition of a KA is divided into two parts: the *triggering part* and the *Context Part* [14]. The triggering part of an invocation condition is a logic expression that describes the events that must take place in order for the KA to be executed. These events may consist in the acquisition of new goals, in which case the reasoning is goal oriented, or the modification of the agent's beliefs, in which case the reasoning is data oriented or reactive. The context part of an invocation condition specifies the conditions that must be true regarding the current system state for the associated KA to be executed. KAs, as beliefs and goals, are not limited to dealing with the environment that surrounds the agent, they can also be used to manipulate beliefs, desires and intentions of PRS itself. These KAs are therefore called *meta-level* KAs [14, 51]. One of the objectives of such KAs is the modification of PRS interpreter standard behavior in dealing with reasoning. This might include the modification of plans during execution, establish new goals or even the modification of beliefs during the execution of a meta-level KA.

**Intention Structure.** The intention structure contains the tasks that the system has chosen for immediate or later execution, these tasks are called intentions [14, 27]. An intention is comprised of a KA chosen to fulfill a goal, along with all the KAs needed for the initial KA to be completed. Intentions in the intention structure may be active, suspended or delayed, for example, waiting for a condition to become true. It is important to point out that for every intrinsic goal being pursued there is a proper intention, and that for each intention there is a stack of KAs to be executed in order to achieve the KA.

The KAs included in the intention structure are partially ordered, with possibly more than one KA as the root of the intention structure. The order established for the executing KAs is followed so that for a given KA to be executed it is necessary for all the preceding KAs to be either executed or dropped. Within the intention structure there is no differentiation regarding the nature of KAs. Therefore, meta-level KAs are not treated differently than regular KAs during processing. The commitment to intentions defined by Bratman *et al.* in [11] is implemented in PRS so that once a given KA is chosen to fulfill a goal, no other KAs will be selected to fulfill the same goal, even if their activation condition is satisfied [14]. That is, PRS is committed to the plan chosen by a given KA, and it will only consider other means to achieving the goal in case the initial way becomes impossible.

**System Interpreter.** The interpreter is responsible for the interaction of PRS components, and its operation process as simplified as possible so that the minimum system reaction time can be attained [14, 27, 52]. Considering the existing goals and beliefs in a given moment, one or more KAs may become eligible for execution, and one or more of these KAs will be chosen for inclusion to the intention structure (*i.e.* chosen for execution). In order for the interpreter to check whether the KAs will be executed, the interpreter only tries to unify the KA execution condition with the system's beliefs. The authors point out that, if any other more complex inference process were used, it would not be possible to prove that the execution time for the KA selection process is limited [14]. In order for more complex inference processes to be executed, meta-level KAs should be used [14]. The use of meta-level KAs does not violates the system's reactive capability since these are treated just like any other KAs, therefore being possible that new KAs take precedence over the meta-level KAs used to make complex inferences.

### 4.2.2 PRS functioning

PRS operation was designed so that, even without any high-level information regarding how the system reasoning process should function, it would be possible for it to work in some kind of pre-defined way. On the other hand, it is possible to add meta-level information to the system so as to refine system behavior, while keeping a bounded reaction time for the basic process.

**Interpreter Cycle.** The main process that controls PRS operation is depicted in the center of Figure 5 by the Interpreter Box. This process is responsible for selecting KAs so as to react to changes in the environment, achieve the goals established by the goal acquisition and dropping process, or to further the execution of a previously selected KA. The Interpreter initiates its execution cycle based on the set of goals and beliefs contained within its database. The known KAs are analyzed while the system tries to unify the KAs invocation conditions to the current beliefs and goals; from this unification, a set of eligible KAs is formed. Among these KAs, some are selected for actual execution. The selected KAs are inserted into the intention structure based on the following criteria: if the KA was selected due to the acquisition of a new intrinsic goal or a new belief, then this KA will be inserted in the intention structure as a new intention, otherwise (*i.e.* the KA was selected as the result of an operational goal) the KA will be inserted in the stack of KAs that comprise the corresponding intention. The next step is the selection of an intention in the root of the intention structure so that another step from this intention can be executed. This step might be the establishment of one or more new goals or the execution of a primitive action. The execution of a primitive action may result in the modification of the system's beliefs either by being a meta-level action or by modifying the environment in such a way that it will be later reflected by the beliefs. At the end of this step, the cycle repeats itself. The interpreter execution may be summarized as follows:

1. The interpreter receives new beliefs and goals;
2. Based on that new data, the interpreter selects appropriate plans, or KA;
3. The KAs selected for execution will be inserted in the intention structure;
4. Among the roots of the intention structure, one intention is selected;
5. On step of the KA selected in the previous step is executed;
6. This step might be the execution of a primitive action or the establishment of a new objective.

**Intention States.** An intention in PRS can be in three possible states: active, suspended or conditionally suspended [14]. An active intention can be executed as soon as it becomes a root in the intention structure. A suspended intention has been adopted by the system, but it has no definition about when it is supposed to be executed, therefore it must be explicitly activated before it can be executed. A conditionally suspended intention is temporarily suspended until a given activation condition is met. The suspension of an intention can be performed through meta-level KAs. When a suspended intention is reactivated, it is necessary for the system to decide whether the intention structure needs to be reorganized or not. This reorganization can be performed through meta-level KAs, though PRS interpreter's default behavior is to prioritize the execution of re-activated intentions, so as to minimize reaction time for the event that cause such re-activation.

**Goal Establishment and Dropping.** As described in before, intentions are inserted into the intention structure due to changes in the objectives or beliefs. As a result of this insertion, other sub-goals might also be inserted into the intention structure. The possibility that these goals may fail must be dealt with accordingly [14]. The authors state it is necessary to determine how the system should react in case a failure occurs, deciding which alternate course of action must be taken in order to reach the goal. On the other hand, it must be established when a failed goal becomes impossible to be fulfilled, *i.e.* that there is no other alternate way to achieve it. It will hardly be possible to establish at run time that a given goal has become impossible, therefore meta-level KAs can be included in order to deal with the question of alternate execution paths. When these are lacking, PRS will execute every KA that could possibly fulfill a given goal exactly once. In this case, it is still possible to include a meta-level KA to deal with a failure of every plan available to fulfill the given goal, perhaps reconsidering a specific prior plan whose re-execution might seem more promising in a second try.

### 4.2.3 PRS Descendants

PRS has been used to underpin a variety of implementations of the BDI agent model, it has also evolved to a number of other works aiming to solve some of its original shortcomings. Two of the most notable direct PRS descendants are dMARS and AgentSpeak, whose main objectives were, respectively, to create a formal definition of an agent system that was suitable for implementation, and to formally define an agent specification language and its semantics. These works are briefly described below.

**dMARS.** The distributed Multi-Agent Reasoning System (dMARS) [43] is a PRS implementation that was used as a reference for formal specification in the Z specification language. This specification models an *ideal* dMARS implementation, therefore implying that no equivalence guarantee is given between the specification and the implementation used as base. The system described in [43] uses the same notion of beliefs as PRS, which is the use of PROLOG-like ground literals of classical first-order logic. dMARS reduces the amount of possible goal types from the original seven in PRS to achievement goals (`!C`), query (test) goals (`?C`), and distinguishes the assertion and retraction goals present in PRS as *internal actions*, naming them, respectively, `add` and `remove` action. It also formalizes agent interaction with the environment through the notion of *external actions*, which are used to perform some arbitrary operation defined by the system programmer. In the same way as the original PRS an agent intentions in dMARS are represented by the currently adopted plans. Though the events that trigger the adoption of a new intention have been expanded from the original addition and removal of beliefs and adoption of a new goal to contain also the receipt of a message [43]. The definition of plans in dMARS is performed through a formalized version of PRS plan graphs, and the two components of PRS invocation condition were refined to the notions of relevance and applicability. One of the greatest contributions of dMARS formalization is the definition of an interpreter algorithm unambiguously.

**AgentSpeak(L).** The AgentSpeak language [19] was created in order to diminish the distance between BDI agent theory and practice. That distance is due to a number of factors [19, 53], for example, implementations are generally conceived in a simplified manner, resulting in the weakening of its theoretical underpinning, or the logics used in those theories are usually very weakly related to practical problems. Thus, the goal to diminish that distance should be attained through a formal specification of agents, which would also be used to underpin their implementation. Hence, AgentSpeak(L) defines an agent specification language whose operational model is formally defined. Such operation should match that of PRS implementation and dMARS. AgentSpeak(L) is a programming language based on a restricted first-order language with events and actions, where BDI model components such as Beliefs, Desires and Intentions are not explicitly represented as modal formulas [19]. Although AgentSpeak(L) was conceived to provide a tighter theoretical underpinning to already implemented systems such as PRS and dMARS, this goal cannot be considered fully achieved, given that AgentSpeak(L) is based on a number of simplifications over PRS. These simplifications were also applied to dMARS, as can be observed in dMARS formal specification [43]. Besides, when dMARS was formally specified later, no effort towards relating it to AgentSpeak(L) can be perceived. Hence, these two works can be considered more as parallel works than completely equivalent formalisms, as can be ascertained in a refined specification of AgentSpeak(L) [53]. The most significant simplifications observed in AgentSpeak(L) regarding PRS are:

- **Goal Types:** In PRS it is possible to specify the following goal types [27, 51]: to reach or test a condition, to wait for a condition to be true or to maintain the condition true, to assert or retract the truth value of a condition and to conclude that the condition is true. In AgentSpeak(L) it is possible to declare goals of achievement and testing of a given condition over the world, also it is possible to assert and retract conditions over the world through AgentSpeak(L) basic actions. This clearly results in diminished expressivity by the language tied to AgentSpeak formal model;
- **Meta-level components:** In AgentSpeak(L) it is not possible to specify meta-level components such as those that are possible in PRS, thus limiting behavior flexibility in the definition of a given agent.

**INTERRAP.** Another architecture that follows PRS philosophy of *to do* goals is the INTERRAP (Integration of Reactive Behavior and Rational Planning) architecture [42]. INTERRAP is not closely related to any formal theory but innovates in its definition as a layered architecture in which single-agent PRS-like reasoning operates concurrently with reasoning regarding communication and cooperation.

### 4.3 X-BDI: Non-monotonic reasoning using explicit negation

The X-BDI agent model was created in order to allow a formal agent specification to be directly executed [40]. That is possible because X-BDI's language is defined in terms of a formalism that has a reference implementation, which is called Extended Logic Programming with explicit negation (ELP) using the Well-Founded Semantics extended for the explicit Negation (WFSX), with a derivation procedure is called Selected Linear Derivation for extended programs (SLX) [54].

X-BDI uses a variety of ELP properties. Specifically, X-BDI uses ELP's ability to deal with contradiction to implement a variety of non-monotonic reasoning processes, necessary for the BDI model. Moreover, a modified form of Event Calculus [55] is used in order to allow X-BDI to deal with a dynamic world.

### 4.3.1 X-BDI Operation

Considering that the X-BDI model is not a complete architecture, its operation processes may be implemented in a multitude of ways. This section will describe X-BDI's most important components and processes and the modifications implemented for this work.

An X-BDI agent has the traditional components of a BDI agent, *i.e.* a set of Beliefs, Desires and Intentions. Besides, given its extended logic definition, it has also a set of time axioms defined through a variation of the *Event Calculus* [40, 55].

The set of beliefs is simply a formalization of facts in ELP, individualized for a specific agent. The belief revision process in X-BDI is the result of the program revision process performed in ELP by the SLX procedure. From the agent's point of view, it is assumed that its beliefs are not always consistent, because whenever an event that makes the beliefs inconsistent, SLX will minimally revise the program, and therefore, the beliefs.

Every desire in an X-BDI agent is conditioned to the body of a logic rule, which is a conjunction of literals called *Body*. Thus, *Body* specifies the pre-conditions that must be satisfied in order for an agent to desire a property. When *Body* is an empty conjunction, property $P$ is unconditionally desired. Desires may be temporally situated, *i.e.* can be desired in a specific moment, or whenever its pre-conditions are valid. Besides, desires have a priority value used in the formation of an order relation among desire sets.

There are two possible types of intentions: Primary Intentions, which refer to the intended properties, and Relative Intentions, which refer to actions able to bring about these properties. An agent may intend something in the past or that is already true. Besides, intentions may not be impossible, *i.e.* there must be a course of action available to the agent whose result is a world state where the intended property is true.

The reasoning process performed by X-BDI initiates with the selection of Eligible Desires, which represent the unsatisfied desires whose pre-conditions were satisfied. The elements of this set are not necessarily consistent among themselves. Candidate Desires are then generated, which represent a set of Eligible Desires that are both consistent and possible and will be later adopted as Primary Intentions. In order to satisfy the properties represented by Primary Intentions, the planning process generates a sequence of temporally ordered actions that constitute the Relative Intentions. This process is summarized in Figure 6.

Eligible desires have rationality constraints that are similar to those imposed over the intentions in the sense that an agent will not desire something in the past or something the agent believes will happen without his interference. Agent beliefs must also support

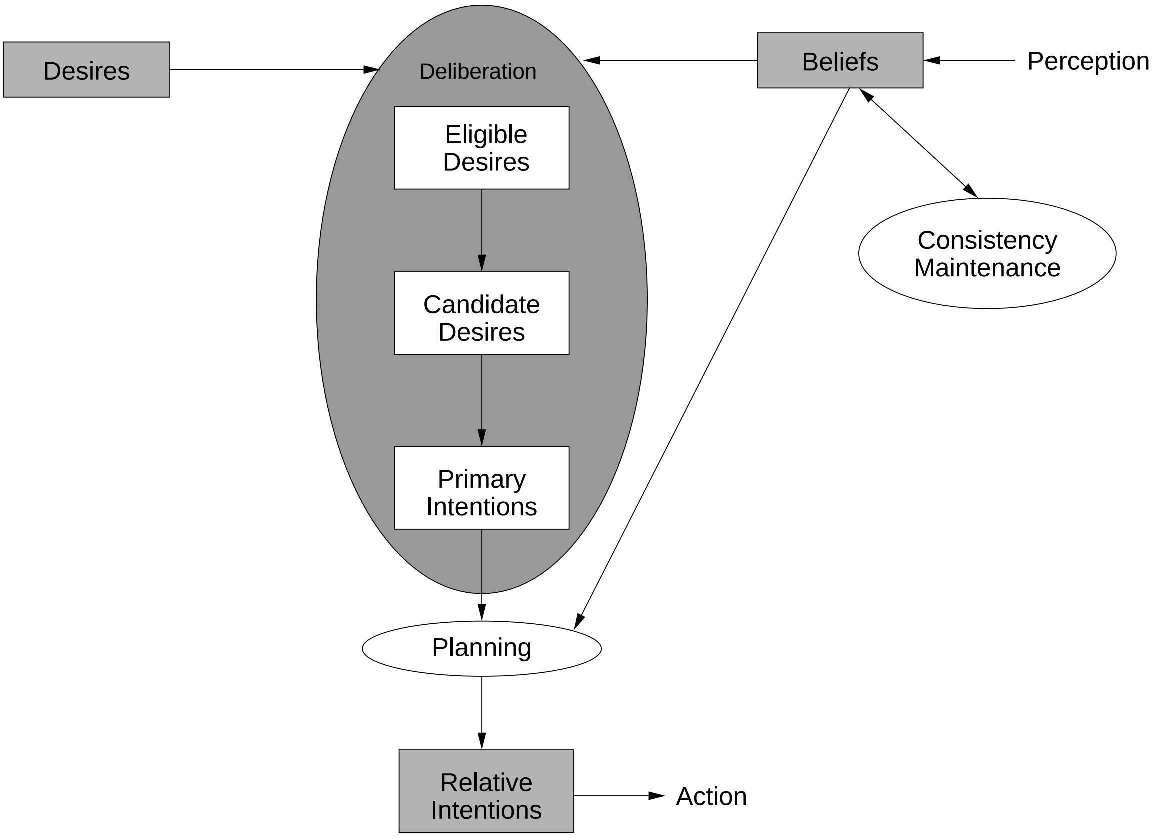

Figure 6: X-BDI operation overview.

the pre-conditions defined in the desire *Body*. Within the agent's reasoning process these desires will originate a set of mutually consistent subsets ordered by a partial order relation.

The process of selecting Candidate Desires seeks to choose among the Eligible Desires one subset that contains only desires that are internally consistent and possible. A possible desire is one that has a property $P$ that can be satisfied through a sequence of actions. In order to choose among multiple sets of Candidate Desires, X-BDI uses ELP constructs that allow the definition of preferred revisions. Thus, X-BDI defines a desire preference relation through a set of preferred revisions generated using the priorities expressed in the desires. Through this preference relation, a desire preference graph that relates all subsets of Eligible Desires is generated. X-BDI verifies the possibility of a desire through the abduction of an Event Calculus Theory in which the belief in the validity of a desired property $P$ could be true. Such abduction process is, actually, a form of planning. A set of Candidate Desires is the subset of Eligible Desires with the greater preference value, and whose properties can be satisfied. Satisfiability is verified through the abduction of a theory in event calculus that supports the belief on the validity of the desired properties at some point in the future. The $P$ properties present in the Candidate Desires are used to generate the set primary intentions.

Primary Intentions can be seen as high-level plans, similar to the intentions in IRMA [11]. Hence, they represent the agent's commitment to a course of action, which will be performed through a series of refinements up to the point where an agent has a temporally ordered set of actions representing a concrete plan towards the satisfaction of its goals. Relative Intentions correspond to the temporally ordered steps of the concrete plans generated to satisfy the agent's Primary Intentions. The notion of agent commitment results from the fact that Relative Intentions must be non-contradictory regarding Primary Intentions.

### 4.3.2 Intention Revision

The computational effort and the time required to reconsider the whole set of intentions of a resource-bounded agent is generally significant regarding the environment change ratio. Therefore, intention reconsideration should not occur constantly, but only when the world changes in such a way as to threaten the plans an agent is executing or when an opportunity to satisfy more important goals is detected. As a consequence, X-BDI uses a set of reconsideration "triggers" generated when intentions are selected, and causes the agent to reconsider its course of action when activated.

If all of the agent's Primary Intentions are satisfied before the time planned for them to be satisfied, the agent will restart the deliberative process, for he has achieved his goals. On the other hand, if one of the Primary Intentions has not been achieved at the time planned for it, the agent will have to reconsider its intentions because its plans have failed. Moreover, if a desire with a higher priority than the currently selected desires becomes possible, the agent will reconsider its desires in order to take advantage of the new opportunity. Reconsideration is completely based on integrity constraints over beliefs. Therefore, considering that beliefs are revised at every sensoring cycle, it is possible that a reconsideration occurs due the "triggering" of a reconsideration restriction.

## 5. Concluding Remarks

In this survey we traced the evolution of the BDI model of agents from its abstract start to some of its concrete instances. Throughout this process it is possible to identify two basic branches of BDI agent research whose aim is to define a functional BDI theory:

- The first one aims at the creation of a complete logic system for rational agents with distinct semantics and proof methods, either by refining existing basic theories of agency or by defining them from the ground up. The major advantage of their results lies in the flexibility of these theories, since they can be more easily modified to accommodate new modifications. Its drawback is that the creation of computationally efficient proof procedures for those logics might take much more time to be developed. This approach can be clearly identified in works such as [18, 39, 56];

- The second branch aims to reach a complete BDI theory through the augmentation of theories that are underpinned in existing logical proof procedures. These theories have can benefit from an existing body of research, even though the resulting theory must abide to some restrictions. Examples of this approach can be clearly identified in works such as [53, 31, 22]

Current research regarding BDI agent systems seems to be directed towards forms of solving the *Agent Design Problem* [57], either by verifying plan libraries used in an agent before it is deployed [58] or by using a form of means-end reasoning that deals appropriately with the problem [22, 59].

Some authors suggest the use of other mental states besides beliefs, desires and intentions. Among these different mental states, some are proposed aiming to facilitate the comprehension and optimize the operation of the models for which they were defined, such as objectives and plans [42], commitments and capacities [60], among others. Due to the fact that an agent's set of desires is potentially inconsistent, some authors suggest the use of an auxiliary mental state between desires and intentions, this intermediate state could, for instance represent an internally consistent subset of an agent's desires (*i.e.* each desire in the set is consistent with every other) and possibly externally (*i.e.* the desires are consistent with the agent's beliefs) [39, 42, 44]. The ability of an agent to decide how to attain its objectives is essential in the deliberative process [11]. Therefore some authors [16] model agent's plans as a sequence of intentions adopted at a given moment, and the origins of those intentions are pre-defined plans stored in a plan library. Besides that approach, it is possible to use a planner that generates completely new plans at run time so that the agent becomes more flexible at the cost of time efficiency [59]. The plans generated this way might be stored in a plan library, imitating somewhat the learning process of human beings in the sense that once you have done something one time, it is easier to perform it faster in the future. In one definition of agent-oriented programming [60], the author proposes the use of two mental states that are conceptually different from those in the BDI model: obligations and capabilities, besides using an alternate mental state called decision, which is just another denomination for intention. Obligations are mental states that perform a social role that have as a goal the creation of a relationship between two agents $a$ and $b$, where $a$ is committed to trying to make the world to maintain a given proposition true to $b$. Capabilities have the role of specifying formally which actions a given agent is able to perform, so that an agent will only try to perform a given action in case it believes he can do so.

## References

[1] Nicholas R. Jennings. On agent-based software engineering. *Artificial Intelligence*, 117(2):277–296, 2000.

[2] Nicholas R. Jennings. Agent-Oriented Software Engineering. In Francisco J. Garijo and Magnus Boman, editors, *Proceedings of the 9th European Workshop on Modelling Autonomous Agents in a Multi-Agent World: Multi-Agent System Engineering (MAAMAW-99)*, volume 1647, pages 1–7. Springer-Verlag: Heidelberg, Germany, 1999.

[3] Michael Wooldridge and Paolo Ciancarini. Agent-oriented software engineering: The state of the art. In P. Ciancarini and M. Wooldridge, editors, *First International Workshop on Agent-Oriented Software Engineering (AOSE)*, volume 1957, pages 1–28, Limerick, Ireland, 2000. Springer-Verlag, Berlin.

[4] Michael Wooldridge. Agent-based software engineering. *The Institution of Electrical Engineers (IEE) Proceedings on Software Engineering*, 144(1):26–37, February 1997.

[5] Les Gasser. Social conceptions of knowledge and action: DAI foundations and open systems semantics. *Artificial Intelligence*, 47(1-3):107–138, 1991.

[6] Ronald Ashri, Michael Luck, and Mark D'Inverno. Infrastructure support for agent-based development. In M. Fisher M. D'Inverno, M. Luck and C. Preist (Eds.), editors, *Foundations and Applications of Multi-Agent Systems*, Lecture Notes in Artificial Intelligence 2403, pages 73–88. Springer-Verlag, 2002.

[7] Stefan Bussmann, Nicholas R. Jennings, and Michael Wooldridge. On the identification of agents in the design of production control systems. In Paolo Ciancarini and Michael Wooldridge, editors, *Proceedings of the First International Workshop on Agent-Oriented Software Engineering (AOSE), LNCS 1957*, pages 141–162, Limerick, Ireland, 2000. Springer Verlag.

[8] Cora B. Excelente-Toledo, Rachel A. Bourne, and Nicholas R. Jennings. Reasoning about commitments and penalties for coordination between autonomous agents. In *Proceedings of the 5th International Conference on Autonomous Agents*, pages 131–138. ACM Press, 2001.

[9] S. Shaheen Fatima and Michael Wooldridge. Adaptive task resources allocation in multi-agent systems. In *Proceedings of the 5th International Conference on Autonomous Agents*, pages 537–544. ACM Press, 2001.

[10] Franco Zambonelli, Nicholas R. Jennings, and Michael Wooldridge. Organisational abstractions for the analysis and design of multi-agent systems. In Paolo Ciancarini and Michael Wooldridge, editors, *Agent-Oriented Software Engineering, First International Workshop, AOSE 2000, Limerick, Ireland, June 10, 2000, Revised Papers*, volume 1957 of *Lecture Notes in Computer Science*. Springer, 2001.

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

[12] Michael P. Georgeff and Amy L. Lansky. Procedural knowledge. *Proceedings of the IEEE, Special Issue on Knowledge Representation*, 74(10):1383–1898, 1986.

[13] Michael P. Georgeff and Amy L. Lansky. Reactive reasoning and planning. In *Proceedings of the American Association for Artificial Intelligence (AAAI)*, pages 677–682, Seattle, WA, 1987. Morgan Kaufmann Publishers.

[14] Michael P. Georgeff and François Félix Ingrand. Decision-making in an embedded reasoning system. In *Proceedings of the 11th International Joint Conference on Artificial Intelligence (IJCAI'89)*, pages 972–978, Detroit, MI, 1989. Morgan Kaufmann.

[15] Michael Georgeff, Barney Pell, Martha Elizabeth Pollack, Milind Tambe, and Michael Wooldridge. The belief-desire-intention model of agency. In Jörg Müller, Munindar P. Singh, and Anand S. Rao, editors, *Proceedings of the 5th International Workshop on*

*Intelligent Agents V : Agent Theories, Architectures, and Languages (ATAL-98)*, volume 1555, pages 1–10. Springer-Verlag: Heidelberg, Germany, 1999.

[16] Anand S. Rao and Michael P. Georgeff. Modeling rational agents within a BDI-architecture. In James Allen, Richard Fikes, and Erik Sandewall, editors, *Proceedings of the 2nd International Conference on Principles of Knowledge Representation and Reasoning (KR'91)*, pages 473–484, San Mateo, CA, USA, 1991. Morgan Kaufmann Publishers.

[17] Anand S. Rao and Michael P. Georgeff. Deliberation and its role in the formation of intentions. In *Proceedings of the Seventh Annual Conference on Uncertainty in Artificial Intelligence (UAI–91)*, pages 300–307, San Mateo, California, 1991. Morgan Kaufmann Publishers.

[18] Anand S. Rao and Michael P. Georgeff. BDI-agents: from theory to practice. In *Proceedings of the First International Conference on Multiagent Systems*, San Francisco, 1995.

[19] Anand S. Rao. AgentSpeak(L): BDI agents speak out in a logical computable language. In Rudy van Hoe, editor, *7th European Workshop on Modelling Autonomous Agents in a Multi-Agent World LNCS 1038*, pages 42–55, Eindhoven, Netherlands, 1996. Springer Verlag.

[20] Rafael H. Bordini, Ana L. C. Bazzan, Rafael de O. Jannone, Daniel M. Basso, Rosa M. Vicari, and Victor R. Lesser. AgentSpeak(XL): efficient intention selection in BDI agents via decision-theoretic task scheduling. In *Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 1294–1302, Bologna, Italy, 2002. ACM Press.

[21] Holger Knublauch. Extreme programming of multi-agent systems. In *Proceedings of the first international joint conference on Autonomous agents and multiagent systems*, pages 704–711. ACM Press, 2002.

[22] Naoyuki Nide and Shiro Takata. Deduction systems for BDI logics using sequent calculus. In *Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 928–935. ACM Press, 2002.

[23] Michael E. Bratman. *Faces of Intention: Selected Essays on Intention and Agency.* Cambridge University Press, Cambridge, MA, 1999.

[24] Phillip R. Cohen and Hector J. Levesque. Intention is choice with commitment. *Artificial Intelligence*, 42(2-3):213–261, 1990.

[25] Anand S. Rao and Michael P. Georgeff. Formal models and decision procedures for multi-agent systems. Technical Report 61, Australian Artificial Intelligence Institute, 171 La Trobe Street, Melbourne, Australia, 1995. Technical Note.

[26] Martha Elizabeth Pollack and Marc Ringuette. Introducing the tileworld: experimentally evaluating agent architectures. In Thomas Dietterich and William Swartout,

editors, *Proceedings of the 8th National Conference on Artificial Intelligence*, pages 183–189, Menlo Park, CA, 1990. AAAI Press.

[27] Michael P. Georgeff and François Félix Ingrand. Monitoring and control of spacecraft systems using procedural reasoning. In *Proceedings of the Space Operations and Robotics Workshop*, Houston, TX, July 1989.

[28] Daniel C. Dennett. *The Intentional Stance.* MIT Press / Bradford Books, Cambridge, MA, 1987.

[29] Daniel C. Dennet. *Brainstorms: Philosophical Essays on Mind and Psychology.* Bradford Books and Hassocks, Montgomery, VT, 1978.

[30] John McCarthy. Ascribing mental qualities to machines. Technical Report 326, Stanford University AI Lab, 1979. Technical Report.

[31] Michael C. Móra. *A Formal and Executable Model of BDI Agents.* PhD thesis, CPGCC/UFRGS, 1999. In Portuguese.

[32] John R. Searle. What is an intentional state? In H. Dreyfuss and H. Hall, editors, *Husserl, Intentionality and Cognitive Science*, volume 42, pages 213–261. 1984.

[33] Yoav Shoham and Steve B. Cousins. Logics of mental attitudes in AI. In G. Lakemeyer and B. Nebel, editors, *Foundations of Knowledge Representation and Reasoning*, pages 296–309. Springer-Verlag, Deutschland, 1994.

[34] George Kiss and Han Reichgelt. Towards a semantics of desires. In E. Werner and Y. Demazeau, editors, *Decentralized AI 3 — Proceedings of the 3rd European Workshop on Modelling Autonomous Agents in a Multi-Agent World (MAAMAW-91)*, pages 115–128, Amsterdam, Netherlands, 1992. Elsevier Science Publishers Ltd.

[35] Donald Davidson. *Actions, Reasons, and Causes*, chapter 1, pages 3–20. Oxford University Press, 1963.

[36] Michael E. Bratman. *Intentions, Plans and Practical Reason.* CLSI Publications, Stanford, CA, 1999.

[37] Michael E. Bratman. What is intention? *Intentions in Communication*, 1990.

[38] Richard Fikes and Nils Nilsson. STRIPS: A new approach to the application of theorem proving to problem solving. *Artificial Intelligence*, 2(3-4):189–208, 1971.

[39] Michael Wooldridge. *Reasoning about Rational Agents.* The MIT Press, 2000.

[40] Michael C. Móra, José G. Lopes, Rosa M. Viccari, and Helder Coelho. BDI models and systems: Reducing the gap. In *Proceedings of the 5th International Workshop on Intelligent Agents V : Agent Theories, Architectures, and Languages (ATAL-98)*, Germany, 1999. Springer Verlag.

[41] Michael Wooldridge and Nicholas Jennings. Intelligent agents: Theory and practice. *The Knowledge Engineering Review*, 10(2):115–152, 1995.

[42] Jörg P. Müller. *The Design of Intelligent Agents: A Layered Approach.* Springer-Verlag, Germany, 1996.

[43] Mark d'Inverno, David Kinny, Michael Luck, and Michael Wooldridge. A formal specification of dMARS. In *Agent Theories, Architectures, and Languages*, pages 155–176. Springer-Verlag, 1998.

[44] Michael Wooldridge. *Intelligent Agents*, chapter 2. The MIT Press, 1999.

[45] Munindar P. Singh. Towards a formal theory of communication for multi-agent systems. In John Mylopoulos and Ray Reiter, editors, *Proceedings of the 12th International Joint Conference on Artificial Intelligence (IJCAI-91)*, pages 69–74, Sydney, Australia, 1991. Morgan Kaufmann publishers Inc.: San Mateo, CA, USA.

[46] Joseph Y. Halpern and Yoram Moses. A guide to completeness and complexity for modal logics of knowledge and belief. *Artificial Intelligence*, 54(2):319–379, 1992.

[47] E. Allen Emerson. *Temporal and Modal Logic*, volume B, chapter 16, pages 996–1072. Elsevier Science Publishers Ltd., 1990. Handbook of Theoretical Computer Science.

[48] Paul R. Cohen, Michael L. Greenberg, David M. Hart, and Adele E. Howe. Trial by fire: Understanding the design requirements for agents in complex environments. *AI Magazine*, 10(3):32–48, 1989.

[49] Afsaneh Haddadi. *Communication and Cooperation in Agent Systems: A Pragmatic Theory*, volume 1056 of *LNCS*. Springer-Verlag, 1996.

[50] Martha Elizabeth Pollack, D. Joslin, A. Nunes, S. Ur, and E. Ephrati. Experimental investigation of an agent commitment strategy. Technical Report 94–31, University of Pittsburgh, Pittsburgh, PA 15260, 1994.

[51] François F. Ingrand, Michael P. Georgeff, and Anand S. Rao. An architecture for real-time reasoning and system control. *IEEE Expert, Knowledge-Based Diagnosis in Process Engineering*, 7(6):33–44, 1992.

[52] François Félix Ingrand and Vianney Coutance. Real-time reasoning using procedural reasoning. Technical Report 93104, LAAS, January 2001. Technical Report.

[53] Mark d'Inverno and Michael Luck. Engineering AgentSpeak(L): A formal computational model. *Journal of Logic and Computation*, 8(3):233–260, 1998.

[54] José Júlio Alferes and Luíz Moniz Pereira. *Reasoning with Logic Programming.* Springer Verlag, 1996.

[55] Robert A. Kowalski and Marek J. Sergot. A logic-based calculus of events. *New Generation Computing*, 4(1):67–95, 1986.

[56] Wiebe Van der Hoek and Michael Wooldridge. Towards a logic of rational agency. *Logic Journal of the IGPL*, 11(2):133–157, March 2003.

[57] Michael Wooldridge. The computational complexity of agent design problems. In E. Durfee, editor, *Proceedings of the Fourth International Conference on Multi-Agent Systems (ICMAS 2000)*, pages 341–348. IEEE Press, 2000.

[58] Rafael H. Bordini, Michael Fisher, Carmen Pardavila, and Michael Wooldridge. Model checking AgentSpeak. In *Proceedings of the 2nd International Conference on Autonomous Agents and Multiagent Systems (AAMAS-03)*, pages 409–416, Melbourne, Australia, July 2003. ACM Press.

[59] Felipe Rech Meneguzzi, Avelino Francisco Zorzo, and Michael Da Costa Móra. Propositional planning in BDI agents. In *Proceedings of the 2004 ACM Symposium on Applied Computing*, Nicosia, Cyprus, 2004. ACM Press. To be published.

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