# Towards a Logic of Rational Agency

Wiebe van der Hoek, *Dept of Computer Science, University of Liverpool, Liverpool L69 7ZF, UK, E-mail: wiebe@csc.liv.ac.uk*

Michael Wooldridge, *Dept of Computer Science, University of Liverpool, Liverpool L69 7ZF, UK, E-mail: mjw@csc.liv.ac.uk*

## Abstract

Rational agents are important objects of study in several research communities, including economics, philosophy, cognitive science, and most recently computer science and artificial intelligence. Crudely, a rational agent is an entity that is capable of acting on its environment, and which chooses to act in such a way as to further its own best interests. There has recently been much interest in the use of mathematical logic for developing formal theories of such agents. Such theories view agents as practical reasoning systems, deciding moment by moment which action to perform next, given the beliefs they have about the world and their desires with respect to how they would like the world to be. In this article, we survey the state of the art in developing logical theories of rational agency. Following a discussion on the dimensions along which such theories can vary, we briefly survey the logical tools available in order to construct such theories. We then review and critically assess three of the best known theories of rational agency: Cohen and Levesque's intention logic, Rao and Georgeff's BDI logics, and the KARO framework of Meyer et al. We then discuss the various roles that such logics can play in helping us to *engineer* rational agents, and conclude with a discussion of open problems.

*Keywords*: rational agents, informational, normative and pro attitudes

## 1 Introduction

Rational agents are the central objects of study in many research disciplines, including economics [52], philosophy [17], cognitive science [73], and most recently, computer science and artificial intelligence [79]. Put crudely, an agent is an entity that is situated in some environment and is capable of acting upon it in order to modify, shape, and control it; a *rational* agent is one that acts in its own best interests. The notion of "best interest" is commonly captured by assuming that an agent has preferences over possible outcomes — an agent is then rational if it chooses to act in such a way as to optimise its outcome with respect to these preferences.

There has recently been much interest in the use of mathematical logic for developing formal theories of such agents. Much of this interest arises from the fact that rational agents are increasingly being recognised as an important concept in computer science and artificial intelligence [78, 40]. Logical theories of rational agency view agents as *practical reasoning systems*, deciding moment by moment which action to perform next, given the beliefs they have about the world and their desires with respect to how they would like the world to be. In this article, we survey the state of the art in developing logical theories of rational agency. Following a discussion on the dimensions along which such theories can vary, we briefly survey the logical tools available in order to construct such theories. We then review and critically assess

 

three of the best known theories of rational agency: Cohen and Levesque's intention logic, Rao and Georgeff's BDI logics, and the KARO framework of Meyer et al. We then discuss the various roles that such logics can play in helping us to *engineer* rational agents, and conclude with a discussion of open problems.

## *1.1 Dimensions of Rational Agency*

Agents are the producers of action. Agents perform actions in order to shape and modify the environment they inhabit; they are not merely the passive recipients of actions, but rather they are active and purposeful. In attempting to understand the behaviour of agents in the everyday world, we frequently make use of *folk psychology*:

> Many philosophers and cognitive scientists claim that our everyday or "folk" understanding of mental states constitutes a theory of mind. That theory is widely called "folk psychology" (sometimes "commonsense" psychology). The terms in which folk psychology is couched are the familiar ones of "belief" and "desire", "hunger", "pain" and so forth. According to many theorists, folk psychology plays a central role in our capacity to predict and explain the behavior of ourselves and others. However, the nature and status of folk psychology remains controversial. [80]

For example, we use statements such as *Michael intends to write a paper* in order to explain Michael's behaviour. Once told this statement, we expect to find Michael shelving other commitments and developing a plan to write the paper; we would expect him to spend a lot of time at his computer; we would not be surprised to find him in a grumpy mood; but we *would* be surprised to find him at a late night party. The philosopher Dennett coined the phrase *intentional system* to refer to an entity that is best understood in terms of folk-psychology notions such as beliefs, desires, and the like [17]. This was also what Hofstadter was referring to already in '81, when one of his characters, Sandy, puts the following forward in a Coffee House Conversation ([36]):

> But eventually, when you put enough feelingless calculations toghether in a huge coordinated organization, you'll get something that has properties on another level. You can see it – in fact you *have* to see it– not as a bunch of little calculations, but as a system of tendencies and desires and beliefs and so on. When things get complicated enough, you're forced to change your level of description. To some extend that' already happening, which is why we use words such as "want," "think," "try," and "hope," to describe chess programs and other attempts at mechanical thought.

The intentional stance is essentially nothing more than an abstraction tool. It is a convenient shorthand for talking about certain complex systems (such as people), which allows us to succinctly predict and explain their behaviour without having to understand or make claims about their internal structure or operation. Note that the intentional stance has been widely discussed in the literature –let us just remark here that Sandy of the Coffeeshop Conversation claims that the really interesting things in AI will only begin to happen, 'when the program *itself* adopts the intentional stance towards itself'– and it is not our intention to add to this debate; see [77] for a discussion and references.

If we accept the usefulness of the intentional stance for characterising the properties of rational agents, then the next step in developing a formal theory of such agents is to identify the components of an agent's state. There are many possible mental states that we might choose to characterise an agent: beliefs, goals, desires, intentions, commitments, fears, hopes, and obligations are just a few. We can identify several important categories of such attitudes, for example:

**Information attitudes:** those attitudes an agent has towards information about its environment. The most obvious members of this category are knowledge and belief.

**Pro attitudes:** those attitudes an agent has that tend to lead it to perform actions. The most obvious members of this category are goals, desires, and intentions.

**Normative attitudes:** including obligations, permissions and authorization.

Much of the literature on developing formal theories of agency has been taken up with the relative merits of choosing one attitude over another, and investigating the possible relationships between these attitudes. While there is no consensus on which attitudes should be chosen as primitive, most formalisms choose knowledge or belief together with at least goals or desires.

Figure 1 gives a summary of the main areas that have contributed to a logical theory of rational agency.

## *1.2 A Logical Toolkit*

In attempting to axiomatise the properties of a rational agent in terms of (say) its beliefs and desires, we will find ourselves attempting to formalise statements such as the following

$$\text{Wiebe believes Ajax are great.} \tag{1.1}$$

$$\text{Wiebe desires that Ajax will win.} \tag{1.2}$$

This suggests that a logical characterisation of these statements must include constructions of the form

$$i \left\{ \begin{array}{l} \text{believes} \\ \text{desires} \end{array} \right\} \varphi$$

where $i$ is a term denoting an agent, and $\varphi$ is a sentence. We immediately encounter difficulties if we attempt to represent such statements in first-order logic. First of all, the constructs mentioned above should definitely not be *extensional*–even if "Agent-research in Utrecht" and "Agent-research in Liverpool" may accidentially both be true, one can believe one without the other, desire the second but not the first, even try to achieve one while hindering the other. Apart from this, representing such statements in first-order logic — as binary predicates of the form $Bel(i, \varphi)$ and $Desire(i, \varphi)$ — will not work, because the second term is a sentence, and not a term. By fixing the domain of the first-order language to be itself a language, we can get around this problem, thereby obtaining a first-order meta-language. The meta-language approach has been successfully adopted by a number of researchers, for example [75]. However,

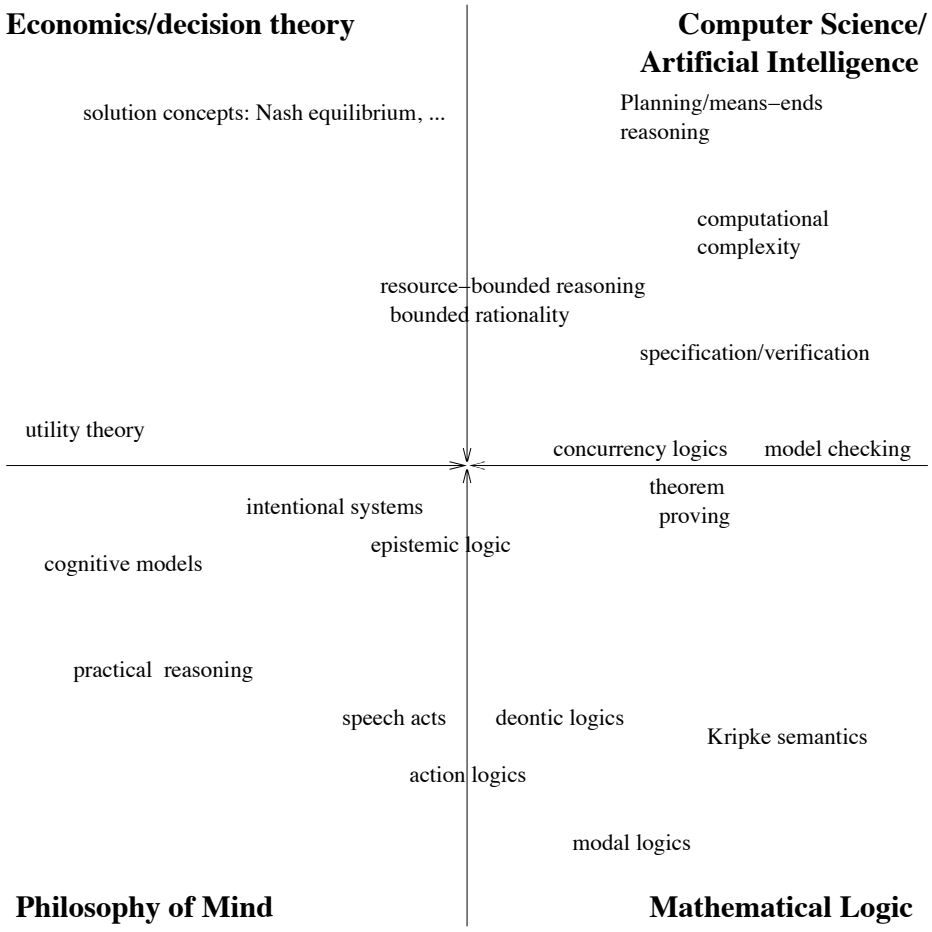


FIG. 1: The dimensions of rational agency: some of the fields that have contributed to the study of rational agents.

meta-language approaches have also been widely criticised (see, e.g., [42] for a detailed critique). Instead of choosing a meta-language approach, most researchers opt for a *modal* approach, thereby folowing Moore's early work on action and knowledge ([51]). In such a modal approach, an agent's beliefs, desires, and the like are represented by an indexed collection of modal operators. The semantics of these operators are generally given in terms or Kripke structures, in the by-now familiar way [11, 59]. The use of Kripke structures and their associated mathematics of correspondence theory makes it possible to quickly generate a number of soundness results for axiomatizations of these logics. However, the *combination* of many modalities into a single framework presents a significant challenge from a logical point of view. Completeness results for logics that incorporate multiple modalities into a single framework are comparatively few and far between, and this area of research is much at the leading edge of contemporary modal logic research [48]. Moreover, for the few combinations

for which completeness results *are* availaible, we also know that satisfiability problem for many such enriched systems can easily become computationally hard [26].

Despite these problems, modal approaches dominate in the literature, and in this article, we focus exclusively on such approaches.

In addition to representing an agent's attitudes, logics of rational agency also typically incorporate some way of representing the *actions* that agents perform, and the effects of these actions. Most researchers adapt techniques from *dynamic* logic in order to represent actions and their effects [29], whereas others confine themselves to a *temporal* set-up. Although there is some work in establishing the exact relation between the two approaches, this issue still deserves a better investigation.

## 2 The State of the Art

In this section, we review three of the best-known formalisms for reasoning about rational agents: Cohen and Levesque's seminal intention logic [14], Rao and Georgeff's BDI framework [63], and the KARO framework of Linder et al [47]. In addition, we give a brief survey of work on what we loosely characterise as "game logics"; the unifying theme of these approaches is that they exploit the close relationship between games (in the sense of game theory) and Kripke-like relational models for modal logics to provide a principled foundation for reasoning about rational agents [6, 4, 19, 55, 2].

### *2.1 Cohen and Levesque's Intention Logic*

One of the best known, and most sophisticated attempts to show how the various components of an agent's cognitive makeup could be combined to form a logic of rational agency is due to Cohen and Levesque [14]. Cohen and Levesque's formalism was originally used to develop a theory of intention (as in "I intended to. . . "), which the authors required as a pre-requisite for a theory of speech acts (see next chapter for a summary, and [15] for full details). However, the logic has subsequently proved to be so useful for specifying and reasoning about the properties of agents that it has been used in an analysis of conflict and cooperation in multi-agent dialogue [24], [23], as well as several studies in the theoretical foundations of cooperative problem solving [45, 38, 39]. This section will focus on the use of the logic in developing a theory of intention. The first step is to lay out the criteria that a theory of intention must satisfy.

When building intelligent agents — particularly agents that must interact with humans — it is important that a *rational balance* is achieved between the beliefs, goals, and intentions of the agents.

> For example, the following are desirable properties of intention: An autonomous agent should act on its intentions, not in spite of them; adopt intentions it believes are feasible and forego those believed to be infeasible; keep (or commit to) intentions, but not forever; discharge those intentions believed to have been satisfied; alter intentions when relevant beliefs change; and adopt subsidiary intentions during plan formation. [14, p214]

Following Bratman [7, 8], Cohen and Levesque identify seven specific properties that must be satisfied by a reasonable theory of intention:

| Operator | Meaning |
|---|---|
| (Bel $i$ $\varphi$) | agent $i$ believes $\varphi$ |
| (Goal $i$ $\varphi$) | agent $i$ has goal of $\varphi$ |
| (Happens $\alpha$) | action $\alpha$ will happen next |
| (Done $\alpha$) | action $\alpha$ has just happened |

TABLE 1. Atomic modalities in Cohen and Levesque's logic

1. Intentions pose problems for agents, who need to determine ways of achieving them.
2. Intentions provide a "filter" for adopting other intentions, which must not conflict.
3. Agents track the success of their intentions, and are inclined to try again if their attempts fail.
4. Agents believe their intentions are possible.
5. Agents do not believe they will not bring about their intentions.
6. Under certain circumstances, agents believe they will bring about their intentions.
7. Agents need not intend all the expected side effects of their intentions.

Given these criteria, Cohen and Levesque adopt a two tiered approach to the problem of formalizing a theory of intention. First, they construct the logic of rational agency, "being careful to sort out the relationships among the basic modal operators" [14, p221]. On top of this framework, they introduce a number of derived constructs, which constitute a "partial theory of rational action" [14, p221]; intention is one of these constructs.

Syntactically, the logic of rational agency is a many-sorted, first-order, multi-modal logic with equality, containing four primary modalities; see Table 1. The semantics of Bel and Goal are given via possible worlds, in the usual way: each agent is assigned a belief accessibility relation, and a goal accessibility relation. The belief accessibility relation is euclidean, transitive, and serial, giving a belief logic of KD45. The goal relation is serial, giving a conative logic KD. It is assumed that each agent's goal relation is a subset of its belief relation, implying that an agent will not have a goal of something it believes will not happen. Worlds in the formalism are a discrete sequence of events, stretching infinitely into past and future. The system is only defined semantically, and Cohen and Levesque derive a number of properties from that. In the semantics, a number of assumptions are implicit, and one might vary on them. For instance, there is a fixed domain assumption, giving us properties as $\forall x(\mathsf{Bel}\ i\ \varphi(x)) \rightarrow (\mathsf{Bel}\ i \forall x \varphi(x))$. Also, agents 'know what thime it is', we immediately obtain from the semantics the validity of formulas like $2:30\mathsf{PM}/3/6/85 \rightarrow \mathsf{Bel}\ i\ 2:30\mathsf{PM}/3/6/85$.

The two basic temporal operators, Happens and Done, are augmented by some operators for describing the structure of event sequences, in the style of dynamic logic [28]. The two most important of these constructors are ";" and "?":

| | |
|---|---|
| $\alpha; \alpha'$ | denotes $\alpha$ followed by $\alpha'$ |
| $\varphi?$ | denotes a "test action" $\varphi$ |

Here, the test must be interpreted as a test by the system; it is not a so-called 'knowledge-producing action' that can be used by the agent to acquire knowledge.

The standard future time operators of temporal logic, "$\Box$" (always), and "$\Diamond$" (sometime) can be defined as abbreviations, along with a "strict" sometime operator, Later:

$$\begin{array}{rcl} \Diamond\alpha & \hat{=} & \exists x \cdot (\mathsf{Happens}\ x; \alpha?) \\ \Box\alpha & \hat{=} & \neg\Diamond\neg\alpha \\ (\mathsf{Later}\ p) & \hat{=} & \neg p \wedge \Diamond p \end{array}$$

A temporal precedence operator, (Before $p$ $q$) can also be derived, and holds if $p$ holds before $q$. An important assumption is that *all* goals are eventually dropped:

$$\Diamond\neg(\mathsf{Goal}\ x\ (\mathsf{Later}\ p))$$

The first major derived construct is a *persistent* goal.

$$\begin{array}{rcll} (\mathsf{P\text{-}Goal}\ i\ p) & \hat{=} & (\mathsf{Goal}\ i\ (\mathsf{Later}\ p)) & \wedge \\ & & (\mathsf{Bel}\ i\ \neg p) & \wedge \\ & & \left[\begin{array}{l} \mathsf{Before} \\ \quad ((\mathsf{Bel}\ i\ p) \vee (\mathsf{Bel}\ i\ \Box\neg p)) \\ \quad \neg(\mathsf{Goal}\ i\ (\mathsf{Later}\ p)) \end{array}\right] & \end{array}$$

So, an agent has a persistent goal of $p$ if:

1. It has a goal that $p$ eventually becomes true, and believes that $p$ is not currently true.
2. Before it drops the goal, one of the following conditions must hold:
   (a) the agent believes the goal has been satisfied;
   (b) the agent believes the goal will never be satisfied.

It is a small step from persistent goals to a first definition of intention, as in "intending to act". Note that "intending that something becomes true" is similar, but requires a slightly different definition; see [14]. An agent $i$ intends to perform action $\alpha$ if it has a persistent goal to have brought about a state where it had just believed it was about to perform $\alpha$, and then did $\alpha$.

$$\begin{array}{rcl} (\mathsf{Intend}\ i\ \alpha) & \hat{=} & (\mathsf{P\text{-}Goal}\ i \\ & & \quad [\mathsf{Done}\ i\ (\mathsf{Bel}\ i\ (\mathsf{Happens}\ \alpha))?; \alpha] \\ & & ) \end{array}$$

Cohen and Levesque go on to show how such a definition meets many of Bratman's criteria for a theory of intention (outlined above). In particular, by basing the definition of intention on the notion of a *persistent goal*, Cohen and Levesque are able to avoid overcommitment or undercommitment. An agent will only drop an intention if it believes that the intention has either been achieved, or is unachievable.

A critique of Cohen and Levesque's theory of intention is presented in [72]; space restrictions prevent a discussion here.

## *2.2 Rao and Georgeff's* BDI *Logics*

One of the best-known (and most widely misunderstood) approaches to reasoning about rational agents is the *belief-desire-intention* (BDI) model [9]. The BDI model

gets its name from the fact that it recognizes the primacy of beliefs, desires, and intentions in rational action. The BDI model is particularly interesting because it combines three distinct components:

- A *philosophical foundation*.
  The BDI model is based on a widely respected theory of rational action in humans, developed by the philosopher Michael Bratman [7].
- A *software architecture*.
  The BDI model of agency does not prescribe a specific implementation. The model may be realized in many different ways, and indeed a number of different implementations of it have been developed. However, the fact that the BDI model *has* been implemented successfully is a significant point in its favor. Moreover, the BDI model has been used to build a number of significant real-world applications, including such demanding problems as fault diagnosis on the space shuttle.
- A *logical formalisation*.
  The third component of the BDI model is a family of logics. These logics capture the key aspects of the BDI model as a set of logical axioms. There are many candidates for a formal theory of rational agency, but BDI logics in various forms have proved to be among the most useful, longest-lived, and widely accepted.

Intuitively, an agent's *beliefs* correspond to information the agent has about the world. These beliefs may be incomplete or incorrect. An agent's *desires* represent states of affairs that the agent would, in an ideal world, wish to be brought about. (Implemented BDI agents require that desires be *consistent* with one another, although *human* desires often fail in this respect.) Finally, an agent's *intentions* represent desires that it has *committed* to achieving. The intuition is that an agent will not, in general, be able to achieve *all* its desires, even if these desires *are* consistent. Ultimately, an agent must therefore fix upon some subset of its desires and commit resources to achieving them. These chosen desires, to which the agent has some commitment, are intentions [14]. The BDI theory of human rational action was originally developed by Michael Bratman [7]. It is a theory of *practical reasoning* — the process of reasoning that we all go through in our everyday lives, deciding moment by moment which action to perform next. Bratman's theory focuses in particular on the role that *intentions* play in practical reasoning. Bratman argues that intentions are important because they constrain the reasoning an agent is required to do in order to select an action to perform. For example, suppose I have an intention to write a book. Then while deciding what to do, I need not expend any effort considering actions that are incompatible with this intention (such as having a summer holiday, or enjoying a social life). This reduction in the number of possibilities I have to consider makes my decision making considerably simpler than would otherwise be the case. Since any real agent we might care to consider — and in particular, any agent that we can implement on a computer — must have resource bounds, an intention-based model of agency, which constrains decision-making in the manner described, seems attractive.

The BDI model has been implemented several times. Originally, it was realized in IRMA, the Intelligent Resource-bounded Machine Architecture [9]. IRMA was intended as a more or less direct realization of Bratman's theory of practical reasoning. However, the best-known implementation is the Procedural Reasoning System (PRS) [25] and its many descendants [21, 60, 18, 37]. In the PRS, an agent has data structures

that explicitly correspond to beliefs, desires, and intentions. A PRS agent's beliefs are directly represented in the form of PROLOG-like facts [13, p.3]. Desires and intentions in PRS are realized through the use of a *plan library*.[1] A plan library, as its name suggests, is a collection of plans. Each plan is a recipe that can be used by the agent to achieve some particular state of affairs. A plan in the PRS is characterized by a *body* and an *invocation condition*. The body of a plan is a course of action that can be used by the agent to achieve some particular state of affairs. The invocation condition of a plan defines the circumstances under which the agent should "consider" the plan. Control in the PRS proceeds by the agent continually updating its internal beliefs, and then looking to see which plans have invocation conditions that correspond to these beliefs. The set of plans made active in this way correspond to the *desires* of the agent. Each desire defines a possible course of action that the agent may follow. On each control cycle, the PRS picks one of these desires, and pushes it onto an execution stack, for subsequent execution. The execution stack contains desires that have been chosen by the agent, and thus corresponds to the agent's *intentions*.

The third and final aspect of the BDI model is the logical component, which gives us a family of tools that allow us to reason about BDI agents. There have been several versions of BDI logic, starting in 1991 and culminating in Rao and Georgeff's 1998 paper on systems of BDI logics [64, 68, 65, 66, 67, 61, 63]; a book-length survey was published as [77]. We focus on [77].

Syntactically, BDI logics are essentially branching time logics (CTL or CTL*, depending on which version you are reading about), enhanced with additional modal operators Bel, Des, and Intend, for capturing the beliefs, desires, and intentions of agents respectively. The BDI modalities are indexed with agents, so for example the following is a legitimate formula of BDI logic.

$$(\mathsf{Bel}\ i\ (\mathsf{Intend}\ j\ \mathsf{A}\Diamond p)) \Rightarrow (\mathsf{Bel}\ i\ (\mathsf{Des}\ j\ \mathsf{A}\Diamond p))$$

This formula says that if $i$ believes that $j$ intends that $p$ is inevitably true eventually, then $i$ believes that $j$ desires $p$ is inevitable. Although they share much in common with Cohen-Levesque's intention logics, the first and most obvious distinction between BDI logics and the Cohen-Levesque approach is the explicit starting point of CTL-like branching time logics. However, the differences are actually much more fundamental than this. The semantics that Rao and Georgeff give to BDI modalities in their logics are based on the conventional apparatus of Kripke structures and possible worlds. However, rather than assuming that worlds are instantaneous states of the world, or even that they are linear sequences of states, it is assumed instead that worlds are themselves branching temporal structures: thus each world can be viewed as a Kripke structure for a CTL-like logic. While this tends to rather complicate the semantic machinery of the logic, it makes it possible to define an interesting array of semantic properties, as we shall see below.

Before proceeding, we summarise the key semantic structures in the logic. Instantaneous states of the world are modelled by *time points*, given by a set $T$; the set of all possible evolutions of the system being modelled is given by a binary relation $R \subseteq T \times T$. A *world* (over $T$ and $R$) is then a pair $\langle T', R' \rangle$, where $T' \subseteq T$ is a non-empty set of time points, and $R' \subseteq R$ is a branching time structure on $T'$. Let

[1] In this description of the PRS, I have modified the original terminology somewhat, to be more in line with contemporary usage; I have also simplified the control cycle of the PRS slightly.

$W$ be the set of all worlds over $T$. A pair $\langle w, t\rangle$, where $w \in W$ and $t \in T_w$, is known as a *situation*. If $w \in W$, then the set of all situations in $w$ is denoted by $S_w$. We have belief accessibility relations $\mathcal{B}$, $\mathcal{D}$, and $\mathcal{I}$, modelled as functions that assign to every agent a relation over situations. Thus, for example:

$$\mathcal{B} : \mathit{Agents} \rightarrow \wp(W \times T \times W)$$

We write $\mathcal{B}_t^w(i)$ to denote the set of worlds accessible to agent $i$ from situation $\langle w, t\rangle$: $B_t^w(i) = \{w' \mid \langle w, t, w'\rangle \in \mathcal{B}(i)\}$. We define $\mathcal{D}_t^w$ and $\mathcal{I}_t^w$ in the obvious way. The semantics of belief, desire and intention modalities are then given in the conventional manner:

- $\langle w, t\rangle \models (\mathsf{Bel}\ i\ \varphi)$ iff $\langle w', t\rangle \models \varphi$ for all $w' \in B_t^w(i)$.
- $\langle w, t\rangle \models (\mathsf{Des}\ i\ \varphi)$ iff $\langle w', t\rangle \models \varphi$ for all $w' \in D_t^w(i)$.
- $\langle w, t\rangle \models (\mathsf{Intend}\ i\ \varphi)$ iff $\langle w', t\rangle \models \varphi$ for all $w' \in I_t^w(i)$.

The primary focus of Rao and Georgeff's early work was to explore the possible interrelationships between beliefs, desires, and intentions from the perspective of semantic characterisation. In order to do this, they defined a number of possible interrelationships between an agent's belief, desire, and intention accessibility relations. The most obvious relationships that can exist are whether one relation is a subset of another: for example, if $D_t^w(i) \subseteq I_t^w(i)$ for all $i, w, t$, then we would have as an interaction axiom $(\mathsf{Intend}\ i\ \varphi) \Rightarrow (\mathsf{Des}\ i\ \varphi)$. However, the fact that worlds themselves have structure in BDI logic also allows us to combine such properties with relations on the *structure* of worlds themselves. The most obvious structural relationship that can exist between two worlds — and the most important for our purposes — is that of one world being a *subworld* of another. Intuitively, a world $w$ is said to be a subworld of world $w'$ if $w$ has the same structure as $w'$ but has fewer paths *and is otherwise identical*. Formally, if $w, w'$ are worlds, then $w$ is a subworld of $w'$ (written $w \sqsubseteq w'$) iff $\mathit{paths}(w) \subseteq \mathit{paths}(w')$ but $w, w'$ agree on the interpretation of predicates and constants in common time points.

The first property we consider is the *structural subset* relationship between accessibility relations. We say that accessibility relation $R$ is a structural subset of accessibility relation $\bar{R}$ if for every $R$-accessible world $w$, there is an $\bar{R}$-accessible world $w'$ such that $w$ is a subworld of $w'$. Formally, if $R$ and $\bar{R}$ are two accessibility relations then we write $R \subseteq_{sub} \bar{R}$ to indicate that if $w' \in R_t^w([\![i]\!])$, then there exists some $w'' \in \bar{R}_t^w([\![i]\!])$ such that $w' \sqsubseteq w''$. If $R \subseteq_{sub} \bar{R}$, then we say $R$ is a *structural subset* of $\bar{R}$.

We write $\bar{R} \subseteq_{sup} R$ to indicate that if $w' \in R_t^w([\![i]\!])$, then there exists some $w'' \in \bar{R}_t^w([\![i]\!])$ such that $w'' \sqsubseteq w'$. If $R \subseteq_{sup} \bar{R}$, then we say $R$ is a *structural superset* of $\bar{R}$. In other words, if $R$ is a structural superset of $\bar{R}$, then for every $R$-accessible world $w$, there is an $\bar{R}$-accessible world $w'$ such that $w'$ is a subworld of $w$.

Finally, we can also consider whether the *intersection* of accessibility relations is empty or not. For example, if $\mathcal{B}_t^w(i) \cap \mathcal{I}_t^w(i) \neq \emptyset$, for all $i, w, t$, then we get the following interaction axiom.

$$(\mathsf{Intend}\ i\ \varphi) \Rightarrow \neg(\mathsf{Bel}\ i\ \neg\varphi)$$

This axiom expresses an *inter-modal consistency* property. Just as we can undertake a more fine-grained analysis of the basic interactions among beliefs, desires, and

| Name | Semantic Condition | Corresponding Formula Schema |
|---|---|---|
| BDI-S1 | $\mathcal{B} \subseteq_{sup} \mathcal{D} \subseteq_{sup} \mathcal{I}$ | $(\mathsf{Intend}\ i\ \mathsf{E}(\varphi)) \Rightarrow (\mathsf{Des}\ i\ \mathsf{E}(\varphi)) \Rightarrow (\mathsf{Bel}\ i\ \mathsf{E}(\varphi))$ |
| BDI-S2 | $\mathcal{B} \subseteq_{sub} \mathcal{D} \subseteq_{sub} \mathcal{I}$ | $(\mathsf{Intend}\ i\ \mathsf{A}(\varphi)) \Rightarrow (\mathsf{Des}\ i\ \mathsf{A}(\varphi)) \Rightarrow (\mathsf{Bel}\ i\ \mathsf{A}(\varphi))$ |
| BDI-S3 | $\mathcal{B} \subseteq \mathcal{D} \subseteq \mathcal{I}$ | $(\mathsf{Intend}\ i\ \varphi) \Rightarrow (\mathsf{Des}\ i\ \varphi) \Rightarrow (\mathsf{Bel}\ i\ \varphi)$ |
| BDI-R1 | $\mathcal{I} \subseteq_{sup} \mathcal{D} \subseteq_{sup} \mathcal{B}$ | $(\mathsf{Bel}\ i\ \mathsf{E}(\varphi)) \Rightarrow (\mathsf{Des}\ i\ \mathsf{E}(\varphi)) \Rightarrow (\mathsf{Intend}\ i\ \mathsf{E}(\varphi))$ |
| BDI-R2 | $\mathcal{I} \subseteq_{sub} \mathcal{D} \subseteq_{sub} \mathcal{B}$ | $(\mathsf{Bel}\ i\ \mathsf{A}(\varphi)) \Rightarrow (\mathsf{Des}\ i\ \mathsf{A}(\varphi)) \Rightarrow (\mathsf{Intend}\ i\ \mathsf{A}(\varphi))$ |
| BDI-R3 | $\mathcal{I} \subseteq \mathcal{D} \subseteq \mathcal{B}$ | $(\mathsf{Bel}\ i\ \varphi) \Rightarrow (\mathsf{Des}\ i\ \varphi) \Rightarrow (\mathsf{Intend}\ i\ \varphi)$ |
| BDI-W1 | $\mathcal{B} \cap_{sup} \mathcal{D} \neq \emptyset$<br>$\mathcal{D} \cap_{sup} \mathcal{I} \neq \emptyset$<br>$\mathcal{B} \cap_{sup} \mathcal{I} \neq \emptyset$ | $(\mathsf{Bel}\ i\ \mathsf{A}(\varphi)) \Rightarrow \neg(\mathsf{Des}\ i\ \neg\mathsf{A}(\varphi))$<br>$(\mathsf{Des}\ i\ \mathsf{A}(\varphi)) \Rightarrow \neg(\mathsf{Intend}\ i\ \neg\mathsf{A}(\varphi))$<br>$(\mathsf{Bel}\ i\ \mathsf{A}(\varphi)) \Rightarrow \neg(\mathsf{Intend}\ i\ \neg\mathsf{A}(\varphi))$ |
| BDI-W2 | $\mathcal{B} \cap_{sub} \mathcal{D} \neq \emptyset$<br>$\mathcal{D} \cap_{sub} \mathcal{I} \neq \emptyset$<br>$\mathcal{B} \cap_{sub} \mathcal{I} \neq \emptyset$ | $(\mathsf{Bel}\ i\ \mathsf{E}(\varphi)) \Rightarrow \neg(\mathsf{Des}\ i\ \neg\mathsf{E}(\varphi))$<br>$(\mathsf{Des}\ i\ \mathsf{E}(\varphi)) \Rightarrow \neg(\mathsf{Intend}\ i\ \neg\mathsf{E}(\varphi))$<br>$(\mathsf{Bel}\ i\ \mathsf{E}(\varphi)) \Rightarrow \neg(\mathsf{Intend}\ i\ \neg\mathsf{E}(\varphi))$ |
| BDI-W3 | $\mathcal{B} \cap \mathcal{D} \neq \emptyset$<br>$\mathcal{D} \cap \mathcal{I} \neq \emptyset$<br>$\mathcal{B} \cap \mathcal{I} \neq \emptyset$ | $(\mathsf{Bel}\ i\ \varphi) \Rightarrow \neg(\mathsf{Des}\ i\ \neg\varphi)$<br>$(\mathsf{Des}\ i\ \varphi) \Rightarrow \neg(\mathsf{Intend}\ i\ \neg\varphi)$<br>$(\mathsf{Bel}\ i\ \varphi) \Rightarrow \neg(\mathsf{Intend}\ i\ \neg\varphi)$ |

TABLE 2. Systems of BDI logic. (Source: [63, p.321].)

intentions by considering the structure of worlds, so we are also able to undertake a more fine-grained characterization of inter-modal consistency properties by taking into account the structure of worlds. We write $R_t^w([\![i]\!]) \cap_{sup} \bar{R}_t^w([\![i]\!])$ to denote the set of worlds $w' \in \bar{R}_t^w([\![i]\!])$ for which there exists some world $w'' \in R_t^w([\![i]\!])$ such that $w' \sqsubseteq w''$. We can then define $\cap_{sub}$ in the obvious way.

Putting all these relations together, we can define a range of BDI logical systems. The most obvious possible systems, and the semantic properties that they correspond to, are summarised in Table 2.

## *2.3 The* KARO *Framework*

The KARO-framework (for Knowledge, Actions, Results and Opportunities) is an attempt to develop and formalise the ideas of Moore [51], who realized that dynamic and epistemic logic can be perfectly combined into one modal framework. The basic framework comes with a sound and complete axiomatization [47]. Also, results on automatic verification of the theory are known, both using translations to first order logic, as well as in a clausal resolution approach. The core of KARO is a combination of epistemic (the standard knowledge operator $\mathbf{K}_i$ is an **S5**-operator) and dynamic logic; many extensions have also been studied.

Along with the notion of the result of events, the notions of ability and opportunity are among the most discussed and investigated in analytical philosophy. Ability plays an important part in various philosophical theories, as for instance the theory of free will and determinism, the theory of refraining and seeing-to-it, and deontic theories. Following Kenny [41], the authors behind KARO consider ability to be the complex

of physical, mental and moral capacities, internal to an agent, and being a positive explanatory factor in accounting for the agent's performing an action. Opportunity on the other hand is best described as circumstantial possibility, i.e., possibility by virtue of the circumstances. The opportunity to perform some action is external to the agent and is often no more than the absence of circumstances that would prevent or interfere with the performance. Although essentially different, abilities and opportunities are interconnected in that abilities can be exercised only when opportunities for their exercise present themselves, and opportunities can be taken only by those who have the appropriate abilities. From this point of view it is important to remark that abilities are understood to be *reliable* (cf. [10]), i.e. having the ability to perform a certain action suffices to take the opportunity to perform the action every time it presents itself. The combination of ability and opportunity determines whether or not an agent has the (practical) possibility to perform an action.

Let $i$ be a variable over a set of agents $\{1, \ldots, n\}$. Actions in the set Ac are either atomic actions ($\mathrm{Ac} = \{a, b, \ldots\}$) or composed ($\alpha, \beta, \ldots$) by means of confirmation of formulas ($\texttt{confirm}\ \varphi$), sequencing ($\alpha; \beta$), conditioning ($\texttt{if}\ \varphi\ \texttt{then}\ \alpha\ \texttt{else}\ \beta$) and repetition ($\texttt{while}\ \varphi\ \texttt{do}\ \alpha$). These actions $\alpha$ can then be used to build new formulas to express the possible *result* of the execution of $\alpha$ by agent $i$ (the formula $[do_i(\alpha)]\varphi$ denotes that $\varphi$ is a result of $i$'s execution of $\alpha$), the *opportunity* for $i$ to perform $\alpha$ ($\langle do_i(\alpha)\rangle\top$) and $i$'s *capability* of performing the action $\alpha$ ($\mathbf{A}_i\alpha$). The formula $\langle do_i(\alpha)\rangle\varphi$ is shorthand for $\neg[do_i(\alpha)]\neg\varphi$, thus expressing that one possible result of performance of $\alpha$ by $i$ implies $\varphi$.

With these tools at hand, one has already a rich framework to reason about agent's knowledge about doing actions. For instance, an infamous properties like *perfect recall*

$$\mathbf{K}_i[do_i(\alpha)]\varphi \rightarrow [do_i(\alpha)]\mathbf{K}_i\varphi$$

can now be enforced for *particular actions* $\alpha$. Also, the core KARO already guarantees a number of properties, of which we list a few:

1. $\mathbf{A}_i\texttt{confirm}\,\varphi \leftrightarrow \varphi$
2. $\mathbf{A}_i\alpha_1; \alpha_2 \leftrightarrow \mathbf{A}_i\alpha_1 \wedge [\mathrm{do}_i(\alpha_1)]\mathbf{A}_i\alpha_2$ or $\mathbf{A}_i\alpha_1; \alpha_2 \leftrightarrow \mathbf{A}_i\alpha_1 \wedge \langle \mathrm{do}_i(\alpha_1)\rangle\mathbf{A}_i\alpha_2$
3. $\mathbf{A}_i\texttt{if}\,\varphi\,\texttt{then}\,\alpha_1\,\texttt{else}\,\alpha_2\,\texttt{fi} \leftrightarrow ((\varphi \wedge \mathbf{A}_i\alpha_1) \vee (\neg\varphi \wedge \mathbf{A}_i\alpha_2))$
4. $\mathbf{A}_i\texttt{while}\,\varphi\,\texttt{do}\,\alpha\,\texttt{od} \leftrightarrow (\neg\varphi \vee (\varphi \wedge \mathbf{A}_i\alpha \wedge [\mathrm{do}_i(\alpha)]\mathbf{A}_i\texttt{while}\,\varphi\,\texttt{do}\,\alpha\,\texttt{od}))$
   or $\mathbf{A}_i\texttt{while}\,\varphi\,\texttt{do}\,\alpha\,\texttt{od} \leftrightarrow (\neg\varphi \vee (\varphi \wedge \mathbf{A}_i\alpha \wedge \langle \mathrm{do}_i(\alpha)\rangle\mathbf{A}_i\texttt{while}\,\varphi\,\texttt{do}\,\alpha\,\texttt{od}))$

For a discussion about the problems with the ability to do a sequential action (the possible behaviour of the items 2 and 4 above), we refer to [47], or to a general solution to this problem that was offered in [35].

*Practical possibility* is considered to consist of two parts, viz. correctness and feasibility: action $\alpha$ is *correct* with respect to $\varphi$ iff $\langle \mathrm{do}_i(\alpha)\rangle\varphi$ holds and $\alpha$ is *feasible* iff $\mathbf{A}_i\alpha$ holds.

$$\mathbf{PracPoss}_i(\alpha, \varphi) \stackrel{\triangle}{=} \langle \mathrm{do}_i(\alpha)\rangle\varphi \wedge \mathbf{A}_i\alpha$$

The importance of practical possibility manifests itself particularly when ascribing — from the outside — certain qualities to an agent. It seems that for the agent itself practical possibilities are relevant in so far as the agent has knowledge of these

possibilities. To formalise this kind of knowledge, KARO comes with a Can-predicate and a Cannot-predicate. The first of these predicates concerns the knowledge of agents about their practical possibilities, the latter predicate does the same for their practical impossibilities.

$$\mathbf{Can}_i(\alpha,\varphi) \stackrel{\triangle}{=} \mathbf{K}_i\mathbf{PracPoss}_i(\alpha,\varphi) \text{ and } \mathbf{Cannot}_i(\alpha,\varphi) \stackrel{\triangle}{=} \mathbf{K}_i\neg\mathbf{PracPoss}_i(\alpha,\varphi)$$

The Can-predicate and the Cannot-predicate integrate knowledge, ability, opportunity and result, and seem to formalise one of the most important notions of agency. In fact it is probably not too bold to say that knowledge like that formalised through the Can-predicate, although perhaps in a weaker form by taking aspects of uncertainty into account, underlies all acts performed by rational agents. For rational agents act only if they have some information on both the possibility to perform the act, and its possible outcome. It therefore seems worthwhile to take a closer look at both the Can-predicate and the Cannot-predicate. The following propperties focus on the behaviour of the *means*-part of the predicates, which is the $\alpha$ in $\mathbf{Can}_i(\alpha,\varphi)$ and $\mathbf{Cannot}_i(\alpha,\varphi)$.

1. $\mathbf{Can}_i(\texttt{confirm}\,\varphi,\psi) \leftrightarrow \mathbf{K}_i(\varphi\wedge\psi)$
2. $\mathbf{Cannot}_i(\texttt{confirm}\,\varphi,\psi) \leftrightarrow \mathbf{K}_i(\neg\varphi\vee\neg\psi)$
3. $\mathbf{Can}_i(\alpha_1;\alpha_2,\varphi) \leftrightarrow \mathbf{Can}_i(\alpha_1,\mathbf{PracPoss}_i(\alpha_2,\varphi))$
4. $\mathbf{Can}_i(\alpha_1;\alpha_2,\varphi) \rightarrow \langle \mathrm{do}_i(\alpha_1)\rangle\mathbf{Can}_i(\alpha_2,\varphi)$ if $i$ has perfect recall regarding $\alpha_1$
5. $\mathbf{Can}_i(\texttt{if}\,\varphi\,\texttt{then}\,\alpha_1\,\texttt{else}\,\alpha_2\,\texttt{fi},\psi)\wedge\mathbf{K}_i\varphi \leftrightarrow \mathbf{Can}_i(\alpha_1,\psi)\wedge\mathbf{K}_i\varphi$
6. $\mathbf{Can}_i(\texttt{if}\,\varphi\,\texttt{then}\,\alpha_1\,\texttt{else}\,\alpha_2\,\texttt{fi},\psi)\wedge\mathbf{K}_i\neg\varphi \leftrightarrow \mathbf{Can}_i(\alpha_2,\psi)\wedge\mathbf{K}_i\neg\varphi$
7. $\mathbf{Can}_i(\texttt{while}\,\varphi\,\texttt{do}\,\alpha\,\texttt{od},\psi)\wedge\mathbf{K}_i\varphi \leftrightarrow \mathbf{Can}_i(\alpha,\mathbf{PracPoss}_i(\texttt{while}\,\varphi\,\texttt{do}\,\alpha\,\texttt{od},\psi))\wedge\mathbf{K}_i\varphi$

In *Actions that make you change your mind* ([46]), the authors of KARO look at specific atomic actions Atthat the agents can perform, i.e., doxastic actions of expanding, contracting or revising its beliefs (we have now both knowledge ($\mathbf{K}_i$) and belief ($\mathbf{B}_i$). Those actions are assumed to have the following general properties:

- $\models \langle \mathrm{do}_i(\alpha)\rangle\top$ *realizability*
- $\models \langle \mathrm{do}_i(\alpha)\rangle\chi \rightarrow [\mathrm{do}_i(\alpha)]\chi$ *determinism*
- $\models \langle \mathrm{do}_i(\alpha;\alpha)\rangle\chi \leftrightarrow \langle \mathrm{do}_i(\alpha)\rangle\chi$ *idempotence*

Realizability of an action implies that agents have the opportunity to perform the action regardless of circumstances; determinism of an action means that performing the action results in a unique state of affairs, and idempotence of an action implies that performing the action an arbitrary number of times has the same effect as performing the action just once.

Then, specific definitions for the three actions are given, and related to the AGM framework of belief revision ([1]). As an illustration, we list some properties, written in one object language, of the action of revising one's beliefs (here, $\varphi$ is an objective formula):

- $[\mathrm{do}_i(\texttt{revise}\ \varphi)]\mathbf{B}_i\varphi$
- $[\mathrm{do}_i(\texttt{revise}\ \varphi)]\mathbf{B}_i\vartheta \rightarrow [\mathrm{do}_i(\texttt{expand}\ \varphi)]\mathbf{B}_i\vartheta$
- $\neg\mathbf{B}_i\neg\varphi \rightarrow ([\mathrm{do}_i(\texttt{expand}\ \varphi)]\mathbf{B}_i\vartheta \leftrightarrow [\mathrm{do}_i(\texttt{revise}\ \varphi)]\mathbf{B}_i\vartheta)$

- $\mathbf{K}_i\neg\varphi \leftrightarrow [\mathrm{do}_i(\texttt{revise}\ \varphi)]\mathbf{B}_i\bot$
- $\mathbf{K}_i(\varphi \leftrightarrow \psi) \rightarrow ([\mathrm{do}_i(\texttt{revise}\ \varphi)]\mathbf{B}_i\vartheta \leftrightarrow [\mathrm{do}_i(\texttt{revise}\ \psi)]\mathbf{B}_i\vartheta)$

In [50], the KARO-authors show how motivational attitudes can be incorporated in their framework. The most primitive notion here is that agent $i$ *wishes* $\varphi$ ($\mathbf{W}_i\varphi$), from which it has to *select* some (if so, $\mathbf{C}_i\varphi$ becomes true). In order to define what a goal is, a higher order notion of *implementability* is first defined:

$$\Diamond_i\varphi \Leftrightarrow \exists k \in IN \exists a_1, \ldots, a_k \in \mathrm{At}\mathbf{PracPoss}_i(a_1; \ldots; a_k, \varphi))$$

Now the notion of a goal in KARO is as follows:

$$\mathbf{Goal}_i\varphi \triangleq \mathbf{W}_i\varphi \wedge \neg\varphi \wedge \Diamond_i\varphi \wedge \mathbf{C}_i\varphi$$

It is easily seen that this definition of a goal does not suffer from effects as being closed under consequence. In [50], these motivational attitudes are also 'dynamized', in the sense that actions, like committing and decommitting are added, with which an agent can change its motivational attitudes. Semantically, this is supported by letting the agents maintain an "agenda". Space does not permit us to investigate this issue further.

## *2.4 Game Logics*

The final class of work we review involves a formulation of rational agents in logic by making use of the techniques of game theory [5, 53]. This class of work is arguably less well established than the alternative tradition as exemplified by our previous reviews, and for this reason we will only give some pointers into the literature. It is hard to precisely identify the origins of this class of work, but many researchers have noted the link between Kripke structures and the extensive form of games; a good example is Parikh's seminal work on the dynamic logic of games [54]; another example is the branching time logic of protocols developed by Ladner et al [44]. More recently, there has been renewed interest in this area. Bonnano has investigated the relationship between various branching time logic and game models [6], while van Benthem has investigated the link between Kripke strcutures, modal logic, and extensive form games (for example by using bisimulation notions to define a notion of equivalence over games) [4], Ditmarsch's work on the use of epistemic dynamic logic to analyse board games [19] and work on modal characterisation of gamne logic concepts such as Nash equilibrium [30, 2].

As an example of the kind of work being done in this area, we will focus on the *coalition logic* of Pauly [55, 57, 56]. Put crudely, coalition logic is a modal logic of *cooperative ability*. Coalition logic thus extends propositional logic with a collection of unary modal connectives of the form $[C]$, where $C$ is an expression denoting a set of agents. A formula $[C]\varphi$ is intended to mean that the coalition $C$ can cooperate to ensure that $\varphi$ is true. The semantics of coalition modalities are given in terms of an *effectivity function*, which defines for every coalition $C$ the sets of states for which $C$ is effective. This leads to the following notion of ability:

- $s \models [C]\varphi$ iff $\exists S \in E(C)$ such that $s' \models \varphi$ for all $s'$ such that $s' \in S$.

The semantics of coalition modalities thus corresponds to our intuitions of coalitions being able to "force" states of affairs. Pauly investigates various classes of coalition logic, corresponding to a variety of constraints on effectivity functions, and gives complete axiomatizations of a range of coalition logics. In addition, shows that the computational complexity of the satisfiability problem for coalition logic varies from NP-complete to PSPACE-complete (depending upon what assumptions are made with respect to the semantics); he also shows that the model checking problem for coalition logics can be solved in deterministic polynomial time.

One of the fascinating aspects of coalition logic is its use in social choice theory, and in particular in the specification, development, and verification of *voting procedures*. Consider the following scenario, adapted from [55].

> Two individuals, $A$ and $B$, are to choose between two outcomes, $p$ and $q$. We want a procedure that will allow them to choose that will satisfy the following requirements. First, we definitely want an outcome to be possible — that is, we want the two agents to bring about either $p$ or $q$. We do not want them to be able to bring about both outcomes simultaneously. Similarly, we do not want either agent to dominate: we want them both to have equal power.

The first point to note is that we can naturally axiomatize these requirements using coalition logic:

$$\begin{array}{ll} [A,B]x & x \in \{p,q\} \\ \neg[A,B](p \wedge q) & \\ \neg[x]p & x \in \{A,B\} \\ \neg[x]q & x \in \{A,B\} \end{array}$$

It should be immediately obvious how these axioms capture the requirements as stated above. Now, given a particular voting procedure, a model checking algorithm can be used to check whether or not this procedure implements the specification correctly. Moreover, a constructive proof of satisfiability for these axioms might be used to *synthesise* a procedure; or else announce that no implementation exists.

## 2.5 Discussion

Undoubtedly, formalizing the informational and motivational attitudes in a contex with evolving time or where agents can do actions, have greatly helped to improve our understanding of complex systems. At the same time, admittedly, there are many weaknesses and open problems with such approaches.

To give one example of how a formalisation can help us to become more clear about the interrelationship between the notions defined here, recall that Rao and Georgeff assume the notion of *belief-goal compatibility*, saying

$$\mathbf{Goal}_i\varphi \rightarrow \mathbf{B}_i\varphi$$

for formulas $\varphi$ that refer to the future.

Cohen and Levesque, however, put a lot of emphasis on their notion of *realizability*, stating exactly the opposite:

$$\mathbf{B}_i\varphi \rightarrow \mathbf{Goal}_i\varphi$$

By analyzing the framework of Cohen and Levesque more closely, it appears that they have a much weaker property in mind, which is

$$\mathbf{Goal}_i\varphi \rightarrow \neg\mathbf{B}_i\neg\varphi$$

To mention just one aspect in which the approach mentioned here are still far from completed, we reall that the three frameworks allow one to reason about many agents, but are in essence still one-agent systems. Where notions as distributed and common knowledge are well understood epistemic notions in multi-agent systems, their motivational analogues seem to be much harder and are yet only partially understood (see Cohen and Levesque's [16] or Tambe's [74] on teamwork). Moreover, there is much research going on in the area that incoroporates epistemic actions and epistemic group notions ([2]) where our property of perfect recall is even more subtle:

$$\mathbf{K}_i[do_i(\alpha)]\varphi \rightarrow [\mathbf{K}_i do_i(\alpha)]\mathbf{K}_i\varphi$$

expressing that, if the agent knows that performing some (learning) action leads it to know $\varphi$, then, if the agent knows that $\alpha$ is performed, it indeed gets to know $\varphi$.

## 3 Agent Theory and Agent Practice

Much of the interest in logical formalisations of rational agency arises because authors are interested in building *artificial agents*. This then begs the question of what these various logics might buy us in the development of such agents. Broadly speaking, logic has played a role in three aspects of software development.

- as a *specification language*;
- as a *programming language*; and
- as a *verification language*.

In the sections that follow, we will discuss the possible use of logics of rational agency in these three processes.

### *3.1 Specification*

The software development process begins by establishing the client's requirements. When this process is complete, a *specification* is developed, which sets out the functionality of the new system. Temporal and dynamic logics have found wide applicability in the specification of systems. An obvious question is therefore whether logics of rational agency might be used as specification languages.

A specification expressed such a logic would be a formula $\varphi$. The idea is that such a specification would express the desirable behavior of a system. To see how this might work, consider the following formula of BDI logic (in fact from [77]), intended to form part of a specification of a process control system.

$$(\mathsf{Bel}\ i\ \mathit{Open}(\mathit{valve}32)) \Rightarrow (\mathsf{Intend}\ i\ (\mathsf{Bel}\ j\ \mathit{Open}(\mathit{valve}32)))$$

This formula says that if $i$ believes valve 32 is open, then $i$ should intend that $j$ believes valve 32 is open. A rational agent $i$ with such an intention can select a speech act

to perform in order to inform $j$ of this state of affairs. It should be intuitively clear how a system specification might be constructed using such formulae, to define the intended behavior of a system.

One of the main desirable features of a software specification language is that it should not dictate *how* a specification should be satisfied by an implementation. It should be clear that the specification above has exactly these properties. It does not dictate how agent $i$ should go about making $j$ aware that valve 32 is open. We simply expect $i$ to behave as a rational agent given such an intention.

There are a number of problems with the use of such logics for specification. The most worrying of these is with respect to their semantics. The semantics for the modal connectives (for beliefs, desires, and intentions) are given in the normal modal logic tradition of possible worlds [11]. So, for example, an agent's beliefs in some state are characterized by a set of different states, each of which represents one possibility for how the world could actually be, given the information available to the agent. In much the same way, an agent's desires in some state are characterized by a set of states that are consistent with the agent's desires. Intentions are represented similarly. There are several advantages to the possible worlds model: it is well studied and well understood, and the associated mathematics of correspondence theory is extremely elegant. These attractive features make possible worlds the semantics of choice for almost every researcher in formal agent theory. However, there are also a number of serious drawbacks to possible worlds semantics. First, possible worlds semantics imply that agents are logically perfect reasoners, (in that their deductive capabilities are sound and complete), and they have infinite resources available for reasoning. No real agent, artificial or otherwise, has these properties.

Second, possible worlds semantics are generally *ungrounded*. That is, there is usually no precise relationship between the abstract accessibility relations that are used to characterize an agent's state, and any concrete computational model. As we shall see in later sections, this makes it difficult to go from a formal specification of a system in terms of beliefs, desires, and so on, to a concrete computational system. Similarly, given a concrete computational system, there is generally no way to determine what the beliefs, desires, and intentions of that system are. If temporal modal logics of rational agency are to be taken seriously as *specification* languages, then this is a significant problem.

## *3.2 Implementation*

Once given a specification, we must implement a system that is correct with respect to this specification. The next issue we consider is this move from abstract specification to concrete computational system. There are at least two possibilities for achieving this transformation that we consider here:

1. somehow directly execute or animate the abstract specification; or
2. somehow translate or compile the specification into a concrete computational form using an automatic translation technique.

In the sub-sections that follow, we shall investigate each of these possibilities in turn.

## Directly Executing Agent Specifications

Suppose we are given a system specification, $\varphi$, which is expressed in some logical language $L$. One way of obtaining a concrete system from $\varphi$ is to treat it as an *executable specification*, and *interpret* the specification directly in order to generate the agent's behaviour. Interpreting an agent specification can be viewed as a kind of constructive proof of satisfiability, whereby we show that the specification $\varphi$ is satisfiable by *building a model* (in the logical sense) for it. If models for the specification language $L$ can be given a computational interpretation, then model building can be viewed as executing the specification. To make this discussion concrete, consider the Concurrent METATEM programming language [22]. In this language, agents are programmed by giving them a temporal logic specification of the behaviour it is intended they should exhibit; this specification is directly executed to generate each agent's behaviour. Models for the temporal logic in which Concurrent METATEM agents are specified are linear discrete sequences of states: executing a Concurrent METATEM agent specification is thus a process of constructing such a sequence of states. Since such state sequences can be viewed as the histories traced out by programs as they execute, the temporal logic upon which Concurrent METATEM is based has a computational interpretation; the actual execution algorithm is described in [3].

Note that executing Concurrent METATEM agent specifications is possible primarily because the models upon which the Concurrent METATEM temporal logic is based are comparatively simple, with an obvious and intuitive computational interpretation. However, agent specification languages in general (e.g., the BDI formalisms of Rao and Georgeff [62]) are based on considerably more complex logics. In particular, they are usually based on a semantic framework known as *possible worlds* [11]. The technical details are somewhat involved for the purposes of this article: the main point is that, *in general*, possible worlds semantics do not have a computational interpretation in the way that Concurrent METATEM semantics do. Hence it is not clear what "executing" a logic based on such semantics might mean.

In response to this, a number of researchers have attempted to develop executable agent specification languages with a simplified semantic basis, that has a computational interpretation. An example is Rao's AgentSpeak(L) language, which although essentially a BDI system, has a simple computational semantics [60]. The 3APL project ([32]) is also an attempt to have a agent programmng language with a well-defined semantics, based on transition systems. One advantage of having a thorough semantics is that it enables one to compare different agent programming languages, such as AgentSpeak(L) with 3APL ([31]) or AGENT-0 and 3APL ([33]). One complication in bridging the gap between the agent programming paradigm and the formal systems of Section 2, is that the former usually take goals to be procedural (a plan), whereas goals in the latter are declarative (a desired state). A programming language that tries to bridge the gap in this respect is the language GOAL ([43]).

## Compiling Agent Specifications

An alternative to direct execution is *compilation*. In this scheme, we take our abstract specification, and transform it into a concrete computational model via some automatic synthesis process. The main perceived advantages of compilation over direct execution are in run-time efficiency. Direct execution of an agent specification, as in

Concurrent METATEM, above, typically involves manipulating a symbolic representation of the specification at run time. This manipulation generally corresponds to reasoning of some form, which is computationally costly. Compilation approaches aim to reduce abstract symbolic specifications to a much simpler computational model, which requires no symbolic representation. The 'reasoning' work is thus done off-line, at compile-time; execution of the compiled system can then be done with little or no run-time symbolic reasoning.

Compilation approaches usually depend upon the close relationship between models for temporal/modal logic (which are typically labeled graphs of some kind), and automata-like finite state machines. For example, Pnueli and Rosner [58] synthesise reactive systems from branching temporal logic specifications. Similar techniques have also been used to develop concurrent system skeletons from temporal logic specifications. Perhaps the best-known example of this approach to agent development is the *situated automata* paradigm of Rosenschein and Kaelbling [70]. They use an epistemic logic (i.e., a logic of *knowledge* [20]) to specify the perception component of intelligent agent systems. They then used an technique based on constructive proof to directly synthesise automata from these specifications [69].

The general approach of automatic synthesis, although theoretically appealing, is limited in a number of important respects. First, as the agent specification language becomes more expressive, then even offline reasoning becomes too expensive to carry out. Second, the systems generated in this way are not capable of *learning*, (i.e., they are not capable of adapting their "program" at run-time). Finally, as with direct execution approaches, agent specification frameworks tend to have no concrete computational interpretation, making such a synthesis impossible.

## *3.3 Verification*

Once we have developed a concrete system, we need to show that this system is correct with respect to our original specification. This process is known as *verification*, and it is particularly important if we have introduced any informality into the development process. We can divide approaches to the verification of systems into two broad classes: (1) *axiomatic*; and (2) *semantic* (model checking). In the subsections that follow, we shall look at the way in which these two approaches have evidenced themselves in agent-based systems.

### Axiomatic Approaches

Axiomatic approaches to program verification were the first to enter the mainstream of computer science, with the work of Hoare in the late 1960s [34]. Axiomatic verification requires that we can take our concrete program, and from this program systematically derive a logical theory that represents the behaviour of the program. Call this the program theory. If the program theory is expressed in the same logical language as the original specification, then verification reduces to a proof problem: show that the specification is a theorem of (equivalently, is a logical consequence of) the program theory. The development of a program theory is made feasible by *axiomatizing* the programming language in which the system is implemented. For example, Hoare logic gives us more or less an axiom for every statement type in a simple PASCAL-like

language. Once given the axiomatization, the program theory can be derived from the program text in a systematic way.

Perhaps the most relevant work from mainstream computer science is the specification and verification of reactive systems using temporal logic, in the way pioneered by Pnueli, Manna, and colleagues [49]. The idea is that the computations of reactive systems are infinite sequences, which correspond to models for linear temporal logic. Temporal logic can be used both to develop a system specification, and to axiomatize a programming language. This axiomatization can then be used to systematically derive the theory of a program from the program text. Both the specification and the program theory will then be encoded in temporal logic, and verification hence becomes a proof problem in temporal logic.

Comparatively little work has been carried out within the agent-based systems community on axiomatizing multi-agent environments. We shall review just one approach. In [76], an axiomatic approach to the verification of multi-agent systems was proposed. Essentially, the idea was to use a temporal belief logic to axiomatize the properties of two multi-agent programming languages. Given such an axiomatization, a program theory representing the properties of the system could be systematically derived in the way indicated above. A temporal belief logic was used for two reasons. First, a temporal component was required because, as we observed above, we need to capture the ongoing behaviour of a multi-agent system. A belief component was used because the agents we wish to verify are each symbolic AI systems in their own right. That is, each agent is a symbolic reasoning system, which includes a representation of its environment and desired behaviour. A belief component in the logic allows us to capture the symbolic representations present within each agent. The two multi-agent programming languages that were axiomatized in the temporal belief logic were Shoham's AGENT0 [71], and Fisher's Concurrent METATEM (see above). Note that this approach relies on the operation of agents being sufficiently simple that their properties can be axiomatized in the logic. It works for Shoham's AGENT0 and Fisher's Concurrent METATEM largely because these languages have a simple semantics, closely related to rule-based systems, which in turn have a simple logical semantics. For more complex agents, an axiomatization is not so straightforward. Also, capturing the semantics of concurrent execution of agents is not easy (it is, of course, an area of ongoing research in computer science generally).

## Semantic Approaches: Model Checking

Ultimately, axiomatic verification reduces to a proof problem. Axiomatic approaches to verification are thus inherently limited by the difficulty of this proof problem. Proofs are hard enough, even in classical logic; the addition of temporal and modal connectives to a logic makes the problem considerably harder. For this reason, more efficient approaches to verification have been sought. One particularly successful approach is that of *model checking* [12]. As the name suggests, whereas axiomatic approaches generally rely on syntactic proof, model checking approaches are based on the semantics of the specification language.

The model checking problem, in abstract, is quite simple: given a formula $\varphi$ of language $L$, and a model $M$ for $L$, determine whether or not $\varphi$ is valid in $M$, i.e., whether or not $M \models_L \varphi$. Model checking-based verification has been studied in connection

with temporal logic. The technique once again relies upon the close relationship between models for temporal logic and finite-state machines. Suppose that $\varphi$ is the specification for some system, and $\pi$ is a program that claims to implement $\varphi$. Then, to determine whether or not $\pi$ truly implements $\varphi$, we take $\pi$, and from it generate a model $M_\pi$ that corresponds to $\pi$, in the sense that $M_\pi$ encodes all the possible computations of $\pi$; determine whether or not $M_\pi \models \varphi$, i.e., whether the specification formula $\varphi$ is valid in $M_\pi$; the program $\pi$ satisfies the specification $\varphi$ just in case the answer is 'yes'. The main advantage of model checking over axiomatic verification is in complexity: model checking using the branching time temporal logic CTL ([12]) can be done in polynomial time, whereas the proof problem for most modal logics is quite complex.

In [67], Rao and Georgeff present an algorithm for model checking agent systems. More precisely, they give an algorithm for taking a logical model for their (propositional) BDI agent specification language, and a formula of the language, and determining whether the formula is valid in the model. The technique is closely based on model checking algorithms for normal modal logics [27]. They show that despite the inclusion of three extra modalities, (for beliefs, desires, and intentions), into the CTL branching time framework, the algorithm is still quite efficient, running in polynomial time. So the second step of the two-stage model checking process described above can still be done efficiently. However, it is not clear how the first step might be realised for BDI logics. Where does the logical model characterizing an agent actually comes from — can it be derived from an arbitrary program $\pi$, as in mainstream computer science? To do this, we would need to take a program implemented in, say, PASCAL, and from it derive the belief, desire, and intention accessibility relations that are used to give a semantics to the BDI component of the logic. Because, as we noted earlier, there is no clear relationship between the BDI logic and the concrete computational models used to implement agents, it is not clear how such a model could be derived.

## 4 Conclusions

In this paper, we have motivated and introduced a number of logics of rational agency; moreover, we have investigated the role(s) that such logics might play in the *development* of artificial agents. We hope to have demonstrated that logics for rational agents are a fascinating area of study, at the confluence of many different research areas, including logic, artificial intelligence, economics, game theory, and the philosophy of mind. We also hope to have illustrated some of the popular approaches to the theory of rational agency.

There are far too many research challenges open to identify in this article. Instead, we simply note that the search for a logic of rational agency poses a range of deep technical, philosophical, and computational research questions for the logic community. We believe that all the disparate research communities with an interest in rational agency can benefit from this search.

## References

[1] C.E. Alchourrón, P. Gärdenfors, and D. Makinson. On the logic of theory change: partial meet contraction and revision functions. *Journal of Symbolic Logic*, 50:510–530, 1985.

[2] A. Baltag. A logic for suspicious players. *Bulletin of Economic Research*, 54(1):1–46, 2002.

[3] H. Barringer, M. Fisher, D. Gabbay, G. Gough, and R. Owens. METATEM: A framework for programming in temporal logic. In *REX Workshop on Stepwise Refinement of Distributed Systems: Models, Formalisms, Correctness (LNCS Volume 430)*, pages 94–129. Springer-Verlag: Berlin, Germany, June 1989.

[4] J. van Benthem. Extensive games as process models. *Journal of Logic, Language, and Information*, 11(3):289–313, 2002.

[5] K. Binmore. *Fun and Games: A Text on Game Theory*. D. C. Heath and Company: Lexington, MA, 1992.

[6] G. Bonanno. Branching time logic, perfect information games and backward induction. *Games and Economic Behavior*, 36(1):57–73, July 2001.

[7] M. E. Bratman. *Intention, Plans, and Practical Reason*. Harvard University Press: Cambridge, MA, 1987.

[8] M. E. Bratman. What is intention? In P. R. Cohen, J. L. Morgan, and M. E. Pollack, editors, *Intentions in Communication*, pages 15–32. The MIT Press: Cambridge, MA, 1990.

[9] M. E. Bratman, D. J. Israel, and M. E. Pollack. Plans and resource-bounded practical reasoning. *Computational Intelligence*, 4:349–355, 1988.

[10] M. A. Brown. On the logic of ability. *Journal of Philosophical Logic*, 17:1–26, 1988.

[11] B. Chellas. *Modal Logic: An Introduction*. Cambridge University Press: Cambridge, England, 1980.

[12] E. M. Clarke, O. Grumberg, and D. A. Peled. *Model Checking*. The MIT Press: Cambridge, MA, 2000.

[13] W. F. Clocksin and C. S. Mellish. *Programming in Prolog*. Springer-Verlag: Berlin, Germany, 1981.

[14] P. R. Cohen and H. J. Levesque. Intention is choice with commitment. *Artificial Intelligence*, 42:213–261, 1990.

[15] P. R. Cohen and H. J. Levesque. Rational interaction as the basis for communication. In P. R. Cohen, J. Morgan, and M. E. Pollack, editors, *Intentions in Communication*, pages 221–256. The MIT Press: Cambridge, MA, 1990.

[16] P. R. Cohen and H. J. Levesque. Teamwork. *Nous*, 25(4):487–512, 1991.

[17] D. C. Dennett. *The Intentional Stance*. The MIT Press: Cambridge, MA, 1987.

[18] M. d'Inverno, D. Kinny, M. Luck, and M. Wooldridge. A formal specification of dMARS. In M. P. Singh, A. Rao, and M. J. Wooldridge, editors, *Intelligent Agents IV (LNAI Volume 1365)*, pages 155–176. Springer-Verlag: Berlin, Germany, 1997.

[19] H. P. van Ditmarsch. *Knowledge Games*. PhD thesis, University of Groningen, Groningen, 2000.

[20] R. Fagin, J. Y. Halpern, Y. Moses, and M. Y. Vardi. *Reasoning About Knowledge*. The MIT Press: Cambridge, MA, 1995.

[21] K. Fischer, J. P. Müller, and M. Pischel. A pragmatic BDI architecture. In M. Wooldridge, J. P. Müller, and M. Tambe, editors, *Intelligent Agents II (LNAI Volume 1037)*, pages 203–218. Springer-Verlag: Berlin, Germany, 1996.

[22] M. Fisher. A survey of Concurrent METATEM — the language and its applications. In D. M. Gabbay and H. J. Ohlbach, editors, *Temporal Logic — Proceedings of the First International Conference (LNAI Volume 827)*, pages 480–505. Springer-Verlag: Berlin, Germany, July 1994.

[23] J. R. Galliers. A strategic framework for multi-agent cooperative dialogue. In *Proceedings of the Eighth European Conference on Artificial Intelligence (ECAI-88)*, pages 415–420, Munich, Federal Republic of Germany, 1988.

[24] J. R. Galliers. *A Theoretical Framework for Computer Models of Cooperative Dialogue, Acknowledging Multi-Agent Conflict*. PhD thesis, Open University, UK, 1988.

[25] M. P. Georgeff and A. L. Lansky. Reactive reasoning and planning. In *Proceedings of the Sixth National Conference on Artificial Intelligence (AAAI-87)*, pages 677–682, Seattle, WA, 1987.

[26] J. Y. Halpern and M. Y. Vardi. The complexity of reasoning about knowledge and time. I. Lower bounds. *Journal of Computer and System Sciences*, 38:195–237, 1989.

[27] J. Y. Halpern and M. Y. Vardi. Model checking versus theorem proving: A manifesto. In V. Lifschitz, editor, *AI and Mathematical Theory of Computation — Papers in Honor of John McCarthy*, pages 151–176. The Academic Press: London, England, 1991.

[28] D. Harel. *First-Order Dynamic Logic (LNCS Volume 68)*. Springer-Verlag: Berlin, Germany, 1979.

[29] D. Harel. Dynamic logic. In D. Gabbay and F. Guenther, editors, *Handbook of Philosophical Logic Volume II — Extensions of Classical Logic*, pages 497–604. D. Reidel Publishing Company: Dordrecht, The Netherlands, 1984. (Synthese library Volume 164).

[30] P. Harrenstein, W. van der Hoek, J.-J Meyer, and C. Witteveen. On modal logic interpretations for games. In *Proceedings of the Fifteenth European Conference on Artificial Intelligence (ECAI-2002)*, pages 28–32, Lyon, France, 2002.

[31] K. Hindriks, F.S. de Boer, W van der Hoek, and J.-J. Ch. Meyer. A formal embedding of agentspeak(l) in 3apl. In G. Antoniou and J. Slaney, editors, *Advanced Topics in Artificial Intelligence*, pages 155–166. Springer, LNAI 1502, 1998.

[32] K. V. Hindriks, F. S. de Boer, W. van der Hoek, and J.-J. Ch. Meyer. Agent programming in 3APL. *Autonomous Agents and Multi-Agent Systems*, 2(4):357–402, 1999.

[33] K.V. Hindriks, F.S. de Boer, W. van der Hoek, and J.-J. Ch. Meyer. A formal semantics for the core of AGENT-0. In E. Postma and M. Gyssens, editors, *Proceedings of Eleventh Belgium-Netherlands Conference on Artificial Intelligence*, pages 27–34, 1999.

[34] C. A. R. Hoare. An axiomatic basis for computer programming. *Communications of the ACM*, 12(10):576–583, 1969.

[35] W. van der Hoek, J.-J.Ch Meyer, and J.W. van Schagen. Formalizing potential of agents: The karo framework revisited. In Martina Falle, Stefan Kaufmann, and Marc Pauly, editors, *Formalizing the Dynamics of Information*, volume 91 of *CSLI Lecture Notes*, pages 51–67. CSLI Publications, Stanford, 2000.

[36] D.R. Hofstadter. Metamagical themas: A coffeehouse conversation on the turing test to determine if a machine can think. *Scientific American*, pages 15–36, 1981.

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

[38] N. R. Jennings. On being responsible. In E. Werner and Y. Demazeau, editors, *Decentralized AI 3 — Proceedings of the Third European Workshop on Modelling Autonomous Agents in a Multi-Agent World (MAAMAW-91)*, pages 93–102. Elsevier Science Publishers B.V.: Amsterdam, The Netherlands, 1992.

[39] N. R. Jennings. Towards a cooperation knowledge level for collaborative problem solving. In *Proceedings of the Tenth European Conference on Artificial Intelligence (ECAI-92)*, pages 224–228, Vienna, Austria, 1992.

[40] N. R. Jennings and M. Wooldridge, editors. *Agent Technology: Foundations, Applications and Markets*. Springer-Verlag: Berlin, Germany, 1998.

[41] A. Kenny. *Will, Freedom and Power*. Basil Blackwell, Oxford, 1975.

[42] K. Konolige. *A Deduction Model of Belief*. Pitman Publishing: London and Morgan Kaufmann: San Mateo, CA, 1986.

[43] W. van der Hoek K.V. Hindriks, F.S. de Boer and J.-J.Ch. Meyer. Agent programming with declarative goals. In C. Castelfranchi and Y. Lespérance, editors, *Intelligent Agents VII, Proceedings of the 6th workshop on Agent Theories, Architectures, and Languages (ATAL)*, number 1986 in LNAI, pages 228–243, 2001.

[44] R. E. Ladner and J. H. Reif. The logic of distributed protocols: preliminary report. In *Proceedings of the 1986 Conference on Theoretical Aspects of Reasoning About Knowledge*, pages 207–222. Morgan Kaufmann Publishers: San Mateo, CA, 1986.

[45] H. J. Levesque, P. R. Cohen, and J. H. T. Nunes. On acting together. In *Proceedings of the Eighth National Conference on Artificial Intelligence (AAAI-90)*, pages 94–99, Boston, MA, 1990.

[46] B. van Linder, W. van der Hoek, and J.-J. Ch. Meyer. Actions that make you change your mind. In A. Laux and H. Wansing, editors, *Knowledge and Belief in Philosophy and AI*, pages 103–146. Akademie-Verlag, 1995.

[47] B. van Linder, W. van der Hoek, and J.J-Ch. Meyer. Formalizing abilities and opportunities of agents. *Fundameta Informaticae*, 34(1,2):53–101, 1998.

[48] A. Lomuscio. *Knowledge Sharing among Ideal Agents*. PhD thesis, School of Computer Science, University of Birmingham, Birmingham, UK, June 1999.

[49] Z. Manna and A. Pnueli. *Temporal Verification of Reactive Systems — Safety*. Springer-Verlag: Berlin, Germany, 1995.

[50] J.-J. Ch. Meyer, W. van der Hoek, and B. van Linder. A logical approach to the dynamics of commitments. *Artificial Intelligence*, 113:1–40, 1999.

[51] R. C. Moore. Reasoning about knowledge and action. In *Proceedings of the Fifth International Joint Conference on Artificial Intelligence (IJCAI-77)*, Cambridge, MA, 1977.

[52] J. Von Neumann and O. Morgenstern. *Theory of Games and Economic Behaviour*. Princeton University Press: Princeton, NJ, 1944.

[53] M. J. Osborne and A. Rubinstein. *A Course in Game Theory*. The MIT Press: Cambridge, MA, 1994.

[54] R. Parikh. The logic of games and its applications. In M. Karpinksy and J. van Leeuwen, editors, *Topics in the Theory of Computation*. Elsevier Science Publishers B.V.: Amsterdam, The Netherlands, 1985.

[55] M. Pauly. *Logic for Social Software*. PhD thesis, University of Amsterdam, 2001. ILLC Dissertation Series 2001-10.

[56] M. Pauly. A logical framework for coalitional effectivity in dynamic procedures. *Bulletin of Economic Research*, 53(4):305–324, 2002.

[57] M. Pauly. A modal logic for coalitional power in games. *Journal of Logic and Computation*, 12(1):149–166, 2002.

[58] A. Pnueli and R. Rosner. On the synthesis of a reactive module. In *Proceedings of the Sixteenth ACM Symposium on the Principles of Programming Languages (POPL)*, pages 179–190, January 1989.

[59] S Popkorn.

*First Steps in Modal Logic*.

Cambridge University Press: Cambridge, England, 1994.

[60] A. S. Rao. AgentSpeak(L): BDI agents speak out in a logical computable language. In W. Van de Velde and J. W. Perram, editors, *Agents Breaking Away: Proceedings of the Seventh European Workshop on Modelling Autonomous Agents in a Multi-Agent World, (LNAI Volume 1038)*, pages 42–55. Springer-Verlag: Berlin, Germany, 1996.

[61] A. S. Rao. Decision procedures for propositional linear-time Belief-Desire-Intention logics.

In M. Wooldridge, J. P. Müller, and M. Tambe, editors, *Intelligent Agents II (LNAI Volume 1037)*, pages 33–48. Springer-Verlag: Berlin, Germany, 1996.

[62] A. S. Rao and M. Georgeff. BDI Agents: from theory to practice. In *Proceedings of the First International Conference on Multi-Agent Systems (ICMAS-95)*, pages 312–319, San Francisco, CA, June 1995.

[63] A. S. Rao and M. Georgeff. Decision procedures for BDI logics. *Journal of Logic and Computation*, 8(3):293–344, 1998.

[64] A. S. Rao and M. P. Georgeff. Asymmetry thesis and side-effect problems in linear time and branching time intention logics. In *Proceedings of the Twelfth International Joint Conference on Artificial Intelligence (IJCAI-91)*, pages 498–504, Sydney, Australia, 1991.

[65] A. S. Rao and M. P. Georgeff. Modeling rational agents within a BDI-architecture. In R. Fikes and E. Sandewall, editors, *Proceedings of Knowledge Representation and Reasoning (KR&R-91)*, pages 473–484. Morgan Kaufmann Publishers: San Mateo, CA, April 1991.

[66] A. S. Rao and M. P. Georgeff. An abstract architecture for rational agents. In C. Rich, W. Swartout, and B. Nebel, editors, *Proceedings of Knowledge Representation and Reasoning (KR&R-92)*, pages 439–449, 1992.

[67] A. S. Rao and M. P. Georgeff. A model-theoretic approach to the verification of situated reasoning systems. In *Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence (IJCAI-93)*, pages 318–324, Chambéry, France, 1993.

[68] A. S. Rao, M. P. Georgeff, and E. A. Sonenberg. Social plans: A preliminary report. In E. Werner and Y. Demazeau, editors, *Decentralized AI 3 — Proceedings of the Third European Workshop on Modelling Autonomous Agents in a Multi-Agent World (MAAMAW-91)*, pages 57–76. Elsevier Science Publishers B.V.: Amsterdam, The Netherlands, 1992.

[69] S. Rosenschein and L. P. Kaelbling. The synthesis of digital machines with provable epistemic properties. In J. Y. Halpern, editor, *Proceedings of the 1986 Conference on Theoretical Aspects of Reasoning About Knowledge*, pages 83–98. Morgan Kaufmann Publishers: San Mateo, CA, 1986.

[70] S. J. Rosenschein and L. P. Kaelbling. A situated view of representation and control. In P. E. Agre and S. J. Rosenschein, editors, *Computational Theories of Interaction and Agency*, pages 515–540. The MIT Press: Cambridge, MA, 1996.

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

[72] M. P. Singh. A critical examination of the Cohen-Levesque theory of intention. In *Proceedings of the Tenth European Conference on Artificial Intelligence (ECAI-92)*, pages 364–368, Vienna, Austria, 1992.

[73] S. P. Stich. *From Folk Psychology to Cognitive Science*. The MIT Press: Cambridge, MA, 1983.

[74] M. Tambe. Towards flexible teamwork. *Journal of AI Research*, 7:83–124, 1997.

[75] R. Turner. *Truth and Modality for Knowledge Representation*. Pitman Publishing: London, 1990.

[76] M. Wooldridge. *The Logical Modelling of Computational Multi-Agent Systems*. PhD thesis, Department of Computation, UMIST, Manchester, UK, October 1992.

[77] M. Wooldridge. *Reasoning about Rational Agents*. The MIT Press: Cambridge, MA, 2000.

[78] M. Wooldridge and N. R. Jennings. Intelligent agents: Theory and practice. *The Knowledge Engineering Review*, 10(2):115–152, 1995.

[79] M. Wooldridge and A. Rao, editors. *Foundations of Rational Agency*. Kluwer Academic Publishers: Boston, MA, 1999.

[80] E. N. Zalta. Stanford encyclopedia of philosophy. See `http://plato.stanford.edu/`.

Received September 2002