Intelligent Agents: Theory and Practice

Michael Wooldridge
Department of Computing
Manchester Metropolitan University
Chester Street, Manchester M1 5GD
United Kingdom
M.Wooldridge@doc.mmu.ac.uk

Nicholas R. Jennings
Department of Electronic Engineering
Queen Mary & Westfield College
Mile End Road, London E1 4NS
United Kingdom
N.R.Jennings@qmw.ac.uk

Submitted to *Knowledge Engineering Review*, October 1994.
Revised January 1995.

Abstract

The concept of an *agent* has become important in both Artificial Intelligence (AI) and mainstream computer science. Our aim in this paper is to point the reader at what we perceive to be the most important theoretical and practical issues associated with the design and construction of intelligent agents. For convenience, we divide these issues into three areas (though as the reader will see, the divisions are at times somewhat arbitrary). *Agent theory* is concerned with the question of what an agent is, and the use of mathematical formalisms for representing and reasoning about the properties of agents. *Agent architectures* can be thought of as software engineering models of agents; researchers in this area are primarily concerned with the problem of designing software or hardware systems that will satisfy the properties specified by agent theorists. Finally, *agent languages* are software systems for programming and experimenting with agents; these languages may embody principles proposed by theorists. The paper is *not* intended to serve as a tutorial introduction to all the issues mentioned; we hope instead simply to identify the most important issues, and point to work that elaborates on them. The article includes a short review of current and potential applications of agent technology.

1 Introduction

We begin our article with descriptions of three events that occur sometime in the future:

1. The key air-traffic control systems in the country of Ruritania suddenly fail, due to freak weather conditions. Fortunately, computerised air-traffic control systems in neighbouring countries negotiate between themselves to track and deal with all affected flights, and the potentially disastrous situation passes without major incident.

2. Upon logging in to your computer, you are presented with a list of email messages, sorted into order of importance by your personal digital assistant (PDA). You are then presented with a similar list of news articles; the assistant draws your attention to one particular article, which describes hitherto unknown work that is very close to your own. After an electronic discussion with a number of other PDAs, your PDA has already obtained a relevant technical report for you from an FTP site, in the anticipation that it will be of interest.

3. You are editing a file, when your PDA requests your attention: an email message has arrived, that contains notification about a paper you sent to an important conference, and the PDA correctly predicted that you would want to see it as soon as possible. The paper has been accepted, and without prompting, the PDA begins to look into travel arrangements, by consulting a number of databases and other networked information sources. A short time later, you are presented with a summary of the cheapest and most convenient travel options.

We shall not claim that computer systems of the sophistication indicated in these scenarios are just around the corner, but serious academic research *is* underway into similar applications: air-traffic control has long been a research domain in distributed artificial intelligence (DAI) (Steeb et al., 1988); various types of information manager, that filter and obtain information on behalf of their users, have been prototyped (Maes, 1994a); and systems such as those that appear in the third scenario are discussed in (McGregor, 1992; Levy et al., 1994). The key computer-based components that appear in each of the above scenarios are known as *agents*. It is interesting to note that one way of defining AI is by saying that it is the subfield of computer science which aims to construct agents that exhibit aspects of intelligent behaviour. The notion of an 'agent' is thus central to AI. It is perhaps surprising, therefore, that until the mid to late 1980s, researchers from mainstream AI gave relatively little consideration to the issues surrounding agent synthesis. Since then, however, there has been an intense flowering of interest in the subject: agents are now widely discussed by researchers in mainstream computer science, as well as those working in data communications and concurrent systems research, robotics, and user interface design. A British national daily paper recently predicted that:

> 'Agent-based computing (ABC) is likely to be the next significant breakthrough in software development.' (Sargent, 1992)

Moreover, the UK-based consultancy firm Ovum has predicted that the *agent technology* industry would be worth some US$3.5 billion worldwide by the year 2000 (Houlder, 1994). Researchers from both industry and academia are thus taking agent technology seriously: our aim in this paper is to survey what we perceive to be the most important issues in the design and construction of intelligent agents, of the type that might ultimately appear in applications such as those suggested by the fictional scenarios above. We begin our article, in the following sub-section, with a discussion on the subject of exactly what an agent *is*.

1.a What *is* an Agent?

Carl Hewitt recently remarked[1] that the question *what is an agent?* is embarrassing for the agent-based computing community in just the same way that the question *what is intelligence?* is embarrassing for the mainstream AI community. The problem is that although the term is widely used, by many people working in closely related areas, it defies attempts to produce a single universally accepted definition. This need not necessarily be a problem: after all, if many people are successfully developing interesting and useful applications, then it hardly matters that they do not agree on potentially trivial terminological details. However, there is also the danger that unless the issue is discussed, 'agent' might become a 'noise' term, subject to both abuse and misuse, to the potential confusion of the research community. It is for this reason that we briefly consider the question.

We distinguish two general usages of the term 'agent': the first is weak, and relatively uncontentious; the second is stronger, and potentially more contentious.

A Weak Notion of Agency

Perhaps the most general way in which the term agent is used is to denote a hardware or (more usually) software-based computer system that enjoys the following properties:

- *autonomy*: agents operate without the direct intervention of humans or others, and have some kind of control over their actions and internal state (Castelfranchi, 1995);
- *social ability*: agents interact with other agents (and possibly humans) via some kind of *agent-communication language* (Genesereth and Ketchpel, 1994);
- *reactivity*: agents perceive their environment, (which may be the physical world, a user via a graphical user interface, a collection of other agents, the INTERNET, or perhaps all of these combined), and respond in a timely fashion to changes that occur in it;

[1] At the thirteenth international workshop on distributed AI.

- *pro-activeness*: agents do not simply act in response to their environment, they are able to exhibit goal-directed behaviour by *taking the initiative*.

A simple way of conceptualising an agent is thus as a kind of UNIX-like software process, that exhibits the properties listed above. This weak notion of agency has found currency with a surprisingly wide range of researchers. For example, in mainstream computer science, the notion of an agent as a self-contained, concurrently executing software process, that encapsulates some state and is able to communicate with other agents via message passing, is seen as a natural development of the object-based concurrent programming paradigm (Agha, 1986; Agha et al., 1993).

This weak notion of agency is also that used in the emerging discipline of *agent-based software engineering*:

> '[Agents] communicate with their peers by exchanging messages in an expressive *agent communication language*. While agents can be as simple as subroutines, typically they are larger entities with some sort of persistent control.' (Genesereth and Ketchpel, 1994, p48)

A *softbot* (software robot) is a kind of agent:

> 'A *softbot* is an agent that interacts with a software environment by issuing commands and interpreting the environment's feedback. A softbot's effectors are commands (e.g., UNIX shell commands such as `mv` or `compress`) meant to change the external environment's state. A softbot's sensors are commands (e.g., `pwd` or `ls` in UNIX) meant to provide … information.' (Etzioni et al., 1994, p10)

A Stronger Notion of Agency

For some researchers — particularly those working in AI — the term 'agent' has a stronger and more specific meaning than that sketched out above. These researchers generally mean an agent to be a computer system that, in addition to having the properties identified above, is either conceptualised or implemented using concepts that are more usually applied to humans. For example, it is quite common in AI to characterise an agent using *mentalistic* notions, such as knowledge, belief, intention, and obligation (Shoham, 1993). Some AI researchers have gone further, and considered *emotional* agents (Bates et al., 1992a; Bates, 1994). (Lest the reader suppose that this is just pointless anthropomorphism, it should be noted that there are good arguments in favour of designing and building agents in terms of human-like mental states — see section 2.) Another way of giving agents human-like attributes is to represent them visually, perhaps by using a cartoon-like graphical icon or an animated face (Maes, 1994a, p36) — for obvious reasons, such agents are of particular importance to those interested in human-computer interfaces.

Other Attributes of Agency

Various other attributes are sometimes discussed in the context of agency. For example:

- *mobility* is the ability of an agent to move around an electronic network (White, 1994);
- *veracity* is the assumption that an agent will not knowingly communicate false information (Galliers, 1988b, pp159–164);
- *benevolence* is the assumption that agents do not have conflicting goals, and that every agent will therefore always try to do what is asked of it (Rosenschein and Genesereth, 1985, p91); and
- *rationality* is (crudely) the assumption that an agent will act in order to achieve its goals, and will not act in such a way as to prevent its goals being achieved — at least insofar as its beliefs permit (Galliers, 1988b, pp49–54).

(A discussion of some of these notions is given below; various other attributes of agency are formally defined in (Goodwin, 1993).)

1.b The Structure of this Article

Now that we have at least a preliminary understanding of what an agent is, we can embark on a more detailed look at their properties, and how we might go about constructing them. For convenience, we identify three key issues, and structure our survey around these (cf. (Seel, 1989, p1)):

- *Agent theories* are essentially *specifications*. Agent theorists address such questions as: How are we to conceptualise agents? What properties should agents have, and how are we to formally represent and reason about these properties?
- *Agent architectures* represent the move from specification to implementation. Those working in the area of agent architectures address such questions as: How are we to construct computer systems that satisfy the properties specified by agent theorists? What software and/or hardware structures are appropriate? What is an appropriate separation of concerns?
- *Agent languages* are programming languages that may embody the various principles proposed by theorists. Those working in the area of agent languages address such questions as: How are we to program agents? What are the right primitives for this task? How are we to effectively compile or execute agent programs?

As we pointed out above, the distinctions between these three areas are occasionally unclear. The issue of agent theories is discussed in the section 2. In section 3, we discuss architectures, and in section 4, we discuss agent languages. A brief discussion of applications appears in section 5, and some concluding remarks appear in section 6. Each of the three major sections closes with a discussion, in which we give a brief critical review of current work and open problems, and a section pointing the reader to further relevant reading.

Finally, some notes on the scope and aims of the article. First, it is important to realise that we are writing very much from the point of view of AI, and the material we have chosen to review clearly reflects this bias. Secondly, the article is *not* a intended as a review of Distributed AI, although the material we discuss arguably falls under this banner. We have deliberately avoided discussing what might be called the *macro* aspects of agent technology (i.e., those issues relating to the agent *society*, rather than the individual (Gasser, 1991)), as these issues are reviewed more thoroughly elsewhere (see (Bond and Gasser, 1988, pp1–56) and (Chaib-draa et al., 1992)). Thirdly, we wish to reiterate that agent technology is, at the time of writing, one of the most active areas of research in AI and computer science generally. Thus, work on agent theories, architectures, and languages is very much *ongoing*. In particular, many of the fundamental problems associated with agent technology can by no means be regarded as solved. This article therefore represents only a snapshot of past and current work in the field, along with some tentative comments on open problems and suggestions for future work areas. Our hope is that the article will introduce the reader to some of the different ways that agency is treated in (D)AI, and in particular to current thinking on the theory and practice of such agents.

2 Agent Theories

In the preceding section, we gave an informal overview of the notion of agency. In this section, we turn our attention to the *theory* of such agents, and in particular, to *formal* theories. We regard an agent theory as a specification for an agent; agent theorists develop formalisms for representing the properties of agents, and using these formalisms, try to develop theories that capture desirable properties of agents. Our starting point is the notion of an agent as an entity 'which appears to be the subject of beliefs, desires, etc.' (Seel, 1989, p1). The philosopher Dennett has coined the term *intentional system* to denote such systems.

2.a Agents as Intentional Systems

When explaining human activity, it is often useful to make statements such as the following:

> Janine took her umbrella because she *believed* it was going to rain.
> Michael worked hard because he *wanted* to possess a PhD.

These statements make use of a *folk psychology*, by which human behaviour is predicted and explained through the attribution of *attitudes*, such as believing and wanting (as in the above examples), hoping, fearing, and so on. This folk psychology is well established: most people reading the above statements would say they found their meaning entirely clear, and would not give them a second glance.

The attitudes employed in such folk psychological descriptions are called the *intentional* notions. The philosopher Daniel Dennett has coined the term *intentional system* to describe entities 'whose behaviour can be predicted by the method of attributing belief, desires and rational acumen' (Dennett, 1987, p49). Dennett identifies different 'grades' of intentional system:

> 'A *first-order* intentional system has beliefs and desires (etc.) but no beliefs and desires *about* beliefs and desires. ... A *second-order* intentional system is more sophisticated; it has beliefs and desires (and no doubt other intentional states) about beliefs and desires (and other intentional states) — both those of others and its own'. (Dennett, 1987, p243)

One can carry on this hierarchy of intentionality as far as required.

An obvious question is whether it is legitimate or useful to attribute beliefs, desires, and so on, to artificial agents. Isn't this just anthropomorphism? McCarthy, among others, has argued that there are occasions when the *intentional stance* is appropriate:

> 'To ascribe *beliefs*, *free will*, *intentions*, *consciousness*, *abilities*, or *wants* to a machine is <u>legitimate</u> when such an ascription expresses the same information about the machine that it expresses about a person. It is <u>useful</u> when the ascription helps us understand the structure of the machine, its past or future behaviour, or how to repair or improve it. It is perhaps never <u>logically required</u> even for humans, but expressing reasonably briefly what is actually known about the state of the machine in a particular situation may require mental qualities or qualities isomorphic to them. Theories of belief, knowledge and wanting can be constructed for machines in a simpler setting than for humans, and later applied to humans. Ascription of mental qualities is <u>most straightforward</u> for machines of known structure such as thermostats and computer operating systems, but is <u>most useful</u> when applied to entities whose structure is incompletely known'. (McCarthy, 1978), (quoted in (Shoham, 1990))

What objects can be described by the intentional stance? As it turns out, more or less anything can. In his doctoral thesis, Seel showed that even very simple, automata-like objects can be consistently ascribed intentional descriptions (Seel, 1989); similar work by Rosenschein and Kaelbling, (albeit with a different motivation), arrived at a similar conclusion (Rosenschein and Kaelbling, 1986). For example, consider a light switch:

> 'It is perfectly coherent to treat a light switch as a (very cooperative) agent with the capability of transmitting current at will, who invariably transmits current when it believes that we want it transmitted and not otherwise; flicking the switch is simply our way of communicating our desires'. (Shoham, 1990, p6)

And yet most adults would find such a description absurd — perhaps even infantile. Why is this? The answer seems to be that while the intentional stance description is perfectly consistent with the observed behaviour of a light switch, and is internally consistent,

> '… it does not *buy us anything*, since we essentially understand the mechanism sufficiently to have a simpler, mechanistic description of its behaviour'. (Shoham, 1990, p6)

Put crudely, the more we know about a system, the less we need to rely on animistic, intentional explanations of its behaviour. However, with very complex systems, even if a complete, accurate picture of the system's architecture and working *is* available, a mechanistic, *design stance* explanation of its behaviour may not be practicable. Consider a computer. Although we might have a complete technical description of a computer available, it is hardly practicable to appeal to such a description when explaining why a menu appears when we click a mouse on an icon. In such situations, it may be more appropriate to adopt an intentional stance description, if that description is consistent, and simpler than the alternatives. The intentional notions are thus *abstraction tools*, which provide us with a convenient and familiar way of describing, explaining, and predicting the behaviour of complex systems.

Being an intentional system seems to be a *necessary* condition for agenthood, but is it a *sufficient* condition? In his Master's thesis, Shardlow trawled through the literature of cognitive science and its component disciplines in an attempt to find a unifying concept that underlies the notion of agenthood. He was forced to the following conclusion:

> 'Perhaps there is something more to an agent than its capacity for beliefs and desires, but whatever that thing is, it admits no unified account within cognitive science'. (Shardlow, 1990)

So, an agent is a system that is most conveniently described by the intentional stance; one whose simplest consistent description requires the intentional stance. Before proceeding, it is worth considering exactly which attitudes are appropriate for representing agents. For the purposes of this survey, the two most important categories are *information attitudes* and *pro-attitudes*:

$$\text{information attitudes} \begin{cases} \text{belief} \\ \text{knowledge} \end{cases} \qquad \text{pro-attitudes} \begin{cases} \text{desire} \\ \text{intention} \\ \text{obligation} \\ \text{commitment} \\ \text{choice} \\ \ldots \end{cases}$$

Thus information attitudes are related to the information that an agent has about the world it occupies, whereas pro-attitudes are those that in some way guide the agent's actions. Precisely which *combination* of attitudes is most appropriate to characterise an agent is, as we shall see later, an issue of some debate. However, it seems reasonable to suggest that an agent must be represented in terms of at least one information attitude, and at least one pro-attitude. Note that pro- and information attitudes are closely linked, as a rational agent will make choices and form intentions, etc., on the basis of the information it has about the world. Much work in agent theory is concerned with sorting out exactly what the relationship between the different attitudes is.

The next step is to investigate methods for representing and reasoning about intentional notions.

2.b Representing Intentional Notions

Suppose one wishes to reason about intentional notions in a logical framework. Consider the following statement (after (Genesereth and Nilsson, 1987, pp210–211)):

Janine believes Cronos is the father of Zeus. (1)

A naive attempt to translate (1) into first-order logic might result in the following:

$$Bel(Janine, Father(Zeus, Cronos)) \tag{2}$$

Unfortunately, this naive translation does not work, for two reasons. The first is syntactic: the second argument to the *Bel* predicate is a *formula* of first-order logic, and is not, therefore, a term. So (2) is not a well-formed formula of classical first-order logic. The second problem is semantic, and is potentially more serious. The constants *Zeus* and *Jupiter*, by any reasonable interpretation, denote the same individual: the supreme deity of the classical world. It is therefore acceptable to write, in first-order logic:

$$(Zeus = Jupiter). \tag{3}$$

Given (2) and (3), the standard rules of first-order logic would allow the derivation of the following:

$$Bel(Janine, Father(Jupiter, Cronos)) \tag{4}$$

But intuition rejects this derivation as invalid: believing that the father of Zeus is Cronos is *not* the same as believing that the father of Jupiter is Cronos. So what is the problem? Why does first-order logic fail here? The problem is that the intentional notions — such as belief and desire — are *referentially opaque*, in that they set up *opaque contexts*, in which the standard substitution rules of first-order logic do not apply. In classical (propositional or first-order) logic, the denotation, or semantic value, of an expression is dependent solely on the denotations of its sub-expressions. For example, the denotation of the propositional logic formula $p \wedge q$ is a function of the truth-values of p and q. The operators of classical logic are thus said to be *truth functional*. In contrast, intentional notions such as belief are *not* truth functional. It is surely not the case that the truth value of the sentence:

$$\text{Janine believes } p \tag{5}$$

is dependent solely on the truth-value of p[2]. So substituting equivalents into opaque contexts is not going to preserve meaning. This is what is meant by referential opacity. Clearly, classical logics are not suitable in their standard form for reasoning about intentional notions: alternative formalisms are required.

The number of basic techniques used for alternative formalisms is quite small. Recall, from the discussion above, that there are two problems to be addressed in developing a logical formalism for intentional notions: a syntactic one, and a semantic one. It follows that any formalism can be characterized in terms of two independent attributes: its *language of formulation*, and *semantic model* (Konolige, 1986a, p83).

There are two fundamental approaches to the syntactic problem. The first is to use a *modal* language, which contains non-truth-functional *modal operators*, which are applied to formulae. An alternative approach involves the use of a *meta-language*: a many-sorted first-order language containing terms that denote formulae of some other *object-language*. Intentional notions can be represented using a meta-language predicate, and given whatever axiomatization is deemed appropriate. Both of these approaches have their advantages and disadvantages, and will be discussed in the sequel.

As with the syntactic problem, there are two basic approaches to the semantic problem. The first, best-known, and probably most widely used approach is to adopt a *possible worlds* semantics, where an agent's beliefs, knowledge, goals, and so on, are characterized as a set of so-called *possible worlds*, with an *accessibility relation* holding between them. Possible worlds semantics have an associated *correspondence theory* which makes them an attractive mathematical tool to work with (Chellas, 1980). However, they also have many associated difficulties, notably the well-known *logical omniscience* problem, which implies that agents are perfect reasoners (we discuss this problem in more detail below). A number of variations on the possible-worlds theme have been proposed, in an attempt to retain the correspondence theory, but without logical omniscience. The commonest alternative to the possible worlds

[2]Note, however, that the sentence (5) is itself a proposition, in that its denotation is the value true or false.

model for belief is to use a *sentential*, or *interpreted symbolic structures* approach. In this scheme, beliefs are viewed as symbolic formulae explicitly represented in a data structure associated with an agent. An agent then believes φ if φ is present in its belief data structure. Despite its simplicity, the sentential model works well under certain circumstances (Konolige, 1986a).

In the subsections that follow, we discuss various approaches in some more detail. We begin with a close look at the basic possible worlds model for logics of knowledge (*epistemic* logics) and logics of belief (*doxastic* logics).

2.c Possible Worlds Semantics

The possible worlds model for logics of knowledge and belief was originally proposed by Hintikka (Hintikka, 1962), and is now most commonly formulated in a normal modal logic using the techniques developed by Kripke (Kripke, 1963)[3]. Hintikka's insight was to see that an agent's beliefs could be characterized as a set of *possible worlds*, in the following way. Consider an agent playing a card game such as poker[4]. In this game, the more one knows about the cards possessed by one's opponents, the better one is able to play. And yet complete knowledge of an opponent's cards is generally impossible, (if one excludes cheating). The ability to play poker well thus depends, at least in part, on the ability to deduce what cards are held by an opponent, given the limited information available. Now suppose our agent possessed the ace of spades. Assuming the agent's sensory equipment was functioning normally, it would be rational of her to believe that she possessed this card. Now suppose she were to try to deduce what cards were held by her opponents. This could be done by first calculating all the various different ways that the cards in the pack could possibly have been distributed among the various players. (This is not being proposed as an actual card playing strategy, but for illustration!) For argument's sake, suppose that each possible configuration is described on a separate piece of paper. Once the process was complete, our agent can then begin to systematically eliminate from this large pile of paper all those configurations which are *not possible, given what she knows*. For example, any configuration in which she did not possess the ace of spades could be rejected immediately as impossible. Call each piece of paper remaining after this process a *world*. Each world represents one state of affairs considered possible, given what she knows. Hintikka coined the term *epistemic alternatives* to describe the worlds possible given one's beliefs. Something true in *all* our agent's epistemic alternatives could be said to be believed by the agent. For example, it will be true in all our agent's epistemic alternatives that she has the ace of spades.

On a first reading, this seems a peculiarly roundabout way of characterizing belief, but

[3]In Hintikka's original work, he used a technique based on 'model sets', which is equivalent to Kripke's formalism, though less elegant. See (Hughes and Cresswell, 1968, pp351–352) for a comparison and discussion of the two techniques.

[4]This example was adapted from (Halpern, 1987).

it has two advantages. First, it remains neutral on the subject of the cognitive structure of agents. It certainly doesn't posit any internalized collection of possible worlds. It is just a convenient way of characterizing belief. Second, the mathematical theory associated with the formalization of possible worlds is extremely appealing (see below).

The next step is to show how possible worlds may be incorporated into the semantic framework of a logic. Epistemic logics are usually formulated as *normal modal logics* using the semantics developed by Kripke (Kripke, 1963). Before moving on to explicitly epistemic logics, we consider a simple normal modal logic. This logic is essentially classical propositional logic, extended by the addition of two operators: '$\Box$' (necessarily), and '$\Diamond$' (possibly). Let $Prop = \{p, q, \ldots\}$ be a countable set of *atomic propositions*. Then the syntax of the logic is defined by the following rules: (i) if $p \in Prop$ then p is a formula; (ii) if φ, ψ are formulae, then so are $\neg\varphi$ and $\varphi \vee \psi$; and (iii) if φ is a formula then so are $\Box\varphi$ and $\Diamond\varphi$. The operators '$\neg$' (not) and '$\vee$' (or) have their standard meanings. The remaining connectives of classical propositional logic can be defined as abbreviations in the usual way. The formula $\Box\varphi$ is read: 'necessarily φ', and the formula $\Diamond\varphi$ is read: 'possibly φ'. The semantics of the modal connectives are given by introducing an *accessibility relation* into models for the language. This relation defines what worlds are considered accessible from every other world. The formula $\Box\varphi$ is then true if φ is true in every world accessible from the current world; $\Diamond\varphi$ is true if φ is true in at least one world accessible from the current world. The two modal operators are *duals* of each other, in the sense that the universal and existential quantifiers of first-order logic are duals:

$$\Box\varphi \Leftrightarrow \neg\Diamond\neg\varphi \qquad \Diamond\varphi \Leftrightarrow \neg\Box\neg\varphi.$$

It would thus have been possible to take either one as primitive, and introduce the other as a derived operator. The two basic properties of this logic are as follows. First, the following axiom schema is valid: $\Box(\varphi \Rightarrow \psi) \Rightarrow (\Box\varphi \Rightarrow \Box\psi)$. This axiom is called K, in honour of Kripke. The second property is as follows: if φ is valid, then $\Box\varphi$ is valid. Now, since K is valid, it will be a theorem of any complete axiomatization of normal modal logic. Similarly, the second property will appear as a rule of inference in any axiomatization of normal modal logic; it is generally called the *necessitation* rule. These two properties turn out to be the most problematic features of normal modal logics when they are used as logics of knowledge/belief (this point will be examined later).

The most intriguing properties of normal modal logics follow from the properties of the accessibility relation, R, in models. To illustrate these properties, consider the following axiom schema: $\Box\varphi \Rightarrow \varphi$. It turns out that this axiom is *characteristic* of the class of models with a *reflexive* accessibility relation. (By characteristic, we mean that it is true in all and only those models in the class.) There are a host of axioms which correspond to certain properties of R: the study of the way that properties of R correspond to axioms is called *correspondence theory*. For our present purposes, we identify just four axioms: the axiom called T, (which corresponds to a reflexive accessibility relation); D (serial accessibility relation); 4 (transitive accessibility relation); and 5 (euclidean accessibility relation):

$$
\begin{array}{llll}
\mathrm{T} & \Box\varphi \Rightarrow \varphi & \mathrm{D} & \Box\varphi \Rightarrow \Diamond\varphi \\
4 & \Box\varphi \Rightarrow \Box\Box\varphi & 5 & \Diamond\varphi \Rightarrow \Box\Diamond\varphi.
\end{array}
$$

The results of correspondence theory make it straightforward to derive completeness results for a range of simple normal modal logics. These results provide a useful point of comparison for normal modal logics, and account in a large part for the popularity of this style of semantics.

To use the logic developed above as an epistemic logic, the formula $\Box\varphi$ is read as: 'it is known that φ'. The worlds in the model are interpreted as epistemic alternatives, the accessibility relation defines what the alternatives are from any given world.

The logic defined above deals with the knowledge of a single agent. To deal with multi-agent knowledge, one adds to a model structure an indexed set of accessibility relations, one for each agent. The language is then extended by replacing the single modal operator '$\Box$' by an indexed set of unary modal operators $\{K_i\}$, where $i \in \{1, \ldots, n\}$. The formula $K_i\varphi$ is read: 'i knows that φ'. Each operator K_i is given exactly the same properties as '$\Box$'.

The next step is to consider how well normal modal logic serves as a logic of knowledge/belief. Consider first the necessitation rule and axiom K, since any normal modal system is committed to these. The necessitation rule tells us that an agent knows all valid formulae. Amongst other things, this means an agent knows all propositional tautologies. Since there are an infinite number of these, an agent will have an infinite number of items of knowledge: immediately, one is faced with a counter-intuitive property of the knowledge operator. Now consider the axiom K, which says that an agent's knowledge is closed under implication. Together with the necessitation rule, this axiom implies that an agent's knowledge is closed under logical consequence: an agent believes all the logical consequences of its beliefs. This also seems counter intuitive. For example, suppose, like every good logician, our agent knows Peano's axioms. Now Fermat's last theorem follows from Peano's axioms — but it took the combined efforts of some of the best minds over the past century to prove it. Yet if our agent's beliefs are closed under logical consequence, then our agent must know it. So consequential closure, implied by necessitation and the K axiom, seems an overstrong property for resource bounded reasoners.

These two problems — that of knowing all valid formulae, and that of knowledge/belief being closed under logical consequence — together constitute the famous *logical omniscience* problem. It has been widely argued that this problem makes the possible worlds model unsuitable for representing resource bounded believers — and any real system is resource bounded.

Axioms for Knowledge and Belief

We now consider the appropriateness of the axioms D, T, 4, and 5 for logics of knowledge/belief. The axiom D says that an agent's beliefs are non-contradictory; it can be re-written as: $K_i\varphi \Rightarrow$

$\neg K_i \neg \varphi$, which is read: 'if i knows φ, then i doesn't know $\neg\varphi$'. This axiom seems a reasonable property of knowledge/belief. The axiom T is often called the *knowledge* axiom, since it says that what is known is true. It is usually accepted as the axiom that distinguishes knowledge from belief: it seems reasonable that one could believe something that is false, but one would hesitate to say that one could *know* something false. Knowledge is thus often defined as true belief: i knows φ if i believes φ and φ is true. So defined, knowledge satisfies T. Axiom 4 is called the *positive introspection axiom*. Introspection is the process of examining one's own beliefs, and is discussed in detail in (Konolige, 1986a, Chapter 5). The positive introspection axiom says that an agent is aware of what it knows. Similarly, axiom 5 is the *negative introspection axiom*, which says that an agent is aware of what it doesn't know. Positive and negative introspection together imply an agent has perfect knowledge about what it does and doesn't know (cf. (Konolige, 1986a, Equation (5.11), p79)). Whether or not the two types of introspection are appropriate properties for knowledge/belief is the subject of some debate. However, it is generally accepted that positive introspection is a less demanding property than negative introspection, and is thus a more reasonable property for resource bounded reasoners.

Given the comments above, the axioms KTD45 are often chosen as a logic of (idealised) *knowledge*, and KD45 as a logic of (idealised) *belief*.

2.d Alternatives to the Possible Worlds Model

As a result of the difficulties with logical omniscience, many researchers have attempted to develop alternative formalisms for representing belief. Some of these are attempts to adapt the basic possible worlds model; others represent significant departures from it. In the subsections that follow, we examine some of these attempts.

Levesque — belief and awareness

In a 1984 paper, Levesque proposed a solution to the logical omniscience problem that involves making a distinction between *explicit* and *implicit* belief (Levesque, 1984). Crudely, the idea is that an agent has a relatively small set of explicit beliefs, and a very much larger (infinite) set of implicit beliefs, which includes the logical consequences of the explicit beliefs. To formalise this idea, Levesque developed a logic with two operators; one each for implicit and explicit belief. The semantics of the explicit belief operator were given in terms of a weakened possible worlds semantics, by borrowing some ideas from situation semantics (Barwise and Perry, 1983; Devlin, 1991). The semantics of the implicit belief operator were given in terms of a standard possible worlds approach. A number of objections have been raised to Levesque's model (Reichgelt, 1989b, p135): first, it does not allow quantification — this drawback has been rectified by Lakemeyer (Lakemeyer, 1991); second, it does not seem to allow for nested beliefs; third, the notion of a situation, which underlies Levesque's logic is, if anything, more mysterious than the notion of a world in possible worlds; and fourth, under certain circumstances, Levesque's proposal still makes unrealistic predictions about agent's reasoning capabilities.

In an effort to recover from this last negative result, Fagin and Halpern have developed a 'logic of general awareness', based on a similar idea to Levesque's but with a very much simpler semantics (Fagin and Halpern, 1985). However, this proposal has itself been criticised by some (Konolige, 1986b).

Konolige — the deduction model

A more radical approach to modelling resource bounded believers was proposed by Konolige (Konolige, 1986a). His *deduction model of belief* is, in essence, a direct attempt to model the 'beliefs' of symbolic AI systems. Konolige observed that a typical knowledge-based system has two key components: a database of symbolically represented 'beliefs', (which may take the form of rules, frames, semantic nets, or, more generally, formulae in some logical language), and some logically incomplete inference mechanism. Konolige modelled such systems in terms of *deduction structures*. A deduction structure is a pair $d = (\Delta, \rho)$, where Δ is a base set of formula in some logical language, and ρ is a set of inference rules, (which may be logically incomplete), representing the agent's reasoning mechanism. To simplify the formalism, Konolige assumed that an agent would apply its inference rules wherever possible, in order to generate the *deductive closure* of its base beliefs under its deduction rules. We model deductive closure in a function *close*:

$$close((\Delta, \rho)) \stackrel{\text{def}}{=} \{\varphi \mid \Delta \vdash_\rho \varphi\}$$

where $\Delta \vdash_\rho \varphi$ means that φ can be proved from Δ using only the rules in ρ. A belief logic can then be defined, with the semantics to a modal belief connective $[i]$, where i is an agent, given in terms of the deduction structure d_i modelling i's belief system: $[i]\varphi$ iff $\varphi \in close(d_i)$.

Konolige went on to examine the properties of the deduction model at some length, and developed a variety of proof methods for his logics, including resolution and tableau systems (Geissler and Konolige, 1986). The deduction model is undoubtedly simple; however, as a direct model of the belief systems of AI agents, it has much to commend it.

Meta-languages and syntactic modalities

A meta-language is one in which it is possible to represent the properties of another language. A first-order meta-language is a first-order logic, with the standard predicates, quantifiers, terms, and so on, whose domain contains formulae of some other language, called the *object* language. Using a meta-language, it is possible to represent a relationship between a meta-language term denoting an agent, and an object language term denoting some formula. For example, the meta-language formula $Bel(Janine, \ulcorner Father(Zeus, Cronos) \urcorner)$ might be used to represent the example

(1) that we saw earlier. The quote marks, $\ulcorner \ldots \urcorner$, are used to indicate that their contents are a meta-language term denoting the corresponding object-language formula.

Unfortunately, meta-language formalisms have their own package of problems, not the least of which is that they tend to fall prey to inconsistency (Montague, 1963; Thomason, 1980). However, there have been some fairly successful meta-language formalisms, including those by Konolige (Konolige, 1982), Haas (Haas, 1986), Morgenstern (Morgenstern, 1987), and Davies (Davies, 1993). Some results on retrieving consistency appeared in the late 1980s (Perlis, 1985; Perlis, 1988; des Rivieres and Levesque, 1986; Turner, 1990).

2.e Pro-attitudes: Goals and Desires

An obvious approach to developing a logic of goals or desires is to adapt possible worlds semantics — see, e.g., (Cohen and Levesque, 1990a; Wooldridge, 1994). In this view, each goal-accessible world represents one way the world might be if the agent's goals were realised. However, this approach falls prey to the *side effect* problem, in that it predicts that agents have a goal of the logical consequences of their goals (cf. the logical omniscience problem, discussed above). This is not a desirable property: one might have a goal of going to the dentist, with the necessary consequence of suffering pain, without having a goal of suffering pain. The problem is discussed, (in the context of intentions), in (Bratman, 1990). The basic possible worlds model has been adapted by some researchers in an attempt to overcome this problem (Wainer, 1994). Other, related semantics for goals have been proposed (Doyle et al., 1991; Kiss and Reichgelt, 1992; Rao and Georgeff, 1991b).

2.f Theories of Agency

All of the formalisms considered so far have focussed on just one aspect of agency. However, it is to be expected that a realistic agent theory will be represented in a logical framework that *combines* these various components. Additionally, we expect an agent logic to be capable of representing the *dynamic* aspects of agency. A complete agent theory, expressed in a logic with these properties, must define how the attributes of agency are related. For example, it will need to show how an agent's information and pro-attitudes are related; how an agent's cognitive state changes over time; how the environment affects an agent's cognitive state; and how an agent's information and pro-attitudes lead it to perform actions. Giving a good account of these relationships is the most significant problem faced by agent theorists.

An all-embracing agent theory is some time off, and yet significant steps have been taken towards it. In the following subsections, we briefly review some of this work.

Moore — knowledge and action

Moore was in many ways a pioneer of the use of logics for capturing aspects of agency (Moore, 1990). His main concern was the study of *knowledge pre-conditions for actions* — the question of what an agent needs to know in order to be able to perform some action. He formalised a model of *ability* in a logic containing a modality for knowledge, and a dynamic logic-like apparatus for modelling action (cf. (Harel, 1984)). This formalism allowed for the possibility of an agent having incomplete information about how to achieve some goal, and performing actions in order to find out how to achieve it. Critiques of the formalism (and attempts to improve on it) may be found in (Morgenstern, 1987; Lespérance, 1989).

Cohen and Levesque — intention

One of the best-known and most influential contributions to the area of agent theory is due to Cohen and Levesque (Cohen and Levesque, 1990a). Their formalism was originally used to develop a theory of intention (as in 'I intend to…'), which the authors required as a pre-requisite for a theory of speech acts (Cohen and Levesque, 1990b). However, the logic has subsequently proved to be so useful for reasoning about agents that it has been used in an analysis of conflict and cooperation in multi-agent dialogue (Galliers, 1988b; Galliers, 1988a), as well as several studies in the theoretical foundations of cooperative problem solving (Levesque et al., 1990; Jennings, 1992; Castelfranchi, 1990; Castelfranchi et al., 1992). Here, we shall review its use in developing a theory of intention.

Following Bratman, (Bratman, 1987; Bratman, 1990), Cohen and Levesque identify seven properties that must be satisfied by a reasonable theory of intention:

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 intention. First, they construct a *logic of rational agency*, 'being careful to sort out the relationships among the basic modal operators' (Cohen and Levesque, 1990a, p221). Over this framework, they introduce a number of derived constructs, which constitute a 'partial theory of rational action' (Cohen and Levesque, 1990a, p221); intention is one of these constructs.

The first major derived construct is the *persistent goal*. An agent has a persistent goal of φ iff:

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

It is a small step from persistent goals to a first definition of intention, as in 'intending to act': an agent intends to do action α iff it has a persistent goal to have brought about a state wherein it believed it was about to do α, and then did α. Cohen and Levesque go on to show how such a definition meets many of Bratman's criteria for a theory of intention (outlined above). A critique of Cohen and Levesque's theory of intention may be found in (Singh, 1992).

Rao and Georgeff — belief, desire, intention architectures

As we observed earlier, there is no clear consensus in either the AI or philosophy communities about precisely which combination of information and pro-attitudes are best suited to characterising rational agents. In the work of Cohen and Levesque, described above, just two basic attitudes were used: beliefs and goals. Further attitudes, such as intention, were defined in terms of these. In related work, Rao and Georgeff have developed a logical framework for agent theory based on three primitive modalities: beliefs, desires, and intentions (Rao and Georgeff, 1991b; Rao and Georgeff, 1991a; Rao and Georgeff, 1993). Their formalism is based on a branching model of time, (cf. (Emerson and Halpern, 1986)), in which belief-, desire- and intention-accessible worlds are themselves branching time structures.

They are particularly concerned with the notion of *realism* — the question of how an agent's beliefs about the future affect its desires and intentions. In other work, they also consider the potential for adding (social) plans to their formalism (Rao and Georgeff, 1992b; Kinny et al., 1992).

Singh

A quite different approach to modelling agents was taken by Singh, who has developed an interesting family of logics for representing intentions, beliefs, knowledge, know-how, and communication in a branching-time framework (Singh, 1990b; Singh, 1991a; Singh and Asher, 1991; Singh, 1991b); these articles are collected and expanded in (Singh, 1994). Singh's formalism is extremely rich, and considerable effort has been devoted to establishing its properties. However, its complexity prevents a detailed discussion here.

Werner

In an extensive sequence of papers, Werner has laid the foundations of a general model of agency, which draws upon work in economics, game theory, situated automata theory, situation semantics, and philosophy (Werner, 1988; Werner, 1989; Werner, 1990; Werner, 1991). At the time of writing, however, the properties of this model have not been investigated in depth.

Wooldridge — modelling multi-agent systems

For his 1992 doctoral thesis, Wooldridge developed a family of logics for representing the properties of multi-agent systems (Wooldridge, 1992; Wooldridge and Fisher, 1992). Unlike the approaches cited above, Wooldridge's aim was not to develop a general framework for agent theory. Rather, he hoped to construct formalisms that might be used in the specification and verification of realistic multi-agent systems. To this end, he developed a simple, and in some sense general, model of multi-agent systems, and showed how the histories traced out in the execution of such a system could be used as the semantic foundation for a family of both linear and branching time temporal belief logics. He then gave examples of how these logics could be used in the specification and verification of protocols for cooperative action.

2.g Communication

Formalisms for representing communication in agent theory have tended to be based on *speech act theory*, as originated by Austin (Austin, 1962), and further developed by Searle (Searle, 1969) and others (Cohen and Perrault, 1979; Cohen and Levesque, 1990a). Briefly, the key axiom of speech act theory is that communicative utterances are *actions*, in just the sense that physical actions are. They are performed by a speaker with the intention of bringing about a desired change in the world: typically, the speaker intends to bring about some particular mental state in a listener. Speech acts may *fail* in the same way that physical actions may fail: a listener generally has control over her mental state, and cannot be guaranteed to react in the way that the speaker intends. Much work in speech act theory has been devoted to classifying the various different types of speech acts. Perhaps the two most widely recognised categories of speech acts are *representatives* (of which *informing* is the paradigm example), and *directives* (of which *requesting* is the paradigm example).

Although not directly based on work in speech acts, (and arguably more to do with architectures than theories), we shall here mention work on *agent communication languages* (Genesereth and Ketchpel, 1994). The best known work on agent communication languages is that by the ARPA knowledge sharing effort (Patil et al., 1992). This work has been largely devoted to developing two related languages: the knowledge query and manipulation language (KQML) and the knowledge interchange format (KIF). KQML provides the agent designer with a standard syntax for messages, and a number of *performatives* that define the *force* of a message. Example performatives include *tell*, *perform*, and *reply*; the inspiration for these message types comes largely from speech act theory. KIF provides a syntax for message *content* — KIF is essentially the first-order predicate calculus, recast in a LISP-like syntax.

2.h Discussion

Formalisms for reasoning about agents have come a long way since Hintikka's pioneering work on logics of knowledge and belief (Hintikka, 1962). Within AI, perhaps the main emphasis of subsequent work has been on attempting to develop formalisms that capture the relationship between the various elements that comprise an agent's cognitive state; the paradigm example of this work is the well-known theory of intention developed by Cohen and Levesque (Cohen and Levesque, 1990a). Despite the very real progress that has been made, there still remain many fairly fundamental problems and issues still outstanding.

On a technical level, we can identify a number of issues that remain open. First, the problems associated with possible worlds semantics (notably, logical omniscience) cannot be regarded as solved. As we observed above, possible worlds remain the semantics of choice for many researchers, and yet they do not in general represent a realistic model of agents with limited resources — and of course all real agents are resource-bounded. One solution is to *ground* possible worlds semantics, giving them a precise interpretation in terms of the world. This was the approach taken in Rosenschein and Kaelbling's situated automata paradigm, and can be very successful. However, it is not clear how such a grounding could be given to pro-attitudes such as desires or intentions (although some attempts have been made (Singh, 1990a; Wooldridge, 1992; Werner, 1990)). There is obviously much work remaining to be done on formalisms for knowledge and belief, in particular in the area of modelling resource bounded reasoners.

With respect to logics that combine different attitudes, perhaps the most important problems still outstanding relate to intention. In particular, the relationship between intention and action has not been formally represented in a satisfactory way. The problem seems to be that having an intention to act makes it more likely that an agent will act, but does not generally guarantee it. While it seems straightforward to build systems that appear to have intentions, (Wooldridge, 1995), it seems much harder to capture this relationship formally. Other problems that have not yet really been addressed in the literature include the management of multiple, possibly conflicting intentions, and the formation, scheduling, and reconsideration of intentions.

The question of exactly *which combination* of attitudes is required to characterise an agent is also the subject of some debate. As we observed above, a currently popular approach is to use a combination of beliefs, desires, and intentions (hence BDI architectures (Rao and Georgeff, 1991b)). However, there are alternatives: Shoham, for example, suggests that the notion of choice is more fundamental (Shoham, 1990). Comparatively little work has yet been done on formally comparing the suitability of these various combinations. One might draw a parallel with the use of temporal logics in mainstream computer science, where the expressiveness of specification languages is by now a well-understood research area (Emerson and Halpern, 1986). Perhaps the obvious requirement for the short term is experimentation with real agent specifications, in order to gain a better understanding of the relative merits of different formalisms.

More generally, the kinds of logics used in agent theory tend to be rather elaborate, typically containing many modalities which interact with each other in subtle ways. Very little work has yet been carried out on the theory underlying such logics (perhaps the only notable exception is (Catach, 1988)). Until the general principles and limitations of such multi-modal logics become understood, we might expect that progress with using such logics will be slow. One area in which work is likely to be done in the near future is theorem proving techniques for multi-modal logics.

Finally, there is often some confusion about the role played by a theory of agency. The view we take is that such theories represent *specifications* for agents. The advantage of treating agent theories as specifications, and agent logics as specification languages, is that the problems and issues we then face are familiar from the discipline of software engineering: How useful or expressive is the specification language? How concise are agent specifications? How does one refine or otherwise transform a specification into an implementation? However, the view of agent theories as specifications is not shared by all researchers. Some intend their agent theories to be used as knowledge representation formalisms, which raises the difficult problem of algorithms to reason with such theories. Still others intend their work to formalise a concept of interest in cognitive science or philosophy (this is, of course, what Hintikka intended in his early work on logics of knowledge of belief). What *is* clear is that it is important to be precise about the role one expects an agent theory to play.

2.i Further Reading

For a recent discussion on the role of logic and agency, which lays out in more detail some contrasting views on the subject, see (Israel, 1993, pp17–24). For a detailed discussion of intentionality and the intentional stance, see (Dennett, 1978; Dennett, 1987). A number of papers on AI treatments of agency may be found in (Allen et al., 1990). For an introduction to modal logic, see (Chellas, 1980); a slightly older, though more wide ranging introduction, may be found in (Hughes and Cresswell, 1968). As for the use of modal logics to model knowledge and belief, see (Halpern and Moses, 1992), which includes complexity results and proof procedures. Related work on modelling knowledge has been done by the distributed systems community, who give the worlds in possible worlds semantics a precise interpretation; for an introduction and further references, see (Halpern, 1987; Fagin et al., 1992). Overviews of formalisms for modelling belief and knowledge may be found in (Halpern, 1986; Konolige, 1986a; Reichgelt, 1989a; Wooldridge, 1992). A variant on the possible worlds framework, called the *recursive modelling method*, is described in (Gmytrasiewicz and Durfee, 1993); a deep theory of belief may be found in (Mack, 1994). *Situation semantics*, developed in the early 1980s and recently the subject of renewed interest, represent a fundamentally new approach to modelling the world and cognitive systems (Barwise and Perry, 1983; Devlin, 1991). However, situation semantics are not (yet) in the mainstream of (D)AI, and it is not obvious what impact the paradigm will ultimately have.

Logics which integrate time with mental states are discussed in (Kraus and Lehmann, 1988; Halpern and Vardi, 1989; Wooldridge and Fisher, 1994); the last of these presents a tableau-based proof method for a temporal belief logic. Two other important references for temporal aspects are (Shoham, 1988; Shoham, 1989). Thomas has developed some logics for representing agent theories as part of her framework for agent programming languages; see (Thomas et al., 1991; Thomas, 1993) and section 4. For an introduction to temporal logics and related topics, see (Goldblatt, 1987; Emerson, 1990). A non-formal discussion of intention may be found in (Bratman, 1987), or more briefly (Bratman, 1990). Further work on modelling intention may be found in (Grosz and Sidner, 1990; Sadek, 1992; Goldman and Lang, 1991; Konolige and Pollack, 1993; Bell, 1995; Dongha, 1995). Related work, focussing less on single-agent attitudes, and more on social aspects, is (Levesque et al., 1990; Jennings, 1993a; Wooldridge, 1994; Wooldridge and Jennings, 1994).

Finally, although we have not discussed formalisms for reasoning about action here, we suggested above that an agent logic would need to incorporate some mechanism for representing agent's actions. Our reason for avoiding the topic is simply that the field is so big, it deserves a whole review in its own right. Good starting points for AI treatments of action are (Allen, 1984; Allen et al., 1990; Allen et al., 1991). Other treatments of action in agent logics are based on formalisms borrowed from mainstream computer science, notably dynamic logic (originally developed to reason about computer programs) (Harel, 1984). The logic of *seeing to it that* has been discussed in the formal philosophy literature, but has yet to impact on (D)AI (Belnap and Perloff, 1988; Perloff, 1991; Belnap, 1991; Segerberg, 1989).

3 Agent Architectures

Until now, this article has been concerned with agent theory — the construction of formalisms for reasoning about agents, and the properties of agents expressed in such formalisms. Our aim in this section is to shift the emphasis from theory to practice. We consider the issues surrounding the construction of computer systems that satisfy the properties specified by agent theorists. This is the area of *agent architectures*. Maes defines an agent architecture as:

> '[A] particular methodology for building [agents]. It specifies how … the agent can be decomposed into the construction of a set of component modules and how these modules should be made to interact. The total set of modules and their interactions has to provide an answer to the question of how the sensor data and

> the current internal state of the agent determine the actions … and future internal state of the agent. An architecture encompasses techniques and algorithms that support this methodology.' (Maes, 1991, p115).

Kaelbling considers an agent architecture to be:

> '[A] specific collection of software (or hardware) modules, typically designated by boxes with arrows indicating the data and control flow among the modules. A more abstract view of an architecture is as a general methodology for designing particular modular decompositions for particular tasks.' (Kaelbling, 1991, p86)

The classical approach to building agents is to view them as a particular type of knowledge-based system. This paradigm is known as *symbolic AI*: we begin our review of architectures with a look at this paradigm, and the assumptions that underpin it.

3.a Classical Approaches: Deliberative Architectures

The foundation upon which the symbolic AI paradigm rests is the *physical-symbol system hypothesis*, formulated by Newell and Simon (Newell and Simon, 1976). A physical symbol system is defined to be a physically realizable set of physical entities (symbols) that can be combined to form structures, and which is capable of running processes that operate on those symbols according to symbolically coded sets of instructions. The physical-symbol system hypothesis then says that such a system is capable of general intelligent action.

It is a short step from the notion of a physical symbol system to McCarthy's dream of a *sentential processing automaton*, or *deliberative agent*. (The term 'deliberative agent' seems to have derived from Genesereth's use of the term 'deliberate agent' to mean a specific type of symbolic architecture (Genesereth and Nilsson, 1987, pp325–327).) We define a deliberative agent or agent architecture to be one that contains an explicitly represented, symbolic model of the world, and in which decisions (for example about what actions to perform) are made via logical (or at least pseudo-logical) reasoning, based on pattern matching and symbolic manipulation. The idea of deliberative agents based on purely logical reasoning is highly seductive: to get an agent to realise some theory of agency one might naively suppose that it is enough to simply give it logical representation of this theory and 'get it to *do a bit of theorem proving*' (Shardlow, 1990, section 3.2). If one aims to build an agent in this way, then there are at least two important problems to be solved:

1. The transduction problem: that of translating the real world into an accurate, adequate symbolic description, in time for that description to be useful.

2. The representation/reasoning problem: that of how to symbolically represent information about complex real-world entities and processes, and how to get agents to reason with this information in time for the results to be useful.

The former problem has led to work on vision, speech understanding, learning, etc. The latter has led to work on knowledge representation, automated reasoning, automatic planning, etc. Despite the immense volume of work that these problems have generated, most researchers would accept that neither is anywhere near solved. Even seemingly trivial problems, such as commonsense reasoning, have turned out to be extremely difficult (cf. the CYC project (Guha and Lenat, 1994)). The underlying problem seems to be the difficulty of theorem proving in even very simple logics, and the complexity of symbol manipulation algorithms in general: recall that first-order logic is not even *decidable*, and modal extensions to it (including representations of belief, desire, time, and so on) tend to be *highly* undecidable. Thus, the idea of building 'agents as theorem provers' — what might be called an extreme logicist view of agency — although it is very attractive in theory, seems, for the time being at least, to be unworkable in practice. Perhaps more troubling for symbolic AI is that many symbol manipulation algorithms of interest are intractable. It seems hard to build useful symbol manipulation algorithms that will be guaranteed to terminate with useful results in an acceptable fixed time bound. And yet such algorithms seem essential if agents are to operate in any real-world, time-constrained domain. Good discussions of this point appear in (Kaelbling, 1986; Russell and Wefald, 1991).

It is because of these problems that some researchers have looked to alternative techniques for building agents; such alternatives are discussed in section 3.b. First, however, we consider efforts made within the symbolic AI community to construct agents.

Planning agents

Since the early 1970s, the AI planning community has been closely concerned with the design of artificial agents; in fact, it seems reasonable to claim that most innovations in agent design have come from this community. Planning is essentially automatic programming: the design of a course of action that, when executed, will result in the achievement of some desired goal. Within the symbolic AI community, it has long been assumed that some form of AI planning system will be a central component of any artificial agent. Perhaps the best-known early planning system was STRIPS (Fikes and Nilsson, 1971). This system takes a symbolic description of both the world and a desired goal state, and a set of action descriptions, which characterise the pre- and post-conditions associated with various actions. It then attempts to find a sequence of actions that will achieve the goal, by using a simple means-ends analysis, which essentially involves matching the post-conditions of actions against the desired goal. The STRIPS planning algorithm was very simple, and proved to be ineffective on problems of even moderate complexity. Much effort was subsequently devoted to developing more effective techniques. Two major innovations were *hierarchical* and *non-linear* planning (Sacerdoti, 1974; Sacerdoti, 1975). However, in the mid 1980s, Chapman established some theoretical results which indicate that even such refined techniques will ultimately turn out to be unusable in any time-constrained system (Chapman, 1987). These results have had a profound influence on subsequent AI planning research; perhaps more than any other, they have caused some researchers to question the whole symbolic AI paradigm, and have thus led to the work on alternative approaches that we discuss in section 3.b.

In spite of these difficulties, various attempts have been made to construct agents whose primary component is a planner. For example: the Integrated Planning, Execution and Monitoring (IPEM) system is based on a sophisticated non-linear planner (Ambros-Ingerson and Steel, 1988); Wood's AUTODRIVE system has planning agents operating in a highly dynamic environment (a traffic simulation) (Wood, 1993); Etzioni has built 'softbots' that can plan and act in a UNIX environment (Etzioni et al., 1994); and finally, Cohen's PHEONIX system includes planner-based agents that operate in the domain of simulated forest fire management (Cohen et al., 1989).

Bratman, Israel and Pollack — IRMA

In section 2, we saw that some researchers have considered frameworks for agent theory based on beliefs, desires, and intentions (Rao and Georgeff, 1991b). Some researchers have also developed agent architectures based on these attitudes. One example is the *Intelligent Resource-bounded Machine Architecture* (IRMA) (Bratman et al., 1988). This architecture has four key symbolic data structures: a plan library, and explicit representations of beliefs, desires, and intentions. Additionally, the architecture has: a reasoner, for reasoning about the world; a means-ends analyser, for determining which plans might be used to achieve the agent's intentions; an *opportunity analyser*, which monitors the environment in order to determine further options for the agent; a *filtering process*; and a *deliberation process*. The filtering process is responsible for determining the subset of the agent's potential courses of action that have the property of being consistent with the agent's current intentions. The choice between competing options is made by the deliberation process. The IRMA architecture has been evaluated in an experimental scenario known as the *Tileworld* (Pollack and Ringuette, 1990).

Vere and Bickmore — HOMER

An interesting experiment in the design of intelligent agents was conducted by Vere and Bickmore (Vere and Bickmore, 1990). They argued that the enabling technologies for intelligent agents are sufficiently developed to be able to construct a prototype autonomous agent, with linguistic ability, planning and acting capabilities, and so on. They developed such an agent, and christened it HOMER. This agent is a simulated robot submarine, which exists in a two-dimensional 'Seaworld', about which it has only partial knowledge. HOMER takes instructions from a user in a limited subset of English with about an 800 word vocabulary; instructions can contain moderately sophisticated temporal references. HOMER can plan how to achieve its instructions, (which typically relate to collecting and moving items around the Seaworld), and can then execute its plans, modifying them as required during execution. The agent has a limited *episodic memory*, and using this, is able to answer questions about its past experiences.

Jennings — GRATE*

GRATE* is a layered architecture in which the behaviour of an agent is guided by the mental attitudes of beliefs, desires, intentions and joint intentions (Jennings, 1993b). Agents are divided into two distinct parts: a domain level system and a cooperation and control layer. The former solves problems for the organisation; be it in the domain of industrial control, finance or transportation. The latter is a meta-level controller which operates on the domain level system with the aim of ensuring that the agent's domain level activities are coordinated with those of others within the community. The cooperation layer is composed of three generic modules: a control module which interfaces to the domain level system, a situation assessment module and a cooperation module. The assessment and cooperation modules provide an implementation of a model of joint responsibility (Jennings, 1992), which specifies how agents should act both locally and towards other agents whilst engaged in cooperative problem solving. The performance of a GRATE* community has been evaluated against agents which only have individual intentions, and agents which behave in a selfish manner, in the domain of electricity transportation management. A significant improvement was noted when the situation became complex and dynamic (Jennings, 1995).

3.b Alternative Approaches: Reactive Architectures

As we observed above, there are many unsolved (some would say insoluble) problems associated with symbolic AI. These problems have led some researchers to question the viability of the whole paradigm, and to the development of what are generally know as *reactive* architectures. For our purposes, we shall define a reactive architecture to be one that does not include any kind of central symbolic world model, and does not use complex symbolic reasoning.

Brooks — behaviour languages

Possibly the most vocal critic of the symbolic AI notion of agency has been Rodney Brooks, a researcher at MIT who apparently became frustrated by AI approaches to building control mechanisms for autonomous mobile robots. In a 1985 paper, he outlined an alternative architecture for building agents, the so called *subsumption architecture* (Brooks, 1986). The review of alternative approaches begins with Brooks' work.

In recent papers, (Brooks, 1990; Brooks, 1991b; Brooks, 1991a), Brooks has propounded three key theses:

1. Intelligent behaviour can be generated *without* explicit representations of the kind that symbolic AI proposes.

2. Intelligent behaviour can be generated *without* explicit abstract reasoning of the kind that symbolic AI proposes.

3. Intelligence is an *emergent* property of certain complex systems.

Brooks identifies two key ideas that have informed his research:

1. Situatedness and embodiment: 'Real' intelligence is situated in the world, not in disembodied systems such as theorem provers or expert systems.

2. Intelligence and emergence: 'Intelligent' behaviour arises as a result of an agent's interaction with its environment. Also, intelligence is 'in the eye of the beholder'; it is not an innate, isolated property.

If Brooks was just a Dreyfus-style critic of AI, his ideas might not have gained much currency. However, to demonstrate his claims, he has built a number of robots, based on the subsumption architecture. A subsumption architecture is a hierarchy of task-accomplishing *behaviours*. Each behaviour 'competes' with others to exercise control over the robot. Lower layers represent more primitive kinds of behaviour, (such as avoiding obstacles), and have precedence over layers further up the hierarchy. It should be stressed that the resulting systems are, in terms of the amount of computation they need to do, *extremely* simple, with no explicit reasoning of the kind found in symbolic AI systems. But despite this simplicity, Brooks has demonstrated the robots doing tasks that would be impressive if they were accomplished by symbolic AI systems. Similar work has been reported by Steels, who described simulations of 'Mars explorer' systems, containing a large number of subsumption-architecture agents, that can achieve near-optimal performance in certain tasks (Steels, 1990).

Agre and Chapman — PENGI

At about the same time as Brooks was describing his first results with the subsumption architecture, Chapman was completing his Master's thesis, in which he reported the theoretical difficulties with planning described above, and was coming to similar conclusions about the inadequacies of the symbolic AI model himself. Together with his co-worker Agre, he began to explore alternatives to the AI planning paradigm (Chapman and Agre, 1986).

Agre observed that most everyday activity is 'routine' in the sense that it requires little — if any — new abstract reasoning. Most tasks, once learned, can be accomplished in a routine way, with little variation. Agre proposed that an efficient agent architecture could be based on the idea of 'running arguments'. Crudely, the idea is that as most decisions are routine, they can be encoded into a low-level structure (such as a digital circuit), which only needs periodic updating, perhaps to handle new kinds of problems. His approach was illustrated with the celebrated PENGI system (Agre and Chapman, 1987). PENGI is a simulated computer game, with the central character controlled using a scheme such as that outlined above.

Rosenschein and Kaelbling — situated automata

Another sophisticated approach is that of Rosenschein and Kaelbling (Rosenschein, 1985; Rosenschein and Kaelbling, 1986; Kaelbling and Rosenschein, 1990; Kaelbling, 1991). In their *situated automata* paradigm, an agent is specified in declarative terms. This specification is then compiled down to a digital machine, which satisfies the declarative specification. This digital machine can operate in a provably time-bounded fashion; it does not do any symbol manipulation, and in fact no symbolic expressions are represented in the machine at all. The logic used to specify an agent is essentially a modal logic of knowledge (see above). The technique depends upon the possibility of giving the worlds in possible worlds semantics a concrete interpretation in terms of the states of an automaton:

> '[An agent] … x is said to carry the information that p in world state s, written $s \models K(x, p)$, if for all world states in which x has the same value as it does in s, the proposition p is true.' (Kaelbling and Rosenschein, 1990, p36)

An agent is specified in terms of two components: perception and action. Two programs are then used to synthesise agents: RULER is used to specify the perception component of an agent; GAPPS is used to specify the action component.

RULER takes as its input three components:

> '[A] specification of the semantics of the [agent's] inputs ("whenever bit 1 is on, it is raining"); a set of static facts ("whenever it is raining, the ground is wet"); and a specification of the state transitions of the world ("if the ground is wet, it stays wet until the sun comes out"). The programmer then specifies the desired semantics for the output ("if this bit is on, the ground is wet"), and the compiler … [synthesises] a circuit whose output will have the correct semantics. … All that declarative "knowledge" has been reduced to a very simple circuit." (Kaelbling, 1991, p86)

The GAPPS program takes as its input a set of *goal reduction rules*, (essentially rules that encode information about how goals can be achieved), and a top level goal, and generates a program that can be translated into a digital circuit in order to realise the goal. Once again, the generated circuit does not represent or manipulate symbolic expressions; all symbolic manipulation is done at compile time.

The situated automata paradigm has attracted much interest, as it appears to combine the best elements of both reactive and symbolic, declarative systems. However, at the time of writing, the theoretical limitations of the approach are not well understood; there are similarities with the automatic synthesis of programs from temporal logic specifications, a complex area of much ongoing work in mainstream computer science (see the comments in (Emerson, 1990)).

Maes — Agent Network Architecture

Pattie Maes has developed an agent architecture in which an agent is defined as a set of *competence modules* (Maes, 1989; Maes, 1990b; Maes, 1991). These modules loosely resemble the behaviours of Brooks' subsumption architecture (above). Each module is specified by the designer in terms of pre- and post-conditions (rather like STRIPS operators), and an *activation level*, which gives a real-valued indication of the *relevance* of the module in a particular situation. The higher the activation level of a module, the more likely it is that this module will influence the behaviour of the agent. Once specified, a set of competence modules is compiled into a *spreading activation network*, in which the modules are linked to one-another in ways defined by their pre- and post-conditions. For example, if module a has post-condition φ, and module b has pre-condition φ, then a and b are connected by a *successor* link. Other types of link include predecessor links and conflicter links. When an agent is executing, various modules may become more active in given situations, and may be executed. The result of execution may be a command to an effector unit, or perhaps the increase in activation level of a successor module.

There are obvious similarities between the agent network architecture and neural network architectures. Perhaps the key difference is that it is difficult to say what the meaning of a node in a neural net is; it only has a meaning in the context of the net itself. Since competence modules are defined in declarative terms, however, it is very much easier to say what their meaning is.

3.c Hybrid Architectures

Many researchers have suggested that neither a completely deliberative nor completely reactive approach is suitable for building agents. They have argued the case for *hybrid* systems, which attempt to marry classical and alternative approaches.

An obvious approach is to build an agent out of two (or more) subsystems: a deliberative one, containing a symbolic world model, which develops plans and makes decisions in the way proposed by mainstream symbolic AI; and a reactive one, which is capable of reacting to events that occur in the environment without engaging in complex reasoning. Often, the reactive component is given some kind of precedence over the deliberative one, so that it can provide a rapid response to important environmental events. This kind of structuring leads naturally to the idea of a *layered* architecture, of which TOURINGMACHINES (Ferguson, 1992a) and INTERRAP (Müller and Pischel, 1994) are good examples. (These architectures are described below.) In such an architecture, an agent's control subsystems are arranged into a hierarchy, with higher layers dealing with information at increasing levels of abstraction. Thus, for example, the very lowest layer might map raw sensor data directly onto effector outputs, while the uppermost layer deals with long-term goals. A key problem in such architectures is what kind control framework to embed the agent's subsystems in, to manage the interactions between the various layers.

Georgeff and Lansky — PRS

One of the best-known agent architectures is the *Procedural Reasoning System* (PRS), developed by Georgeff and Lansky (Georgeff and Lansky, 1987). Like IRMA, (see above), the PRS is a belief-desire-intention architecture, which includes a plan library, as well as explicit symbolic representations of beliefs, desires, and intentions. Beliefs are facts, either about the external world or the system's internal state. These facts are expressed in classical first-order logic. Desires are represented as *system behaviours* (rather than as static representations of goal states). A PRS plan library contains a set of partially-elaborated plans, called *knowledge areas* (KAs), each of which is associated with an *invocation condition*. This condition determines when the KA is to be *activated*. KAs may be activated in a goal-driven or data-driven fashion; KAs may also be *reactive*, allowing the PRS to respond rapidly to changes in its environment. The set of currently active KAs in a system represent its *intentions*. These various data structures are manipulated by a *system interpreter*, which is responsible for updating beliefs, invoking KAs, and executing actions. The PRS has been evaluated in a simulation of maintenance procedures for the space shuttle, as well as other domains (Georgeff and Ingrand, 1989).

Ferguson — TOURINGMACHINES

For his 1992 Doctoral thesis, Ferguson developed the TOURINGMACHINES hybrid agent architecture (Ferguson, 1992b; Ferguson, 1992a)[5]. The architecture consists of *perception* and *action* subsystems, which interface directly with the agent's environment, and three *control layers*, embedded in a *control framework*, which mediates between the layers. Each layer is an independent, activity-producing, concurrently executing process.

The *reactive layer* generates potential courses of action in response to events that happen too quickly for other layers to deal with. It is implemented as a set of situation-action rules, in the style of Brooks' subsumption architecture (see above).

The *planning layer* constructs plans and selects actions to execute in order to achieve the agent's goals. This layer consists of two components: a planner, and a *focus of attention mechanism*. The planner integrates plan generation and execution, and uses a library of partially elaborated plans, together with a topological world map, in order to construct plans that will accomplish the agent's main goal. The purpose of the focus of attention mechanism is to limit the amount of information that the planner must deal with, and so improve its efficiency. It does this by filtering out irrelevant information from the environment.

[5] It is worth noting that Ferguson's thesis gives a good overview of the problems and issues associated with building rational, resource-bounded agents. Moreover, the description given of the TOURINGMACHINES architecture is itself extremely clear. We recommend it as a point of departure for further reading.

The *modelling layer* contains symbolic representations of the cognitive state of other entities in the agent's environment. These models are manipulated in order to identify and resolve *goal conflicts* — situations where an agent can no longer achieve its goals, as a result of unexpected interference.

The three layers are able to communicate with each other (via message passing), and are embedded in a control framework. The purpose of this framework is to mediate between the layers, and in particular, to deal with conflicting action proposals from the different layers. The control framework does this by using *control rules*.

Burmeister *et al.* — COSY

The COSY architecture is a hybrid BDI-architecture that includes elements of both the PRS and IRMA, and was developed specifically for a multi-agent testbed called DASEDIS (Burmeister and Sundermeyer, 1992; Haddadi, 1994). The architecture has five main components: (i) sensors; (ii) actuators; (iii) communications; (iv) cognition; and (v) intention. The first three components are straightforward: the sensors receive non-communicative perceptual input, the actuators allow the agent to perform non-communicative actions, and the communications component allows the agent to send messages. Of the remaining two components, the *intention* component contains 'long-term goals, attitudes, responsibilities and the like … the control elements taking part in the reasoning and decision-making of the cognition component' (Haddadi, 1994, p15), and the cognition component is responsible for mediating between the intentions of the agent and its beliefs about the world, and choosing an appropriate action to perform. Within the cognition component is the knowledge base containing the agent's beliefs, and three procedural components: a *script execution* component, a *protocol execution* component, and a *reasoning, deciding, and reacting* component. A script is very much like a script in Schank's original sense: it is a stereotypical recipe or plan for achieving a goal. Protocols are stereotypical dialogues representing cooperation frameworks such as the contract net (Smith, 1980). The reasoning, deciding and reacting component is perhaps the key component in COSY. It is made up of a number of other subsystems, and is structured rather like the PRS and IRMA (see above). An *agenda* is maintained, that contains a number of active scripts. These scripts may be invoked in a goal-driven fashion (to satisfy one of the agent's intentions), or a data-driven fashion (in response to the agent's current situation). A *filter* component chooses between competing scripts for execution.

Müller *et al.* — INTERRAP

INTERRAP, like Ferguson's TOURINGMACHINES, is a layered architecture, with each successive layer representing a higher level of abstraction than the one below it (Müller and Pischel, 1994; Müller et al., 1995; Müller, 1994). In INTERRAP, these layers are further subdivided into two vertical layers: one containing layers of knowledge bases, the other containing various control components, that interact with the knowledge bases at their level. At the lowest level is the *world interface* control component, and the corresponding *world model* knowledge base. The world interface component, as its name suggests, manages the interface between the agent and its environment, and thus deals with acting, communicating, and perception.

Above the world interface component is the *behaviour-based* component. The purpose of this component is to implement and control the basic reactive capability of the agent. This component manipulates a set of *patterns of behaviour* (PoB). A PoB is a structure containing a pre-condition that defines when the PoB is to be activated, various conditions that define the circumstances under which the PoB is considered to have succeeded or failed, a post-condition (*à la* STRIPS (Fikes and Nilsson, 1971)), and an executable *body*, that defines what action should be performed if the PoB is executed. (The action may be a primitive, resulting in a call on the agent's world interface, or may involve calling on a higher-level layer to generate a plan.)

Above the behaviour-based component in INTERRAP is the *plan-based component*. This component contains a planner that is able to generate single-agent plans in response to requests from the behaviour-based component. The knowledge-base at this layer contains a set of plans, including a plan library. The highest layer in INTERRAP is the *cooperation component*. This component is able to generate joint plans, that satisfy the goals of a number of agents, by elaborating plans selected from a plan library. These plans are generated in response to requests from the plan-based component.

Control in INTERRAP is both data- and goal-driven. Perceptual input is managed by the world-interface, and typically results in a change to the world model. As a result of changes to the world model, various patterns of behaviour may be activated, dropped, or executed. As a result of PoB execution, the plan-based component and cooperation component may be asked to generate plans and joint plans respectively, in order to achieve the goals of the agent. This ultimately results in primitive actions and messages being generated by the world interface.

3.d Discussion

The deliberative, symbolic paradigm is, at the time of writing, the dominant approach in (D)AI. This state of affairs is likely to continue, at least for the near future. There seem to be several reasons for this. Perhaps most importantly, many symbolic AI techniques (such as rule-based systems) carry with them an associated technology and methodology that is becoming familiar to mainstream computer scientists and software engineers. Despite the well-documented problems with symbolic AI systems, this makes symbolic AI agents (such as GRATE* (Jennings, 1993b)) an attractive proposition when compared to reactive systems, which have as yet no associated methodology. The need for a development methodology seems to be one of the most pressing requirements for reactive systems. Anecdotal descriptions of current reactive systems implementations indicate that each such system must be individually hand-crafted through a potentially lengthy period of experimentation (Wavish and Graham, 1995). This kind of approach seems unlikely to be usable for large systems. Some researchers have suggested that techniques from the domain of genetic algorithms or machine learning might be used to get around these development problems, though this work is at a very early stage.

There is a pressing need for research into the capabilities of reactive systems, and perhaps in particular to the types of application for which these types of system are best suited; some preliminary work has been done in this area, using a problem domain known as the TILEWORLD (Pollack and Ringuette, 1990). With respect to reactive systems, Ferguson suggests that:

> '[T]he strength of purely non-deliberative architectures lies in their ability to exploit *local* patterns of activity in their current surroundings in order to generate more or less hardwired action responses … for a given set of stimuli. Successful operation using this method pre-supposes: (i) that the complete set of environmental stimuli required for unambiguously determining action sequences is always present and readily identifiable — in other words, that the agent's activity can be *situationally determined*; (ii) that the agent has no *global* task constraints … which need to be reasoned about at run time; and (iii) that the agent's goal or desire system is capable of being represented *implicitly* in the agent's structure according to a fixed, pre-compiled ranking scheme.' (Ferguson, 1992a, pp29–30)

Hybrid architectures, such as the PRS, TOURINGMACHINES, INTERRAP, and COSY, are currently a very active area of work, and arguably have some advantages over both purely deliberative and purely reactive architectures. However, an outstanding problem with such architectures is that of combining multiple interacting subsystems (deliberative and reactive) *cleanly*, in a well-motivated control framework. Humans seem to manage different levels of abstract behaviour with comparative ease; it is not clear that current hybrid architectures can do so.

Another area where as yet very little work has been done is the *generation* of goals and intentions. Most work in AI assumes that an agent has a single, well-defined goal that it must achieve. But if agents are ever to be really autonomous, and act pro-actively, then they must be able to generate their own goals when either the situation demands, or the opportunity arises. Some preliminary work in this area is (Norman and Long, 1995). Similarly, little work has yet been done into the management and scheduling of multiple, possibly conflicting goals; some preliminary work is reported in (Dongha, 1995).

Finally, we turn to the relationship between agent theories and agent architectures. To what extent do the agent architectures reviewed above correspond to the theories discussed in section 2? What, if any, is the theory that underpins an architecture? With respect to purely deliberative architectures, there is a wealth of underlying theory. The close relationship between symbolic processing systems and mathematical logic means that the semantics of such architectures can often be represented as a logical system of some kind. There is a wealth of work establishing such relationships in AI, of which a particularly relevant example is (Rao and Georgeff, 1992a). This article discusses the relationship between the abstract BDI logics developed by Rao *et al.* for reasoning about agents, and an abstract 'agent interpreter', based on the PRS. However, the relationship between the logic and the architecture is not formalised; the BDI logic is not used to give a formal semantics to the architecture, and in fact it is difficult to see how such a logic could be used for this purpose. A serious attempt to define the semantics of a (somewhat simple) agent architecture is presented in (Wooldridge, 1995), where a formal model of the system MYWORLD, in which agents are directly programmed in terms of beliefs and intentions, is used as the basis upon which to develop a logic for reasoning about MYWORLD systems. Although the logic contains modalities for representing beliefs and intentions, the semantics of these modalities are given in terms of the agent architecture itself, and the problems associated with possible worlds do not, therefore, arise; this work builds closely on Konolige's models of the beliefs of symbolic AI systems (Konolige, 1986a). However, more work needs to be done using this technique to model more complex architectures, before the limitations and advantages of the approach are well-understood.

Like purely deliberative architectures, some reactive systems are also underpinned by a relatively transparent theory. Perhaps the best example is the situated automata paradigm, where an agent is specified in terms of a logic of knowledge, and this specification is compiled down to a simple digital machine that can be realistically said to realise its corresponding specification. However, for other purely reactive architectures, based on more *ad hoc* principles, it is not clear that there is any transparent underlying theory. It could be argued that hybrid systems also tend to be *ad hoc*, in that while their structures are well-motivated from a design point of view, it is not clear how one might reason about them, or what their underlying theory is. In particular, architectures that contain a number of independent activity producing subsystems, which compete with each other in real time to control the agent's activities, seem to defy attempts at formalisation. It is a matter of debate whether this need be considered a serious disadvantage, but one argument is that unless we have a good theoretical model of a particular agent or agent architecture, then we shall never really understand *why* it works. This is likely to make it difficult to generalise and reproduce results in varying domains.

3.e Further Reading

Most introductory textbooks on AI discuss the physical symbol system hypothesis; a good recent example of such a text is (Ginsberg, 1993). A detailed discussion of the way that this hypothesis has affected thinking in symbolic AI is provided in (Shardlow, 1990). There are many objections to the symbolic AI paradigm, in addition to those we have outlined above. Again, introductory textbooks provide the stock criticisms and replies.

There is a wealth of material on planning and planning agents. See (Georgeff, 1987) for an overview of the state of the art in planning (as it was in 1987), (Allen et al., 1990) for a thorough collection of papers on planning, (many of the papers cited above are included), and (Wilkins, 1988) for a detailed description of SIPE, a sophisticated planning system used in a real-world application (the control of a brewery!) Another important collection of planning papers is (Georgeff and Lansky, 1986). The book by Dean and Wellman and the book by Allen *et al.* contain much useful related material (Dean and Wellman, 1991; Allen et al., 1991). There is now a regular international conference on planning; the proceedings of the first were published as (Hendler, 1992).

The collection of papers edited by Maes (Maes, 1990a) contains many interesting papers on alternatives to the symbolic AI paradigm. Kaelbling (Kaelbling, 1986) presents a clear discussion of the issues associated with developing resource-bounded rational agents, and proposes an agent architecture somewhat similar to that developed by Brooks. A proposal by Nilsson for *teleo reactive programs* — goal directed programs that nevertheless respond to their environment — is described in (Nilsson, 1992). The proposal draws heavily on the situated automata paradigm; other work based on this paradigm is described in (Shoham, 1990; Kiss and Reichgelt, 1992). Schoppers has proposed compiling plans in advance, using traditional planning techniques, in order to develop *universal plans*, which are essentially decision trees that can be used to efficiently determine an appropriate action in any situation (Schoppers, 1987). Another proposal for building 'reactive planners' involves the use of *reactive action packages* (Firby, 1987).

Other hybrid architectures are described in (Hayes-Roth, 1990; Downs and Reichgelt, 1991; Aylett and Eustace, 1994; Bussmann and Demazeau, 1994).

4 Agent Languages

As agent technology becomes more established, we might expect to see a variety of software tools become available for the design and construction of agent-based systems; the need for software support tools in this area was identified as long ago as the mid-1980s (Gasser et al., 1987). The emergence of a number of prototypical *agent languages* is one sign that agent technology is becoming more widely used, and that many more agent-based applications are likely to be developed in the near future. By an *agent language*, we mean a system that allows one to program hardware or software computer systems in terms of some of the concepts developed <by agent theorists. At the very least, we expect such a language to include some structure corresponding to an agent. However, we might also expect to see some other attributes of agency (beliefs, goals, or other mentalistic notions) used to program agents. Some of the languages we consider below embody this strong notion of agency; others do not. However, all have properties that make them interesting from the point of view of this review.

Concurrent Object Languages

Concurrent object languages are in many respects the ancestors of agent languages. The notion of a self-contained concurrently executing object, with some internal state that is not directly accessible to the outside world, responding to messages from other such objects, is very close to the concept of an agent as we have defined it. The earliest concurrent object framework was Hewitt's Actor model (Hewitt, 1977; Agha, 1986); another well-known example is the ABCL system (Yonezawa, 1990). For a discussion on the relationship between agents and concurrent object programming, see (Gasser and Briot, 1992).

Shoham — agent-oriented programming

Yoav Shoham has proposed a 'new programming paradigm, based on a societal view of computation' (Shoham, 1990, p4),(Shoham, 1993). The key idea that informs this *agent-oriented programming* (AOP) paradigm is that of directly programming agents in terms of the mentalistic, intentional notions that agent theorists have developed to represent the properties of agents. The motivation behind such a proposal is that, as we observed in section 2, humans use the intentional stance as an *abstraction* mechanism for representing the properties of complex systems. In the same way that we use the intentional stance to describe humans, it might be useful to use the intentional stance to program machines.

Shoham proposes that a fully developed AOP system will have three components:

- a logical system for defining the mental state of agents;
- an interpreted programming language for programming agents;
- an 'agentification' process, for compiling agent programs into low-level executable systems.

At the time of writing, Shoham has only published results on the first two components. (In (Shoham, 1990, p12) he wrote that 'the third is still somewhat mysterious to me', though later in the paper he indicated that he was thinking along the lines of Rosenschein and Kaelbling's situated automata paradigm (Rosenschein and Kaelbling, 1986).) Shoham's first attempt at an AOP language was the AGENT0 system. The logical component of this system is a quantified multi-modal logic, allowing direct reference to time. No semantics are given, but the logic appears to be based on (Thomas et al., 1991). The logic contains three modalities: belief, commitment and ability. The following is an acceptable formula of the logic, illustrating it's key properties:

$$CAN_a^5\ open(door)^8 \Rightarrow B_b^5\ CAN_a^5\ open(door)^8.$$

This formula is read: 'if at time 5 agent a can ensure that the door is open at time 8, then at time 5 agent b believes that at time 5 agent a can ensure that the door is open at time 8'.

Corresponding to the logic is the AGENT0 programming language. In this language, an agent is specified in terms of a set of capabilities (things the agent can do), a set of initial beliefs and commitments, and a set of *commitment rules*. The key component, which determines how the agent acts, is the commitment rule set. Each commitment rule contains a *message condition*, a *mental condition*, and an action. In order to determine whether such a rule fires, the message condition is matched against the messages the agent has received; the mental condition is matched against the beliefs of the agent. If the rule fires, then the agent becomes committed to the action. Actions may be *private*, corresponding to an internally executed subroutine, or *communicative*, i.e., sending messages. Messages are constrained to be one of three types: 'requests' or 'unrequests' to perform or refrain from actions, and 'inform' messages, which pass on information — Shoham indicates that he took his inspiration for these message types from speech act theory (Searle, 1969; Cohen and Perrault, 1979). Request and unrequest messages typically result in the agent's commitments being modified; inform messages result in a change to the agent's beliefs.

Thomas — PLACA

AGENT0 was only ever intended as a prototype, to illustrate the principles of AOP. A more refined implementation was developed by Thomas, for her 1993 doctoral thesis (Thomas, 1993). Her Planning Communicating Agents (PLACA) language was intended to address one severe drawback to AGENT0: the inability of agents to plan, and communicate requests for action via high-level goals. Agents in PLACA are programmed in much the same way as in AGENT0, in terms of *mental change* rules. The logical component of PLACA is similar to AGENT0's, but includes operators for planning to do actions and achieve goals. The semantics of the logic and its properties are examined in detail. However, PLACA is not at the 'production' stage; it is an experimental language.

Fisher — Concurrent METATEM

One drawback with both AGENT0 and PLACA is that the relationship between the logic and interpreted programming language is only loosely defined: in neither case can the programming language be said to truly *execute* the associated logic. The Concurrent METATEM language developed by Fisher can make a stronger claim in this respect (Fisher, 1994). A Concurrent METATEM system contains a number of concurrently executing agents, each of which is able to communicate with its peers via asynchronous broadcast message passing. Each agent is programmed by giving it a temporal logic specification of the behaviour that it is intended the agent should exhibit. An agent's specification is executed directly to generate its behaviour. Execution of the agent program corresponds to iteratively building a logical model for the temporal agent specification. It is possible to prove that the procedure used to execute an agent specification is correct, in that if it is possible to satisfy the specification, then the agent will do so (Barringer et al., 1989).

The logical semantics of Concurrent METATEM are closely related to the semantics of temporal logic itself. This means that, amongst other things, the specification and verification of Concurrent METATEM systems is a realistic proposition (Fisher and Wooldridge, 1993). At the time of writing, only prototype implementations of the language are available; full implementations are expected soon.

The IMAGINE Project — APRIL and MAIL

APRIL (McCabe and Clark, 1995) and MAIL (Haugeneder et al., 1994) are two languages for developing multi-agent applications that were developed as part of the ESPRIT project IMAGINE (Haugeneder, 1994). The two languages are intended to fulfill quite different roles. APRIL was designed to provide the core features required to realise most agent architectures and systems. Thus APRIL provides facilities for multi-tasking (via processes, which are treated as first-class objects, and a UNIX-like *fork* facility), communication (with powerful message-passing facilities supporting network-transparent agent-to-agent links); and pattern matching and symbolic processing capabilities. The generality of APRIL comes at the expense of powerful abstractions — an APRIL system builder must implement an agent or system architecture from scratch using APRIL's primitives. In contrast, the MAIL language provides a rich collection of pre-defined abstractions, including plans and multi-agent plans. APRIL was originally envisaged as the implementation language for MAIL. The MAIL system has been used to implement several prototype multi-agent systems, including an urban traffic management scenario (Haugeneder and Steiner, 1994).

General Magic, Inc. — TELESCRIPT

TELESCRIPT is a language-based environment for constructing agent societies that has been developed by General Magic, Inc.: it is perhaps the first commercial agent language.

TELESCRIPT technology is the name given by General Magic to a family of concepts and techniques they have developed to underpin their products. There are two key concepts in TELESCRIPT technology: *places* and *agents*. Places are virtual locations that are occupied by agents. Agents are the providers and consumers of goods in the *electronic marketplace* applications that TELESCRIPT was developed to support. Agents are software processes, and are mobile: they are able to move from one place to another, in which case their program and state are encoded and transmitted across a network to another place, where execution recommences. Agents are able to communicate with one-another: if they occupy different places, then they can connect across a network, in much the standard way; if they occupy the same location, then they can *meet* one another.

Four components have been developed by General Magic to support TELESCRIPT technology. The first is the TELESCRIPT language. This language 'is designed for carrying out complex communication tasks: navigation, transportation, authentication, access control, and so on' (White, 1994, p17). The second component is the TELESCRIPT engine. An engine acts as an interpreter for the TELESCRIPT language, maintains places, schedules agents for execution, manages communication and agent transport, and finally, provides an interface with other applications. The third component is the TELESCRIPT protocol set. These protocols deal primarily with the encoding and decoding of agents, to support transport between places. The final component is a set of software tools to support the development of TELESCRIPT applications.

Connah and Wavish — ABLE

A group at Philips research labs in the UK have developed an *Agent Behaviour Language*, (ABLE), in which agents are programmed in terms of simple, rule-like *licences* (Connah and Wavish, 1990; Wavish, 1992). Licences may include some representation of time (though the language is not based on any kind of temporal logic): they loosely resemble behaviours in the subsumption architecture (see above). ABLE can be compiled down to a simple digital machine, realised in the 'C' programming language. The idea is similar to situated automata, though there appears to be no equivalent theoretical foundation. The result of the compilation process is a very fast implementation, which has been used to control a Compact Disk-Interactive (CD-I) application. ABLE has recently been extended to a version called Real-Time ABLE (RTA) (Wavish and Graham, 1995).

4.a Discussion

The emergence of various language-based software tools for building agent applications is clearly an important development for the wider acceptance and use of agent technology. The release of TELESCRIPT, a commercial agent language (albeit one that does not embody the strong notion of agency discussed in this paper) is particularly important, as it potentially makes agent technology available to a user base that is industrially (rather than academically) oriented.

While the development of various languages for agent-based applications is of undoubted importance, it is worth noting that all of the academically produced languages mentioned above are in some sense prototypes. Each was designed either to illustrate or examine some set of principles, and these languages were not, therefore, intended as production tools. Work is thus needed, both to make the languages more robust and usable, and to investigate the usefulness of the concepts that underpin them. As with architectures, work is also needed to investigate the kinds of domain for which the different languages are appropriate.

Finally, we turn to the relationship between an agent language and the corresponding theories that we discussed in section 2. As with architectures, it is possible to divide agent languages into various different categories. Thus AGENT0, PLACA, Concurrent METATEM, APRIL, and MAIL are deliberative languages, as they are all based on traditional symbolic AI techniques. ABLE, on the other hand, is a purely *reactive* language. With AGENT0 and PLACA, there is a clear (if informal) relationship between the programming language and the logical theory the language is intended to realise. In both cases, the programming language represents a subset of the corresponding logic, which can be interpreted directly. However, the relationship between logic and language is not formally defined. Like these two languages, Concurrent METATEM is intended to correspond to a logical theory. But the relationship between Concurrent METATEM and the corresponding logic is much more closely defined, as this language is intended to be a directly executable version of the logic. Agents in Concurrent METATEM, however, are not defined in terms of mentalistic constructs. For a discussion on the relationship between Concurrent METATEM and AGENT0-like languages, see (Fisher, 1995).

4.b Further Reading

A recent collection of papers on concurrent object systems is (Agha et al., 1993). Various languages have been proposed that marry aspects of object-based systems with aspects of Shoham's agent-oriented proposal. Two examples are AGENTSPEAK and DAISY. AGENTSPEAK is loosely based on the PRS agent architecture, and incorporates aspects of concurrent-object technology (Weerasooriya et al., 1995). In contrast, DAISY is based on the concurrent-object language CUBL (Adorni and Poggi, 1993), and incorporates aspects of the agent-oriented proposal (Poggi, 1995).

Other languages of interest include OZ (Henz et al., 1993) and IC PROLOG II (Chu, 1993). The latter, as its name suggests, is an extension of PROLOG, which includes multiple-threads, high-level communication primitives, and some object-oriented features.

5 Applications

Although this article is not intended primarily as an applications review, it is nevertheless worth pausing to examine some of the current and potential applications of agent technology.

Cooperative Problem Solving and Distributed AI

As we observed in section 1, there has been a marked flowering of interest in agent technology since the mid-1980s. This interest is in part due to the upsurge of interest in Distributed AI. Although DAI encompasses most of the issues we have discussed in this paper, it should be stressed that the classical emphasis in DAI has been on *macro* phenomena (the *social* level), rather than the *micro* phenomena (the *agent* level) that we have been concerned with in this paper. DAI thus looks at such issues as how a group of agents can be made to cooperate in order to efficiently solve problems, and how the activities of such a group can be efficiently coordinated. DAI researchers have applied agent technology in a variety of areas. Example applications include power systems management (Wittig, 1992; Varga et al., 1994), air-traffic control (Steeb et al., 1988), particle accelerator control (Jennings et al., 1993), intelligent document retrieval (Mukhopadhyay et al., 1986), patient care (Huang et al., 1995), telecommunications network management (Weihmayer and Velthuijsen, 1994), spacecraft control (Schwuttke and Quan, 1993), computer integrated manufacturing (Parunak, 1995), concurrent engineering (Cutkosky et al., 1993), transportation management (Fischer et al., 1993), job shop scheduling (Morley and Schelberg, 1993), and steel coil processing control (Mori et al., 1988). The classic reference to DAI is (Bond and Gasser, 1988), which includes both a comprehensive review article and a collection of significant papers from the field; a more recent review article is (Chaib-draa et al., 1992).

Interface Agents

Maes defines interface agents as:

> '[C]omputer programs that employ artificial intelligence techniques in order to provide assistance to a user dealing with a particular application. … The metaphor is that of a *personal assistant* who is *collaborating with the user* in the same work environment.' (Maes, 1994b, p71)

There are many interface agent prototype applications: for example, the NEWT system is an USENET news filter, (along the lines mentioned in the second scenario that introduced this article) (Maes, 1994a, pp38–39). A NEWT agent is trained by giving it a series of examples, illustrating articles that the user would and would not choose to read. The agent then begins to make suggestions to the user, and is given feedback on its suggestions. NEWT agents are not intended to remove human choice, but to represent an extension of the human's wishes: the aim is for the agent to be able to bring to the attention of the user articles of the type that the user has shown a consistent interest in. Similar ideas have been proposed by McGregor, who imagines *prescient agents* — intelligent administrative assistants, that predict our actions, and carry out routine or repetetive administrative procedures on our behalf (McGregor, 1992).

There is much related work being done by the computer supported cooperative work (CSCW) community. CSCW is informally defined by Baecker to be 'computer assisted coordinated activity such as problem solving and communication carried out by a group of collaborating individuals' (Baecker, 1993, p1). The primary emphasis of CSCW is on the development of (hardware and) software tools to support collaborative human work — the term *groupware* has been coined to describe such tools. Various authors have proposed the use of agent technology in groupware. For example, in his *participant systems* proposal, Chang suggests systems in which humans collaborate with not only other humans, but also with artificial agents (Chang, 1987). We refer the interested reader to the collection of papers edited by Baecker (Baecker, 1993) and the article by Greif (Greif, 1994) for more details on CSCW.

Information Agents and Cooperative Information Systems

An *information agent* is an agent that has access to at least one, and potentially many information sources, and is able to collate and manipulate information obtained from these sources in order to answer queries posed by users and other information agents (the network of inter-operating information sources are often referred to as intelligent and cooperative information systems (Papazoglou et al., 1992)). The information sources may be of many types, including, for example, traditional databases as well as other information agents. Finding a solution to a query might involve an agent accessing information sources over a network. A typical scenario is that of a user who has heard about somebody at Stanford who has proposed something called agent-oriented programming. The agent is asked to investigate, and, after a careful search of various FTP sites, returns with an appropriate technical report, as well as the name and contact details of the researcher involved. A number of studies have been made of information agents, including a theoretical study of how agents are able to incorporate information from different sources (Levy et al., 1994; Gruber, 1991), as well a prototype system called IRA (information retrieval agent) that is able to search for loosely specificed articles from a range of document repositories (Voorhees, 1994). Another important system in this area is called Carnot (Huhns et al., 1992), which allows pre-existing and heterogeneous database systems to work together to answer queries that are outside the scope of any of the individual databases.

Believable Agents

There is obvious potential for marrying agent technology with that of the cinema, computer games, and virtual reality. The Oz project[6] was initiated to develop:

> '… artistically interesting, highly interactive, simulated worlds … to give users the experience of living in (not merely watching) dramatically rich worlds that include moderately competent, emotional agents.' (Bates et al., 1992b, p1)

In order to construct such simulated worlds, one must first develop *believable agents*: agents that 'provide the illusion of life, thus permitting the audience's suspension of disbelief' (Bates, 1994, p122). A key component of such agents is *emotion*: agents should not be represented in a computer game or animated film as the flat, featureless characters that appear in current computer games. They need to show emotions; to act and react in a way that resonates in tune with our empathy and understanding of human behaviour. The Oz group have investigated various architectures for emotion (Bates et al., 1992a), and have developed at least one prototype implementation of their ideas (Bates, 1994).

6 Concluding Remarks

This paper has reviewed the main concepts and issues associated with the theory and practice of intelligent agents. It has drawn together a very wide range of material, and has hopefully

[6]Not to be confused with the OZ programming language (Henz et al., 1993).

provided an insight into what an agent is, how the notion of an agent can be formalised, how appropriate agent architectures can be designed and implemented, how agents can be programmed, and the types of applications for which agent-based solutions have been proposed. The subject matter of this review is important because it is increasingly felt, both within academia and industry, that intelligent agents will be a key technology as computing systems become ever more distributed, interconnected, and open. In such environments, the ability of agents to autonomously plan and pursue their actions and goals, to cooperate, coordinate, and negotiate with others, and to respond flexibly and intelligently to dynamic and unpredictable situations will lead to significant improvements in the quality and sophistication of the software systems that can be conceived and implemented, and the application areas and problems which can be addressed.

Acknowledgements

Much of this paper was adapted from the first author's 1992 PhD thesis (Wooldridge, 1992), and as such this work was supported by the UK Science and Engineering Research Council (now the EPSRC). We are grateful to those people that read and commented on earlier drafts of this article, and in particular to the participants of the 1994 workshop on agent theories, architectures, and languages for their encouragement, enthusiasm, and helpful feedback. Finally, we would like to thank the referees of this paper for their perceptive and helpful comments.

References

Adorni, G. and Poggi, A. (1993). An object-oriented language for distributed artificial intelligence. *International Journal of Man-Machine Studies*, 38:435–453.

Agha, G. (1986). *ACTORS: A Model of Concurrent Computation in Distributed Systems*. The MIT Press: Cambridge, MA.

Agha, G., Wegner, P., and Yonezawa, A., editors (1993). *Research Directions in Concurrent Object-Oriented Programming*. The MIT Press: Cambridge, MA.

Agre, P. and Chapman, D. (1987). PENGI: An implementation of a theory of activity. In *Proceedings of the Sixth National Conference on Artificial Intelligence (AAAI-87)*, pages 268–272, Seattle, WA.

Allen, J. F. (1984). Towards a general theory of action and time. *Artificial Intelligence*, 23(2):123–154.

Allen, J. F., Hendler, J., and Tate, A., editors (1990). *Readings in Planning*. Morgan Kaufmann Publishers: San Mateo, CA.

Allen, J. F., Kautz, H., Pelavin, R., and Tenenberg, J. (1991). *Reasoning About Plans*. Morgan Kaufmann Publishers: San Mateo, CA.

Ambros-Ingerson, J. and Steel, S. (1988). Integrating planning, execution and monitoring. In *Proceedings of the Seventh National Conference on Artificial Intelligence (AAAI-88)*, pages 83–88, St. Paul, MN.

Austin, J. L. (1962). *How to Do Things With Words*. Oxford University Press: Oxford, England.

Aylett, R. and Eustace, D. (1994). Multiple cooperating robots — combining planning and behaviours. In Deen, S. M., editor, *Proceedings of the 1993 Workshop on Cooperating Knowledge Based Systems (CKBS-93)*, pages 3–11. DAKE Centre, University of Keele, UK.

Baecker, R. M., editor (1993). *Readings in Groupware and Computer-Supported Cooperative Work*. Morgan Kaufmann Publishers: San Mateo, CA.

Barringer, H., Fisher, M., Gabbay, D., Gough, G., and Owens, R. (1989). 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: Heidelberg, Germany.

Barwise, J. and Perry, J. (1983). *Situations and Attitudes*. The MIT Press: Cambridge, MA.

Bates, J. (1994). The role of emotion in believable agents. *Communications of the ACM*, 37(7):122–125.

Bates, J., Bryan Loyall, A., and Scott Reilly, W. (1992a). An architecture for action, emotion, and social behaviour. Technical Report CMU–CS–92–144, School of Computer Science, Carnegie-Mellon University, Pittsburgh, PA.

Bates, J., Bryan Loyall, A., and Scott Reilly, W. (1992b). Integrating reactivity, goals, and emotion in a broad agent. Technical Report CMU–CS–92–142, School of Computer Science, Carnegie-Mellon University, Pittsburgh, PA.

Bell, J. (1995). Changing attitudes. In Wooldridge, M. and Jennings, N. R., editors, *Intelligent Agents: Theories, Architectures, and Languages (LNAI Volume 890)*, pages 40–55. Springer-Verlag: Heidelberg, Germany.

Belnap, N. (1991). Backwards and forwards in the modal logic of agency. *Philosophy and Phenomenological Research*, LI(4):777–807.

Belnap, N. and Perloff, M. (1988). Seeing to it that: a canonical form for agentives. *Theoria*, 54:175–199.

Bond, A. H. and Gasser, L., editors (1988). *Readings in Distributed Artificial Intelligence*. Morgan Kaufmann Publishers: San Mateo, CA.

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

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

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

Brooks, R. A. (1986). A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation*, 2(1):14–23.

Brooks, R. A. (1990). Elephants don't play chess. In Maes, P., editor, *Designing Autonomous Agents*, pages 3–15. The MIT Press: Cambridge, MA.

Brooks, R. A. (1991a). Intelligence without reason. In *Proceedings of the Twelfth International Joint Conference on Artificial Intelligence (IJCAI-91)*, pages 569–595, Sydney, Australia.

Brooks, R. A. (1991b). Intelligence without representation. *Artificial Intelligence*, 47:139–159.

Burmeister, B. and Sundermeyer, K. (1992). Cooperative problem solving guided by intentions and perception. In Werner, E. and Demazeau, Y., editors, *Decentralized AI 3 — Proceedings of the Third European Workshop on Modelling Autonomous Agents and Multi-Agent Worlds (MAAMAW-91)*, pages 77–92. Elsevier Science Publishers B.V.: Amsterdam, The Netherlands.

Bussmann, S. and Demazeau, Y. (1994). An agent model combining reactive and cognitive capabilities. In *Proceedings of the IEEE International Conference on Intelligent Robots and Systems (IROS-94)*, Munich, Germany.

Castelfranchi, C. (1990). Social power. In Demazeau, Y. and Müller, J.-P., editors, *Decentralized AI — Proceedings of the First European Workshop on Modelling Autonomous Agents in Multi-Agent Worlds (MAAMAW-89)*, pages 49–62. Elsevier Science Publishers B.V.: Amsterdam, The Netherlands.

Castelfranchi, C. (1995). Guarantees for autonomy in cognitive agent architecture. In Wooldridge, M. and Jennings, N. R., editors, *Intelligent Agents: Theories, Architectures, and Languages (LNAI Volume 890)*, pages 56–70. Springer-Verlag: Heidelberg, Germany.

Castelfranchi, C., Miceli, M., and Cesta, A. (1992). Dependence relations among autonomous agents. In Werner, E. and Demazeau, Y., editors, *Decentralized AI 3 — Proceedings of the Third European Workshop on Modelling Autonomous Agents and Multi-Agent*

Worlds (MAAMAW-91), pages 215–231. Elsevier Science Publishers B.V.: Amsterdam, The Netherlands.

Catach, L. (1988). Normal multimodal logics. In *Proceedings of the Seventh National Conference on Artificial Intelligence (AAAI-88)*, pages 491–495, St. Paul, MN.

Chaib-draa, B., Moulin, B., Mandiau, R., and Millot, P. (1992). Trends in distributed artificial intelligence. *Artificial Intelligence Review*, 6:35–66.

Chang, E. (1987). Participant systems. In Huhns, M., editor, *Distributed Artificial Intelligence*, pages 311–340. Pitman Publishing: London and Morgan Kaufmann: San Mateo, CA.

Chapman, D. (1987). Planning for conjunctive goals. *Artificial Intelligence*, 32:333–378.

Chapman, D. and Agre, P. (1986). Abstract reasoning as emergent from concrete activity. In Georgeff, M. P. and Lansky, A. L., editors, *Reasoning About Actions & Plans — Proceedings of the 1986 Workshop*, pages 411–424. Morgan Kaufmann Publishers: San Mateo, CA.

Chellas, B. (1980). *Modal Logic: An Introduction*. Cambridge University Press: Cambridge, England.

Chu, D. (1993). I.C. PROLOG II: A language for implementing multi-agent systems. In Deen, S. M., editor, *Proceedings of the 1992 Workshop on Cooperating Knowledge Based Systems (CKBS-92)*, pages 61–74. DAKE Centre, University of Keele, UK.

Cohen, P. R., Greenberg, M. L., Hart, D. M., and Howe, A. E. (1989). Trial by fire: Understanding the design requirements for agents in complex environments. *AI Magazine*, 10(3):32–48.

Cohen, P. R. and Levesque, H. J. (1990a). Intention is choice with commitment. *Artificial Intelligence*, 42:213–261.

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

Cohen, P. R. and Perrault, C. R. (1979). Elements of a plan based theory of speech acts. *Cognitive Science*, 3:177–212.

Connah, D. and Wavish, P. (1990). An experiment in cooperation. In Demazeau, Y. and Müller, J.-P., editors, *Decentralized AI — Proceedings of the First European Workshop on Modelling Autonomous Agents in Multi-Agent Worlds (MAAMAW-89)*, pages 197–214. Elsevier Science Publishers B.V.: Amsterdam, The Netherlands.

Cutkosky, M. R., Engelmore, R. S., Fikes, R. E., Genesereth, M. . R., Gruber, T., Mark, W. S., Tenenbaum, J. M., and Weber, J. C. (1993). PACT: An experiment in integrating concurrent engineering systems. *IEEE Computer*, 26(1):28–37.

Davies, N. J. (1993). *Truth, Modality, and Action*. PhD thesis, Department of Computer Science, University of Essex, Colchester, UK.

Dean, T. L. and Wellman, M. P. (1991). *Planning and Control*. Morgan Kaufmann Publishers: San Mateo, CA.

Dennett, D. C. (1978). *Brainstorms*. The MIT Press: Cambridge, MA.

Dennett, D. C. (1987). *The Intentional Stance*. The MIT Press: Cambridge, MA.

des Rivieres, J. and Levesque, H. J. (1986). The consistency of syntactical treatments of knowledge. In Halpern, J. Y., editor, *Proceedings of the 1986 Conference on Theoretical Aspects of Reasoning About Knowledge*, pages 115–130. Morgan Kaufmann Publishers: San Mateo, CA.

Devlin, K. (1991). *Logic and Information*. Cambridge University Press: Cambridge, England.

Dongha, P. (1995). Toward a formal model of commitment for resource-bounded agents. In Wooldridge, M. and Jennings, N. R., editors, *Intelligent Agents: Theories, Architectures, and Languages (LNAI Volume 890)*, pages 86–101. Springer-Verlag: Heidelberg, Germany.

Downs, J. and Reichgelt, H. (1991). Integrating classical and reactive planning within an architecture for autonomous agents. In Hertzberg, J., editor, *European Workshop on Planning (LNAI Volume 522)*, pages 13–26.

Doyle, J., Shoham, Y., and Wellman, M. P. (1991). A logic of relative desire. In Ras, Z. W. and Zemankova, M., editors, *Methodologies for Intelligent Systems — Sixth International Symposium, ISMIS-91 (LNAI Volume 542)*. Springer-Verlag: Heidelberg, Germany.

Emerson, E. A. (1990). Temporal and modal logic. In van Leeuwen, J., editor, *Handbook of Theoretical Computer Science*, pages 996–1072. Elsevier Science Publishers B.V.: Amsterdam, The Netherlands.

Emerson, E. A. and Halpern, J. Y. (1986). ‘Sometimes’ and ‘not never’ revisited: on branching time versus linear time temporal logic. *Journal of the ACM*, 33(1):151–178.

Etzioni, O., Lesh, N., and Segal, R. (1994). Building softbots for UNIX. In Etzioni, O., editor, *Software Agents — Papers from the 1994 Spring Symposium (Technical Report SS–94–03)*, pages 9–16. AAAI Press.

Fagin, R. and Halpern, J. Y. (1985). Belief, awareness, and limited reasoning. In *Proceedings of the Ninth International Joint Conference on Artificial Intelligence (IJCAI-85)*, pages 480–490, Los Angeles, CA.

Fagin, R., Halpern, J. Y., and Vardi, M. Y. (1992). What can machines know? on the properties of knowledge in distributed systems. *Journal of the ACM*, 39(2):328–376.

Ferguson, I. A. (1992a). *TouringMachines: An Architecture for Dynamic, Rational, Mobile Agents*. PhD thesis, Clare Hall, University of Cambridge, UK. (Also available as Technical Report No. 273, University of Cambridge Computer Laboratory).

Ferguson, I. A. (1992b). Towards an architecture for adaptive, rational, mobile agents. In Werner, E. and Demazeau, Y., editors, *Decentralized AI 3 — Proceedings of the Third European Workshop on Modelling Autonomous Agents and Multi-Agent Worlds (MAAMAW-91)*, pages 249–262. Elsevier Science Publishers B.V.: Amsterdam, The Netherlands.

Fikes, R. E. and Nilsson, N. (1971). STRIPS: A new approach to the application of theorem proving to problem solving. *Artificial Intelligence*, 5(2):189–208.

Firby, J. A. (1987). An investigation into reactive planning in complex domains. In *Proceedings of the Tenth International Joint Conference on Artificial Intelligence (IJCAI-87)*, pages 202–206, Milan, Italy.

Fischer, K., Kuhn, N., Müller, H. J., Müller, J. P., and Pischel, M. (1993). Sophisticated and distributed: The transportation domain. In *Proceedings of the Fifth European Workshop on Modelling Autonomous Agents and Multi-Agent Worlds (MAAMAW-93)*, Neuchatel, Switzerland.

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

Fisher, M. (1995). Representing and executing agent-based systems. In Wooldridge, M. and Jennings, N. R., editors, *Intelligent Agents: Theories, Architectures, and Languages (LNAI Volume 890)*, pages 307–323. Springer-Verlag: Heidelberg, Germany.

Fisher, M. and Wooldridge, M. (1993). Specifying and verifying distributed intelligent systems. In Filgueiras, M. and Damas, L., editors, *Progress in Artificial Intelligence — Sixth Portuguese Conference on Artificial Intelligence (LNAI Volume 727)*, pages 13–28. Springer-Verlag: Heidelberg, Germany.

Galliers, J. R. (1988a). 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.

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

Gasser, L. (1991). Social conceptions of knowledge and action: DAI foundations and open systems semantics. *Artificial Intelligence*, 47:107–138.

Gasser, L., Braganza, C., and Hermann, N. (1987). MACE: A flexible testbed for distributed AI research. In Huhns, M., editor, *Distributed Artificial Intelligence*, pages 119–152. Pitman Publishing: London and Morgan Kaufmann: San Mateo, CA.

Gasser, L. and Briot, J. P. (1992). Object-based concurrent programming and DAI. In *Distributed Artificial Intelligence: Theory and Praxis*, pages 81–108. Kluwer Academic Publishers: Boston, MA.

Geissler, C. and Konolige, K. (1986). A resolution method for quantified modal logics of knowledge and belief. In Halpern, J. Y., editor, *Proceedings of the 1986 Conference on Theoretical Aspects of Reasoning About Knowledge*, pages 309–324. Morgan Kaufmann Publishers: San Mateo, CA.

Genesereth, M. R. and Ketchpel, S. P. (1994). Software agents. *Communications of the ACM*, 37(7):48–53.

Genesereth, M. R. and Nilsson, N. (1987). *Logical Foundations of Artificial Intelligence*. Morgan Kaufmann Publishers: San Mateo, CA.

Georgeff, M. P. (1987). Planning. *Annual Review of Computer Science*, 2:359–400.

Georgeff, M. P. and Ingrand, F. F. (1989). Decision-making in an embedded reasoning system. In *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence (IJCAI-89)*, pages 972–978, Detroit, MI.

Georgeff, M. P. and Lansky, A. L., editors (1986). *Reasoning About Actions & Plans — Proceedings of the 1986 Workshop*. Morgan Kaufmann Publishers: San Mateo, CA.

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

Ginsberg, M. (1993). *Essentials of Artificial Intelligence*. Morgan Kaufmann Publishers: San Mateo, CA.

Gmytrasiewicz, P. and Durfee, E. H. (1993). Elements of a utilitarian theory of knowledge and action. In *Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence (IJCAI-93)*, pages 396–402, Chambéry, France.

Goldblatt, R. (1987). *Logics of Time and Computation*. Centre for the Study of Language and Information — Lecture Notes Series. (Distributed by Chicago University Press).

Goldman, R. P. and Lang, R. R. (1991). Intentions in time. Technical Report TUTR 93–101, Tulane University.

Goodwin, R. (1993). Formalizing properties of agents. Technical Report CMU–CS–93–159, School of Computer Science, Carnegie-Mellon University, Pittsburgh, PA.

Greif, I. (1994). Desktop agents in group-enabled products. *Communications of the ACM*, 37(7):100–105.

Grosz, B. J. and Sidner, C. L. (1990). Plans for discourse. In Cohen, P. R., Morgan, J., and Pollack, M. E., editors, *Intentions in Communication*, pages 417–444. The MIT Press: Cambridge, MA.

Gruber, T. R. (1991). The role of common ontology in achieving sharable, reusable knowledge bases. In Fikes, R. and Sandewall, E., editors, *Proceedings of Knowledge Representation and Reasoning (KR&R-91)*. Morgan Kaufmann Publishers: San Mateo, CA.

Guha, R. V. and Lenat, D. B. (1994). Enabling agents to work together. *Communications of the ACM*, 37(7):127–142.

Haas, A. (1986). A syntactic theory of belief and knowledge. *Artificial Intelligence*, 28(3):245–292.

Haddadi, A. (1994). A hybrid architecture for multi-agent systems. In Deen, S. M., editor, *Proceedings of the 1993 Workshop on Cooperating Knowledge Based Systems (CKBS-93)*, pages 13–26, DAKE Centre, University of Keele, UK.

Halpern, J. Y. (1986). Reasoning about knowledge: An overview. In Halpern, J. Y., editor, *Proceedings of the 1986 Conference on Theoretical Aspects of Reasoning About Knowledge*, pages 1–18. Morgan Kaufmann Publishers: San Mateo, CA.

Halpern, J. Y. (1987). Using reasoning about knowledge to analyze distributed systems. *Annual Review of Computer Science*, 2:37–68.

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

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

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

Haugeneder, H. (1994). IMAGINE final project report.

Haugeneder, H. and Steiner, D. (1994). A multi-agent approach to cooperation in urban traffic. In Deen, S. M., editor, *Proceedings of the 1993 Workshop on Cooperating Knowledge Based Systems (CKBS-93)*, pages 83–98. DAKE Centre, University of Keele, UK.

Haugeneder, H., Steiner, D., and McCabe, F. G. (1994). IMAGINE: A framework for building multi-agent systems. In Deen, S. M., editor, *Proceedings of the 1994 International Working Conference on Cooperating Knowledge Based Systems (CKBS-94)*, pages 31–64, DAKE Centre, University of Keele, UK.

Hayes-Roth, B. (1990). Architectural foundations for real-time performance in intelligent agents. *The Journal of Real-Time Systems*, 2:99–125.

Hendler, J., editor (1992). *Artificial Intelligence Planning: Proceedings of the First International Conference*. Morgan Kaufmann Publishers: San Mateo, CA.

Henz, M., Smolka, G., and Wuertz, J. (1993). Oz — a programming language for multi-agent systems. In *Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence (IJCAI-93)*, pages 404–409, Chambéry, France.

Hewitt, C. (1977). Viewing control structures as patterns of passing messages. *Artificial Intelligence*, 8(3):323–364.

Hintikka, J. (1962). *Knowledge and Belief*. Cornell University Press: Ithaca, NY.

Houlder, V. (1994). Special agents. In *Financial Times*, 15 August 1994, page 12.

Huang, J., Jennings, N. R., and Fox, J. (1995). An agent architecture for distributed medical care. In Wooldridge, M. and Jennings, N. R., editors, *Intelligent Agents: Theories, Architectures, and Languages (LNAI Volume 890)*, pages 219–232. Springer-Verlag: Heidelberg, Germany.

Hughes, G. E. and Cresswell, M. J. (1968). *Introduction to Modal Logic*. Methuen and Co., Ltd.

Huhns, M. N., Jacobs, N., Ksiezyk, T., Shen, W. M., Singh, M. P., and Cannata, P. E. (1992). Integrating enterprise information models in Carnot. In *Proceedings of the International Conference on Intelligent and Cooperative Information Systems*, pages 32–42, Rotterdam, The Netherlands.

Israel, D. J. (1993). The role(s) of logic in artificial intelligence. In Gabbay, D. M., Hogger, C. J., and Robinson, J. A., editors, *Handbook of Logic in Artificial Intelligence and Logic Programming*, pages 1–29. Oxford University Press: Oxford, England.

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

Jennings, N. R. (1993a). Commitments and conventions: The foundation of coordination in multi-agent systems. *Knowledge Engineering Review*, 8(3):223–250.

Jennings, N. R. (1993b). Specification and implementation of a belief desire joint-intention architecture for collaborative problem solving. *Journal of Intelligent and Cooperative Information Systems*, 2(3):289–318.

Jennings, N. R. (1995). Controlling cooperative problem solving in industrial multi-agent systems using joint intentions. *Artificial Intelligence*, 74(2). (To appear).

Jennings, N. R., Varga, L. Z., Aarnts, R. P., Fuchs, J., and Skarek, P. (1993). Transforming standalone expert systems into a community of cooperating agents. *International Journal of Engineering Applications of Artificial Intelligence*, 6(4):317–331.

Kaelbling, L. P. (1986). An architecture for intelligent reactive systems. In Georgeff, M. P. and Lansky, A. L., editors, *Reasoning About Actions & Plans — Proceedings of the 1986 Workshop*, pages 395–410. Morgan Kaufmann Publishers: San Mateo, CA.

Kaelbling, L. P. (1991). A situated automata approach to the design of embedded agents. *SIGART Bulletin*, 2(4):85–88.

Kaelbling, L. P. and Rosenschein, S. J. (1990). Action and planning in embedded agents. In Maes, P., editor, *Designing Autonomous Agents*, pages 35–48. The MIT Press: Cambridge, MA.

Kinny, D., Ljungberg, M., Rao, A. S., Sonenberg, E., Tidhar, G., and Werner, E. (1992). Planned team activity. In Castelfranchi, C. and Werner, E., editors, *Artificial Social Systems — Selected Papers from the Fourth European Workshop on Modelling Autonomous Agents and Multi-Agent Worlds, MAAMAW-92 (LNAI Volume 830)*, pages 226–256. Springer-Verlag: Heidelberg, Germany.

Kiss, G. and Reichgelt, H. (1992). Towards a semantics of desires. In Werner, E. and Demazeau, Y., editors, *Decentralized AI 3 — Proceedings of the Third European Workshop on Modelling Autonomous Agents and Multi-Agent Worlds (MAAMAW-91)*, pages 115–128. Elsevier Science Publishers B.V.: Amsterdam, The Netherlands.

Konolige, K. (1982). A first-order formalization of knowledge and action for a multi-agent planning system. In Hayes, J. E., Michie, D., and Pao, Y., editors, *Machine Intelligence 10*, pages 41–72. Ellis Horwood: Chichester, England.

Konolige, K. (1986a). *A Deduction Model of Belief*. Pitman Publishing: London and Morgan Kaufmann: San Mateo, CA.

Konolige, K. (1986b). What awareness isn't: A sentential view of implicit and explicit belief (position paper). In Halpern, J. Y., editor, *Proceedings of the 1986 Conference on Theoretical Aspects of Reasoning About Knowledge*, pages 241–250. Morgan Kaufmann Publishers: San Mateo, CA.

Konolige, K. and Pollack, M. E. (1993). A representationalist theory of intention. In *Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence (IJCAI-93)*, pages 390–395, Chambéry, France.

Kraus, S. and Lehmann, D. (1988). Knowledge, belief and time. *Theoretical Computer Science*, 58:155–174.

Kripke, S. (1963). Semantical analysis of modal logic. *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik*, 9:67–96.

Lakemeyer, G. (1991). A computationally attractive first-order logic of belief. In *JELIA-90: Proceedings of the European Workshop on Logics in AI (LNAI Volume 478)*, pages 333–347. Springer-Verlag: Heidelberg, Germany.

Lespérance, Y. (1989). A formal account of self knowledge and action. In *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence (IJCAI-89)*, pages 868–874, Detroit, MI.

Levesque, H. J. (1984). A logic of implicit and explicit belief. In *Proceedings of the Fourth National Conference on Artificial Intelligence (AAAI-84)*, pages 198–202, Austin, TX.

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

Levy, A. Y., Sagiv, Y., and Srivastava, D. (1994). Towards efficient information gathering agents. In Etzioni, O., editor, *Software Agents — Papers from the 1994 Spring Symposium (Technical Report SS–94–03)*, pages 64–70. AAAI Press.

Mack, D. (1994). A new formal model of belief. In *Proceedings of the Eleventh European Conference on Artificial Intelligence (ECAI-94)*, pages 573–577, Amsterdam, The Netherlands.

Maes, P. (1989). The dynamics of action selection. In *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence (IJCAI-89)*, pages 991–997, Detroit, MI.

Maes, P., editor (1990a). *Designing Autonomous Agents*. The MIT Press: Cambridge, MA.

Maes, P. (1990b). Situated agents can have goals. In Maes, P., editor, *Designing Autonomous Agents*, pages 49–70. The MIT Press: Cambridge, MA.

Maes, P. (1991). The agent network architecture (ANA). *SIGART Bulletin*, 2(4):115–120.

Maes, P. (1994a). Agents that reduce work and information overload. *Communications of the ACM*, 37(7):31–40.

Maes, P. (1994b). Social interface agents: Acquiring competence by learning from users and other agents. In Etzioni, O., editor, *Software Agents — Papers from the 1994 Spring Symposium (Technical Report SS–94–03)*, pages 71–78. AAAI Press.

McCabe, F. G. and Clark, K. L. (1995). `April` — agent process interaction language. In Wooldridge, M. and Jennings, N. R., editors, *Intelligent Agents: Theories, Architectures, and Languages (LNAI Volume 890)*, pages 324–340. Springer-Verlag: Heidelberg, Germany.

McCarthy, J. (1978). Ascribing mental qualities to machines. Technical report, Stanford University AI Lab., Stanford, CA 94305.

McGregor, S. L. (1992). Prescient agents. In Coleman, D., editor, *Proceedings of Groupware-92*, pages 228–230.

Montague, R. (1963). Syntactical treatments of modality, with corollaries on reflexion principles and finite axiomatizations. *Acta Philosophica Fennica*, 16:153–167.

Moore, R. C. (1990). A formal theory of knowledge and action. In Allen, J. F., Hendler, J., and Tate, A., editors, *Readings in Planning*, pages 480–519. Morgan Kaufmann Publishers: San Mateo, CA.

Morgenstern, L. (1987). Knowledge preconditions for actions and plans. In *Proceedings of the Tenth International Joint Conference on Artificial Intelligence (IJCAI-87)*, pages 867–874, Milan, Italy.

Mori, K., Torikoshi, H., Nakai, K., and Masuda, T. (1988). Computer control system for iron and steel plants. *Hitachi Review*, 37(4):251–258.

Morley, R. E. and Schelberg, C. (1993). An analysis of a plant-specific dynamic scheduler. In *Proceedings of the NSF Workshop on Dynamic Scheduling*, Cocoa Beach, Florida.

Mukhopadhyay, U., Stephens, L., and Huhns, M. (1986). An intelligent system for document retrieval in distributed office environments. *Journal of the American Society for Information Science*, 37:123–135.

Müller, J. P. (1994). A conceptual model for agent interaction. In Deen, S. M., editor, *Proceedings of the Second International Working Conference on Cooperating Knowledge Based Systems (CKBS-94)*, pages 213–234, DAKE Centre, University of Keele, UK.

Müller, J. P. and Pischel, M. (1994). Modelling interacting agents in dynamic environments. In *Proceedings of the Eleventh European Conference on Artificial Intelligence (ECAI-94)*, pages 709–713, Amsterdam, The Netherlands.

Müller, J. P., Pischel, M., and Thiel, M. (1995). Modelling reactive behaviour in vertically layered agent architectures. In Wooldridge, M. and Jennings, N. R., editors, *Intelligent Agents: Theories, Architectures, and Languages (LNAI Volume 890)*, pages 261–276. Springer-Verlag: Heidelberg, Germany.

Newell, A. and Simon, H. A. (1976). Computer science as empirical enquiry. *Communications of the ACM*, 19:113–126.

Nilsson, N. J. (1992). Towards agent programs with circuit semantics. Technical Report STAN–CS–92–1412, Computer Science Department, Stanford University, Stanford, CA 94305.

Norman, T. J. and Long, D. (1995). Goal creation in motivated agents. In Wooldridge, M. and Jennings, N. R., editors, *Intelligent Agents: Theories, Architectures, and Languages (LNAI Volume 890)*, pages 277–290. Springer-Verlag: Heidelberg, Germany.

Papazoglou, M. P., Laufman, S. C., and Sellis, T. K. (1992). An organizational framework for cooperating intelligent information systems. *Journal of Intelligent and Cooperative Information Systems*, 1(1):169–202.

Parunak, H. V. D. (1995). Applications of distributed artificial intelligence in industry. In O'Hare, G. M. P. and Jennings, N. R., editors, *Foundations of Distributed AI*. John Wiley & Sons: Chichester, England. (To appear).

Patil, R. S., Fikes, R. E., Patel-Schneider, P. F., McKay, D., Finin, T., Gruber, T., and Neches, R. (1992). The DARPA knowledge sharing effort: Progress report. In Rich, C., Swartout, W., and Nebel, B., editors, *Proceedings of Knowledge Representation and Reasoning (KR&R-92)*, pages 777–788.

Perlis, D. (1985). Languages with self reference I: Foundations. *Artificial Intelligence*, 25:301–322.

Perlis, D. (1988). Languages with self reference II: Knowledge, belief, and modality. *Artificial Intelligence*, 34:179–212.

Perloff, M. (1991). *STIT* and the language of agency. *Synthese*, 86:379–408.

Poggi, A. (1995). DAISY: An object-oriented system for distributed artificial intelligence. In Wooldridge, M. and Jennings, N. R., editors, *Intelligent Agents: Theories, Architectures, and Languages (LNAI Volume 890)*, pages 341–254. Springer-Verlag: Heidelberg, Germany.

Pollack, M. E. and Ringuette, M. (1990). Introducing the Tileworld: Experimentally evaluating agent architectures. In *Proceedings of the Eighth National Conference on Artificial Intelligence (AAAI-90)*, pages 183–189, Boston, MA.

Rao, A. S. and Georgeff, M. P. (1991a). 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.

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

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

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

Rao, A. S. and Georgeff, M. P. (1993). 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.

Reichgelt, H. (1989a). A comparison of first-order and modal logics of time. In Jackson, P., Reichgelt, H., and van Harmelen, F., editors, *Logic Based Knowledge Representation*, pages 143–176. The MIT Press: Cambridge, MA.

Reichgelt, H. (1989b). Logics for reasoning about knowledge and belief. *Knowledge Engineering Review*, 4(2):119–139.

Rosenschein, J. S. and Genesereth, M. R. (1985). Deals among rational agents. In *Proceedings of the Ninth International Joint Conference on Artificial Intelligence (IJCAI-85)*, pages 91–99, Los Angeles, CA.

Rosenschein, S. (1985). Formal theories of knowledge in AI and robotics. *New Generation Computing*, pages 345–357.

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

Russell, S. J. and Wefald, E. (1991). *Do the Right Thing — Studies in Limited Rationality*. The MIT Press: Cambridge, MA.

Sacerdoti, E. (1974). Planning in a hierarchy of abstraction spaces. *Artificial Intelligence*, 5:115–135.

Sacerdoti, E. (1975). The non-linear nature of plans. In *Proceedings of the Fourth International Joint Conference on Artificial Intelligence (IJCAI-75)*, pages 206–214, Stanford, CA.

Sadek, M. D. (1992). A study in the logic of intention. In Rich, C., Swartout, W., and Nebel, B., editors, *Proceedings of Knowledge Representation and Reasoning (KR&R-92)*, pages 462–473.

Sargent, P. (1992). Back to school for a brand new ABC. In *The Guardian*, 12 March 1992, page 28.

Schoppers, M. J. (1987). Universal plans for reactive robots in unpredictable environments. In *Proceedings of the Tenth International Joint Conference on Artificial Intelligence (IJCAI-87)*, pages 1039–1046, Milan, Italy.

Schwuttke, U. M. and Quan, A. G. (1993). Enhancing performance of cooperating agents in real-time diagnostic systems. In *Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence (IJCAI-93)*, pages 332–337, Chambéry, France.

Searle, J. R. (1969). *Speech Acts: An Essay in the Philosophy of Language*. Cambridge University Press: Cambridge, England.

Seel, N. (1989). *Agent Theories and Architectures*. PhD thesis, Surrey University, Guildford, UK.

Segerberg, K. (1989). Bringing it about. *Journal of Philosophical Logic*, 18:327–347.

Shardlow, N. (1990). Action and agency in cognitive science. Master's thesis, Department of Psychlogy, University of Manchester, Oxford Rd., Manchester M13 9PL, UK.

Shoham, Y. (1988). *Reasoning About Change: Time and Causation from the Standpoint of Artificial Intelligence*. The MIT Press: Cambridge, MA.

Shoham, Y. (1989). Time for action: on the relation between time, knowledge and action. In *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence (IJCAI-89)*, pages 954–959, Detroit, MI.

Shoham, Y. (1990). Agent-oriented programming. Technical Report STAN–CS–1335–90, Computer Science Department, Stanford University, Stanford, CA 94305.

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

Singh, M. P. (1990a). Group intentions. In *Proceedings of the Tenth International Workshop on Distributed Artificial Intelligence (IWDAI-90)*.

Singh, M. P. (1990b). Towards a theory of situated know-how. In *Proceedings of the Ninth European Conference on Artificial Intelligence (ECAI-90)*, pages 604–609, Stockholm, Sweden.

Singh, M. P. (1991a). Group ability and structure. In Demazeau, Y. and Müller, J.-P., editors, *Decentralized AI 2 — Proceedings of the Second European Workshop on Modelling Autonomous Agents and Multi-Agent Worlds (MAAMAW-90)*, pages 127–146. Elsevier Science Publishers B.V.: Amsterdam, The Netherlands.

Singh, M. P. (1991b). Towards a formal theory of communication for multi-agent systems. In *Proceedings of the Twelfth International Joint Conference on Artificial Intelligence (IJCAI-91)*, pages 69–74, Sydney, Australia.

Singh, M. P. (1992). 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.

Singh, M. P. (1994). *Multiagent Systems: A Theoretical Framework for Intentions, Know-How, and Communications (LNAI Volume 799)*. Springer-Verlag: Heidelberg, Germany.

Singh, M. P. and Asher, N. M. (1991). Towards a formal theory of intentions. In *Logics in AI — Proceedings of the European Workshop JELIA-90 (LNAI Volume 478)*, pages 472–486. Springer-Verlag: Heidelberg, Germany.

Smith, R. G. (1980). *A Framework for Distributed Problem Solving*. UMI Research Press.

Steeb, R., Cammarata, S., Hayes-Roth, F. A., Thorndyke, P. W., and Wesson, R. B. (1988). Distributed intelligence for air fleet control. In Bond, A. H. and Gasser, L., editors, *Readings in Distributed Artificial Intelligence*, pages 90–101. Morgan Kaufmann Publishers: San Mateo, CA.

Steels, L. (1990). Cooperation between distributed agents through self organization. In Demazeau, Y. and Müller, J.-P., editors, *Decentralized AI — Proceedings of the First European Workshop on Modelling Autonomous Agents in Multi-Agent Worlds (MAAMAW-89)*, pages 175–196. Elsevier Science Publishers B.V.: Amsterdam, The Netherlands.

Thomas, S. R. (1993). *PLACA, an Agent Oriented Programming Language*. PhD thesis, Computer Science Department, Stanford University, Stanford, CA 94305. (Available as technical report STAN–CS–93–1487).

Thomas, S. R., Shoham, Y., Schwartz, A., and Kraus, S. (1991). Preliminary thoughts on an agent description language. *International Journal of Intelligent Systems*, 6:497–508.

Thomason, R. (1980). A note on syntactical treatments of modality. *Synthese*, 44:391–395.

Turner, R. (1990). *Truth and Modality for Knowledge Representation*. Pitman Publishing: London.

Varga, L. Z., Jennings, N. R., and Cockburn, D. (1994). Integrating intelligent systems into a cooperating community for electricity distribution management. *International Journal of Expert Systems with Applications*, 7(4):563–579.

Vere, S. and Bickmore, T. (1990). A basic agent. *Computational Intelligence*, 6:41–60.

Voorhees, E. M. (1994). Software agents for information retrieval. In Etzioni, O., editor, *Software Agents — Papers from the 1994 Spring Symposium (Technical Report SS–94–03)*, pages 126–129. AAAI Press.

Wainer, J. (1994). Yet another semantics of goals and goal priorities. In *Proceedings of the Eleventh European Conference on Artificial Intelligence (ECAI-94)*, pages 269–273, Amsterdam, The Netherlands.

Wavish, P. (1992). Exploiting emergent behaviour in multi-agent systems. In Werner, E. and Demazeau, Y., editors, *Decentralized AI 3 — Proceedings of the Third European Workshop on Modelling Autonomous Agents and Multi-Agent Worlds (MAAMAW-91)*, pages 297–310. Elsevier Science Publishers B.V.: Amsterdam, The Netherlands.

Wavish, P. and Graham, M. (1995). Roles, skills, and behaviour: a situated action approach to organising systems of interacting agents. In Wooldridge, M. and Jennings, N. R., editors, *Intelligent Agents: Theories, Architectures, and Languages (LNAI Volume 890)*, pages 371–385. Springer-Verlag: Heidelberg, Germany.

Weerasooriya, D., Rao, A., and Ramamohanarao, K. (1995). Design of a concurrent agent-oriented language. In Wooldridge, M. and Jennings, N. R., editors, *Intelligent Agents: Theories, Architectures, and Languages (LNAI Volume 890)*, pages 386–402. Springer-Verlag: Heidelberg, Germany.

Weihmayer, R. and Velthuijsen, H. (1994). Application of distributed AI and cooperative problem solving to telecommunications. In Liebowitz, J. and Prereau, D., editors, *AI Approaches to Telecommunications and Network Management*. IOS Press.

Werner, E. (1988). Toward a theory of communication and cooperation for multiagent planning. In Vardi, M. Y., editor, *Proceedings of the Second Conference on Theoretical Aspects of Reasoning About Knowledge*, pages 129–144. Morgan Kaufmann Publishers: San Mateo, CA.

Werner, E. (1989). Cooperating agents: A unified theory of communication and social structure. In Gasser, L. and Huhns, M., editors, *Distributed Artificial Intelligence Volume II*, pages 3–36. Pitman Publishing: London and Morgan Kaufmann: San Mateo, CA.

Werner, E. (1990). What can agents do together: A semantics of co-operative ability. In *Proceedings of the Ninth European Conference on Artificial Intelligence (ECAI-90)*, pages 694–701, Stockholm, Sweden.

Werner, E. (1991). A unified view of information, intention and ability. In Demazeau, Y. and Müller, J.-P., editors, *Decentralized AI 2 — Proceedings of the Second European Workshop on Modelling Autonomous Agents and Multi-Agent Worlds (MAAMAW-90)*, pages 109–126. Elsevier Science Publishers B.V.: Amsterdam, The Netherlands.

White, J. E. (1994). Telescript technology: The foundation for the electronic marketplace. White paper, General Magic, Inc., 2465 Latham Street, Mountain View, CA 94040.

Wilkins, D. (1988). *Practical Planning: Extending the Classical AI Planning Paradigm*. Morgan Kaufmann Publishers: San Mateo, CA.

Wittig, T., editor (1992). *ARCHON: An Architecture for Multi-Agent Systems*. Ellis Horwood: Chichester, England.

Wood, S. (1993). *Planning and Decision Making in Dynamic Domains*. Ellis Horwood: Chichester, England.

Wooldridge, M. (1992). *The Logical Modelling of Computational Multi-Agent Systems*. PhD thesis, Department of Computation, UMIST, Manchester, UK. (Also available as Technical Report MMU–DOC–94–01, Department of Computing, Manchester Metropolitan University, Chester St., Manchester, UK).

Wooldridge, M. (1994). Coherent social action. In *Proceedings of the Eleventh European Conference on Artificial Intelligence (ECAI-94)*, pages 279–283, Amsterdam, The Netherlands.

Wooldridge, M. (1995). This is MYWORLD: The logic of an agent-oriented testbed for DAI. In Wooldridge, M. and Jennings, N. R., editors, *Intelligent Agents: Theories, Architectures, and Languages (LNAI Volume 890)*, pages 160–178. Springer-Verlag: Heidelberg, Germany.

Wooldridge, M. and Fisher, M. (1992). A first-order branching time logic of multi-agent systems. In *Proceedings of the Tenth European Conference on Artificial Intelligence (ECAI-92)*, pages 234–238, Vienna, Austria.

Wooldridge, M. and Fisher, M. (1994). A decision procedure for a temporal belief logic. In Gabbay, D. M. and Ohlbach, H. J., editors, *Temporal Logic — Proceedings of the First International Conference (LNAI Volume 827)*, pages 317–331. Springer-Verlag: Heidelberg, Germany.

Wooldridge, M. and Jennings, N. R. (1994). Formalizing the cooperative problem solving process. In *Proceedings of the Thirteenth International Workshop on Distributed Artificial Intelligence (IWDAI-94)*, pages 403–417, Lake Quinalt, WA.

Yonezawa, A., editor (1990). *ABCL — An Object-Oriented Concurrent System*. The MIT Press: Cambridge, MA.