A Model-Theoretic Approach to the Verification of Situated Reasoning Systems

Anand 5. *Rao* and *Michael P. Georgeff*
Australian Artificial Intelligence Institute
1 Grattan Street, Carlton
Victoria 3053, Australia

Phone: (+61 3) 663-7922, Fax: (+61 3) 663-7937
Email: anand@aaii.oz.au and georgeff@aaii.oz.au

Abstract

The study of situated systems that are capable of reactive and goal-directed behaviour has received increased attention in recent years. One approach to the design of such systems is based upon *agent-oriented* architectures. This approach has led to the development of expressive, but computationally intractable, logics for describing or specifying the behaviours of agent-oriented systems. In this paper, we present three propositional variants of such logics, with different expressive power, and analyze the computational complexity of verifying if a given property is satisfied by a given abstract agent-oriented system. We show the complexity to be linear time for one of these logics and polynomial time for another, thus providing encouraging results with respect to the practical use of such logics for verifying agent-oriented systems.

1 Introduction

The study of systems that are situated or embedded in a changing environment has been receiving considerable attention within the knowledge representation and planning communities. The primary characteristic of these systems is their dynamic and resource-bounded nature. In particular, situated systems need to provide an appropriate balance between time spent deliberating and time spent acting. If the time spent on deliberation is too long, the ability of the system to complete its tasks may be seriously affected. On the other hand, too little deliberation may lead to a system that is short-sighted and reactive.

A number of different architectures have emerged as a possible basis for such systems [Bratman *ct* al., 1988; Rao and Georgeff, 1991b; Rosenschein and Kaelbling, 1986; Shoham, 1991]. Some of the most interesting of these are agent-oriented architectures, in which the system is viewed as a rational agent having certain *77ien-tal attitudes* that influence its decision making and determine its behaviour. The simplest of these architectures, called a BDI architecture, is based on attitudes of *belief, desire,* and *intention.* The first two attitudes represent, respectively, the information and evaluative states of the agent. The last represents decisions the agent has made at a previous time, and is critical for achieving adequate or optimal performance when deliberation is subject to resource bounds [Bratman, 1987; Kinny and Georgeff, 1991]. Recently, a number of attempts have been made to formalize these mental attitudes and to show how these attitudes determine the actions of an agent [Cohen and Levesque, 1990; Rao and Georgeff, 1991b; Singh, 1991].

Most of these studies on agent-oriented systems concentrate on the specification or characterization of rational agents and their behaviours under different environmental conditions. They introduce logics that use linear or branching temporal structures, are often first-order, and tend to have a rich repertoire of modal operators to model beliefs, desires, goals, intentions, commitment, ability, actions, and plans.

However, the design of agent-oriented systems has so far had little connection with these formalisms. Although some systems have been designed and built upon the philosophy of rational agents [Georgeff and Lansky, 1986], the linkage between the formal specification and the design is weak. Similarly, little has been done in the verification of agent-oriented systems. As more and more of these systems are being tested and installed in safety-critical applications, such as air-traffic management, real-time network management, and power-system management, the need to verify and validate such systems is becoming increasingly important.

This paper addresses the issue of verification of situated systems based on the the theory of rational agents. Issues related to the specification and practical design of agent-oriented systems are dealt with elsewhere [Rao and Georgeff, 1991b; Rao and Georgeff, 1992]. The outline of the paper is as follows. Section 2 describes the semantic model. Section 3 presents three branching-time BDI logics with increasing expressive power and introduces the notion of commitment. The problem of verification in these logics and their complexity is described in Section 4. Using an example, Section 5 shows how one can verify temporal and commitment properties of agent-oriented systems in polynomial time. Finally, we conclude in Section 6 by comparing our work with related effort and highlighting the contributions of this paper.

2 Overview

Situated agents can be viewed as concurrent systems of processes. The execution of such processes can be modeled by the nondeterministic interleaving of the atomic actions of each process. In such a model, the nondeterministic choice of a concurrent program is represented as a time point with multiple successors in a branching-time tree structure. Each possibly infinite execution sequence of a concurrent program is represented as a computation path of the tree structure.

For systems based on the notion of a rational agent, however, such a model of the system's behaviour is too abstract. In this case, one is interested in analyzing how such agents choose to bring about the future that they desire. In so doing, the agent needs to model the uncertainty or *chance* inherent in the environment as well as the *choice* of actions available to it at each and every time point. As the agent does not have direct control over the environment, but has direct control over the actions it can perform, it is desirable to separate the agent's choice of action (over which it has control) from its view of the environment (over which it has no control). Also, unlike concurrency theory, there is no single view of the environment; each agent can have its own view of the environment and of other agents' mental states which may not coincide with the actual environment nor the actual mental states of these agents. These different views of the world can be more effectively modeled within a possible-worlds framework.

Hence, we adopt a *possible worlds branching-time tree structure* in which there are multiple possible worlds and each possible world is a branching-time tree structure. Multiple possible worlds model the chance inherent in the environment as viewed by the agent and are a result of the agent's lack of knowledge about the environment. But within each of these possible worlds, the branching future represents the choice of actions available to the agent.

A particular time point in a particular world is called a *situation.* For each situation we associate a set of *belief-accessible, desire-accessible,* and *intention-accessible* worlds intuitively, those worlds that the agent *believes* to be possible, *desires* to bring about, and *commits* to achieving, respectively. We require that an agent's intentions be consistent with its adopted desires, and that its desires be believed to be achievable [Rao and Georgeff, 1991a].

One of the important properties in reasoning about concurrent programs is the notion *of fairness.* Fairness or fair scheduling assumptions specify when an individual process in a family of concurrent processes must be scheduled to execute next. A number of different fairness assumptions have been analyzed in the literature.[1] A commonly used fairness assumption is that a process must be executed infinitely often. A concurrent program can thus be viewed as a branching-time tree structure, with fairness and starting conditions. The verification of a property is equivalent to checking if the property is satisfied in the model corresponding to the concurrent program under the fairness and starting conditions. As described by Emerson [1990], concurrency can be expressed by the following equation: concurrency = nondeterminism + fairness.

Analogously, an important aspect of rational agency is the notion of *commitment.* The commitment condition specifies when and for how long an agent should pursue a chosen course of action and under what conditions it should give up its commitment. Thus, a commitment condition embodies the balance between reactivity and goal-directedness of an agent-oriented system. An abstract agent-oriented system can thus be viewed as a possible worlds branching-time tree structure with commitment and starting conditions. The verification of a property of the agent-oriented system is equivalent to checking if the property is satisfied in the model corresponding to the system under the commitment and starting conditions. We could therefore express agent-oriented reasoning as follows: agent-oriented reasoning = chance + choice -+ commitment. In the next two sections we formalize these notions.

*See EmeTson [1990] for an overview on this topic.

3 Propositional BDI Logics

3.1 Syntax

We define three languages CTLBDI, CCTLBDI (Committed $\mathbf{CTL_{BDI}}$), and $\mathbf{CTL^{*}_{BDI}}$, which are propositional modal logics based on the branching temporal logics CTL, Fair CTL, and CTL* [Emerson and Lei, 1987], respectively, with increasing expressive power. The primitives of these languages include a non-empty set φ of *primitive propositions;* propositional *connectives* V and -; *modal operators* BEL (agent believes), DESIRE (agent desires), and INTEND (agent intends); and *temporal operators* X (next), U (until), F (sometime in the future or eventually), E (some path in the future or optionally), and $\mathbf{E}_{\xi}$ (some *committed* path ξ in the future). Other connectives and operators such as $\wedge$, $\supset$, $\equiv$, $\mathbf{G}$ (all times in the future or always), A (all paths in the future or inevitably), $\mathbf{A}_{\xi}$ (all *committed* paths ξ in the future), $\overset{\infty}{\mathbf{F}}$ (infinitely often), and $\overset{\infty}{\mathbf{G}}$ (almost always) can be defined in terms of the above primitives. The last two operators are defined only for $\mathbf{CTL^{*}_{BDI}}$.

There are two types of well-formed formulas in these languages: *state formulas* (which are true in a particular world at a particular time point) and *path formulas* (which are true in a particular world along a certain path). State formulas are defined in the standard way as propositional formulas, modal formulas, and their conjunctions and negations. The objects of E and A are path formulas. Path formulas for $\mathbf{CTL^{*}_{BDI}}$ can be any arbitrary combination of a linear-time temporal formula, containing negation, disjunction, and the linear-time operators X and U. Path formulas of $\mathbf{CTL_{BDI}}$ are restricted to be primitive linear-time temporal formulas, with no negations or disjunctions and no nesting of linear-time temporal operators. For example, $\mathbf{AF}(p \vee q)$ is a state formula and $\mathbf{GF}p$ is a path formula of $\mathrm{CTL}_{\mathrm{BDI}}$ but not of $\mathbf{CTL_{BDI}}$.

In contrast to the language CTLBDI, Committed

CTL_{BDI} uses the path operators E_ξ and A_ξ followed by one of the linear-time temporal operators X, U, F, and G. The symbol ξ is used to emphasize the fact that these operators range over paths that meet a *commitment constraint*; namely, that ξ is of the canonical form $\bigwedge_{i=1}^{n}(\overset{\infty}{F}(\alpha_i) \vee \overset{\infty}{G}(\beta_i))$, where α_i and β_i are commitment formulas (described in Section 3.3). The formula $A_\xi\xi_1$ can be viewed as a CTL^*_{BDI} formula $A(\xi \supset \xi_1)$ and similarly $E_\xi\xi_1$ can be viewed as a CTL^*_{BDI} formula $E(\xi \wedge \xi_1)$. We shall call the above conversion **-conversion*.

The *length* of a formula ϕ is denoted by $|\phi|$ and is defined recursively as follows [Emerson and Lei, 1987]: (a) the length of a primitive proposition is zero; and (b) the length of conjunctive, negated, modal, and temporal formulas is one more than the sum of the sizes of their component formulas. The formula ψ is said to be a *subformula* of ϕ if ψ is a substring of ϕ. Let $Sub(\phi)$ be the set of all subformulas of ϕ.

For example, the formula $\neg p \wedge BEL(p \wedge q)$ has a length of 4. The subformulas of the above formula is given by $\{\neg p \wedge BEL(p \wedge q), \neg p, BEL(p \wedge q), p, p \wedge q, q\}$.

3.2 Possible-Worlds Semantics

We define a *structure* M to be a tuple, $M = (W, T, \mathcal{R}, \mathcal{B}, \mathcal{G}, \mathcal{I}, L)$ where T is a set of *time points*; $\mathcal{R} \subseteq T \times T$ is a total binary *temporal accessibility* relation; Φ is the set of primitive propositions; W is a set of *possible worlds*, where each world w is a tuple of the form $(T_w, \mathcal{R}_w, L_w)$ in which $T_w \in T$ is a set of *time points* in w, $\mathcal{R}_w$ is a restriction of $\mathcal{R}$ to the time points T_w, and L_w is the truth assignment function that assigns to each time point in w a set of propositional formulas, i.e., L_w: $T_w \rightarrow 2^\Phi$. Finally, $\mathcal{B} \subseteq W \times T \times W$ is a *belief accessibility* relation; and $\mathcal{G}$ and $\mathcal{I}$ are *desire* and *intention accessibility* relations, respectively, that are defined in the same way as $\mathcal{B}$.

Sometimes, we shall view the arcs of the time tree as being labeled with a primitive event, i.e., $\mathcal{R} \subseteq T \times E \times T$, where E is the set of primitive events that is added to the structure M.

A *fullpath*, $x = (s_0, s_1, ...)$ in w is an infinite sequence of time points such that $(s_i, s_{i+1}) \in \mathcal{R}_w$ for all i. The suffix fullpath $(s_i, s_{i+1}, ...)$ is denoted by x^i. Satisfaction of a state formula ϕ is given with respect to a structure M, a world w and a time point t, denoted by $M, w, t \models \phi$. Satisfaction of path formulas is given with respect to a structure M, world w, and a fullpath x in world w.

$M, w, t \models BEL(\phi)$ iff $M, w', t \models \phi$ for all w' satisfying $(w, t, w') \in \mathcal{B}$;
$M, w, t \models E(\xi)$ iff $M, w, x \models \xi$ where x is a fullpath in world w starting at t;
$M, w, x \models X(\phi)$ iff $M, w, x^1 \models \phi$;
$M, w, x \models \phi_1 U \phi_2$ iff for some $i \geq 0$, $M, w, x^i \models \phi_2$ and for all $0 \leq j < i$, $M, w, x^j \models \phi_1$;

The semantics of primitive propositions, negations, and conjunctions of formulas are defined in a standard manner and the semantics of temporal formulas are as defined for CTL* [Emerson, 1990]. Semantics of formulas in $CCTL_{BDI}$ containing committed path operators E_ξ and A_ξ is given by converting them into CTL^*_{BDI} formulas and then using the above satisfaction conditions. We use $M, w, t \models_\xi \phi$ as an abbreviation for $M, w, t \models \phi^*$, where ϕ^* is a CTL^*_{BDI} formula obtained from ϕ using the *-conversion.

Desires and intentions are defined as for beliefs but with respect to their corresponding accessibility relations. The relationships between beliefs, desires, and intentions impose different restrictions on these accessibility relations. These relationships have been discussed by us elsewhere [Rao and Georgeff, 1991a; Rao and Georgeff, 1991b]. Finally, using the above basic modal operators we can define the additional modal operators as follows: $F\phi$ as *true* U ϕ; $G\phi$ as $\neg F \neg \phi$; $A\phi$ as $\neg E \neg \xi$; $\overset{\infty}{F}\phi$ as $GF\phi$; and $\overset{\infty}{G}\phi$ as $FG\phi$.

3.3 Commitment

Commitment plays an important role in agent-oriented reasoning. In a continuously changing environment, commitment lends a certain sense of stability to the reasoning process of an agent. For example, if John is committed to going to the bank at 2pm, he is unlikely to re-evaluate this decision at every clock tick; instead, he would probably re-evaluate his decision (or give up his commitment) only if there were a significant change in circumstances. In other words, commitment and its relative stability with respect to changes in beliefs results in savings in computational effort and hence better overall performance [Bratman, 1987; Kinny and Georgeff, 1991; Rao and Georgeff, 1991b].

A commitment usually has two parts to it: one is the condition that the agent is committed to maintain, called the *commitment condition*, and the second is the condition under which the agent gives up the commitment, called the *termination condition*. More formally, we define a commitment operator C as follows:

$\phi_1 C \phi_2 \equiv A(\phi_1 \; U \; \phi_2)$;

where ϕ_1 is a commitment condition and ϕ_2 is a termination condition. The *commitment formula* $\phi_1 C \phi_2$ (read as ϕ_1 is inevitably committed until ϕ_2) states that, if the commitment condition ϕ_1 is true, in all future paths the agent will commit to (or maintain) ϕ_1 until the termination condition ϕ_2 is true. Note that ϕ_1 and ϕ_2 can be arbitrary state formulas of the language CTL_{BDI} and the commitment formulas also belong to CTL_{BDI}. Within this basic framework one can express a number of different types of commitment in the language CTL_{BDI}.

As the agent has no direct control over its beliefs and desires, there is no way that it can adopt or effectively realize a commitment strategy over these attitudes. However, an agent can choose what to do with its intentions. Thus, we restrict the commitment condition ϕ_1 to be of the form INTEND(Aξ) or INTEND(Eξ). The former we call *full* commitment and the latter a *partial* commitment. The exact form of the termination condition yields different types of commitment. We review three types of commitment that were described elsewhere [Rao and Georgeff, 1991b]: *blind commitment* in which the termination condition ϕ_2 is of the form $BEL(\phi)$; *single-minded commitment* in which ϕ_2 is of the form $BEL(\phi)$

$\vee \neg BEL(EF(\phi))$; and *open-minded commitment* in which ϕ_2 is of the form $BEL(\phi) \vee \neg DESIRE(EF(\phi))$.

An agent that is blindly committed will give up its commitment only when it believes in φ, where φ is usually a proposition that the agent is striving to achieve. In addition to this, an agent who is single-mindedly committed will give up its commitment when it no longer believes that there exists an option of satisfying the proposition some time in the future. An agent that is open-mindedly committed will give up its commitment either when it believes in the proposition or when it no longer has the desire to eventually achieve the proposition.

One can combine the above forms of commitment in various ways. For example, the formula $(INTEND(AFp))C(BEL(p))$ denotes an agent that is blindly and fully committed to achieving p until it believes in p. Similarly, the formula $(INTEND(EFp))C(BEL(p) \vee \neg BEL(EFp))$ is an example of an agent that is single-mindedly partially committed to achieving p (i.e., has decided not to rule out the posibility of not being able to achieve p in the future).

For an agent to eventually achieve its desires, it needs to maintain its commitment to bring about these desires. Although an agent that only occasionally maintains its commitment may serendipitously fulfill its desires, as designers of these systems we cannot *guarantee* this. To do so, we need to impose stronger maintenance conditions; namely, that the commitment formula is true "infinitely often" or "almost always". Hence, in Committed CTLBDI WE take the commitment constraint ξ to be of the canonical form $\bigwedge_{i=1}^{n}(\overset{\infty}{F}(\alpha_i) \vee \overset{\infty}{G}(\beta_i))$, where a, and β_i are commitment formulas.

4 Verification

Our interest is in determining what properties hold of a given agent, in a given environment, under certain initial conditions and under certain commitment conditions. For example, given a robot that is programmed to single-mindedly commit to a certain set of intentions, we may need to prove that, in a particular environment and under particular initial conditions, it will never harm a human.

Given some specification of the agent and the environment, we can generate the branching-tree structure corresponding to all possible evolutions of that agent in that environment.[2] This structure represents the model M of the agent and its environment. For the purposes of this paper, we consider only *finite* structures. The size of a finite structure M is given by the size of the different components of the structure. More formally, $|M| = O(|W| \cdot (|\mathcal{R}| + |\mathcal{B}| + |\mathcal{G}| + |\mathcal{I}|))$. The size of W is equal to the number of worlds and the size of the relations is equal to the number of elements in the relation.

[2]We do not address this process of model generation in this paper. Methods for generating models used in concurrency theory [Emerson, 1990] can be extended for this purpose. The notion of *plans* as abstract specifications [Rao and Georgeff, 1992] is similar to that of finite-state transitions and can be used to generate a partial model.

We assume that the initial environment-agent configuration of the system is given by a state formula ϕ_{START}. We shall refer to the tuple (M, ϕ_{START}) as an *abstract agent-oriented system*. As designers of these systems, we want to be able to verify that given an abstract agent-oriented system, certain properties of the system, expressed as state formulas, are true. The abstract system and the properties can be expressed in either CTL_{BDI} or CTL^{*}_{BDI}. More formally we have the following definition.

Definition 1:
Verification of abstract agent-oriented systems $(M, \phi_{START}) \models \phi$ iff $\forall\, w, t$ in M such that $M, w, t \models \phi_{START}$ we have $M, w, t \models \phi$.

Hence, the verification problem for CTL_{BDI} reduces to the *Model Checking Problem for* CTL_{BDI} (MCP), defined as follows: Given a structure $M = (W, T, \mathcal{R}, \mathcal{B}, \mathcal{G}, \mathcal{I}, L)$ and a state formula ϕ, determine for each world w and time point t whether $M, w, t \models \phi$.

Informally, an algorithm, AMCP for solving the Model Checking Problem can be given as follows: Start with subformulas of ϕ that are of length 0, determine the worlds and time points where they are true, and then progressively process subformulas of length greater than 0. After i such steps where $|\phi| \leq i$, the set of worlds and time points where ϕ and all its subformulas are true will be known.

This algorithm is a modification of the algorithm given by Emerson and Lei [1987] for their Fair Computation Tree Logic (FCTL). The main difference in model checking is the presence of multiple possible worlds. The complexity of the algorithm AMCP is stated below; its details can be found elsewhere [Rao and Georgeff, 1993].

Theorem 1 *Algorithm AMCP correctly solves MCP by labeling each world w and time point t of the structure M with the set of subformulas of ϕ true at w and t, and takes $O(|\phi| \cdot |M|)$ time to run.*

Although CTL_{BDI} can capture different forms of commitment, it is still not expressive enough for our purposes. For example, the language is not expressive enough to state that if a robot is always committed to serving its master, then no matter what tasks it does it will in all cases eventually satisfy its master. In particular, we cannot state that in all paths that satisfy a certain commitment formula, say ξ_1, a property, say ξ_2, holds. More formally, we cannot express $A[\xi_1 \supset \xi_2]$.

As was discussed earlier, Committed CTL_{BDI} can express such statements. Now we analyze the complexity of verifying abstract agent-oriented systems based on Committed CTL_{BDI}. In $CCTL_{BDI}$, an abstract agent-oriented system is taken to be a tuple (M, ϕ_{START}, ξ), where M and ϕ_{START} are as defined before and ξ is a commitment constraint. More formally, verification of these systems can be defined as follows:

Definition 2:
Verification of abstract agent-oriented systems $(M, \phi_{START}, \xi) \models \phi$ iff $\forall\, w, t$ in M such that $M, w, t \models \phi_{START}$ we have $M, w, t \models_{\xi} \phi$.

Similar to the model checking problem of CTL_{BDI}, one can analogously define the *Model Checking Problem for*

Committed CTLBDI (CMCP). Complexity results given for FCTL by Emerson and Lei [1987] can be extended to CMCP. In particular, CMCP can be reduced to the problem of model checking for committed states. This reduction exploits the nature of the commitment constraint ξ; namely, the fact that $\overset{\infty}{F}$ and $\overset{\infty}{G}$ are oblivious to the addition and deletion of finite prefixes. Also, formulas of the form $E_\xi X\phi$, $E_\xi[\phi U\psi]$, $E_\xi[\neg(\phi U\psi)]$, and $\phi C\psi$ can be reduced to model checking of primitive propositions, formulas of the form $E_\xi X\ true$ and $E_\xi\ G(\psi)$ [Rao and GeorgefT, 1993].

The details of the algorithm ACMCP are given elsewhere [Rao and GeorgefT, 1993]. Its complexity is given below.

Theorem 2 *Solving the model checking problem for committed branching temporal BDI logic,* CCTLBDI *will take* $O(|\phi| \cdot |W| \cdot |M| \cdot |\xi|^2)$ *time to run.*

The extensions of CCTLBDI over FCTL are twofold: (i) the introduction of possible worlds extends the expressive power of CTL and results in a complex structure on which to perform model checking; and (ii) the commitment constraint is more complex involving modal operators and path quantifiers.

The language CTL^*_{BDI} subsumes the language CTL*, which in turn subsumes the linear-time temporal language LTL. Hence, the complexity of model checking for CTL_{BDI} has to be the same or greater than that of the model checking for LTL. It has been shown [Lichtenstein and Pnueli, 1985] that the complexity of model checking in LTL is linear in the size of the structure and exponential in the size of the given formula.

5 Example

Consider a robot, Mark I, that can perform two tasks, each involving two actions. For the first task, the robot can go to the refrigerator and take out a can of beer (denoted by gf) and bring it to the living room (bb). For the second task, the robot can go to the main door of the house (gd) and open the door (od). The only uncertainty in the environment is the presence or absence of a beer can in the refrigerator. For simplicity, we assume that the act of going to the refrigerator also involves opening the door and checking for the can of beer. If there is no can in the refrigerator, the act gf is said to have failed and the next act of bringing beer cannot be performed. We assume that all other acts succeed when executed.

Given appropriate specifications of such a robot and its environment and some commitment constraint, as designers of these robots we will need to guarantee that they satisfy certain properties. For example, we may need to guarantee that (a) when the robot has a desire to serve beer it will inevitably eventually serve beer; or (b) when the robot has a desire to serve beer and a desire to answer the door, and there is beer in the fridge, it will inevitably eventually realize both desires, rather than shifting from one task to the other without completing either of them.[3]

[3] This could happen if the tasks of going to the refrigerator and going to the door involve taking multiple steps; the agent could then take one step towards the door, change its mind, take the next step towards the refrigerator, again change its mind and keep alternating between these tasks forever.

We consider two model structures *M1* and A/2. First, we start by specifying directly the external model structure *M1*. Generation of the external model structure from the agent and environment specifications is beyond the scope of this paper. A partial description of the structure *M1* is shown in Figure 1. World w1 depicts the alternatives available to the robot when it can choose to perform both the tasks and the environment is such that there is a beer can in the refrigerator. The dotted lines refer to additional future paths, which can be described in an analogous manner. One can view worlds w2 and w3 as world wl after the agent has executed the act of either going to the refrigerator or going towards the door, respectively. Similarly, w4 and w5 are evolutions of w2; w6 and w7 are evolutions of w3.

We introduce two propositions: beer-in-refrigerator and served-beer. The proposition beer-in-refrigerator is true at all times in the worlds **w1–w7**. The proposition served-beer will be true in worlds **w1–w7** after the act of bringing the beer (bb).

Next we examine the belief, desire, and intention relations of the agent. The world wl of Figure 1 shows the various time points. The belief relations for world **w1** at various time points are given as follows: **(w1, t1, w1), (w1, t2, w2), (w1, t3, w3), (w1, t4, w4), (w1, t5, w5), (w1, t6, w6), (w1, t7, w7),...** $\in \mathcal{B}$. Desire and intention relations can be defined similarly. Further, we assume that the belief relations do not change when actions are performed. In other words, we also have **(w2, t2, w2), (w2, t4, w4), (w2, t5, w5), ...** $\in \mathcal{B}$. Similar relationships hold for worlds **w3–w7**. This completes our description of the structure *M*\.

Consider a starting state in which the robot believes that there is beer in the refrigerator and has the intention to inevitably eventually have served beer.[4]

We consider two instances of the commitment constraint; the first instance is a blind commitment towards an intention to have served beer sometime in the future and the other is a single-minded commitment towards the same intention. More formally, we have:

$$\xi_1 \equiv \overset{\infty}{F}(\text{INTEND}(\text{AF}(\texttt{served-beer}))\,C\,\text{BEL}(\texttt{served-beer}));$$

$$\xi_2 \equiv \overset{\infty}{F}(\text{INTEND}(\text{AF}(\texttt{served-beer}))\,C\,(\text{BEL}(\texttt{served-beer}) \vee \neg\text{BEL}(\text{EF}(\texttt{served-beer}))));$$

Using Definition 4 and algorithm ACMCP we can show that in all paths where the robot is blindly or single-mindedly committed to its intention, it will achieve its desire of serving beer. More formally,

[4] We assume that the agent has the desire to have inevitably eventually served beer and to have inevitably eventually opened the door. In this example, we consider the case where the agent has only adopted an intention to serve beer; in the full paper [Rao and Georgeff, 1993], we consider the intention to open the door as well.

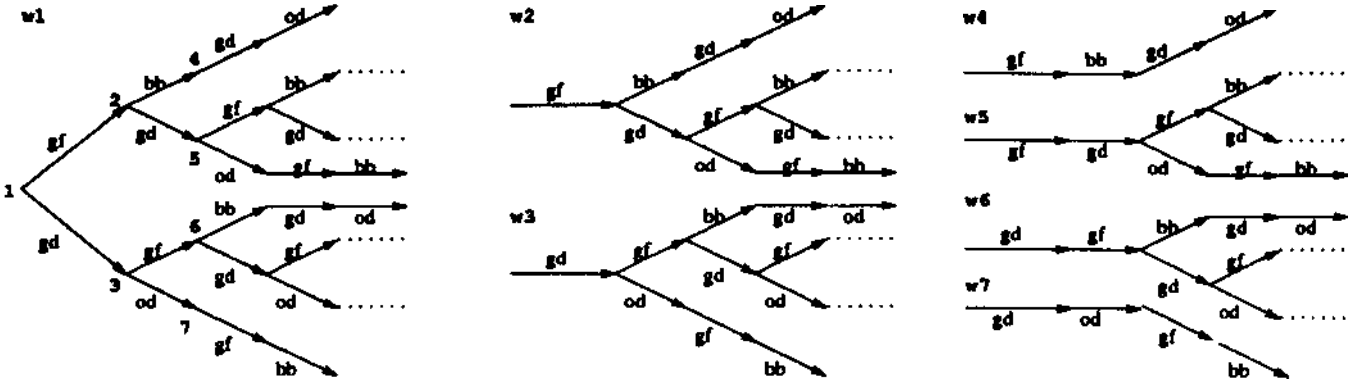

Figure 1: A partial structure of M_1

$(M_1, \phi_{START}, \xi_1) \models A_{\xi_1} F(\text{served-beer})$;
$(M_1, \phi_{START}, \xi_2) \models A_{\xi_2} F(\text{served-beer})$.

Next, consider two robots, Mark I and Mark II, and the situation in which there is no beer in the refrigerator.[5] Intuitively, Mark I does not change its belief about there being beer in the refrigerator at some time point in the future, even if it notices at this time point that there is no beer in the refrigerator. On the other hand, Mark II changes its belief about the beer being in the refrigerator as soon as it notices that there is none.

Now consider the structure M_2 which consists of worlds w1-w7 shown in Figure 1 and additional worlds where the proposition beer-in-refrigerator is false at all time points. Transitions between these worlds are similar to worlds w1-w7 except that the act gf fails (as there is no beer can in the refrigerator) and is followed by the act of going to the main door, namely gd, rather than the act of bringing the beer, namely bb.

With the structure M_2 we can show that a single-mindedly committed Mark II agent will drop its commitment to maintain the intention of inevitably eventually serving beer. On the other hand, a single-mindedly committed Mark I agent will maintain this commitment forever. More formally, we can show the following:

$(M_2, \phi_{START}, \xi_2) \models$
$\neg A_{\xi_2} F(\text{served-beer}) \wedge \text{BEL}(\text{I}, A_{\xi_2} F(\text{served-beer}))$
$A_{\xi_2} G(\text{INTEND}(\text{I}, A_{\xi_2} F(\text{served-beer})))$;
$(M_2, \phi_{START}, \xi_2) \models$
$\neg A_{\xi_2} F(\text{served-beer}) \wedge \text{BEL}(\text{II}, \neg A_{\xi_2} F(\text{served-beer}))$
$\neg A_{\xi_2} G(\text{INTEND}(\text{II}, A_{\xi_2} F(\text{served-beer})))$.

In summary, we have considered two different model structures, one where the robot completes its task, the second where it is impossible for the robot to complete its task, but yet one of the robots maintains its commitment to this task forever, while the other robot reconciles itself to the impossibility of completing the task and gives it up. The purpose of this exercise has been to show how global properties of agent-oriented systems can be verified under a variety of rational behaviours obtained by varying the model structure and the commitment constraint.

[5] Although we have not described a multi-agent CTLBDI logic, the modifications required to do so are straightforward. Also, as long as we do not introduce common knowledge operators, the complexity of model checking in such multi-agent modal logics will be of the same order as single-agent modal logics [Halpern and Moses, 1992].

6 Comparisons and Conclusions

Cohen and Levesque [1990] describe agents by adopting a possible worlds structure in which each world is a linear-time temporal structure and consider fanatical and relativized forms of commitment. A fanatical commitment is similar to our definition of a single-minded agent committed to its intention, i.e., $(\text{INTEND}(AF\phi))C(\text{BEL}(\phi) \vee \text{BEL}(AG\neg\phi))$. A relativized commitment is one in which the agent has a persistent intention towards a proposition until it believes in the proposition or until some other proposition is believed. This can be expressed as $(\text{INTEND}(AF\phi))C(\text{BEL}(\phi) \vee \text{BEL}(AG\neg\phi) \vee \text{BEL}(\psi))$. Cohen and Levesque do not address the issue of model checking in their logic. However, as their logic subsumes linear-time temporal logic (LTL), the process of model checking in their logic will be at least as hard as the model checking for LTL; namely, linear in the size of the structure and exponential in the size of the given formula [Lichtenstein and Pnueli, 1985] .

Singh [1991] presents a branching-time intention logic based on CTL*. Various rationality postulates relating to beliefs, intentions, and actions are analyzed. Also, like Cohen and Levesque, Singh uses his logic only as a specification to characterize different behaviours and does not provide any guidelines for the design or verification of such rational agents. Shoham's work [Shoham, 1991] spans both theory and language design, but does not address the issue of verification either.

This paper goes beyond this earlier work and provides a methodology for formally verifying properties of agent-oriented systems. Starting from a reasonably rich model structure, we have described three propositional logics and analyzed their relative expressive power. Furthermore, the linear time and polynomial time complexity of model checking in two of these logics makes them potentially useful for verifying practical agent-oriented systems.

Our work draws its inspiration from the field of concurrency theory [Emerson, 1990], especially that field's contribution to the techniques of model checking. We have extended the results of Emerson and Lei [1987] by showing that the linear time and polynomial time complexities of model checking hold for logics more expressive than CTL and Fair CTL logics. Also, the complexities are not greatly affected by the number of different modalities - the complexity seems to be dependent on the underlying temporal structure. More importantly, this paper demonstrates the generality of the model-checking technique [Halpern and Vardi, 1991] and extends it to a new domain; namely, the verification of agent-oriented systems. The close correspondence between fairness and commitment, and concurrency theory and rational agency, lays a strong theoretical foundation for the design and verification of agent-oriented systems.

However, a number of open problems with respect to this approach remain. First, we need to address the process of model generation whereby, given an agent specification and/or environment specification, the appropriate model structure is automatically generated. Second, we have used model checking as a means of verifying global properties, i.e., from an external observer viewpoint. Similar techniques can be used by the agent internally. In this case, we may want to build the model incrementally, rather than assuming that the entire model structure is given to us. Third, the size of the structures we are dealing with is likely to be large and techniques to reduce this would be valuable.

Although a number of issues in the model-theoretic design and verification of agent-oriented systems are yet to be resolved, our work indicates, for the first time, that the expressive multi-modal, branching-time logics can possibly be used in practice to verify the properties of these systems.

Acknowledgements

This research was supported by the Cooperative Research Centre for Intelligent Decision Systems under the Australian Government's Cooperative Research Centres Program.

References

[Bratman et ai, 1988] M. E. Bratman, D. Israel, and M. E. Pollack. Plans and resource-bounded practical reasoning. Computational Intelligence, 4, 1988.

[Bratman, 1987] M. E. Bratman. Intentions, Plans, and Practical Reason. Harvard University Press, Cambridge, MA, 1987.

[Cohen and Levesque, 1990] P. R. Cohen and H. J. Levesque. Intention is choice with commitment. Artificial Intelligence, 42(3), 1990.

[Emerson and Lei, 1987] E. A. Emerson and C-L Lei. Modalities for Model Checking: Branching Time Logic Strikes Back. Science of Computer Programming, 8:275-306, 1987.

[Emerson, 1990] E. A. Emerson. Temporal and modal logic. In J. van Leeuwen, editor, Handbook of Theoretical Computer Science: Volume B, Formal Models and Semantics, pages 995-1072. Elsevier Science Publishers and The MIT Press, Cambridge, MA, 1990.

[Georgeff and Lansky, 1986] M. P. GeorgerT and A. L. Lansky. Procedural knowledge. In Proceedings of the IEEE Special Issue on Knowledge Representation, volume 74, pages 1383-1398, 1986.

[Halpern and Moses, 1992] J. Y. Halpern and Y. Moses. A guide to completeness and complexity for modal logics of knowledge and belief. Artificial Intelligence, 54:319-379, 1992.

[Halpern and Vardi, 1991] J. Y. Halpern and M. Y. Vardi. Model checking vs. theorem proving: A manifesto. In J. Allen, R. Fikes, and E. Sandewall, editors, Proceedings of KR&R-91, pages 325-334. Morgan Kaufmann Publishers, San Mateo, CA, 1991.

[Kinny and Georgeff, 1991] D. Kinny and M. P. Georgeff. Commitment and effectiveness of situated agents. In Proceedings of 1JCAI-91, Sydney, Australia, 1991.

[Lichtenstein and Pnueli, 1985] O. Lichtenstein and A. Pnueli. Checking that finite state concurrent programs satisfy their linear specification. In Proceedings of the 12th Annual ACM Symposium on Principles of Programming Languages, pages 97 107, 1985.

[Rao and Georgeff, 1991a] A. S. Rao and M. P. Georgeff. Asymmetry thesis and side-effect problems in linear time and branching time intention logics. In Proceedings of IJCAI-91, Sydney, Australia, 1991.

[Rao and Georgeff, 1991b] A. S. Rao and M. P. Georgeff. Modeling rational agents within a BDI-architecture. In J. Allen, R. Fikes, and E. Sandewall, editors, Proceedings of KR&R-91 Morgan Kaufmann Publishers, San Mateo, CA, 1991.

[Rao and Georgeff, 1992] A. S. Rao and M. P. Georgeff. An abstract architecture for rational agents. In C. Rich, W. Swartout, and B. Nebel, editors, Proceedings of KR&R-92. Morgan Kaufmann Publishers, San Mateo, CA, 1992.

[Rao and Georgeff, 1993] A. S. Rao and M. P. Georgeff. A model-theoretic approach to the verification of agent-oriented systems. Technical Report 37, Australian Artificial Intelligence Institute, Carlton, Australia, 1993.

[Rosenschein and Kaelbling, 1986] S. J. Rosenschein and L. P. Kaelbling. The synthesis of digital machines with provable epistemic properties. In J. Y. Halpern, editor, Proceedings of the First Conference on Theoretical Aspects of Reasoning about Knowledge. Morgan Kaufmann Publishers, San Mateo, CA, 1986.

[Shoham, 1991] Y. Shoham. AgentO: A simple agent language and its interpreter. In Proceedings of AAAI-91, pages 704-709, 1991.

[Singh, 1991] M. P. Singh. A logic of situated know-how. In Proceedings of AAAI-91, pages 343-348, 1991.