

Modelling PRS-Like Agents' Mental States

Wayne Wobcke

School of Computer Science and Engineering
University of New South Wales
Sydney NSW 2052, Australia
`wobcke@cse.unsw.edu.au`

Abstract. In recent years, there have been increased efforts towards defining rigorous operational semantics for a range of agent programming languages. At the same time, there have been increased efforts to develop logical frameworks for modelling belief, desire and intention (and related notions) that make closer connections to the workings of particular architectures, thus aiming to provide some computational interpretation of these abstract models. However, there remains a substantial gap between the more abstract logical approaches and the more computationally oriented operational approaches. In this paper, we present a modelling of the mental states of PRS-like agents developed using a combination of dynamic logic and BDI logic that allows a mapping between the operational semantics and the model-theoretic semantics, considering the statics, though not the dynamics, of mental states. This represents a first step towards bridging the gap between theory and practice for an agent programming language that includes a simple notion of intention.

1 Introduction

As more agent programming languages are designed and more implementations of existing languages developed, there is an increased need to provide such languages with precise definitions. For agent languages and systems, two types of semantic definition have been the focus of much research: operational semantics, usually in Plotkin-style transition systems, and “denotational” semantics, usually based on modal logic with Kripke-style possible worlds semantics, including such descendants as temporal logic and dynamic logic. However, especially for languages based on a BDI (Belief, Desire, Intention) agent architecture, there remains a substantial gap between the two levels of semantic description. This gap means that “cognitive” properties of agents, such as rationality and commitment, that are typically modelled at the higher “denotational” level of abstraction, are not systematically connected to the properties of implemented agents as described at the operational level.

In recent years, there have been increased efforts towards defining rigorous operational semantics for a range of agent programming languages. In this paper, we focus on languages based on the BDI architecture as embodied in PRS and its variants, Georgeff and Lansky [8], Georgeff and Ingrand [7], Lee *et al.* [12], and the abstract architecture of Rao and Georgeff [15]. However, precisely because operational definitions typically dispense with “cognitive” concepts such

as rationality, commitment and goal-directedness, in favour of notions such as process, concurrency and (computational) environment and state, an operational semantics provides only limited assistance to developers of agent programs in reasoning about these higher-level properties. A contributing factor is that, whereas for the case of knowledge, there is a standard way of *ascribing* knowledge to an agent derived from the computational interpretation of its program, as in the approach of Fagin *et al.* [6], the notions of belief, desire and intention do not enjoy this property – at least not for the specific notions employed in PRS style languages.¹

On the other hand, starting with the work of Cohen and Levesque [3], there has been a strand of research aiming to formalize the logical properties of intention – both as understood by Bratman [1] and as instantiated in various BDI agent architectures, e.g. Rao and Georgeff [14], Konolige and Pollack [11], Wooldridge [18]. One major problem with such semantic modelling from the point of view of agent programming is that there is no clear way of mapping the abstract BDI models onto the computational states of an implemented agent (i.e. the converse problem to that described above), although the work of Singh [17] on strategies, and of Cavedon and Rao [2] on plans is a step in this direction. The consequence is that any properties of an agent's mental states that are shown to hold in virtue of some semantic modelling do not necessarily apply to the implemented agent being modelled, potentially making the higher-level analysis irrelevant for practical considerations.

In this paper, we provide a modelling for the semantics of PRS-like agents' mental states, focusing on the statics of such states (we do not consider belief revision or intention update in this paper). We present a logical modelling of PRS-like agent programs based on a new logic which we call Agent Dynamic Logic (ADL), that combines elements from Emerson and Clarke's Computation Tree Logic [5], Pratt's Propositional Dynamic Logic [13] and Rao and Georgeff's BDI Logic [14].

We begin with brief summaries of Computation Tree Logic and BDI Logic.

2 Computation Tree Logic

In Computation Tree Logic (CTL), Emerson and Clarke [5], the basic idea is that temporal logic formulae are evaluated over a tree-like branching time structure whose states correspond to the internal states of a computation, and whose branches represent possible computation sequences. A distinction is made between *path* formulae and *state* formulae: path formulae are evaluated with respect to paths, i.e. single branches in the time tree (so refer to the properties of a single possible computation sequence), while state formulae are evaluated with respect to states (so refer to properties of the set of possible computation sequences emanating from a state).

¹ *ConGolog*, de Giacomo *et al.* [4], is exceptional in having both an operational semantics and a semantics based on the situation calculus; however, this higher-level semantics does not capture a notion of intention.

Definition 1. The CTL path formulae are defined as follows. If α and β are state formulae, then $\Box\alpha$ (henceforth α holds), $\Diamond\alpha$ (eventually α holds), $\bigcirc\alpha$ (α holds in the next state) and $\alpha\mathcal{U}\beta$ (eventually β holds and α holds until β holds) are path formulae.

Definition 2. The CTL state formulae are defined as follows. First, any formula of the base propositional language \mathcal{L} is a state formula. Second, if α is a path formula, then $A\alpha$ (α holds on all paths) and $E\alpha$ (α holds on some path) are state formulae.

Definition 3. A CTL time tree $\langle\mathcal{T}, \prec\rangle$ is a nonempty set of time points \mathcal{T} and a binary relation \prec on \mathcal{T} that is irreflexive, transitive, discrete, serial, backwards linear and rooted.

Definition 4. A path in a CTL time tree $\langle\mathcal{T}, \prec\rangle$ is a maximal subset \mathcal{T}' of \mathcal{T} for which \prec restricted to \mathcal{T}' is connected (for all $t, u \in \mathcal{T}'$, $t \prec u$, $u \prec t$ or $t = u$).

Definition 5. A subtree of a CTL time tree $\langle\mathcal{T}, \prec\rangle$ is a time tree $\langle\mathcal{T}', \prec'\rangle$ where $\mathcal{T}' \subseteq \mathcal{T}$, $\prec' \subseteq \prec$ is \prec restricted to the elements of \mathcal{T}' and the following condition is satisfied.

path complete for all $t, u \in \mathcal{T}'$, $v \in \mathcal{T}$, if $t \prec v \prec u$ then $v \in \mathcal{T}'$

The following definition of a “world” is from Rao and Georgeff [14].

Definition 6. Let \mathcal{S} be a nonempty set of states. A world w over a CTL time tree $\langle\mathcal{T}, \prec\rangle$ based on \mathcal{S} is a function on \mathcal{T} giving a state $w_t \in \mathcal{S}$ for each time point $t \in \mathcal{T}$ (in the context of a particular world w , w_t is called a situation). For convenience, say that time points and paths in \mathcal{T} are also time points and paths in w .

Finally, the semantics of CTL state and path formulae can be defined with respect to (time points and paths in) worlds using the following definitions. The properties of trees ensure that the first two auxiliary definitions are well defined.

Definition 7. For any point t in a CTL time tree $\langle\mathcal{T}, \prec\rangle$, the subtree of $\langle\mathcal{T}, \prec\rangle$ generated from t , denoted $\langle\mathcal{T}_t, \prec_t\rangle$, is defined to be that subtree of $\langle\mathcal{T}, \prec\rangle$ consisting of the set of points $\mathcal{T}_t = \{u \in \mathcal{T} : t \preceq u\}$, where \prec_t is defined as \prec restricted to \mathcal{T}_t .

Definition 8. Let $\langle\mathcal{T}, \prec\rangle$ be a CTL time tree, let p be a path in \mathcal{T} and let t be a time point in p . The successor $s_p(t)$ of t in p is that state $u \in p$ for which $t \prec u$ but there is no $v \in p$ with $t \prec v \prec u$.

Definition 9. Let \mathcal{S} be a set of states and let \mathcal{L} be a language for expressing time-independent properties of states (we assume there is a satisfaction relation \models between states and formulae of \mathcal{L}). Let w be a world over a CTL time tree $\langle\mathcal{T}, \prec\rangle$ based on \mathcal{S} . Then w satisfies a CTL formula at a time point t in \mathcal{T} as follows.

$w \models_t \alpha$	if $w_t \models \alpha$, for α a formula of \mathcal{L}
$w \models_t \mathbf{A}\alpha$	if $w \models_p \alpha$ for every path p in $\langle \mathcal{T}_t, \prec_t \rangle$
$w \models_t \mathbf{E}\alpha$	if $w \models_p \alpha$ for some path p in $\langle \mathcal{T}_t, \prec_t \rangle$
$w \models_p \Box\alpha$	if $w \models_u \alpha$ for every $u \in p$
$w \models_p \Diamond\alpha$	if $w \models_u \alpha$ for some $u \in p$
$w \models_p \bigcirc\alpha$	if $w \models_{s_p(t)} \alpha$
$w \models_p \alpha\mathcal{U}\beta$	if there is $u \in p$ with $w \models_u \beta$, and $w \models_v \alpha$ for every $v \in p$ with $v \prec u$

3 BDI Logic

To enable Computation Tree Logic to be used for modelling rational agents, Rao and Georgeff [14] extended CTL with modal operators for modelling beliefs, desires (goals) and intentions. In this section, we give a reconstruction of this framework more suited to modelling PRS-like agents. The basic notion is a BDI interpretation, which in Rao and Georgeff's framework, is a set of worlds over the subtrees of a single time tree, where this time tree is a branching time structure as described above, and each "world" consists of an assignment of a state to each time point in the tree over which it is based. The semantics is augmented to include (accessibility) relations on situations corresponding to each modality, see also Wooldridge [18].

The following definition is a simplification of Rao and Georgeff's in that only the propositional version is given, and also a slight modification, in that the set of states \mathcal{S} is made explicit. However, in our version of the logic, which we call BDI-CTL, we introduce a modification to the language that makes it more suitable for reasoning about PRS-like agents. In particular, the language BDI-CTL includes three modal operators, **B** (belief), **G** (goal) and **I** (intention), but the operators **G** and **I** are defined in terms of other primitives that are based on elements of dynamic logic.

The formal definitions of the accessibility relations in Rao and Georgeff's framework rely on the notion of a subworld of a world. A subworld of a world w contains, for each time point t , a subset of the possible futures of t that w admits, according to whether or not the corresponding path is contained in the subtree from which the subworld is derived.

Definition 10. *A subworld of a world w over a time tree $\langle \mathcal{T}, \prec \rangle$ based on a set of states \mathcal{S} is the world w restricted to a subtree of $\langle \mathcal{T}, \prec \rangle$ whose root is the root of w .*

Definition 11. *A BDI interpretation is a tuple $\langle \mathcal{T}, \prec, \mathcal{S}, \mathcal{W}, \mathcal{B}, \mathcal{I} \rangle$ where $\langle \mathcal{T}, \prec \rangle$ is a time tree, \mathcal{S} is a nonempty set of states, \mathcal{W} is a nonempty set of worlds based on \mathcal{S} with each world over a subtree of $\langle \mathcal{T}, \prec \rangle$, \mathcal{B} is a subset of $\mathcal{W} \times \mathcal{T} \times \mathcal{W}$ defined only for tuples (w, t, w') for which t is a time point in w and w' , and \mathcal{I} is a function $\mathcal{W} \times \mathcal{T} \rightarrow \mathcal{W}$ mapping each time point t in a world w to a subworld of w whose root is t .*

Any particular world $w \in \mathcal{W}$ defines a set of futures the agent considers possible starting from the initial situation in that world. Since each world is based on a branching time structure, the actions are modelled as being non-deterministic. However, since the agent has only partial information about its environment, it may be ignorant of exactly which situation it is embedded in; the agent considers only that it is embedded in a situation in which its current beliefs are true. This ignorance is captured using the relation \mathcal{B} . More formally, the beliefs of the agent in some situation w_t (at a time point t in a world w) are precisely those propositions holding at all epistemic alternatives w' , i.e. in those situations w'_t such that $\mathcal{B}(w, t, w')$. The relation on situations is assumed to be symmetric, transitive and Euclidean, and moreover, the relations \mathcal{B} and \mathcal{I} are assumed to satisfy the condition that $\mathcal{B}(w, t, w')$ iff $\mathcal{B}(\mathcal{I}(w), t, \mathcal{I}(w'))$, i.e. the epistemic alternatives of $\mathcal{I}(w)$ at t are the intended subworlds of the epistemic alternatives of w at t .

The intentions of the agent are those actions the agent considers that eventually it will successfully perform, according its current view of the world, i.e. not taking into account the potential for the world to change so as to force the agent to revise or abandon its intentions. Intentions are modelled using the function \mathcal{I} , which defines, for any particular world, which futures in that world are intended by the agent: in the “intended” futures, all the agent’s actions are performed successfully. More formally, the intentions of an agent with respect to a world w are those action formulae π for which on all possible futures in $\mathcal{I}(w)$, the agent does π , i.e. the intentions of an agent are represented by formulae of the form $\text{BA}\Diamond do(\pi)$, and this is the basis of the definition below. For the purposes of exposition, in this section, we will simply assume that each formula $do(\pi)$ is an atomic propositional formula satisfiable at a situation. Finally, the goals of the agent are simply those formulae α such that the agent intends to perform the action *achieve* γ .

Definition 12. *Let $\langle \mathcal{T}, \prec, \mathcal{S}, \mathcal{W}, \mathcal{B}, \mathcal{I} \rangle$ be a BDI interpretation. Then a world $w \in \mathcal{W}$ satisfies a BDI logic formula at a time point t in w as follows.*

$$\begin{aligned} w \models_t \text{B}\alpha & \quad \text{if } w' \models_t \alpha \text{ whenever } \mathcal{B}(w, t, w') \\ w \models_t \text{I}\pi & \quad \text{if } \mathcal{I}(w) \models_t \text{BA}\Diamond do(\pi) \\ w \models_t \text{G}\gamma & \quad \text{if } w \models_t \text{I}(\text{achieve } \gamma) \end{aligned}$$

4 Agent Dynamic Logic

In this section, we adapt BDI logic to include a modelling of PRS-like agent programs using dynamic logic. In Propositional Dynamic Logic (PDL), Pratt [13], see also Goldblatt [9], the execution of a standard computer program is modelled using transition functions over internal machine states. For such programs, execution is assumed always to be successful (though not necessarily terminating) and machine states are assumed to change only as the result of program execution. However, in modelling agent programs, both these assumptions need to be relaxed: success is not always guaranteed and the environment is subject

to change due to forces external to the agent. Moreover, though this is not considered in this paper, the agent's mental states (beliefs, goals and intentions) also change through time.

Analogous to PDL, the language ADL (Agent Dynamic Logic) includes modal operators corresponding to each program (concrete hierarchical plan) ϕ , and the semantics is based on computation trees, as in the approach of Harel [10]. However, the semantics is much more closely aligned to the behaviour of the PRS-like interpreter than more general earlier frameworks for modelling action, e.g. Segerberg [16], Singh [17] and Cavedon and Rao [2]. In particular, we aim, following Cavedon and Rao's work, to make explicit reference to the dynamic logic of action in modelling the agent's intentions.

The original semantics of PDL was defined in terms of a family of binary state transition relations \mathcal{R}_π , one for each program π . The relation \mathcal{R}_π is intended to capture the possible input-output pairs corresponding to π : more precisely, $\langle s, t \rangle \in \mathcal{R}_\pi$ iff t is a state that can be reached through executing π starting at the initial state s . In ADL, an interpretation consists of a family of BDI interpretations (sets of worlds) \mathcal{R}_π , one for each (agent) program, with the important modification that the worlds may be over trees with finite branches. This derives from the semantics of PDL based on computation trees developed by Harel [10].

The reason for using BDI interpretations to capture the meaning of programs, rather than just worlds (or computation trees in Harel's sense), is that the meaning of the test statement in the PRS-like language must be defined with respect to the agent's belief states, not with respect to world states. In our semantics, *states* are understood as world states external to the agent, not machine states internal to the agent. Now consider the conditional and iterative constructs. In standard PDL, the *if-then-else* and *while* statements are defined in terms of the sequencing, union, iteration and test primitives as follows.

$$\begin{aligned} \text{if } \alpha \text{ then } \pi \text{ else } \chi &\equiv (\alpha?; \pi) \cup (\neg\alpha?; \chi) \\ \text{while } \alpha \text{ do } \pi &\equiv (\alpha?; \pi)^*; \neg\alpha? \end{aligned}$$

But the conditional and iterative constructs in PRS agent programs behave very differently: the tests in the PRS statements are tests, not on the state of the world, but on the agent's belief state (a distinction not needed for standard PDL). Thus the set of possible worlds that realize an execution of a PRS-like agent's program must ultimately be defined with respect to the agent's beliefs (at execution time).

Definition 13. A PRS interpretation is a pair $\langle \mathcal{S}, \mathcal{R} \rangle$, where \mathcal{S} is a set of states and \mathcal{R} is a family of sets of BDI interpretations \mathcal{R}_π based on \mathcal{S} , one such BDI interpretation for each program π .

Definition 14. Let $\langle \mathcal{S}, \mathcal{R} \rangle$ be a PRS interpretation and $\langle \mathcal{T}, \prec, \mathcal{S}, \mathcal{W}, \mathcal{B}, \mathcal{I} \rangle$ be a BDI interpretation. Then a world $w \in \mathcal{W}$ satisfies the formula $do(\pi)$ at a time point t in w as follows.

$$w \models_t do(\pi) \quad \text{if some subworld of } w \text{ is the prefix of a world in } \mathcal{R}_\pi$$

Definition 15. A subworld w is a prefix of a world w' if for each end node n of w , there is a world w_n such that replacing each n in w by w_n results in the world w' .

Each primitive action π (except for the special action *achieve* γ – see below), is assumed to be modelled by a set of worlds each over a computation tree of depth 1, one world for each possible initial state $s \in \mathcal{S}$: call such a world a π -world. Each π -world with root s has one child state for each possible outcome of executing the action π in s . In addition, a subworld of each π -world is assumed to define the successful executions of π (note that this makes the notion of success state dependent). This definition is meant to capture both the nondeterminism inherent in action execution, a feature of standard programs that can also be modelled in PDL, and also the notion of intended successful execution of an action. Now each BDI interpretation contains a set of worlds, with the situations in those worlds possibly related under the epistemic alternative and intended future relations \mathcal{B} and \mathcal{I} . The meaning of each \mathcal{R}_π is a set of BDI interpretations where the initial situations of all the worlds in the interpretation form a single equivalence class of epistemic alternatives, and where the relation \mathcal{I} is defined by inheriting this relation from each world (this makes the notion of success independent of the agent's beliefs).

We need to define operations on sets of BDI interpretations that correspond to the program construction operators, in particular for sequencing, alternation and iteration, which, in standard PDL, are expressed using composition, set union and reflexive, transitive closure (of binary state transitions relations). The operation for sequencing is a kind of “concatenation” of worlds, analogous to concatenation of computation sequences. Let w_1 and w_2 be worlds over time trees \mathcal{T}_1 and \mathcal{T}_2 , let s be an end point of \mathcal{T}_1 and let r be the root of \mathcal{T}_2 . Say w_2 extends w_1 at s if the state associated with s , $w_1(s)$, equals that associated with r in \mathcal{T}_2 , $w_2(r)$, and the set of states associated with the epistemic alternatives of $w_1(s)$ is the same as that associated with the epistemic alternatives of $w_2(r)$. Then the concatenation of w_1 and w_2 , denoted $w_1 \oplus w_2$, is a world over a time tree in which, intuitively, each state at each end point s of \mathcal{T}_1 is replaced by a copy of the world w_2 whenever w_2 extends w_1 at s . The definition of $w_1 \oplus w_2$ can be made more precise as follows.

Definition 16. Let w_1 and w_2 be worlds (in BDI interpretations) over time trees $\langle \mathcal{T}_1, \prec_1 \rangle$ and $\langle \mathcal{T}_2, \prec_2 \rangle$. Let S be the set of end points of \mathcal{T}_1 , and let S' be the subset of S for which $w_1(s) = w_2(r_2)$ where r_2 is the root of \mathcal{T}_2 , and $\{\text{root}(w) : \mathcal{B}(w_1, s, w)\} = \{\text{root}(w) : \mathcal{B}(w_2, r, w)\}$, where $\text{root}(w)$ is the state associated with the root of w . For each element s of S' , let w_2^s be a world over a tree \mathcal{T}_2^s structurally isomorphic to \mathcal{T}_2 , whose valuation, precedence and accessibility relations are the same as w_2 on corresponding elements. Then the concatenation of w_1 and w_2 , denoted $w_1 \oplus w_2$ is defined over a tree consisting of $\mathcal{T}_1 - S'$ and all the sets \mathcal{T}_2^s with a precedence ordering \prec extending \prec_1 and all the \prec_2^s by also defining $t_1 \prec t_2$ if $t_1 \in \mathcal{T}_1 - S'$, $t_1 \prec_1 s$ and $t_2 \in \mathcal{T}_2^s$.

If W_1 and W_2 are sets of worlds, $W_1 \oplus W_2$ is the set of worlds formed by simultaneously concatenating, for each end point s of each world w_1 in W_1 , w_1

and a world w_2 in W_2 that extends w_1 at s (if one exists). We assume that for each distinct state, there is only one world in W_2 whose root is associated with that state, hence there is at most one world in W_2 that extends w_1 at s , for any given end point s of w_1 .

For alternation, we utilize an operation that merges two worlds if they have identical initial states. Let w_1 and w_2 be worlds over time trees \mathcal{T}_1 and \mathcal{T}_2 with roots r_1 and r_2 , and suppose that $w_1(r_1) = w_2(r_2)$ and the set of states associated with the epistemic alternatives of both situations are the same. Then the merger of w_1 and w_2 , denoted $w_1 \uplus w_2$, is that world formed by identifying the two roots and joining together w_1 and w_2 at this situation.

Definition 17. *Let w_1 and w_2 be worlds (in BDI interpretations) over time trees $\langle \mathcal{T}_1, \prec_1 \rangle$ and $\langle \mathcal{T}_2, \prec_2 \rangle$ with roots r_1 and r_2 , such that $w_1(r_1) = w_2(r_2)$ and $\{root(w) : \mathcal{B}(w_1, r_1, w)\} = \{root(w) : \mathcal{B}(w_2, r_2, w)\}$, where $root(w)$ is the state associated with the root of w . Let the tree \mathcal{T} be defined as the set of time points $\mathcal{T}_1 \cup \mathcal{T}_2$ in which r_1 (and its epistemic alternatives) and r_2 (and its corresponding epistemic alternatives) are identified, and with \prec defined as $\prec_1 \cup \prec_2$] (so that the identified r_1 and r_2 is the root of \mathcal{T} , and the children of this node are the children of r_1 from \mathcal{T}_1 and of r_2 from \mathcal{T}_2). Then the merger of w_1 and w_2 , denoted $w_1 \uplus w_2$, is the world defined over the tree $\langle \mathcal{T}, \prec \rangle$ that is inherited from w_1 and w_2 , i.e. $(w_1 \uplus w_2)(t)$ is $w_1(t)$ if $t \in \mathcal{T}_1$ and is $w_2(t)$ if $t \in \mathcal{T}_2$.*

If W_1 and W_2 are sets of worlds (each set containing at most one world with any given initial state), $W_1 \uplus W_2$ is the set of worlds formed by merging all pairs of worlds w_1 and w_2 in $W_1 \cup W_2$ whose roots have identical states and equivalent sets of epistemic alternatives.

We can finally give the constraints on the sets of computation trees \mathcal{R}_π that ensure that each respects the operational semantics of the program construction operators.

$$\begin{aligned} \mathcal{R}_{\pi;\chi} &= \mathcal{R}_\pi \oplus \mathcal{R}_\chi \\ \mathcal{R}_{\pi \cup \chi} &= \mathcal{R}_\pi \uplus \mathcal{R}_\chi \\ \mathcal{R}_{\pi^*} &= \mathcal{R}_\pi^* \text{ (the reflexive, transitive closure of } \mathcal{R}_\pi \text{ under } \oplus) \\ \mathcal{R}_{\alpha?} &= \wp(\{\langle s, s \rangle : s \in \mathcal{S}, s \models \alpha\}) - \emptyset \end{aligned}$$

In the definition for the test statement $\alpha?$, $\langle s, s \rangle$ denotes a tree with two elements, a root node and a child node, both with the associated state s , and \wp denotes the powerset constructor, assuming all initial situations of the trees in any element of the powerset are belief equivalent (so such situations satisfy $B\alpha$).

The family of relations \mathcal{R}_π must be extended from programs to (concrete hierarchical) plans. Let \mathcal{W} denote the set of all possible worlds, B_0 the set of BDI interpretations all of whose worlds are of depth 0, and define $b \models \alpha$ for a BDI interpretation with set of worlds \mathcal{W} if for each world $w \in \mathcal{W}$, $w \models_t \alpha$ where t is the root of w . Then the family of relations R_π must be extended to sets of concrete hierarchical plans. The definition below is based on the definition of concurrency as sequence interleaving. The operator $|$ is used to denote list concatenation, i.e. $[\pi|P]$ denotes a plan whose highest level program is π and which has a suffix P .

Definition 18. *The relations \mathcal{R}_π for the empty program Λ and for programs achieve γ corresponding to a subgoal γ are defined as follows.*

$$\begin{aligned} \mathcal{R}_\Lambda &= B_0 \\ \mathcal{R}_{\text{achieve } \gamma} &= \biguplus \{ \mathcal{R}_\pi \cap \{b : b \models \text{pre}(\pi) \wedge \text{context}(\pi)\} : \pi \in L, \text{post}(\pi) \vdash \gamma \} \end{aligned}$$

Here $\text{pre}(\pi)$, $\text{post}(\pi)$ and $\text{context}(\pi)$ are the precondition, postcondition and context of a plan π in the plan library L . Thus Λ is executable in every situation in every world, and leaves the state unchanged, while *achieve* γ is modelled as a set of PRS interpretations in which γ is achieved by the successful execution of a (hierarchical) plan built from plans in the plan library. Note carefully that this condition is actually a set of recursive definitions of $\mathcal{R}_{\text{achieve } \gamma}$ for all formulae γ at once; it is recursive because the plans π may include subgoals of the form *achieve* δ (and here, of course, δ may be γ). Thus the meaning of *achieve* γ involves a least fixpoint construction.

Definition 19. *Let P be a concrete hierarchical plan. Then the relation \mathcal{R}_P is defined as follows.*

- $\mathcal{R}_{[\pi]} = \mathcal{R}_\pi$
- if π is *achieve* $\gamma; \chi$ then $\mathcal{R}_{[\pi|P]} = \mathcal{R}_P \oplus \mathcal{R}_\chi$, otherwise $\mathcal{R}_{[\pi|P]} = \mathcal{R}_P \oplus \mathcal{R}_\pi$

Definition 20. *Let I be a set of concrete hierarchical plans. Then the set of worlds R_I corresponding to I is defined as $\mathcal{R}_I = \bigotimes_{P \in I} \mathcal{R}_P$, where \bigotimes denotes interleaving of a set of worlds indexed by a set \mathbb{I} : $\bigotimes_{i \in \mathbb{I}} W_i$ is the smallest set of worlds satisfying the following condition.*

$$\bigotimes_{i \in \mathbb{I}} W_i = \{ \text{head}(W_i) \oplus W'_i : i \in \mathbb{I} \text{ and } W'_i \in \bigotimes_{i \in \mathbb{I}} \{W - W_i \cup \text{tails}(W_i)\} \}$$

Here, $\text{head}(w)$ is the subworld of w derived from the subtree of depth 1 starting at the root of w , and $\text{tails}(w)$ is the set of worlds that are derived from those subtrees of w whose root nodes are the children of the root node of w .

5 Conclusion

We presented a formal modelling of the mental states of PRS-like agents based on a combination of Computation Tree Logic, Propositional Dynamic Logic and BDI Logic called Agent Dynamic Logic. This provides a modelling much more closely aligned to the operational semantics of PRS-like agents than in previous work, and to some extent explains how the notions of belief, desire and intention are employed in PRS-like architectures. The formalism also enables a rigorous semantic definition of subgoals *achieve* γ as corresponding to plans which, when successfully executed within the PRS-like architecture, actually will achieve the subgoal γ provided that no contingency arises during execution that forces the plan's abandonment. One limitation of the present work is that only the statics of mental states has been modelled – the temporal structures used in the modelling capture the “future directedness” of the agent's intentions as viewed from a single point in time, but not the changes in the agent's mental state as time progresses; a more complete model would include such mental state dynamics.

References

1. Bratman, M.E. (1987) *Intention, Plans and Practical Reason*. Harvard University Press, Cambridge, MA.
2. Cavedon, L. & Rao, A.S. (1996) 'Bringing About Rationality: Incorporating Plans Into a BDI Agent Architecture.' in Foo, N.Y. & Goebel, R.G. (Eds) *PRICAI'96: Topics in Artificial Intelligence*. Springer-Verlag, Berlin.
3. Cohen, P.R. & Levesque, H.J. (1990) 'Intention is Choice with Commitment.' *Artificial Intelligence*, **42**, 213–261.
4. De Giacomo, G., Lespérance, Y. & Levesque, H.J. (2000) 'ConGolog, a Concurrent Programming Language Based on the Situation Calculus.' *Artificial Intelligence*, **121**, 109–169.
5. Emerson, E.A. & Clarke, E.M. (1982) 'Using Branching Time Temporal Logic to Synthesize Synchronization Skeletons.' *Science of Computer Programming*, **2**, 241–266.
6. Fagin, R., Halpern, J.Y., Moses, Y. & Vardi, M.Y. (1995) *Reasoning About Knowledge*. MIT Press, Cambridge, MA.
7. Georgeff, M.P. & Ingrand, F.F. (1989) 'Decision-Making in an Embedded Reasoning System.' *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence*, 972–978.
8. Georgeff, M.P. & Lansky, A.L. (1987) 'Reactive Reasoning and Planning.' *Proceedings of the Sixth National Conference on Artificial Intelligence (AAAI-87)*, 677–682.
9. Goldblatt, R. (1992) *Logics of Time and Computation. Second Edition*. Center for the Study of Language and Information, Stanford, CA.
10. Harel, D. (1979) *First-Order Dynamic Logic*. Springer-Verlag, Berlin.
11. Konolige, K. & Pollack, M.E. (1993) 'A Representationalist Theory of Intention.' *Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence*, 390–395.
12. Lee, J., Huber, M.J., Kenny, P.G. & Durfee, E.H. (1994) 'UM-PRS: An Implementation of the Procedural Reasoning System for Multirobot Applications.' *Conference on Intelligent Robotics in Field, Factory, Service, and Space*, 842–849.
13. Pratt, V.R. (1976) 'Semantical Considerations on Floyd-Hoare Logic.' *Proceedings of the Seventeenth IEEE Symposium on Foundations of Computer Science*, 109–121.
14. Rao, A.S. & Georgeff, M.P. (1991) 'Modeling Rational Agents within a BDI-Architecture.' *Proceedings of the Second International Conference on Principles of Knowledge Representation and Reasoning (KR'91)*, 473–484.
15. Rao, A.S. & Georgeff, M.P. (1992) 'An Abstract Architecture for Rational Agents.' *Proceedings of the Third International Conference on Principles of Knowledge Representation and Reasoning (KR'92)*, 439–449.
16. Segerberg, K. (1989) 'Bringing It About.' *Journal of Philosophical Logic*, **18**, 327–347.
17. Singh, M.P. (1994) *Multiagent Systems*. Springer-Verlag, Berlin.
18. Wooldridge, M.J. (2000) *Reasoning About Rational Agents*. MIT Press, Cambridge, MA.