An Extension of BDI_{CTL} with Functional Dependencies and Components

Mehdi Dastani[1] and Leendert van der Torre[2]

[1] Institute of Information and Computer Sciences, Utrecht University
mehdi@cs.uu.nl
[2] Department of Artificial Intelligence, Vrije Universiteit Amsterdam
torre@cs.vu.nl

Abstract. This paper discusses the formal specification of properties that determine the behavior of component based BDI agents, i.e. classical BDI agents in which the mental attitudes are conditional and represented by interconnected components. Some properties, such as realism and commitment strategies, have already been discussed in the BDI literature and can be formally specified by for example Rao and Georgeff's BDI_{CTL} formalism. Other properties are specific to component based cognitive agents and cannot be specified by existing BDI_{CTL} formalisms. We focus here on the so-called functional dependencies between mental attitudes where a mental attitude is considered to be a function of one or more other mental attitudes. To formally specify the properties of functional dependencies we extend Rao and Georgeff's BDI_{CTL} formalism. In particular, for functional dependencies we introduce 'only belief', 'only desire' and 'only intend' operators in the tradition of Levesque's 'all I know' operator, and for components we distinguish between 'belief in' and 'belief out', 'desire in' and 'desire out', and 'intention in' and 'intention out' operators. We show how our extended formalism can be used to specify functionality properties such as conservativity, monotonicity, and self-boundedness, as well as properties related to the connections between and control of the components.

1 Introduction

A cognitive agent is a computational entity that has a mental state consisting of mental attitudes such as beliefs, desires, and intentions. Such an agent senses the environment and reacts to it based on several factors such as a rational balance between its mental attitudes [6, 13], how these mental attitudes are initiated and processed [3, 4, 16], and how information and control flow in such a computational entity [8, 17, 18].

In agent oriented software methodology there is no consensus on the tools and concepts to specify and verify the behavior of cognitive agents [6, 8, 13, 15]. The BDI approach [6, 13] provides a specification and verification framework for cognitive agents. The central theme in specifying the behavior of cognitive agents is the rational balance between mental attitudes. For example, for an agent to have realistic behavior, its desires should not conflict with its beliefs. The BDI_{CTL} formalism [6, 7, 13, 15] is a system that can be used to formally specify and reason about the behavior of cognitive agents. BDI_{CTL} is a multi-modal logic that includes temporal operators of so-called computational tree logic (CTL) combined with three epistemic modal operators for the agent's beliefs, desires, and intentions. The behavior of a cognitive agent is then specified by characterizing these modalities and their relations. For example, the axiom $D(\phi) \rightarrow \neg B(\neg\phi)$ expresses what Rao and Georgeff call weak realism. This axiom states that if an agent desires to bring about a proposition, then it does not believe the negation of it.

A drawback of the existing specification and verification tools for cognitive agents, such as BDI_{CTL} formalism, is that the specification of agent behavior is at an abstract level and merely in terms of a rational balance between mental attitudes. These approaches abstract from properties that may influence the correct behavior of cognitive agents such as how mental attitudes are initiated and processed, and how data and control flow when agents deliberate. We are motivated by drawbacks of the BDI_{CTL} formalism to specify properties for the component-based cognitive agents [3, 8] where agents mental attitudes are conditional and represented as components. Note that motivational attitudes such as desires are usually represented by conditionals (or rules), which has been studied in particular in deontic logic [1, 12].

One type of properties that are abstracted from in the BDI approach is based on the functional dependency between mental attitudes. We say that there is a functional dependency between two mental attitudes if one of them completely determines the other. For example, the dependency between the agent's beliefs (B) and its intentions (I) is functional if the intentions are determined by the beliefs, i.e. if there is a function f such that regardless of the states of the other components we have $I = f(B)$. Note that this particular function may be required to be non-monotonic since intentions do not persists under growing beliefs. For instance, an agent who believes the weather is hot and intends to go to the beach if the weather is hot may intend to go to the beach, but if the agent also believes that he must work and that he cannot go to the beach if he works, then the agent may not intend to go to the beach anymore. In general, the functional dependence between agent's mental attitudes can be specified by functional properties such as conservativity, monotonicity, and self-boundedness [18].

Another type of properties that is abstracted from in the BDI approach is based on the design details of the agent, i.e. how data and control flow in an agent. To illustrate this type of properties, consider the interpreter of the proposed PRS architecture for BDI agents [9]. The PRS interpreter assumes that at any particular time certain goals are active in the system and certain beliefs are held in the system database. Given these beliefs and goals, a subset of plans may be relevant. The exact way in which goals are activated, beliefs are generated, and plans are selected influences the agent's correct behavior. Therefore, it is important to specify properties concerning the flow of data and control in such agents. We are interested in particular in component-based BDI agents, such as Broersen et al.'s BOID system [4], in which each component, when activated, determines its representing mental attitude for a given context.

The aim of this paper is to extend the BDI_{CTL} formalism to specify functional dependencies between agent's mental attitudes as well as properties related to components such as the flow of data and control among components. We extend the BDI_{CTL} formalism with:

- additional modal operators that enable the specification of functional dependencies, based on all-I-know like operators [11].
- additional modal operators that enable the distinction between input and output of components.

This paper is organized as follows. In section 2, we discuss some examples of component based cognitive agents, their possible traces, and various types of properties of these traces. The examples motivate functional dependencies between mental attitudes and the use of components to represent them. They illustrate also the lack of expressiveness of BDI_{CTL} which forms our motivation to extend it. In section 3, we introduce an extension of Rao and Georgeff's BDI_{CTL} formalism that can express various types of trace properties including some properties of functional dependencies among mental attitudes. This formalism can be used as the specification and verification language for component based cognitive agents. In section 4, we recapitulate suggested trace properties of component based cognitive agents and formalize these properties in terms of the extended BDI_{CTL} formalism.

2 Motivating Examples

In component based cognitive agents [2, 3] each mental attitude, which is represented by a component, can be considered as a process having functional behavior, i.e. it generates some output based on the input that it receives from other mental attitudes. Some properties of the functional behavior can be described by functionality descriptions introduced in [18]. Typical properties are whether the input of a component is included in its output, or whether it supports reasoning by cases. Components are related to each other in the sense that their inputs are the outputs of other components. The properties of these relations may involve the possibilities for information exchange between components. For example, in some cases it may make sense for the belief inputs to include the desire outputs while it may make no sense for the desire inputs to include the belief outputs. A typical example is when an agent desires to go to the dentist and believes that going to the dentist implies pain. In this case, the belief inputs can include the desire outputs to generate the consequence of going to the dentist (i.e. having pain), but the desire inputs should not include the belief outputs since the agent does not desire the belief consequences, i.e. the agent does not desire to have pain. Another more complex property of the relation between components is the realism property according to which desires are overridden by beliefs [5, 16]. This property characterize an ordering relation between possible inputs of the belief component. Realism is defined as the property that allows the belief inputs to include all belief outputs before including a desire output.

The functional behavior of mental attitudes gives rise to a variety of functional properties that influence the behavior of agents. For example, in certain applications one may want to specify the belief component as having monotonic behavior with respect to the desire component, i.e. if the belief component outputs $cannot\_work$ for the input $visit\_family$ at any time point, then it should also output $cannot\_work$ for the input $visit\_family \wedge eat$ or in general for the input $visit\_family \wedge x$ for any propositional formula x and at any time point. Note that we can also require the monotonicity of mental attitudes through time in the sense that if a component outputs p at time t, then it will output p at all time moments $t' > t$. This monotonicity property is not a functional property of components since the time moment is not the input of components.

2.1 Functional Dependencies between Mental Attitudes

The behavior of a BDI agent is specified in terms of properties of its possible traces, i.e. sequences of states. We distinguish between concrete and abstract trace properties. Concrete properties specify relations between particular propositions (if I believe that it is hot then I desire a beer') whereas abstract properties specify relations between mental attitudes for all possible propositions ('for any belief ϕ and desire ψ, $\phi \wedge \psi$ is consistent'). Concrete properties are formalized as assumptions or premises, whereas abstract properties are formalized as axioms or theorems. The formal difference is that we may apply uniform substitution to theorems but not to premises.

Definition 1. *A state of a cognitive agent consists of the states of its mental attitudes. A trace or run of a cognitive agent is a sequence of agent states through time. Trace properties can be either concrete or abstract. A concrete property is a property that is used as a premise, and an abstract property is a property which is used as a theorem.*

Example 1 illustrates two concrete properties. In the following examples we ignore the nesting of mental attitudes (e.g. agent believes that he desires that etc.) such that the states are characterized by propositional formulae.

Example 1. Let S^B and S^D indicate the belief and desire states. The following trace starts at time t. The agent believes p at t, it believes $\neg p$ at $t+3$, it desires $\neg r \wedge s$ at $t+1$, etc.

$time$	t	$t+1$	$t+2$	$t+3$	$t+4$
S^B	p	p	p	$\neg p$	$\neg p$
S^D	$\neg r \wedge \neg s$	$\neg r \wedge s$	$r \wedge s$	$\neg r \wedge s$	$\neg r \wedge \neg s$

A concrete property can be stated as: for all traces if at some state the agent believes $\neg p$ and desires s, then the agent desires $\neg s$ at the next state. Another concrete property can be stated as: for all traces if at some state the agent desires $\neg r$ and believes p, then the agent desires r at the next state. Obviously, the first concrete property holds for this trace example while the second does not hold.

The following example illustrates an abstract property, which is the realism property defined by Cohen and Levesque [6].

Example 2. Consider the following trace.

$time$	t	$t+1$	$t+2$	$t+3$
S^B	p	$\neg p$	r	r
S^D	$p \wedge q$	$\neg p \wedge r$	$p \wedge r$	$\neg p \wedge r$

If we represent states by logically closed sets, then the realism property can be stated as: for all traces and at every state the desires S^D are a superset of the beliefs S^B.

Note that this kind of realism, in which the desires S^D are a superset of the beliefs S^B, may be counterintuitive in many applications. Rao and Georgeff therefore introduce several other kinds of realism, such as their weak realism mentioned in the introduction, see [14] for a discussion. In this paper we no further discuss realism, but we are primarily interested in functional dependencies.

The following example illustrates a functional dependency between beliefs and desires.

Example 3. Consider the following trace of an agent that believes q if it desires p.

$time$	t	$t+1$	$t+2$	$t+3$
S^D	s	p	p	$r \wedge s$
S^B	u	q	q	u

This trace can be described by a function $S^B(t) = f(S^D(t))$, because for identical desires the agent has identical beliefs. Note that the agent's beliefs are determined by its desires, which seems to suggest that the agent is not capable of making observations.

The most interesting functional dependencies are the ones where a mental attitude depends on several other mental attitudes. The following example illustrates such a case.

Example 4. Consider the following trace, in which the beliefs are either a function of the desires or a function of the intentions.

$time$	t	$t+1$	$t+2$	$t+3$
S^D	s	p	p	s
S^I	p	s	p	p
S^B	q	q	q	r

The beliefs are not a function of both the desires and the intentions, because there is no function f such that $S^B(t) = f(S^D(t), S^I(t))$. The counterexample is that at moment t the desire is s, the intention is p and the belief is q, whereas at moment $t+3$ for identical desires and intentions the belief is r.

However, there may be another way in which the trace of the latter example is based on a functional dependency between beliefs, desires, intentions. This is the case when the belief at the moment t depends on the intention, at the moment $t+1$ on the desire, at the moment $t+2$ on either the desire or the intention, and at the moment $t+3$ on the desire. That is, we have two functions f and g such that $S^B(t) = f(S^D(t))$ or $S^B(t) = g(S^I(t))$. This is what we consider in the following subsection, using the notion of component.

2.2 Components Based Cognitive Agents

Like the states of BDI agents, a state of a component based cognitive agent with conditional mental attitudes consists of the state of its mental attitudes. However, unlike the states of BDI agents, a state of component based agent consists of input and output states of each mental attitude.

Definition 2. *A state of a component-based cognitive agent consists of the input and output states of its mental attitudes. Traces and their (concrete and abstract) properties are as defined before. We write S^C_{in} and S^C_{out} for the input and output states of the mental attitude that is represented by component C, respectively.*

The output of a component is assumed to represent the mental attitude that correspond with that component, e.g. S^B_{out} represents the belief state S^B. The input of a component is the output of another component. It is considered as the context of the component's output and thus as the context of the mental attitude represented by the component's output. For the previous examples, this consideration implies that we replace S^B by S^B_{out}, S^D by S^D_{out} and S^I by S^I_{out}. The output of a component can be the input of another component. For example, in Example 3 the input of the belief component was the output of the desire component. Note that this trace seems to suggest that the belief component outputs q whenever it receives p as input. This is illustrated by the following trace.

$time$	t	$t+1$	$t+2$	$t+3$
S^D_{out}	s	p	p	$r \wedge s$
S^B_{in}	s	p	p	$r \wedge s$
S^B_{out}	u	q	q	u

The input of a component can originate from various components. For example, in Example 4 the input of the belief component can be the output of the desire or the intention component. The crucial property of the following trace is that S^B_{in} is either S^D_{out} or S^I_{out}, and that S^B_{out} functionally depends on S^B_{in}. Note that in this trace the belief component outputs q whenever it receives p as input.

$time$	t	$t+1$	$t+2$	$t+3$
S^D_{out}	s	p	p	s
S^I_{out}	p	t	p	p
S^B_{in}	p	p	p	s
S^B_{out}	q	q	q	r

In this example, the belief output state depends on two other output states: the desire and intention output states. In such cases, one may abstract over the output-states on which a component C depends and consider only the input-state of the component C. In example 4, one may use the input-state of the belief component S^B_{in} instead of S^D_{out} or S^I_{out}. This implies that agent traces reflect choices that have been made by the agent's control mechanism, i.e. which component provides the input of a component at a certain moment.

Finally, the functional dependencies between mental attitudes may have certain properties. A functional property, such as monotonicity, is a property of the input-states S^C_{in} and the output-states S^C_{out} of component C. For example, belief is a function of desire (i.e. desire is the input of the belief) when the consequence of desires are considered. One may then want to specify that this function is monotonic.

Example 5. The following trace satisfies monotonicity of beliefs as a function of desires. If for any time moment t we have $\phi = S^B_{in}(t)$ and $\psi = S^B_{out}(t)$, then we have for any t' that if $S^B_{in}(t')$ implies ϕ then $S^B_{out}(t')$ implies ψ. If the input grows, then the output grows too.

$time$	t	$t+1$	$t+2$	$t+3$
$S^B_{in} = S^D_{out}$	p	$\neg q$	$p \wedge q$	$\neg q \wedge r$
S^B_{out}	s	u	s	$s \wedge u$

The examples discussed in this section illustrated the functional nature of mental attitudes and their properties. Moreover, we illustrated that a mental attitude may depend on more than one mental attitude and that agent control mechanism determines the dependence of mental attitudes at each state. In the following section we consider the formalization of these examples.

3 The Specification Language for Component Based Agents

In this section we briefly repeat Rao and Georgeff's formalism [14] and we extend it with additional modalities to capture the mutual dependence of the agent's mental attitudes. The introduction of these modalities makes it possible to specify the functional dependencies between agents mental attitudes.

3.1 $\mathrm{BDI}_{\mathrm{CTL}}$ Formalism

We use an equivalent reformulation of $\mathrm{BDI}_{\mathrm{CTL}}$ presented by Schild [15]. In contrast to Schild, we only consider the single agent case. The CTL operators are used to quantify over possible traces and states, while the BDI operators are used to quantify over mental states. Intuitively, the CTL operator A stands for all possible traces, E for at least one possible trace, X for the next state, and U for until.

Definition 3. *[15, Def.4,6] The admissible formulae of $\mathrm{BDI}_{\mathrm{CTL}}$ are categorized into two classes, state formulae and path formulae.*

- *S1 Each primitive proposition is a state formula.*
- *S2 If α and β are state formulae, then so are $\alpha \wedge \beta$ and $\neg\alpha$.*
- *S3 If α is a path formula, then $E\alpha$ and $A\alpha$ are state formulae.*
- *S4 If α is a state formula, then $B(\alpha), D(\alpha), I(\alpha)$ are state formulae as well.*
- *P If α and β are state formulae, then $X\alpha$ and $\alpha \cup \beta$ are path formulae.*

The semantics of $\mathrm{BDI}_{\mathrm{CTL}}$ involves two dimensions: an epistemic and a temporal dimension. The truth of a formula depends on both the epistemic world w and the temporal state s. A pair $\langle w, s\rangle$ is called a situation in which $\mathrm{BDI}_{\mathrm{CTL}}$ formulae are evaluated. The relation between situations is traditionally called an accessibility relation (for beliefs) or a successor relation (for time).

Definition 4. *[15, Def.2,7] A Kripke structure $M = \langle W, \mathcal{R}_1, \ldots, \mathcal{R}_n, L\rangle$ is comprised of three components. The first component is an arbitrary non-empty set W containing all worlds relevant to M. The second component is a family of relations $\mathcal{R}_i \subseteq W \times W$. The remaining third component is an assignment function L. This function assigns a* *particular set of worlds to each primitive proposition. $L(p)$ contains all those worlds in which p holds.*

A Kripke structure $M = \langle \mathcal{S}, \mathcal{R}, \mathcal{B}, \mathcal{D}, \mathcal{I}, L\rangle$ forms a situation structure if each of the following three conditions is met.

1. *$\mathcal{S}$ is a set of situations.*
2. *$w = w'$ whenever $\langle w, s\rangle \mathcal{R} \langle w', s'\rangle$.*
3. *$s = s'$ whenever $\langle w, s\rangle \mathcal{B} \langle w', s'\rangle$ and similarly for $\mathcal{D}$ and $\mathcal{I}$.*

Schild [15, Section 3] does not present the semantic relation of CTL, but only the one of its extension CTL* (as well as the one of the μ-calculus). This extension is not considered in this paper. A speciality of both CTL and CTL* is that some formulae are not interpreted relative to a particular situation. These are the path formulae. What is relevant here are full paths. The reference to M is omitted whenever it is understood.

Definition 5. *A full path in $M = \langle \mathcal{S}, \mathcal{R}, \mathcal{B}, \mathcal{D}, \mathcal{I}, L\rangle$ is an infinite sequence $\chi = \delta_0, \delta_1, \delta_2, \ldots$ such that for every $i \geq 0$, δ_i is an element of $\mathcal{S}$ and $\delta_i \mathcal{R} \delta_{i+1}$. We say that a full path starts at δ iff $\delta_0 = \delta$. We use the following convention. If $\chi = \delta_0, \delta_1, \delta_2, \ldots$ is a full path in M, then χ^i $(i \geq 0)$ denotes δ_i.*

Let $M = \langle \mathcal{S}, \mathcal{R}, \mathcal{B}, \mathcal{D}, \mathcal{I}, L\rangle$ be a situation structure, δ a situation, and χ a full path. The semantic relation $\models$ for BDI$_{CTL}$ is then defined as follows:

S1 $\delta \models p$ *iff* $\delta \in L(p)$.
S2 $\delta \models \alpha \wedge \beta$ *iff* $\delta \models \alpha$ *and* $\delta \models \beta$.
$\delta \models \neg\alpha$ *iff* $\delta \models \alpha$ *does not hold.*
S3 $\delta \models E\alpha$ *iff there is at least one full path* χ *in* M *starting at* δ *s.t.* $\chi \models \alpha$.
$\delta \models A\alpha$ *iff for every full path* χ *in* M *starting at* $\delta, \chi \models \alpha$.
S4 $\delta \models B(\alpha)$ *iff for every* $\delta' \in \mathcal{S}$ *such that* $\delta \mathcal{B} \delta', \delta' \models \alpha$.
$\delta \models D(\alpha)$ *iff for every* $\delta' \in \mathcal{S}$ *such that* $\delta \mathcal{D} \delta', \delta' \models \alpha$.
$\delta \models I(\alpha)$ *iff for every* $\delta' \in \mathcal{S}$ *such that* $\delta \mathcal{I} \delta', \delta' \models \alpha$.
P $\chi \models X\alpha$ *iff* $\chi^1 \models \alpha$.
$\chi \models \alpha \cup \beta$ *iff there is a* $i \geq 0$ *s.t.* $\chi^i \models \beta$ *and for all* $j (0 \leq j < i), \chi^j \models \alpha$.

We write $\Diamond\alpha$ (or $F\alpha$) for $\top \mathbf{U} \alpha$ (read as '*for at least one state in the future*') and $\Box\alpha$ (or $G\alpha$) for $\neg\Diamond\neg\alpha$ (read as '*for all states in the future*').

3.2 Specifying Functional Dependencies

In order to specify the functional relation between two mental attitudes of a component based agent, we should guarantee the following property: if in two situations one mental attitude is the same (the domain), then the other mental attitude should be the same too (the scope). For example, such axioms should be able to express that agent's belief is a function of agent's desire (see example 3), i.e.

$$\{\psi \mid M, \delta \models B(\psi)\} = f(\{\phi \mid M, \delta \models D(\phi)\})$$

It seems that the property that agent's belief depends functionally on its desire can be formally specified by the following axiom.

$$E\Diamond(\ D(\phi) \wedge B(\psi)\ ) \rightarrow A\Box(\ D(\phi) \rightarrow B(\psi)\ )$$

or alternatively

$$A\Box(\ D(\phi) \rightarrow \neg B(\psi)\ ) \vee A\Box(\ D(\phi) \rightarrow B(\psi)\ )$$

However, this formula is too strong. For example, substitute a tautology for ϕ, then assuming the usually accepted seriality axiom $D(true)$, the above formula would entail the following highly problematic property, which says that beliefs cannot change.

$$E\Diamond B(\psi) \rightarrow A\Box B(\psi)$$

There are several solutions to this problem. Here we follow a proposal of Rao and Georgeff which uses 'only belief' operators [10]. We also extend BDI_{CTL} with distinct epistemic modal operators for input and output. In particular, we split each epistemic modal operator X of the BDI_{CTL} formalism into two modal operators: X_{in} and X_{out}. Note that X_{in} can be considered as the context of the mental attitude that is represented by the epistemic modal operator X and, thus, $X_{in}(\phi)$ can be interpreted as ϕ is the context of X. For uniformity, we use X_{out} as the alternative notation for the modal operator X.

Definition 6. *The admissible formulae of extended BDI_{CTL} are generated by the rules of BDI_{CTL}, where S4 is replaced by the following rule:*

S4 If α is a state formula, then $B_{in}(\alpha)$, $D_{in}(\alpha)$, $I_{in}(\alpha)$, $B^o_{in}(\alpha)$, $D^o_{in}(\alpha)$, $I^o_{in}(\alpha)$, $B_{out}(\alpha)$, $D_{out}(\alpha)$, $I_{out}(\alpha)$, $B^o_{out}(\alpha)$, $D^o_{out}(\alpha)$, and $I^o_{out}(\alpha)$ are state formulae as well.

In order to define the semantics of these additional modal operators we extend the BDI_{CTL} situation structure to include an additional specific accessibility relation for each mental attitude in each state. These accessibility relations are denoted by $\mathcal{B}_{in}, \mathcal{D}_{in}$, and $\mathcal{I}_{in}$. These accessibility relations capture the input states of each mental attitude. Note that this accessibility relation formalizes parts of the agent's control that determine the input of a certain component at a certain time. Moreover, we use accessibility relations $\mathcal{B}_{out}, \mathcal{D}_{out}$, and $\mathcal{I}_{out}$ instead of the accessibility relations $\mathcal{B}, \mathcal{D}$ and $\mathcal{I}$ from BDI_{CTL}. Extended structures thus contain accessibility relations for input and output of each component.

Definition 7. *A Kripke structure $M = \langle \mathcal{S}, \mathcal{R}, \mathcal{B}_{in}, \mathcal{D}_{in}, \mathcal{I}_{in}, \mathcal{B}_{out}, \mathcal{D}_{out}, \mathcal{I}_{out}, L\rangle$ forms an extended situation structure if each of the following three conditions is met.*

1. *$\mathcal{S}$ is a set of situations.*
2. *$w = w'$ whenever $\langle w, s\rangle \mathcal{R} \langle w', s'\rangle$.*
3. *$s = s'$ whenever $\langle w, s\rangle \mathcal{B}_{in} \langle w', s'\rangle$ and similarly for $\mathcal{D}_{in}$, $\mathcal{I}_{in}$, $\mathcal{B}_{out}$, $\mathcal{D}_{out}$ and $\mathcal{I}_{out}$.*

The semantic relation of extended BDI_{CTL} contains clauses for the twelve epistemic modal operators that we have defined.

Definition 8. *The semantic relation $\models$ of extended BDI_{CTL} is defined like the semantic relation for BDI_{CTL}, where $S4$ is replaced by the following clauses:*

S4 $\delta \models B_{in}(\alpha)$ *iff for every* $\delta' \in \mathcal{S}$ *such that* $\delta \mathcal{B}_{in} \delta', \delta' \models \alpha$.
$\delta \models D_{in}(\alpha)$ *iff for every* $\delta' \in \mathcal{S}$ *such that* $\delta \mathcal{D}_{in} \delta', \delta' \models \alpha$.
$\delta \models I_{in}(\alpha)$ *iff for every* $\delta' \in \mathcal{S}$ *such that* $\delta \mathcal{I}_{in} \delta', \delta' \models \alpha$.
$\delta \models B^o_{in}(\alpha)$ *iff for every* $\delta' \in \mathcal{S}, \delta \mathcal{B}_{in} \delta'$ *iff* $\delta' \models \alpha$.
$\delta \models D^o_{in}(\alpha)$ *iff for every* $\delta' \in \mathcal{S}, \delta \mathcal{D}_{in} \delta'$ *iff* $\delta' \models \alpha$.
$\delta \models I^o_{in}(\alpha)$ *iff for every* $\delta' \in \mathcal{S}, \delta \mathcal{I}_{in} \delta'$ *iff* $\delta' \models \alpha$.
$\delta \models B_{out}(\alpha)$ *iff for every* $\delta' \in \mathcal{S}$ *such that* $\delta \mathcal{B}_{out} \delta', \delta' \models \alpha$.
$\delta \models D_{out}(\alpha)$ *iff for every* $\delta' \in \mathcal{S}$ *such that* $\delta \mathcal{D}_{out} \delta', \delta' \models \alpha$.
$\delta \models I_{out}(\alpha)$ *iff for every* $\delta' \in \mathcal{S}$ *such that* $\delta \mathcal{I}_{out} \delta', \delta' \models \alpha$.
$\delta \models B^o_{out}(\alpha)$ *iff for every* $\delta' \in \mathcal{S}, \delta \mathcal{B}_{out} \delta'$ *iff* $\delta' \models \alpha$.
$\delta \models D^o_{out}(\alpha)$ *iff for every* $\delta' \in \mathcal{S}, \delta \mathcal{D}_{out} \delta'$ *iff* $\delta' \models \alpha$.
$\delta \models I^o_{out}(\alpha)$ *iff for every* $\delta' \in \mathcal{S}, \delta \mathcal{I}_{out} \delta'$ *iff* $\delta' \models \alpha$.

Rao and Georgeff [10] use the ‘only’ operators to define the following intention persistence axioms. We may again read I as I_{out} and B as B_{out}.

$$(IAXE\epsilon \wedge AXB^oA\epsilon) \rightarrow AXIE\epsilon$$

$$(I^oAXE\epsilon \wedge AXBA\epsilon) \rightarrow AXIE\epsilon$$

We use these new epistemic modal operators to formalize the functional dependence between mental attitudes in terms of corresponding input and output modal operators.

Definition 9. *The functional dependence of agent’s belief on other mental attitudes is represented by the following axiom:*

$$E\Diamond(\ B^o_{in}(\phi) \wedge B_{out}(\psi)\ ) \rightarrow A\Box(\ B^o_{in}(\phi) \rightarrow B_{out}(\psi)\ )$$

This holds for other mental attitudes as well.

In $\mathrm{BDI}_{\mathrm{CTL}*}$ we can also express the following weaker variant of functional dependence called trace functional dependence. It says that within traces similar inputs lead to similar outputs. The formula is not well formed in the limited language of $\mathrm{BDI}_{\mathrm{CTL}}$.

Definition 10. *The trace functional dependence of agent’s belief on other mental attitudes is represented in* $\mathrm{BDI}_{\mathrm{CTL}*}$ *by the following axiom:*

$$A(\Diamond(\ B^o_{in}(\phi) \wedge B_{out}(\psi)\ ) \rightarrow \Box(\ B^o_{in}(\phi) \rightarrow B_{out}(\psi)\ ))$$

Note that Rao and Georgeff’s notion of ‘only belief’ which we used in definition 8 is very strong, because it also contains nested beliefs. For some applications, it may be more accurate to restrict the set of beliefs considered in the ‘only’ operator to the propositional fragment. In this paper we do not consider this option. Moreover, in this paper we do not consider complexity issues (see for example the work of Rao and Georgeff for this issue). In the remainder of this paper, we consider which kind of properties can be formalized in the extended $\mathrm{BDI}_{\mathrm{CTL}}$ logic.

4 Properties of Component Based Cognitive Agents

Properties of the traces of component based cognitive agents are related to a rational balance between mental attitudes, their functional dependence, and other properties that are specific to this type of agents. Some existing properties concerning a rational balance between mental attitudes can be formulated as expressions of the extended BDI_{CTL} formalism in a straightforward way in terms of the output states. For example, the Cohen and Levesque's realism in Example 2 can be expressed by $B_{out}(\phi) \rightarrow D_{out}(\phi)$. Also properties introduced by Rao and Georgeff [13] can be specified by using B_{out}, D_{out}, and I_{out} instead of B, D, and I. For example, single minded commitment strategy can be specified as follows:

$$I_{out}(\mathbf{A}\Diamond\phi) \rightarrow \mathbf{A}(\ I_{out}(\mathbf{A}\Diamond\phi)\ \mathbf{U}\ (B_{out}(\phi) \vee\ \neg B_{out}(\mathbf{E}\Diamond\phi)))$$

In this section, we discuss functional ad control properties taken from [18] and we discuss how they can be formally specified in the extended BDI_{CTL} formalism.

4.1 Functional Properties

The functional dependencies between mental attitudes may be required to have certain properties. The first property we discuss here is conservativity which is the property of a mental attitude whose input is included in its output.

Definition 11 (Conservative). *The belief of an agent is conservative iff the following formula is a theorem.*

$$B_{in}(\phi) \rightarrow B_{out}(\phi)$$

The same holds for other desire and intention.

The following example illustrates conservativity.

Example 6. Let f_B be a function that maps belief input to belief output, i.e. $S^B_{out} = f_B(S^B_{in})$. The following trace illustrates that the output of the agent's beliefs functionally depends on its input and that this function (f_B) is conservative. At every moment, the output implies the input.

$time$	t	$t+1$	$t+2$
S^B_{in}	p	r	$r \wedge p$
S^B_{out}	$p \wedge q$	$\neg s \wedge r$	$r \wedge p \wedge q$

The second property of functional dependence is the monotonicity property. According to this property a mental attitude persists under the grow of its context (input). The monotonicity property is defined on the basis of the modal 'only belief' operator. However, in contrast to the definition of functional dependence, the 'only' operator is used only once. Note that monotonicity implies functional dependence, because B^o_{in} implies B_{in} (but not vice versa!).

Definition 12 (Monotonic). *The belief of an agent is monotonic iff the following formula is a theorem.*

$$E\Diamond(B^o_{in}\phi \wedge B_{out}\psi) \rightarrow A\Box(B_{in}\phi \rightarrow B_{out}\psi).$$

The monotonicity of other mental attitudes can be specified in a similar way.

Monotonicity is illustrated by the following example.

Example 7. The following trace satisfies monotonicity of belief: the input at moment $t+2$ implies the input at moment t, and therefore the output at moment $t+2$ implies the output at t. Similarly for $t+3$ and $t+1$.

$time$	t	$t+1$	$t+2$	$t+3$
S^B_{in}	p	$\neg p$	$p \wedge r$	$\neg p \wedge s$
S^B_{out}	q	$\neg q$	q	$\neg q$

Another property of functional dependence is called self-boundedness. The self-bounded property of mental attitudes can be specified as follows. This property is a reformulation of transitivity.

Definition 13 (Self-bounded). *The belief of an agent is self-bounded iff the following formula is a theorem.*

$$(E\Diamond(B^o_{in}\phi \wedge B_{out}\psi) \wedge E\Diamond(B^o_{in}\psi \wedge B_{out}\theta)) \rightarrow A\Box(B^o_{in}\phi \rightarrow B_{out}\theta)$$

It is weakly self-bounded iff the following formula is a theorem.

$$(E\Diamond(B^o_{in}\phi \wedge B_{out}\psi) \wedge E\Diamond(B^o_{in}(\phi \wedge \psi) \wedge B_{out}\theta)) \rightarrow A\Box(B^o_{in}\phi \rightarrow B_{out}\theta)$$

The self-boundedness of other mental attitudes can be specified in a similar way.

The following example illustrates self-boundedness.

Example 8. The following trace illustrates the self-bounded property of the belief component: q is in the output of p (moment t) and r is in the output of q (moment $t+1$), and therefore q is also in the output of p (moment $t+2$).

$time$	t	$t+1$	$t+2$
S^B_{in}	p	q	p
S^B_{out}	q	r	r

Finally, the last functional property we discuss is called well-informedness. This property is related to reasoning by cases.

Definition 14 (Reasoning by cases). *The belief of an agent supports reasoning by cases iff the following formula is a theorem.*

$$(E\Diamond(B^o_{in}\phi \wedge B_{out}\theta) \wedge E\Diamond(B^o_{in}\psi \wedge B_{out}\theta)) \rightarrow A\Box(B^o_{in}(\phi \vee \psi) \rightarrow B_{out}\theta)$$

Reasoning by cases for other mental attitudes can be specified in a similar way.

The following example illustrates reasoning by cases.

Example 9. The following trace illustrates the reasoning by cases property of the belief component: r is in the output of p (moment t) and q (moment $t+1$), and therefore also in the output of $p \vee q$ (moment t and $t+2$).

$time$	t	$t+1$	$t+2$
S^B_{in}	p	q	p
S^B_{out}	$q \wedge r$	r	$q \wedge r$

4.2 Control Properties

The flow of information is determined by the agent's control mechanism. For example, the output of the belief component may become the input of the desire component in the sense that the desire formulae are activated based on the active beliefs. This type of control can be specified in terms of the input and output modal operators. It is assumed that one mental attitude is the input of a second mental attitude at the next state.

Definition 15. *The following are properties of the agent's control mechanism.*

Desires Contextualizes Beliefs: The output of desires is the input of beliefs at the next state, which implies that belief consequences of desires can be generated, i.e.

$$D_{out}\phi \rightarrow XB_{in}\phi.$$

Beliefs Contextualizes Beliefs: The output of beliefs is its own input at the next state, which implies that all consequences of the active beliefs are generated (closure), i.e.

$$B_{out}\phi \rightarrow XB_{in}\phi.$$

These control properties can be adapted by stating that the output of a component should be in the input of a component at some moment in the future. Moreover, the the control properties can be extended with conflict checks, analogous to the checks in persistence of intentions (see e.g. [10]).

4.3 A Classification of Component Based Agent Properties

The properties of component-based cognitive agents can be specified along several dimensions. Three important dimensions are time, input/output dependency, and component dimensions.

The time dimension determines the time aspect involved in the properties. Properties can be distinguished along this dimension into three types: No-time, Adjacent-time-points, future directed types. For example, functional properties of a single component is a property that does not involved time, some static properties do involve adjacent time points, and commitment strategy properties involve future directed time.

The component dimension determines the aspect of properties that are related to the number of components involved in a property. Some properties are properties of one single component while others involve more than one component. For example, functionality properties involve one component while some of control properties involve two components.

The input/output dependency dimension determines those aspects of properties that are related to the input/output dependency of components. Four types of properties that can be distinguished along this dimension are distinguished: input-output, output-input, input-input, and output-output dependencies. For example, the functionality properties have the input-output type while standard BDICTL properties such as static and dynamic properties have the output-output types.

Let C_{in} and C_{out} be the input and output of the component C. The following table shows some examples of properties specified along the proposed three dimensions.

Time	Component	Input/Output	Property Type
No	C	C_{in}, C_{out}	Functional property
Adjacent	C^1, C^2	C^1_{out}, C^2_{in}	Control property
Future-directed	C^1, C^2	C^1_{out}, C^2_{out}	Commitment Strategies

5 Conclusion

The aim of this paper is to develop a specification language for component based cognitive agents. The mental attitudes of such cognitive agents are conditional, i.e. each mental attitude is represented by a component that receives an input and generates an output. The conditional nature of these mental attitudes implies functional dependencies between them. In fact, one mental attitude becomes the context of another mental attitude such that each mental attitude can be considered as a function of its context.

Several types of properties for component based cognitive agents are discussed, including functional properties of the mental attitudes. To specify the behavior of such agents the $\mathrm{BDI}_{\mathrm{CTL}}$ formalism is extended with additional epistemic modal operators to capture the input and output-states of each component. Based on these additional modalities functional dependencies between mental attitudes are specified. The extended $\mathrm{BDI}_{\mathrm{CTL}}$ formalism can thus be used to specify and verify the behavior of component based cognitive agents.

A question for further research is the specification of compositional BOID agents [4] in the proposed formalism or an extension of it. It seems that the language has to be extended with at least obligations and goals (for the latter, see [7]). One particular problematic issue here is that in the BOID architecture several conflicting goals sets or extensions can exist in a state at the same time. How this can be formalized efficiently in an extension of $\mathrm{BDI}_{\mathrm{CTL}}$ is still an open problem.

References

1. C. Alchourrón. Philosophical foundations of deontic logic and the logic of defeasible conditionals. In J.-J. Meyer and R. Wieringa, editors, *Deontic Logic in Computer Science: Normative System Specification*, pages 43–84. John Wiley & Sons, 1993.

2. F. Brazier, C. Jonker, and J. Treur. Principles of component-based design of intelligent agents. *Data and Knowledge Engineering*, 41:1–28, 2002.
3. J. Broersen, M. Dastani, Z. Huang, J. Hulstijn, and L. van der Torre. The BOID architecture: Conflicts between beliefs, obligations, intentions and desires. In *Proceedings of he Fifth International Conference on Autonomous Agents (AA2001)*, pages 9–16. ACM Press, 2001.
4. J. Broersen, M. Dastani, J. Hulstijn, and L. van der Torre. Goal generation in the BOID architecture. *Cognitive Science Quarterly*, to appear. Special issue on Desires, goals, intentions, and values: Computational architectures.
5. J. Broersen, M. Dastani, and L. van der Torre. Realistic desires. *Journal of Applied Non-Classical Logics*, To appear.
6. P. Cohen and H. Levesque. Intention is choice with commitment. *Artificial Intelligence*, 42:213–261, 1990.
7. M. Dastani and L. van der Torre. Specifying the merging of desires into goals in the context of beliefs. In *Proceedings of The First Eurasian Conference on Advances in Information and Communication Technology (EurAsia ICT 2002)*, LNCS. Springer, 2002.
8. J. Engelfriet, C.M. Jonker, , and J. Treur. Compositional verification of multi-agent systems in temporal multi-epistemic logic. *Journal of Logic, Language and Information*, 11:195–225, 2002.
9. M. Georgeff and A. Lansky. Reactive reasoning and planning. *In Proceedings of the Sixth National Conference on Artificial Intelligence (AAAI-87)*, pages 677–682, 1987.
10. M. Georgeff and A. Rao. The semantics of intention maintenance for rational agents. In *Proceedings of 14th Int. Joint Conference on Artificial Intelligence (IJCAI–95)*, pages 704–710, 1995.
11. H. Levesque. All I know: a study in auto-epistemic logic. *Articial Intelligence*, 42(3):263–309, 1990.
12. D. Makinson. On a fundamental problem of deontic logic. In P. McNamara and H. Prakken, editors, *Norms, Logics and Information Systems. New Studies on Deontic Logic and Computer Science*, pages 29–54. IOS Press, 1999.
13. A. Rao and M. Georgeff. Modeling rational agents within a BDI architecture. In R. Fikes and E. Sandewall, editors, *Proceedings of the Second Conference on Knowledge Representation and Reasoning*, pages 473–484. Morgan Kaufman, 1991.
14. A. Rao and M. Georgeff. Decision procedures for BDI logics. *Journal of Logic and Computation*, 8:293–342, 1998.
15. K. Schild. On the relationship between BDI logics and standard logics of concurrency. *Autonomous Agents and Multi agent Systems*, 3:259–283, 2000.
16. R. Thomason. Desires and defaults: A framework for planning with inferred goals. In *Proceedings of Seventh International Conference on Knowledge Representation and Reasoning (KR'2000)*, pages 702–713, 2000.
17. J. Treur. Declarative functionality descriptions of interactive reasoning modules. In H. Boley and M.M. Richter, editors, *Processing Declarative Knowledge*, volume 567 of *LNAI*, pages 221–236. Springer, 1991.
18. J. Treur. Semantic formalisation of interactive reasoning functionality. *International Journal of Intelligent Systems*, 17:645–686, 2002. Revised and extended version of [17].