

The Interpreted System Model of Knowledge, Belief, Desire and Intention

Kaile Su¹, Xiangyu Luo¹, Abdul Sattar¹, Mehmet A Orgun²

¹Institute for Integrated and Intelligent Systems, Griffith University, Brisbane, Australia

²Department of Computing, Macquarie University, Sydney, Australia
{k.su, x.luo, a.sattar}@griffith.edu.au; mehmet@comp.mq.edu.au

ABSTRACT

We present a new model of knowledge, belief, desire and intention, called the interpreted KBDI-system model (or KBDI-model for short). The key point of the interpreted KBDI-system model is that we express an agent's knowledge, belief, desire and intention as a set of runs (computing paths), which is exactly a *system* in the interpreted system model, a well-known agent model due to Halpern and his colleagues. Our KBDI-model is *computationally grounded* in that we are able to associate a KBDI-model with a computer program, and formulas, involving agents' knowledge, belief, desire (goal) and intention, can be understood as properties of program computations. We present a sound and complete proof system with respect to our KBDI-model and explore how symbolic model checking techniques can be applied to model checking multi-agent systems with KBDI-models.

1. INTRODUCTION

The *possible worlds semantics* [6] is a fruitful approach to formalizing agent systems via modal logics. The well-known theory of intention [3] and the formalism of the belief-desire-intention paradigm [10], for example, are along this line. However, it is still not very clear how to obtain concrete agent models with the belief, desire and intention accessibility relations from specific agent programs.

The *interpreted system* model [4] offers a natural interpretation, in terms of the states of computer processes, to S5 epistemic logic. The salient point of the interpreted system model is that we are able to associate an interpreted system with a computer program, and formulas in epistemic logic that are valid with respect to the interpreted system can be understood as valid properties of program computations. In this sense, the interpreted system model is *computationally grounded* [14].

The aim of this paper is to present a computationally grounded model of knowledge, belief, desire and intention, called the interpreted KBDI-system model (or KBDI-model for short), by extend-

This work was supported by the Australian Research Council grant DP0452628, and partially by the NSFC grants 60496327, 10410638 and 60473004.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

AAMAS'06 May 8–12 2006, Hakodate, Hokkaido, Japan.

Copyright 2006 ACM 1-59593-303-4/06/0005 ...\$5.00.

ing the interpreted system model. The key point of the KBDI-model is that an agent's beliefs, desires and intentions as well as its knowledge is characterized as a set of runs (computing paths), which is exactly a computationally grounded *system* in the interpreted system model.

Intuitively, a KBDI-model includes a system $\mathcal{K}$ as in the interpreted system model, as well as, for each agent i , its beliefs $\mathcal{B}_i$, desires $\mathcal{D}_i$ and intentions $\mathcal{I}_i$, which are subsystems or subsets of $\mathcal{K}$. Agent i 's knowledge is determined by the set of those runs that are consistent with its local *observations* (or local state). Similarly, agent i 's beliefs are defined by the set of runs in $\mathcal{B}_i$ that are consistent with agent i 's observations. Agent i 's desires and intentions can also be defined in the same way. Intuitively, runs in $\mathcal{B}_i$ are possible computing paths from the viewpoint of the agent and those in $\mathcal{D}_i$ are the computing paths that the agent desires. Thus, it is reasonable to assume that $\mathcal{D}_i \subseteq \mathcal{B}_i$ because every desired computing path should be possible. Nevertheless, we need not assume that $\mathcal{I}_i \subseteq \mathcal{D}_i$ or even $\mathcal{I}_i \subseteq \mathcal{B}_i$ because an agent's intention may fail to achieve its goal and the actual computing path may be beyond the agent's belief even though the agent has chosen and completed an intentional series of actions.

2. KBDI LOGIC

This section introduces a multimodal logic of knowledge, belief, desire and intention, referred to as Observation-based KBDI logic (KBDI). As shown below, the semantics of KBDI logic is given in terms of the interpreted KBDI-system model. According to this semantics, the computation of agents' knowledge, belief, desire, and intention is based on agents' *observations*, that is, local states.

2.1 Interpreted systems

Consider a system composed of multiple (say n) agents in an environment. We represent the system's state or the *global state* as a tuple $(s_e, s_1, \dots, s_n)$, where s_e is the *environment's local state* and, for $1 \leq i \leq n$, s_i is *agent i 's local state*.

Let L_e be a set of possible local states of the environment and L_i a set of possible local states for agent i , for $i = 1, \dots, n$. We take $G = L_e \times L_1 \times \dots \times L_n$ to be the set of *reachable global states* of the system. A *run* r over G is a function from the time domain (the natural numbers in our case) to G . Thus, a run over G can be identified with a sequence of global states in G . A *point* is a pair (r, m) consisting of a run r and time m . Given a point (r, m) , we denote the first component of the tuple $r(m) = (s_e, s_1, \dots, s_n)$ by $r_e(m) (= s_e)$ and, for each i ($1 \leq i \leq n$), the $i + 1$ 'th component of the tuple $r(m)$ by $r_i(m) (= s_i)$. Thus, $r_i(m)$ is the local state of agent i in run r at time m .

The idea of the interpreted system semantics is that a run represents one possible computation of a system and a system may have

a number of possible runs, so we say a *system* is a set of runs. Assume that we have a set Φ of primitive propositions, which we can think of as describing basic facts about the system. An *interpreted system* I consists of a pair $(\mathcal{R}, \cdot)$, where $\mathcal{R}$ is a set of runs over a set of global states and $\cdot$ is a valuation function, which gives the set of primitive propositions true at each point in $\mathcal{R}$ [4].

For every agent i , let the notation $(r, u) \sim_i (r', v)$ denote that $r_i(u) = r'_i(v)$. Intuitively, $(r, u) \sim_i (r', v)$ means that (r, u) and (r', v) are indistinguishable to agent i . We also use the notation $(r, u) \stackrel{spr}{\sim}_i (r', v)$ to denote that $u = v$ and, for every $j \neq i$, $r_i(j) = r'_i(j)$ (here *spr* stands for *synchronous systems with perfect recall*).

Let KL_n denote the language of propositional logic augmented by the future-time connectives $\bigcirc$ (next) and $\mathbf{U}$ (until) and a modal (knowledge) operator K_i for each agent i . The language KL_n can be interpreted by using an interpreted system. The related satisfaction relationship $\models_{KL_n}$ is as follows: Given $I = (\mathcal{R}, \cdot)$ and a point (r, u) in $\mathcal{R}$, we define $(I, r, u) \models_{KL_n} \varphi$ by induction on the structure of φ . When φ is of the form $K_i\varphi$, $(I, r, u) \models_{KL_n} \varphi$ iff $(I, r', v) \models_{KL_n} \varphi$ for all (r', v) such that $(r, u) \sim_i (r', v)$.

2.2 Interpreted KBDI Systems

Given a set G of global states and system $\mathcal{K}$ over G , an agent's *mental state* over system $\mathcal{K}$ is a tuple $\langle \mathcal{B}, \mathcal{D}, \mathcal{I} \rangle$, where $\mathcal{B}$, $\mathcal{D}$ and $\mathcal{I}$ are systems (sets of runs over G) such that $\mathcal{I} \subseteq \mathcal{K}$ and $\mathcal{D} \subseteq \mathcal{B} \subseteq \mathcal{K}$. A *KBDI-system* is a structure $\langle \mathcal{K}, \mathcal{M}_1, \dots, \mathcal{M}_n \rangle$, where $\mathcal{K}$ is a system and for every i , $\mathcal{M}_i$ is agent i 's mental state over $\mathcal{K}$.

Assume that we have a set Φ of primitive propositions which describe basic facts about agents and their environment. An *interpreted KBDI-system* I consists of a pair $(\mathcal{S}, \cdot)$, where $\mathcal{S}$ is a KBDI-system and $\cdot$ is a valuation function, which gives the set of primitive propositions true at each point in G .

2.3 Syntax

Given a set Φ of propositional atoms, the language of KBDI logic is defined by the following BNF notations:

$$\begin{aligned} \langle wff \rangle ::= & \text{any element of } \Phi \mid \neg \langle wff \rangle \mid \langle wff \rangle \wedge \langle wff \rangle \mid \\ & \bigcirc \langle wff \rangle \mid \langle wff \rangle \mathbf{U} \langle wff \rangle \mid \\ & K_i \langle wff \rangle \mid B_i \langle wff \rangle \mid D_i \langle wff \rangle \mid I_i \langle wff \rangle \end{aligned}$$

Informally, $K_i\varphi$, $B_i\varphi$ and $D_i\varphi$ means that agent i knows, believes and desires φ , respectively, while $I_i\varphi$ denotes that φ holds under the assumption that agent i acts based on his intention. The formulas not containing modalities K_i , B_i , D_i and I_i ($i = 1, \dots, n$) are called linear-temporal logic (*LT*L) formulas.

2.4 Semantics

We now proceed to interpret KBDI logic formulas in terms of interpreted KBDI-systems. In the following, we inductively define the satisfaction relation $\models_O$ between a formula φ and a pair of interpreted KBDI-system and a point. Given an interpreted KBDI-system $I = (\mathcal{S}, \cdot)$, suppose that $\mathcal{S} = \langle \mathcal{K}, \mathcal{M}_1, \dots, \mathcal{M}_n \rangle$ and for every i , $\mathcal{M}_i = \langle \mathcal{B}_i, \mathcal{D}_i, \mathcal{I}_i \rangle$. Let r be a run in $\mathcal{K}$ and u a natural number, then we have that:

- $(I, r, u) \models_O K_i\varphi$ iff $(I, r', v) \models_O \varphi$ for those (r', v) such that $r' \in \mathcal{K}$ and $(r, u) \sim_i (r', v)$;
- $(I, r, u) \models_O B_i\varphi$ iff $(I, r', v) \models_O \varphi$ for those (r', v) such that $r' \in \mathcal{B}_i$ and $(r, u) \sim_i (r', v)$;
- $(I, r, u) \models_O D_i\varphi$ iff $(I, r', v) \models_O \varphi$ for those (r', v) such that $r' \in \mathcal{D}_i$ and $(r, u) \sim_i (r', v)$;
- $(I, r, u) \models_O I_i\varphi$ iff $(I, r', v) \models_O \varphi$ for those (r', v) such that $r' \in \mathcal{I}_i$ and $(r, u) \sim_i (r', v)$;

The semantics of atomic formulas or formulas of the form $\neg\varphi$, $\varphi \wedge \varphi'$, $\bigcirc\varphi$ or $\varphi \mathbf{U} \varphi'$ can be dealt with in the usual manner.

We use $\models_O \varphi$ to denote that φ is valid in every interpreted KBDI-system.

According to our definition, $D_i\varphi$ is true iff φ is true along those runs that are desirable to agent i and consistent with agent i 's observations. Thus, $D_i\varphi$ intuitively means that agent i 's goal implies that formula φ holds.

For those agents with perfect recall and a global clock, we may use $\stackrel{spr}{\sim}_i$ instead of $\sim_i$ to interpret those formals with modalities B_i , D_i and I_i and get an alternative satisfaction relationship $\models_O^{spr}$.

PROPOSITION 1. *The following axioms are valid with respect to both $\models_O$ and $\models_O^{spr}$:*

- $K_i\varphi \Rightarrow \varphi$
- $\Delta_i(\varphi \Rightarrow \psi) \Rightarrow (\Delta_i\varphi \Rightarrow \Delta_i\psi)$
 $\Delta_i\varphi \Rightarrow K_i\Delta_i\varphi \quad \neg\Delta_i\varphi \Rightarrow K_i\neg\Delta_i\varphi$
where Δ stands for K , B , D or I .
- *Relationship between knowledge, belief, desire and intention*
 $B_i\varphi \Rightarrow D_i\varphi \quad K_i\varphi \Rightarrow B_i\varphi \quad K_i\varphi \Rightarrow I_i\varphi$
- *Temporal operators*
 $\bigcirc(\varphi \Rightarrow \psi) \Rightarrow (\bigcirc\varphi \Rightarrow \bigcirc\psi)$
 $\bigcirc(\neg\varphi) \Rightarrow \neg\bigcirc\varphi$
 $\varphi \mathbf{U} \psi \Leftrightarrow \psi \vee (\varphi \wedge \bigcirc(\varphi \mathbf{U} \psi))$

PROPOSITION 2. *The following axioms are valid with respect to $\models_O^{spr}$: $\Delta_i\bigcirc\varphi \Rightarrow \bigcirc\Delta_i\varphi$, where Δ stands for K , B , D and I .*

The formula $D_i\bigcirc\varphi \Rightarrow \bigcirc D_i\varphi$ says that if agent i 's current goal implies φ holds at the next point in time, then at the next point in time her goal will imply φ , that is, agent i persists on her goal.

2.5 Proof system

We now discuss a proof system, called the KBDI proof system, for those agents with perfect recall and a global clock. The proof system contains the axioms of propositional calculus plus those in Propositions 1 and 2. It is closed under the propositional inference rules plus for every agent i : $\frac{\vdash_O \varphi}{\vdash_O K_i\varphi}$.

THEOREM 3. *The KBDI proof system for agents with perfect recall and a global clock is sound and complete with respect to interpreted KBDI-systems with satisfaction relation $\models_O^{spr}$.*

3. MODEL CHECKING KBDI LOGIC

The model checking problem is to give an algorithm to test whether a model satisfies a concerned specification expressed by a formula. However, in order to make our model checking algorithm practically useful, we must consider from where our model, an interpreted KBDI-system, comes. It would be satisfactory if we can derive our model from a program implemented in, say C or Java. However, to simplify the matter, we may consider some abstract programs such as *nite-state programs*, which are expressive enough from the standpoint of theoretical computer science. Moreover, to make our model checking system practically efficient, we use symbolic model checking techniques. Thus, a *nite-state program* in our approach is represented in a symbolic way.

3.1 Symbolic Representation of Interpreted KBDI System Model

A system as a set of infinite runs is not well suited to model checking directly as it is generally applied to finite state systems. However, we can represent an interpreted system as a *nite-state program* (G, G_0, R, V) , where G is a set of states, G_0 a set of initial states, and R a total next time relation, and V associates each state with a truth assignment function. A set of infinite runs is then obtained by unwinding the relation R starting from initial states in G_0 .

We can present a symbolic representation of a nite-state program (G, G_0, R, V) as $(\mathbf{x}, \theta(\mathbf{x}), (\mathbf{x}, \mathbf{x}'))$. Also, we represent a system as a (symbolic) nite-state program with n agents, which is defined as a tuple $\mathcal{P} = (\mathbf{x}, \theta(\mathbf{x}), (\mathbf{x}, \mathbf{x}'), O_1, \dots, O_n)$, where

1. $\mathbf{x}$ is a set of system variables;
2. θ is a boolean formula over $\mathbf{x}$, called the *initial condition*;
3. $\mathbf{x}, \mathbf{x}'$ is a boolean formula over $\mathbf{x} \cup \mathbf{x}'$, called the *transition relation*; and
4. for each i , O_i is a boolean formula over $\mathbf{x}$, containing agent i 's *local variables*, or *observable variables*.

For convenience, we may use $\mathcal{P}(\theta, \mathbf{x}, \mathbf{x}')$ to denote a nite-state program with n agents $(\mathbf{x}, \theta(\mathbf{x}), (\mathbf{x}, \mathbf{x}'), O_1, \dots, O_n)$, if $\mathbf{x}$ and $O_1, \dots, O_n$ are clear from the context.

Given a nite-state program $\mathcal{P}(\theta, \mathbf{x}, \mathbf{x}')$ with n agents, we define an agent's internal program over $\mathcal{P}(\theta, \mathbf{x}, \mathbf{x}')$ as a tuple

$$\langle \mathcal{P}(\theta_1, \mathbf{x}_1), \mathcal{P}(\theta_2, \mathbf{x}_2), \mathcal{P}(\theta_3, \mathbf{x}_3) \rangle$$

where, for $j = 1, 2, 3$, $(\theta_j \Rightarrow \theta) \wedge \mathbf{x}_j \Rightarrow \mathbf{x}$ is valid, and $(\theta_2 \Rightarrow \theta_1) \wedge \mathbf{x}_2 \Rightarrow \mathbf{x}_1$ is valid.

Clearly, an agent's internal program is exactly related to an agent's mental state. Thus, we define a (symbolic) *BDI-program with n agents* as a tuple $P_A = (\mathcal{P}_K, P_1, \dots, P_n)$, where $\mathcal{P}_K$ is a nite-state program with n agents and for each agent i , P_i is agent i 's internal program over $\mathcal{P}_K$. We use I_{P_A} to denote the corresponding interpreted KBDI-system.

3.2 Simplified BDI Programs

To obtain a BDI-program with n -agents, one should define $1 + 3n$ nite-state programs. Our experience with a previous version of **MCKBDI** indicated that it is inconvenient for a programmer to code a BDI-program with n -agents because there may be many nite-state programs involved. Therefore, we should simplify the definition of a BDI-program with n -agents.

DEFINITION 4. Let $\mathcal{P} = (\mathbf{x}, \theta(\mathbf{x}), (\mathbf{x}, \mathbf{x}'), O_1, \dots, O_n)$ be a nite-state program with n -agents, where, for each agent i , O_i contains at least 3 particular boolean variables WB_i , WD_i and WI_i . A simplified BDI-program with n -agents derived from $\mathcal{P}$ is a tuple $\langle \mathcal{P}, P_1, \dots, P_n \rangle$, where for each agent i , $P_i = \langle \mathcal{P}(\theta_1^i, \mathbf{x}_1^i), \mathcal{P}(\theta_2^i, \mathbf{x}_2^i), \mathcal{P}(\theta_3^i, \mathbf{x}_3^i) \rangle$ such that

1. $\theta_1^i = \theta \wedge WB_i$ and $\mathbf{x}_1^i = \mathbf{x} \wedge WB_i$;
2. $\theta_2^i = \theta \wedge WB_i \wedge WD_i$ and $\mathbf{x}_2^i = \mathbf{x} \wedge WB_i \wedge WD_i$;
3. $\theta_3^i = \theta \wedge WI_i$ and $\mathbf{x}_3^i = \mathbf{x} \wedge WI_i$.

It is more convenient for a programmer to code a simplified BDI-program with n -agents, and it seems that the notion of a simplified BDI-Program does not lose much generality of that of a BDI-Program.

4. CONCLUDING REMARKS

We have proposed a new KBDI-model by using interpreted systems, and developed a computationally grounded BDI logic, called KBDI logic. We interpret KBDI logic via two different semantics. One of them is based on the assumption that agents have perfect recall and there is a global clock, with respect to which a sound and complete proof system has been presented. We have explored how symbolic model checking techniques can be applied to model checking BDI-agents and obtained encouraging preliminary experimental results.

This paper is inspired by Su [11] and Su *et al* [12], which also provide agent models by extending the interpreted system model. However, Su [11] does not consider agents' belief, desire and intention. The main advantages of the present paper over Su *et al* [12] are: (1) We introduce, for each agent i , not only belief, desire and intention modalities B_i , D_i and I_i , but also a knowledge modality K_i . Moreover, the resulting axiomatic system is more succinct and proved to be sound and complete. (2) A model checker with KBDI-models, called **MCKBDI**, is developed.¹ Empirical results are more comprehensive, supported by convincing and motivating examples, where agents' observations and actions are explicitly defined.

Model checking multi-agent systems has become an active research topic in the community of multi-agent systems. Many efforts have been devoted to model checking knowledge in multi-agent systems [5, 13, 11, 7]. However, comparatively little work has been carried out on model checking BDI-agents. Some general approaches to model checking BDI-agents were proposed in Rao and Georgeff [9] and Benerecetti and Cimatti [1], but no method was given for generating models from actual systems, and so the techniques given there could not easily be applied to verifying real multi-agent systems. In Bordini *et al* [2], model checking techniques for AgentSpeak(L) [8] have been reported; however, AgentSpeak(L) has very simple semantics. The salient point of our work is that we present a general form of a BDI agent program, from which BDI agent models are generated and specifications in full BDI logics can be verified by symbolic model checking techniques.

5. REFERENCES

- [1] M. Benerecetti and A. Cimatti. Symbolic model checking for multi-agent systems. In *Proc. MoChart-02*, pages 1–8, Lyon, France, 2002.
- [2] R. Bordini, M. Fisher, C. Pardavila, and M. Wooldridge. Model checking agentspeak. In *Proc. AAMAS-03*, pages 14–18, Melbourne, Australia, 2003.
- [3] P. Cohen and H. Levesque. Intension is choice with commitment. *Artificial Intelligence*, 42:23–261, 1990.
- [4] R. Fagin, J. Halpern, Y. Moses, and M. Vardi. *Reasoning about knowledge*. MIT Press, Cambridge, MA, 1995.
- [5] W. v. d. Hoek and M. Wooldridge. Model checking knowledge and time. In *9th Workshop on SPIN (Model Checking Software)*, 2002.
- [6] S. Kripke. A semantical analysis of modal logic. i: Normal modal propositional calculi. *Z. Math. Logik Grundl. Math.*, 9:67–96, 1963.
- [7] F. Raimondi and A. Lomuscio. Verification of multiagent system via ordered binary decision diagrams: an algorithm and its implementation. In *Proc. of AAMAS-04*, 2004.
- [8] A. Rao. Bdi agent speak out in a logical computable language. In *LNAI*, volume 1038, pages 42–55, 1996.
- [9] A. Rao and M. Georgeff. A model theoretic approach to the verification of situated reasoning systems. In *Proc. 13th International Joint Conference on Artificial Intelligence*, pages 318–324, 1993.
- [10] A. Rao and M. Georgeff. Decision procedures for BDI logics. *Journal of Logic and Computation*, 8(3):293–344, 1998.
- [11] K. Su. Model checking temporal logics of knowledge in distributed systems. In *AAAI-04*, pages 98–103. AAAI, 2004.
- [12] K. Su, A. Sattar, K. Wang, and et al. The observation-based model for bdi-agents. In *AAAI-05*, pages 190–195. AAAI, 2005.
- [13] R. van der Meyden and K. Su. Symbolic model checking the knowledge of the dining cryptographers. In *Proc. of IEEE CSFW-04*, pages 280–291, 2004.
- [14] M. Wooldridge. Computationally grounded theories of agency. In E. Durfee, editor, *ICMAS-00*, pages 13–22. IEEE Press, 2000.

¹The source codes and the user manual of **MCKBDI** are available at <http://www.cit.gu.edu.au/~s360126/arc-mm1/>