

# Modeling Multiagent Systems with Local Model Semantics

Massimo Benerecetti<sup>1</sup>, Fausto Giunchiglia<sup>1,2</sup> and Luciano Serafini<sup>2</sup>

<sup>1</sup> DISA - Università degli Studi di Trento,  
Via Inama 5, 38100 Trento, Italy.

<sup>2</sup> IRST - Istituto Trentino di Cultura,  
38050 Povo, Trento, Italy.

e-mail: bene@cs.unitn.it, fausto@irst.itc.it, serafini@irst.itc.it

## Abstract

Local Model Semantics (Giunchiglia & Ghidini 1998) has been proposed as a formal framework to model contextual reasoning. The basic ideas underlying this semantics are the principles of *locality* of representations and of *compatibility* among representations. In this paper shows how Local Model Semantics can be effectively used to model dynamic Multiagent Systems. Intuitively, each agent have local representations (views) of (mental attitudes of) other agents. Thanks to its *locality* property, the resulting semantics allows us to design and model multiagent systems in a modular and incremental way.

## Introduction

The goal of this paper is to show how Local Model Semantics (Giunchiglia & Ghidini 1998) can be effectively employed in modeling dynamic multiagent systems in a way to allow for a modular specification of multiagent systems, a feature which seems necessary when dealing with real world complex systems (see (Giunchiglia & Giunchiglia 1996) for a discussion on this topic).

We model *agents* as concurrent reactive non-terminating finite state processes able to have what we call BDI attitudes, i.e., beliefs, desires and intentions (Bratman 1990; Rao & Georgeff 1991). The specification of an agent has therefore two orthogonal aspects: a temporal aspect and a “mental attitudes” aspect. The key idea underlying our approach is to keep these two aspects separated. In practice things work as follows:

- when we consider the temporal evolution of an agent we treat BDI atoms (i.e. atomic formulas expressing belief, desire, or intention) as atomic propositions. The fact that these formulas talk about BDI attitudes is not taken into consideration.
- The fact that an agent  $a_1$  has BDI attitudes about another agent  $a_2$  is modeled as the fact that  $a_1$  has access to a representation of  $a_2$  as a process (one representation for each BDI attitude). Then, any time it needs to verify the truth value of some BDI atom about  $a_2$ , e.g.,  $B_2AF \phi$ ,  $a_1$  simply tests whether, e.g.,  $AF \phi$  holds in its (appropriate) representation of  $a_2$ . BDI attitudes are essentially used to control the

“jumping” among processes. This operation is iterated in the obvious way in case of nested BDI attitudes.

An agent is therefore modeled as a set of processes, each representing a “view” on some other agent’s mental attitude evolving over time. Each such view can be thought of as a local (contextual) representation of that agent and formally captured by a suitable extension of the Local Model Semantics. This allows us to achieve modularity as these processes (and the way BDI attitudes enforce relations among them) can be independently defined. This modeling approach has been applied to make model checking-based verification of multiagent systems possible (see (Benerecetti, Giunchiglia, & Serafini 1998) which describes a complete model checking algorithm based on this framework).

The first section of the paper describes the context-based extension of a propositional temporal logic, called MultiAgent Temporal Logic (MATL), that allows us to express properties of multiagent systems. The description is given incrementally over the standard model checking notions. In particular, we adopt CTL (Clarke, Grumberg, & Long 1994) as the propositional temporal logic used to state specifications. The following section describe a finite state presentation of the semantics of MATL that can be used to specify multiagent systems in a modular way. We conclude with some remarks.

## Multiagent temporal logic

The logic we propose, called MATL (MultiAgent Temporal Logic), is the composition of two logics, one formalizing temporal evolution, the other formalizing BDI attitudes. We start from the temporal component CTL (for more details see (Clarke, Grumberg, & Long 1994)). We then present the Hierarchical MetaLogic (called HML) formalizing BDI attitudes. HML is a variation of the logics introduced in (Giunchiglia & Serafini 1994) (syntax and proof-theory) and (Giunchiglia & Ghidini 1998) (semantics). Finally, in the third subsection, we integrate the two logics into MATL.

## CTL

CTL is a branching time propositional temporal logic. Let us consider in turn the language and semantics.

**Language.** Given a set  $P$  of propositional atoms, the set of CTL formulas  $\phi$  is defined inductively as follows:

$$\phi, \psi ::= p \mid \neg\phi \mid \phi \wedge \psi \mid \text{EX } \phi \mid \text{A } (\phi \mathcal{U} \psi) \mid \text{E } (\phi \mathcal{U} \psi)$$

where  $p \in P$ . We have the following intuitive meanings:  $\text{EX } \phi$  means that there is a path such that  $\phi$  will be true in the next step;  $\text{A } (\phi \mathcal{U} \psi)$  means that  $\psi$  will be true in a state in the future and that  $\phi$  will be true in all the states before, for all paths;  $\text{E } (\phi \mathcal{U} \psi)$  means that there exists a path such that  $\psi$  will be true in a state in the future and that  $\phi$  will be true in all the states before. The following abbreviations are used:

$$\begin{array}{ll} \phi \supset \psi \stackrel{\text{def}}{=} \neg(\phi \wedge \neg\psi) & \text{AF } \phi \stackrel{\text{def}}{=} \text{A } (\top \mathcal{U} \phi) \\ \text{EF } \phi \stackrel{\text{def}}{=} \text{E } (\top \mathcal{U} \phi) & \text{AG } \phi \stackrel{\text{def}}{=} \neg \text{E } (\top \mathcal{U} \neg\phi) \\ \text{EG } \phi \stackrel{\text{def}}{=} \neg \text{A } (\top \mathcal{U} \neg\phi) & \text{AX } \phi \stackrel{\text{def}}{=} \neg \text{EX } \neg\phi \end{array}$$

**Semantics.** The semantics for CTL formulas is the standard branching-time temporal semantics based on Kripke-structures. A CTL structure is a tuple  $m = \langle S, s_0, R, L \rangle$ , where  $S$  is a set states,  $s_0 \in S$  is the *initial state*,  $R$  is a total binary relation on  $S$ , and  $L : S \rightarrow \mathcal{P}(P)$  is a *labeling function*, which associates to each state  $s \in S$  the set  $L(s)$  of propositional atoms true at  $s$ . A *path*  $x$  in  $m$  is an infinite sequence of states  $s_1, s_2, \dots$  such that for every  $i \geq 1$ ,  $s_i R s_{i+1}$ .

Satisfiability of a formula  $\phi$  in a CTL structure  $m$  at a state  $s$  is defined as follows:

- $m, s \models p$  iff  $p \in L(s)$ ;
- $m, s \models \neg\phi$  iff  $m, s \not\models \phi$ ;
- $m, s \models \phi \wedge \psi$  iff  $m, s \models \phi$  and  $m, s \models \psi$ ;
- $m, s \models \text{EX } \phi$  iff there's a  $s'$  with  $sRs'$ , such that  $m, s' \models \phi$ ;
- $m, s \models \text{A } (\phi \mathcal{U} \psi)$  iff for every path  $x = (s = s_1, s_2, \dots)$  there's a  $k \geq 1$  such that  $m, s_k \models \psi$  and, for every  $1 \leq j < k$ ,  $m, s_j \models \phi$ ;
- $m, s \models \text{E } (\phi \mathcal{U} \psi)$  iff there's a path  $x = (s = s_1, s_2, \dots)$  and a  $k \geq 1$  such that  $m, s_k \models \psi$  and for every  $1 \leq j < k$ ,  $m, s_j \models \phi$ .
- $m \models \phi$  iff  $m, s_0 \models \phi$ .

## HML

Let us consider in turn the language(s) and the semantics.

**Language(s).** Suppose we are modeling a situation with a set  $I$  of agents. Each agent has its own beliefs, desires, and intentions about itself and the other agents. Let us adopt the standard notation to express attitudes.  $B_i$ ,  $D_i$ , and  $I_i$ , for any  $i \in I$ , are called BDI operators for agent  $i$  (or simply BDI operators).  $O_i$  denotes any BDI operator for agent  $i$ .  $B_i\phi$  ( $D_i\phi$ ,  $I_i\phi$ ) means that agent  $i$  believes (desires or intends)  $\phi$ . Let

$O = \{B, D, I\}$  be a set of symbols, one for each BDI attitude. Let  $OI^*$  be the set  $(O \times I)^*$ , i.e., the set of finite (possibly empty) strings of the form  $O_1 i_1 \dots O_n i_n$  with  $O_k \in O$  and  $i_k \in I$ . We call any  $\alpha \in OI^*$ , a *view*. Intuitively, each view in  $OI^*$  represents a possible nesting of BDI attitudes. We also allow for the empty string,  $\epsilon$ . The intuition is that  $\epsilon$  represents the view of an external observer which, from the outside, "sees" the behavior of the overall multiagent system. Depending on the goals, the external observer can represent the person designing the system, or a selected process of the multiagent system which is given this privileged status. Figure 1 shows part of the tree of views for a multiagent system with two agents  $s$  and  $r$ .

It is important to notice that the crucial notion is that of view. An agent, e.g.,  $i$ , is thus represented by three trees rooted in the views that the external observer has of  $i$ 's beliefs, desires and intentions respectively (e.g. the views  $Bi$ ,  $Di$  and  $Ii$ ). Notice also that the view that an agent has of another agent is in general different from the agent itself. This allows us for instance to model the fact that agent  $i$  might have false beliefs about agent  $j$ .

We associate a logical language  $\mathcal{L}_\alpha$  to each view  $\alpha \in OI^*$ . Intuitively, each  $\mathcal{L}_\alpha$  is the language used to express what is true (and false) in the representation corresponding to  $\alpha$ . In particular, the language  $\mathcal{L}_\epsilon$  is used to speak about the whole multiagent system. Thus, intuitively, a formula  $p \wedge B_i \neg p \in \mathcal{L}_\epsilon$ , (denoted by  $\epsilon : p \wedge B_i \neg p$ ) means that  $p$  is true and that agent  $i$  believes that  $p$  is false. The languages  $\mathcal{L}_{Bi}$ ,  $\mathcal{L}_{Di}$ , and  $\mathcal{L}_{Ii}$  are the languages that  $i$  adopts to represent its beliefs, desires and intentions, respectively. The language  $\mathcal{L}_{BiIj}$  is used to specify  $i$ 's beliefs about  $j$ 's intentions, and so on. Intuitively, the formula  $p \wedge B_i \neg p \in \mathcal{L}_{Dj}$  (denoted by  $Dj : p \wedge B_i \neg p$ ) means that agent  $j$  desires two things: that  $p$  is true and that agent  $i$  believes that  $p$  is false.

Notice that we do not put any restriction on the languages  $\mathcal{L}_\alpha$ , except that  $O_i\phi$  must be an atomic formula of  $\mathcal{L}_\alpha$  if and only if  $\phi$  is a formula of  $\mathcal{L}_{\alpha O_i}$  (see (Giunchiglia & Giunchiglia 1996) for a study of how this condition can be modified in order to capture various interesting properties). We allow also for empty languages. However  $\mathcal{L}_\epsilon$  cannot be empty as we need to be able to talk about the whole multiagent system.

**Semantics.** We need to define the semantics of the family of languages  $\{\mathcal{L}_\alpha\}_{\alpha \in OI^*}$  (hereafter we drop the index  $\alpha$ ). To understand the semantics we need to understand two key facts. On the one hand the semantics of formulas depend on the view. For instance, the formula  $p$  at the view  $Bi$  expresses the fact that  $i$  believes that  $p$  is true. The same formula in the view  $Bj$  expresses the fact that  $j$  believes that  $p$  is true. As a consequence, the semantics associates *locally* to each view  $\alpha$  a set  $M_\alpha$  of interpretations of  $\mathcal{L}_\alpha$ . On the other hand there are formulas in different views which have the same intended meaning. For instance  $B_j p$  in view  $Bi$ , and  $p$  in view  $BiBj$  both mean that  $i$  believes that  $j$  believes that  $p$  is true. This implies that only certain

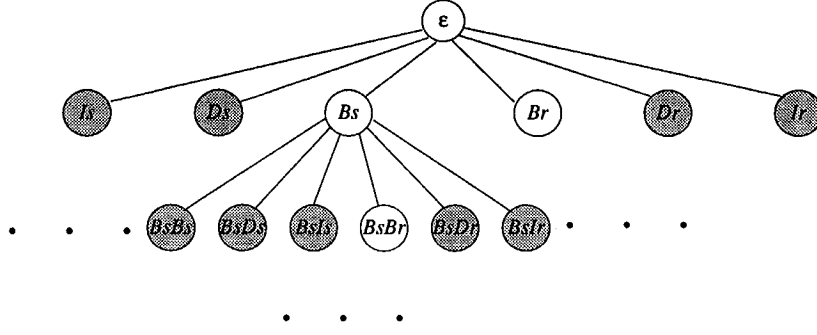


Figure 1: A tree of views.

subsets of interpretations of different views are *compatible*, and these are those which agree on the truth values of the formulas with the same intended meaning. To capture this notion of compatibility we introduce the notion of chain.

Let  $\alpha$  be any view, a  $\alpha$ -chain  $c$  is a finite sequence  $\langle c_\epsilon, \dots, c_\beta, \dots, c_\alpha \rangle$ , where  $c_\beta = m \in M_\beta$  is an interpretation for  $\mathcal{L}_\beta$  and  $\beta$  is a prefix of  $\alpha$  (i.e.  $\alpha = \beta\gamma$  for some  $\gamma$ ). A *compatibility relation*  $C$  on  $\{\mathcal{L}_\alpha\}$  is a set of  $\alpha$ -chains, for every  $\alpha$ . Intuitively,  $C$  will contain all those  $c$ 's whose elements  $c_\alpha, c_\beta$  (where  $\alpha, \beta$  are two views in  $OI^*$ ) assign the same truth values to the formulas with the same intended meaning. (For a detailed discussion about these intuitions and also for a technical presentation, see (Giunchiglia, Serafini, & Simpson 1992) and (Giunchiglia & Ghidini 1998).)

Let us now define the semantics of HML. We start with satisfiability local to views (first step) and suppose that for each view  $\alpha$  there is a satisfiability relation between  $M_\alpha$  and formulas of  $\mathcal{L}_\alpha$ . With an abuse of notation, we denote all these satisfiability relations with the same symbol  $\models$ . The context always makes clear which relation we mean. The second step is to define (global) satisfiability taking into account chains. To do this we need some notation. Let  $\models$  denote satisfiability also for chains.

For any  $\alpha$ -chain  $c$  and for any formula in  $\mathcal{L}_\beta$ , satisfiability relation  $\models$  is defined only when either  $\alpha$  is a prefix of  $\beta$  or  $\beta$  is a prefix of  $\alpha$ . (i.e. when either  $\alpha = \beta\gamma$  or  $\beta = \alpha\gamma$ ). If  $\alpha = \beta\gamma$  then  $c_\beta \models \phi$  iff  $\phi$  is true in  $c_\beta$ . If  $\beta = \alpha\gamma$  then  $c_\beta \models \phi$  for any  $\phi$ .

Let us extend the satisfiability relation to sets of formulas:  $x \models Y$  if and only if for any  $y \in Y$ ,  $x \models y$ .

We are now ready to define the notion of model for HML (called HM structure), and then that of satisfiability between HM structures and formulas of a view.

**Definition 1 (HM structure)** A nonempty compatibility relation  $C$  on  $\{\mathcal{L}_\alpha\}$  is a Hierarchical MetaStructure (HM structure) on  $\{\mathcal{L}_\alpha\}$  if given any  $\alpha\beta$ -chain  $c \in C$ ,  $c_\alpha \models O_i\phi$  iff for every  $\alpha\gamma$ -chain  $c' \in C$ ,  $c'_\alpha = c_\alpha$  implies  $c'_{\alpha O_i} \models \phi$ .

The intuitions underlying Definition 1 are described in detail in (Giunchiglia & Ghidini 1998). Briefly: the

nonemptiness condition for  $C$  guarantees that the external observer has a consistent view of the world; the only if part in the definition guarantees that each view has correct BDI attitudes, i.e. any time  $O_i\phi$  holds at a view then  $\phi$  holds in the view one level down in the chain; the if part is the dual property and ensures the completeness of each view.

Given an HM structure  $C$ , a formula  $\phi$  and a view  $\alpha$ ,  $C \models \alpha : \phi$  is read as  $\phi$  is true in  $C$  (or equivalently,  $\phi$  holds in  $C$ , or  $\phi$  is satisfied by  $C$ ) at view  $\alpha$ , and it is defined as follows:

$$C \models \alpha : \phi \text{ iff for all } \alpha\beta\text{-chain } c \in C, c_\alpha \models \phi \quad (1)$$

The intuition is that in order to check the satisfiability of  $\phi$  at the view  $\alpha$  we need to check all the interpretations of  $\mathcal{L}_\alpha$  allowed by the compatibility imposed by the chains we are considering.

## MATL

Let us consider in turn the language(s) and the semantics.

**Language(s).** We define MATL as a kind of HML where each language  $\mathcal{L}_\alpha$  is a CTL language. We want to allow for (agents and) views with different languages. Let  $\{P_\alpha\}$  be a family of sets of propositional atoms. Each  $P_\alpha$  allows for the definition of a different language (also called an MATL language (on  $\{P_\alpha\}$ )). Since (agents and) views have BDI attitudes, we need to extend the propositional atoms of each MATL with the appropriate BDI atoms. Therefore, the family of MATL languages on  $\{P_\alpha\}$  is the family of CTL languages  $\{\mathcal{L}_\alpha\}$  where  $\mathcal{L}_\alpha$  is the smallest CTL language containing the set of propositional atoms  $P_\alpha$  and the BDI atoms  $O_i\phi$  for any  $\mathcal{L}_{\alpha O_i}$  formula  $\phi$ .

**Semantics.** The semantics for a family of MATL languages is defined in terms of HM structures, where the semantics local to each view is a set of CTL structures. Given a CTL structure  $m = \langle S, s_0, R, L \rangle$  and a state  $s$  of  $m$ , let  $m[s/s_0]$  be the CTL structure  $\langle S, s_0, R, L \rangle$  obtained by replacing the initial state  $s_0$  of  $m$  by  $s$ .

**Definition 2 (MATL structure)** A MATL structure on  $\{P_\alpha\}$  is an HM structure  $C$  for a family of MATL languages on  $\{P_\alpha\}$ , such that for any  $\alpha\beta$ -chain  $c \in C$ ,

if the CTL structure  $c_\alpha = m$ , then for any state  $s$  of  $m$ , there is a  $\alpha\beta$ -chain  $c' \in C$  such that  $c'_\alpha = m[s/s_0]$ .

A MATL structure is a particular kind of HM structure.<sup>1</sup> Therefore it keeps the same notion of satisfiability given by (1).

Satisfiability in a MATL structure can be understood on the basis of two crucial observations, concerning the mutual nesting of CTL operators and BDI operators. The first, concerning the nesting of CTL operators inside BDI operators is that  $c_\alpha \models \phi$  is computed using the notion of satisfiability in a CTL structure. Therefore, a chain links the fact that a BDI atom holds in the initial state of a CTL structure in one view with the fact that its argument holds in the initial state of a CTL structure in the view below. The second observation concerns the nesting of BDI operators inside temporal operators. (Temporal operators which involve no BDI atoms are treated as in CTL structures, that is, without jumping among views). Consider for instance the formula  $EX B_i p$ . To assess the truth of  $EX B_i p$  we need to be able to assess the truth of  $B_i p$  in some future state  $s$  of the CTL structure we are considering, e.g.,  $m = \langle S, s_0, R, L \rangle$ . The only way to establish this is to request that in  $s$  we have a chain  $c'$  which gives access to a CTL structure in the view below. Given the fact that chains connect CTL structures only for what holds in their initial state, the only solution is to request that  $s$  is the initial state of a CTL structure  $c'_\epsilon = m' = m[s/s_0]$  with  $c' \in C$ . Given the fact that temporal operators allow us to state facts about all the states in a CTL structure, this operation must be repeated for each state  $s \in S$ . But this is exactly what Definition 2 says.

### Multiagent finite state machines

MATL structures are in general infinite structures. In fact, CTL structures can have an infinite number of states, and also a labeling function which maps to an infinite number of atoms, while HM structures can have an infinite number of chains (corresponding to an infinite compatibility relation), chains with infinite branching (corresponding to an infinite number of attitudes per agent and/or an infinite number of agents) or infinitely long chains (corresponding to the case of unbounded nested BDI attitudes).

In many applications, like automatic model checking-based verification of systems, we are interested in having a finite presentation of the semantics. For instance in model checking one deals with *finite CTL structures*, i.e., CTL structures which have a finite set of states, and also a labeling function mapping to a finite number of atoms. The crucial observation is that finite CTL structures can be seen as *finite state machines* (FSMs), an

<sup>1</sup>Each view of a MATL structure contains many CTL structures which differ only for the initial state. This redundancy is admitted for the sake of simplicity. In Section we show how these structures can be presented with only one structure.

FSM being an object  $f = \langle S, s_0, R, L \rangle$  (with everything finite). Our solution is to extend the notion of FSM to that of *MultiAgent Finite State Machine (MAFSM)*, where, roughly speaking, a MAFSM is a finite set of FSMs. A first step in this direction is to restrict ourselves to *finite HM structures*, i.e., those HM structures which have a finite number of chains, and a finite number of views  $\alpha$  such that there is a  $\alpha$ -chain (notice that this limits both the number and the depth of chains). Thus, let  $OI^n$  denote a finite subset of  $OI^*$  obtained by taking the views in any finite subtree of  $OI^*$  rooted at view  $\epsilon$ . However this is not enough as finite HM structures allow for an infinite number of BDI atoms. Even if we have a finite number of processes we cannot model them as FSMs. We solve this problem by introducing the notion of explicit BDI atom. Formally if  $\{\mathcal{L}_\alpha\}$  is a family of MATL languages, then  $Expl(o_i, \alpha)$  is a (possibly empty) *finite* subset of the BDI atoms of  $\mathcal{L}_\alpha$ . The elements of  $Expl(o_i, \alpha)$  are called *explicit BDI atoms*. We have the following.

**Definition 3** Let  $\{\mathcal{L}_\alpha\}$  be a family of MATL languages on  $\{P_\alpha\}$ . A MultiAgent Finite State Machine (MAFSM)  $F = \{F_\alpha\}$  for  $\{\mathcal{L}_\alpha\}$  is a recursive total function such that:

1.  $F_\epsilon \neq \emptyset$ ;
2. for all views  $\alpha \in OI^n \subset OI^*$  (with  $OI^n$  finite), it associates a nonempty finite set  $F_\alpha$  of FSMs on the MATL language on the following atoms:  $P_\alpha$ ,  $Expl(b_i, \alpha)$ ,  $Expl(d_i, \alpha)$  and  $Expl(r_i, \alpha)$ , for all  $i \in I$ ;
3. for all views  $\alpha \in OI^* \setminus OI^n$ ,  $F_\alpha = \emptyset$ .

The first condition (dual to the condition  $\mathcal{L}_\epsilon \neq \emptyset$  imposed in Section , and to Condition 1 in Definition 1 of HM structure in Section ) is needed as otherwise there is nothing we can reason about; the second allows us to deal with finite chains, and the third allows us to deal with finite sets of atoms.

Given the notion of MAFSM, the next step is give a notion of satisfiability in a MAFSM. We start from the notion of satisfiability of CTL formulas in an FSM at a state. This notion is defined as in CTL structures. This allows us to determine the satisfiability of all the propositional and explicit BDI atoms (and all the formulas belonging to the corresponding MATL language). For these formulas we do not need to use the machinery associated to BDI attitudes. However, this machinery is needed in order to deal with the (infinite) number of BDI atoms which are not memorized anywhere in MAFSM.

Let the set of *implicit BDI atoms* of a view  $\alpha$ , written  $Impl(o_i, \alpha)$ , be defined as the (infinite) subset of all BDI atoms of  $\mathcal{L}_\alpha$  which are not explicit BDI atoms, i.e.  $Impl(o_i, \alpha) = \{O_i \phi \in \mathcal{L}_\alpha \setminus Expl(o_i, \alpha)\}$ . Let  $ArgExpl(o_i, \alpha, s)$  be defined as follows.

$$ArgExpl(o_i, \alpha, s) = \{\phi \in \mathcal{L}_{\alpha O_i} \mid O_i \phi \in L(s) \cap Expl(o_i, \alpha)\}$$

Intuitively,  $ArgExpl(Oi, \alpha, s)$  consists of all the formulas  $\phi \in \mathcal{L}_{\alpha Oi}$  such that the explicit BDI atom  $O_i\phi$  is true in  $s$ . Restricted to  $s$  and to the explicit BDI atoms,  $ArgExpl(Oi, \alpha, s)$  is the set of formulas which satisfies Conditions 1 and 2 (of correctness and completeness) in the definition of HM (and MATL) structure, Definition 1 in Section . At this point, to define the satisfiability in a MAFSM, it is sufficient to use the fact that we know how to compute  $ArgExpl(Oi, \alpha, s)$  (it is sufficient to use CTL satisfiability and then to compare the results of the relevant CTL structures) and exploit  $ArgExpl(Oi, \alpha, s)$  to compute the implicit BDI atoms which satisfy the two conditions of correctness and completeness.

**Definition 4 (Satisfiability in a MAFSM)** *Let  $F$  be a MAFSM,  $\alpha$  a view,  $f = \langle S, s_0, R, L \rangle \in F_\alpha$  an FSM, and  $s \in S$  a state. Then, for any formula  $\phi$  of  $\mathcal{L}_\alpha$ ,  $F, \alpha, f, s \models \phi$  is defined as follows:*

1.  $F, \alpha, f, s \models p$ , where  $p$  is a propositional atom or an explicit BDI atom: the same as FSM satisfiability;
2. satisfiability of propositional connectives and CTL operators: the same as FSM satisfiability;
3.  $F, \alpha, f, s \models O_i\phi$ , where  $O_i\phi$  is an implicit BDI atom, iff for all  $f' \in F_{\alpha Oi}$  and  $s'$  state of  $f'$ ,  $F, \alpha Oi, f', s' \models \bigwedge ArgExpl(Oi, \alpha, s) \supset \phi$

We have furthermore:

4.  $F, \alpha, f \models \phi$  iff  $F, \alpha, f, s_0 \models \phi$ ;
5.  $F, \alpha \models \phi$  iff for all  $f \in F_\alpha$ ,  $F, \alpha, f \models \phi$ ;
6.  $F \models \alpha : \phi$  iff  $F, \alpha \models \phi$ .

In the definition of  $F, \alpha, f, s \models \phi$ , item 3 is the crucial step.  $\bigwedge ArgExpl(Oi, \alpha, s)$  is the conjunction of all the elements of  $ArgExpl(Oi, \alpha, s)$ . We need to use  $ArgExpl(Oi, \alpha, s)$  in order to compute the formulas  $\phi$  such that  $O_i\phi$  is an implicit BDI atom as, as said at the end of Section , in MATL BDI operators have the same strength as modal  $K(3m)$ . In particular, we have that if  $\Gamma \supset \phi$  is a theorem in a view then  $O_i\Gamma \supset O_i\phi$  is a theorem in the (appropriate) view above. The remaining items are the natural counterpart of the respective definitions given for MATL structures. In particular, item 4 states that a FSM satisfies a formula if the formula is satisfied in its initial state. Item 5 states that a formula is satisfied in a view if it is satisfied by all the FSMs of that view. Finally item 6 states that a labeled formula  $\alpha : \phi$  is satisfied if  $\phi$  is satisfied in the view corresponding to the label.

Finally, the last step is to relate the notion of satisfiability in MAFSMs to the notion of satisfiability in MATL structures. We say that a MATL structure is *equivalent* to a MAFSM  $F$  if in every view they satisfy the same set of formulas.

**Proposition 1** *For each MATL structure which is a finite HM structure there is an equivalent MAFSM and vice versa.*

## Conclusion

In this paper we have defined a logic for multiagent systems based on Local Model semantics for contextual reasoning. We have shown the specification logic and the language for specifying finite state automata. This modeling framework has been used as a basis for a model checking algorithm (Benerecetti, Giunchiglia, & Serafini 1998). The approach allows us to specify multiagent systems incrementally and to reuse, in principle, technology and tools already developed in model checking as the problem of modeling and verifying multiagent system specifications has been reduced to the problem of modeling and verifying processes. In doing this, the notion of context (view) plays a crucial role. It allows us to view agents as collections of local representation modeling beliefs, desires and intentions of other agents connected among them by compatibility relations. The intrinsic locality of this semantics allows us to design and specify multiagent systems in a modular and incremental way.

## References

- Benerecetti, M.; Giunchiglia, F.; and Serafini, L. 1998. Model Checking Multiagent Systems. *Journal of Logic and Computation, Special Issue on Computational & Logical Aspects of Multi-Agent Systems* 8(3):401–423.
- Bratman, M. E. 1990. *Intention, plan and practical reason*. Harvard University Press.
- Clarke, E.; Grumberg, O.; and Long, D. 1994. Model Checking. In *Proceedings of the International Summer School on Deductive Program Design*.
- Giunchiglia, F., and Ghidini, C. 1998. Local Models Semantics, or Contextual Reasoning = Locality + Compatibility. In *Proceedings of the Sixth International Conference on Principles of Knowledge Representation and Reasoning (KR'98)*, 282–289. Trento: Morgan Kaufmann. Short version presented at the AAAI Fall 1997 symposium on context in KR and NL.
- Giunchiglia, E., and Giunchiglia, F. 1996. Ideal and Real Belief about Belief. In *Practical Reasoning, International Conference on Formal and Applied Practical Reasoning, FAPR'96*, number 1085 in Lecture Notes in Artificial Intelligence, 261–275. Springer Verlag.
- Giunchiglia, F., and Serafini, L. 1994. Multilanguage hierarchical logics (or: how we can do without modal logics). *Artificial Intelligence* 65:29–70.
- Giunchiglia, F.; Serafini, L.; and Simpson, A. 1992. Hierarchical meta-logics: intuitions, proof theory and semantics. In *Proc. of META-92, Workshop on Metaprogramming in Logic*, number 649 in LNCS, 235–249. Uppsala, Sweden: Springer Verlag.
- Rao, A. S., and Georgeff, M. P. 1991. Modeling rational agents within a BDI architecture. In Allen, J.; Fikes, R.; and Sandewall, E., eds., *Proceedings of the 2nd International Conference on Principle of Knowledge Representation and Reasoning*. Morgan Kaufmann Publishers.