A Multi-agent Modal Language for Concurrency with Non-communicating Agents

Stefano Borgo

Computer Science, Indiana University, Bloomington, IN 47405 (USA)
LACL, Université de Paris XII, 94010 Creteil Cedex (FR)
stborgo@indiana.edu

Abstract. We introduce a formal language for multi-agent systems based on new modal operators. The modal operators express concurrency at the syntactic level. Operators containing quantifiers describe the evolution of a system where each agent has knowledge of other agents' attitude toward a goal but not of their actions. This result is obtained without introducing standard epistemic operators. The semantics presents a mixture of Tarskian and game-theoretical elements. We apply game-theory to interpret the quantified modalities and to determine which information is available to the agents as well as their reasoning capabilities.

Keywords: modal logic, multi-agent systems, independence, concurrency, game-theoretic semantics

1 Introduction

Nowadays, multi-agent systems [6, 13] are at the center of many research areas like computer science, philosophy, mathematics, linguistics, social science, and economics. The approaches to *multi-agent systems* (*MAS*) vary considerably in these areas but even if one limits the analysis to logical systems, one finds that different languages are applied to very similar problems. Generally speaking, these languages contain a variety of modal operators; epistemic operators are included to express the knowledge of the agents, deontic operators to express obligations, and so on. It is now common to find complex languages for multi-agent systems with the result that we need to deal with complex and different structures for phenomena that do not seem very different. This highlights, we believe, a need for logical tools explicitly developed to describe multi-agent systems.

Starting from this observation, in [3] we motivated the introduction of a new modal language for multi-agents systems and described in details the propositional system with its main properties.

In this paper, we continue to develop this approach introducing one quantified extension of the propositional system. Our aim is to present a new expressive tool, namely quantified modalities, that can deal with some features peculiar to any system with several entities like software agents, biological substances, or human beings. We hope to obtain a language which, using only one type of operator, can already express many features of systems in *MAS*, in particular concurrency and information independence among the agents. The formalism, it is believed, is well suited for description of agents and multi-agent systems properties as well as for reasoning within such systems.

Such a language may provide the starting point for the development of uniform languages for *MAS*. In this way it may become possible to reduce the number of modalities in the language thereby avoiding the need for combining operators from disparate logical approaches (dynamic, temporal, deontic, epistemic logic and the like). Indeed, it is well known that the formal interaction of these operators is often quite complex [2].

The formalism we are going to introduce describes an evolving system with a fixed number of independent agents. We do not give a precise definition of agent. For our purposes, it suffices to say that an *agent* is a rational and autonomous entity that has the power to choose and execute actions. Our notion of rational agent is deliberately broad. (It includes computational and human agents as well as biological substances.) By *state* of the system we refer to the usual notion of global state. The system state is not partitioned with respect to the agents, i.e., there are no "local" states for individual agents. Time is not considered explicitly and there is no reasoning about history. The formalism considers only the present state and the future.

For lack of space, in this paper we are not going to attempt a deductive characterization of the language nor to provide applications of the formalism.

In section 2 we give an example to show the potentiality of the formalism. Section 3 presents one quantified modal logic based on the operators introduced in the example. In the next section, we give the semantics. We conclude with a brief discussion of the literature in section 4.

2 A guiding example

Suppose that Friday morning Bill finally agreed to go tonight to a ballet that his wife Laure would love to see.

Since Bill is not interested in ballet, he would be happy to change the evening plans. However, he made a promise that if nothing comes up to prevent them from going, he will accompany her. According to his attitude, in the afternoon Bill does whatever he needs to do without thinking about the ballet. When he is with his wife Laure tonight, he will go with her to the ballet as agreed. Only some important problem could force him to change this schedule.

Laure loves ballets. She is enthusiastic about tonight's plan and she really wants to attend this one with Bill. Also, she is so into it that this afternoon she does everything she can think of to avoid possible obstacles. In particular, choosing what to do in the afternoon, she takes into account the fact that Bill is not enthusiastic as she is and so she cannot really count on him to avoid possible obstacles. However, she knows his attitude and, in particular, she knows he will not find any excuse tonight unless there is a problem (at that point unavoidable). Furthermore, as much as she loves Bill, she knows she will consent to do something different in the evening if, for some reason, Bill will not be able to attend the ballet.

This being the circumstances, we wonder how the situation evolves and if Bill and Laure will attend the ballet this Friday. Of course, for this we need to state the possible evolutions of the system and the actions available to our agents. To introduce these elements, we begin with an informal description of our quantified modal operators.

In order to state that Bill and Laure choose two successive actions with no knowledge of each other's choices, we write a 2×2 matrix with squared brackets like the following[1]

$$\begin{bmatrix} Q_1 x & Q_3 z \\ Q_2 y & Q_4 v \end{bmatrix} \text{ where } Q_i \text{ is either } \forall \text{ or } \exists \tag{1}$$

We use the first row to list in the right order the variables Bill has to instantiate (choose a value for them). In particular, first Bill chooses a value for x (in our example corresponding to what he does in the afternoon), then a value for z (in the example, his action for the evening). Similarly, the second row lists, in the right order, the variables reserved for Laure.

The intended reading is that Bill, while choosing an instance for x and z, has no knowledge of Laure's choices for y and v, and vice versa. The agents become aware of the effects of their combined actions only after both of them have chosen and executed their actions.

The situation of Bill and Laure is described by formula $\begin{bmatrix} \forall x & \exists z \\ \exists y & \forall v \end{bmatrix} \varphi$ where the first column is related to what the agents do in the afternoon, the latter to what they do in the evening (a column is a sort of time-step), and φ stands for "Bill and Laure attended the ballet".

The first row describes Bill's attitude during the day (not committed in the afternoon but wanting to please Laure in the evening doing everything he can to go to the ballet) while the other describes Laure's attitude (wanting to go in the afternoon but ready to give up in the evening to please Bill). Here we can recognize the specific role of existential and universal quantifiers. Roughly speaking, existential quantifiers mark entries where the agent chooses wanting to make φ true, i.e., looking for an action that leads to this result. The universal quantifiers, on the contrary, mark entries where agents choose with no commitment to the truth-value of φ, i.e., the agent can select any action. In general, it is intended that the agents choose and execute their actions simultaneously at each column.

We now introduce constant operators describing the possible evolutions of the system. Then, using this information, we discuss the options for Bill and Laure.

[1] In general, we use a $2 \times n$ matrix for a sequence of n successive actions and a $k \times n$ matrix if the system has k agents.

Suppose that the actions available to the agents are: $a \equiv$ "visiting Bill's parents", $b \equiv$ "going shopping", $c \equiv$ "working at the office", $d \equiv$ "going to the theater"; and that the possible combinations in the social environment are (these operators are possible instances of the quantified operator $\begin{bmatrix} \forall x & \exists z \\ \exists y & \forall v \end{bmatrix}$):

$$\begin{bmatrix} a & d \\ a & d \end{bmatrix}, \begin{bmatrix} a & d \\ b & d \end{bmatrix}, \begin{bmatrix} a & d \\ c & d \end{bmatrix}, \begin{bmatrix} b & d \\ a & d \end{bmatrix}, \begin{bmatrix} c & d \\ a & d \end{bmatrix}, \begin{bmatrix} a & b \\ a & d \end{bmatrix}, \begin{bmatrix} a & b \\ b & d \end{bmatrix},$$

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix}, \begin{bmatrix} b & b \\ a & d \end{bmatrix}, \begin{bmatrix} c & b \\ a & d \end{bmatrix}, \begin{bmatrix} b & a \\ b & a \end{bmatrix}, \begin{bmatrix} c & a \\ c & a \end{bmatrix}, \begin{bmatrix} b & a \\ c & a \end{bmatrix}, \begin{bmatrix} c & a \\ b & a \end{bmatrix}$$

That is, if in the afternoon at least one of them goes for the traditional visit to Bill's parents on Friday (action a), then they are free to spend the evening by going to the theater or shopping together (actions b and d). However, if in the afternoon neither goes to pay a visit to Bill's parents, then they have both to go there for the evening. Furthermore, going to the theater is allowed in the evening but not in the afternoon, while working at the office is allowed in the afternoon only.

What should Laure choose to do in the afternoon? Since she wants to go to the theater together with Bill, she looks at the operators where the second column has only d's. There are five operators with this characteristic, these describe all the system evolutions compatible with Bill and Laure attending the ballet.

Certainly, Laure wants one of these to be used to instantiate the quantified modality. Among these five operators, the second and the third require Bill to perform action a in the afternoon. Since at least one of the entries in the first column has to be action a and since Laure cannot trust Bill to do it during the afternoon, she will do a herself. This means that she is going to choose a as value for y. Executing a in the afternoon, Laure restricts the possible instances to the following sublist $\begin{bmatrix} a & d \\ a & d \end{bmatrix}, \begin{bmatrix} b & d \\ a & d \end{bmatrix}, \begin{bmatrix} c & d \\ a & d \end{bmatrix} \begin{bmatrix} a & b \\ a & d \end{bmatrix}, \begin{bmatrix} b & b \\ a & d \end{bmatrix}, \begin{bmatrix} c & b \\ a & d \end{bmatrix}$.

Since Bill is willing to make formula φ true when choosing a value for z, he has to choose d if this is possible. Thus, Laure's first choice and Bill's second choice will necessarily isolate one instance for the quantified modal operator among these $\begin{bmatrix} a & d \\ a & d \end{bmatrix}, \begin{bmatrix} b & d \\ a & d \end{bmatrix}, \begin{bmatrix} c & d \\ a & d \end{bmatrix}$. That is, Laure has a strategy to reach her goal, no matter what Bill does in the afternoon.

Notice that the case presented above has several interesting features of multi-agent systems: concurrency, independence in choosing actions, knowledge of some behavioral factors, ignorance of other agents' actions, commitment (or indifference) to a goal, change of attitude toward a goal, and so on, and that these are all captured in the given setting by a unique quantified operator.

With operators like (1) we have enriched the meaning of the modal symbols '[' and ']' with respect to more traditional logics. In systems of modal logic, even elaborated as Dynamic Logic [7], brackets are used to represent grouping and are generally associated with the meaning of the necessity modality. In our formalism, the brackets mantain these roles and also receive an epistemic touch. In an operator of form $\begin{bmatrix} Q_1x & Q_3z \\ Q_2y & Q_4v \end{bmatrix}$, the brackets mark a situation where the agents perform a sequence of actions all the while receiving no feedback either about the evolution of the system or about other agents' actions. These brackets are like epistemic boundaries, and the agents, during the time spanned by the modal operator, act like isolated agents.

In the next section we formally introduce the language and its semantics. It is always assumed that actions in the same column of an operator are performed concurrently by different agents and all actions in the same row are executed, in the order they occur, by the same agent. In the rest of the paper, we posit a fixed number k of agents.

3 Multi-agent Basic Logic (*MBL*)

We fix countably many *constants* denoting actions a_0, $a_1, \ldots$, variables x_0, $x_1, \ldots$, and *atomic sentences* p_0, $p_1, \ldots$ *Const* is the set of constants in the language, *Var* the set of variables. (Sometimes we write $a, b, c, \ldots$ for arbitrary constants.)

Beside the propositional operators $\neg$ and $\rightarrow$ (from which $\vee, \wedge$, and $\leftrightarrow$ are defined in the usual way), there are unary (constant) modal operators (multi-agent operators). A *modal operator* $[M_n]$, where $n \geq 1$, is syntactically a matrix with k rows and n columns whose entries are constants. We call *cOP* the set of constant modal operators (from now on, all the operators and matrices have k rows unless otherwise stated, sometimes we drop the index n). Given any $k \times n$ ($n \geq 1$) matrix M, we write $[M]$ if it is in *cOP*.

We call *vOP* the set of all modal operators obtained from an operator in *cOP* putting at least one free variable in one entry of the operator but no quantified variables. We call *qOP* the set of all modal operators obtained from an operator in *cOP* $\cup$ *vOP* such that at least one entry contains one quantified variable. It is required that no variable, quantified or not, occurs more than once in an operator. Thus, operators in *vOP* are like operators in *cOP* with the only exception that some or all entries have form x_i. Instead, operators in *qOP* are like operators in *cOP* $\cup$ *vOP* with the only exception that some or all entries have form $\forall x_i$ or $\exists x_i$. When necessary, we refer to entries containing $\forall x_i$ or $\exists x_i$ as *quantified entries* while a *constant entry* is any entry containing a constant. Furthermore, we write $M(i, j)$ or $[M](i, j)$ for entry (i, j) in operator $[M]$. Operators in *qOP* are said *quantified* and operators in *vOP* are said *free*. *Constant* operators are the operators in *cOP*. Note that *cOP*, *vOP*, and *qOP* are pairwise disjoint. Finally, we put *OP* $=$ *cOP* $\cup$ *vOP* $\cup$ *qOP*.

An operator $[A_n] \in$ *cOP* is said to be an *instance* of an operator $[M_n]$ if all the constants in $[M_n]$ match the constants in the corresponding entries of $[A_n]$. (We abuse the notation and sometimes talk of instances of matrices as well.)

For the sake of simplicity, in this paper we assume *cOP* is *total*, that is, it contains all possible operators with constants in *Const*. As a consequence, *cOP* is closed under juxtaposition: let $k = 2$ and $[A] = \begin{bmatrix} a_1 & a_3 \\ a_2 & a_4 \end{bmatrix}$, $[B] = \begin{bmatrix} a_5 \\ a_6 \end{bmatrix}$, then the juxtaposition of $[A]$ and $[B]$ is operator $[C] = [A \mid B] = \begin{bmatrix} a_1 & a_3 & a_5 \\ a_2 & a_4 & a_6 \end{bmatrix}$.

Formulas are inductively generated by the following clauses:

1. all atomic sentences are formulas
2. $\varphi \rightarrow \psi$ is a formula if φ and ψ are formulas
3. $y = z$ is a formula if y, z are constants or variables
4. $\neg\varphi$ is a formula if φ is a formula
5. $[M_n]\varphi$ is a formula if φ is a formula and $[M_n] \in OP$

The scope of a column operator is the formula to which it is applied and the scope of a quantifier in an operator is the same as the scope of the operator itself. An occurrence of a variable x is said to be *bound* in a formula φ if either it occurs quantified in a modal operator[2] or it lies within the scope of an operator where x occurs quantified. Otherwise, the occurrence is said to be *free*. A *sentence* is a *closed formula*, i.e., a formula with no free occurrences of variables.

A *model* for *MBL* is a 4-tuple $\langle W, P; \{R^{k\times n} \mid n \in N^+\}; [\![\cdot]\!]\rangle$ such that:

- W is a non-empty set of states;
- P is a set of actions in 1-1 correspondence with the constants of the language;
- for all $n \in N^+$ and for all matrices Γ, if there exists $[A] \in cOP$ with $\Gamma = [\![A]\!]$ (see below), then $R^{k\times n}(\Gamma) \subseteq W \times W$ and, given $R^{k,n}(\Gamma)$ and $R^{k,m}(\Gamma')$, we have $R^{k,n}(\Gamma) \circ R^{k,m}(\Gamma') = R^{k\times(n+m)}(\Gamma \mid \Gamma')$;
- $[\![\cdot]\!]$ is a valuation function mapping atomic sentences to sets of states; distinct constants to distinct elements in P; operators $[A] = \begin{bmatrix} a_{1,1} & \dots & a_{1,n} \\ \vdots & \dots & \vdots \\ a_{k,1} & \dots & a_{k,n} \end{bmatrix}$ to matrices $[\![\,[A]\,]\!] = \begin{pmatrix} [\![a_{1,1}]\!] & \dots & [\![a_{1,n}]\!] \\ \vdots & \dots & \vdots \\ [\![a_{k,1}]\!] & \dots & [\![a_{k,n}]\!] \end{pmatrix}$. We write $[\![A]\!]$ for $[\![\,[A]\,]\!]$.

Given an environment $\Im$, that is, a function from variables to P, if $A = \begin{bmatrix} a_{1,1} & \dots & a_{1,n} \\ \vdots & \dots & \vdots \\ a_{k,1} & \dots & a_{k,n} \end{bmatrix}$ is an element of vOP, we write $[\![\Im(A)]\!]$ for $\begin{pmatrix} b_{1,1} & \dots & b_{1,n} \\ \vdots & \dots & \vdots \\ b_{k,1} & \dots & b_{k,n} \end{pmatrix}$, where $b_{i,j} = \Im(a_{i,j})$ if $a_{i,j}$ is a variable, $[\![a_{i,j}]\!]$ otherwise. Given a formula or operator χ, let χ_{sub} be obtained from χ substituting each free occurrence of a variable x (including those in modal operators) with constant a such that $\Im(x) = [\![a]\!]$.

k-Game Structure. Each formula with quantified operators receives a truth-value through a game, called k-game, played by the agents in the system. Here we describe the rules of the k-game.

[2] A quantified entry, for instance $\forall x_i$, in an operator stands for the quantified variable $\forall x_i$ as well as for a bound occurrence of x_i.

Fix a model $\mathcal{M}$, a state s, an environment $\Im$. A *k-game* over sentence $[N_n]\psi$ is a sequence of choices that singles out a constant $k \times n$ operator. Let $([N_n]\psi)_{sub} = [M_n]\varphi$.

To begin the k-game, we fix k matrices $M^{1,1}, \ldots, M^{k,1}$, one matrix per agent, called *information-matrices*. In *MBL* an agent i knows only the attitude of other agents, thus at the beginning of the game $M^{i,1}$ (the matrix containing the information available to agent i) is as $[M_n]$ except that only the variables occurring in row i are shown; in other words, $M^{i,1}$ has all entries of row i and all constant entries as in $[M_n]$, but in the remaining entries it shows $\forall$ ($\exists$) where $[M_n]$ has $\forall x$ ($\exists x$), see Fig. 1. Also, put $\varphi^{i,1} = \varphi$ for all i.

$$\begin{bmatrix} \forall x & \exists z & b \\ \exists y & a & \forall w \end{bmatrix} ; \quad \begin{bmatrix} \forall x & \exists z & b \\ \exists & a & \forall \end{bmatrix} , \quad \begin{bmatrix} \forall & \exists & b \\ \exists y & a & \forall w \end{bmatrix}$$

Fig. 1. A 3-column operator and the corresponding $M^{1,1}$, $M^{2,1}$ in the 2-game.

The k-game in *MBL* begins with index $j = 1$ according to the rules given below. When all the choices for $(1, j), \ldots, (k, j)$ have been made, the index increases by one and the k-game continues according to the same rules. Let j be fixed. At entry (i, j), with $i = 1, \ldots, k$, agent i chooses a constant as follows:

– if entry (i, j) of $M^{i,j}$ is constant a, then agent i chooses a for (i, j). $M^{i,j+1}$ is put equal to $M^{i,j}$ and $\varphi^{i,j+1}$ equal to $\varphi^{i,j}$.
– if entry (i, j) of $M^{i,j}$ is $\forall x$ (or $\exists x$), agent i chooses a constant, say a, for x.[3] $M^{i,j+1}$ is put equal to $M^{i,j}$ with a substituted for $\forall x$ (or $\exists x$) and $\varphi^{i,j+1}$ is put equal to $\varphi^{i,j}$ with a substituted for the free occurrences of x.

Note that entry (i, j) of the information-matrix $M^{i,j}$ is always equal to entry (i, j) of $[M_n]$. Furthermore, $M^{i,j+1}$ is always equal to $M^{i,j}$ with the choice made for (i, j) shown. Formula $\varphi^{i,j}$ is used by agent i when deciding its j-th action. The formula represents the original φ as modified by previous choices made by agent i itself. Other agents' choices are not known by agent i and so they do not affect formula $\varphi^{i,j}$. This explains why only variables that occur in entries $(i, 1), \ldots, (i, j - 1)$ are instantiated in $\varphi^{i,j}$ by the corresponding values while other variables are left unchanged.

The k-game in *MBL* ends when $j = n + 1$. The *output* of the k-game is the unique constant operator A such that, for all i and j, $A(i, j)$ contains the choice made at (i, j) during the k-game.

There are a few issues pervading all of game-theory. Among these knowledge and memory are particularly important. These are key elements to understand our semantics. In the framework presented here, the agent in charge of choosing for the given entry has perfect knowledge of the general elements of the k-game, that is, $\mathcal{M}$, s, $\Im$, and of all the information in $M^{i,j}$. Also, from the description

[3] The constant constrains the interpretation of $[M_n]$ and is used to instantiate all the occurrences of x in the scope of $[M_n]$, if any.

above, each agent is perfectly aware of the changes due to its previous choices (through $\varphi^{i,j}$). These facts affect the agent strategy as we show next.

i-Strategies and Semantics. Having the structure of the k-game, we turn to the strategies for the agents, i.e., we explain how choices in quantified entries are made. A rational agent playing a k-game in *MBL* needs a *strategy*, that is, a rule telling at every step which choice(s) better fits its goal. Note that the strategy for agent i, or i-*strategy*, depends on the knowledge of agent i in the k-game.

In *MBL* we focus on choices for making a formula true. Thus, i-strategies in *MBL* define what to do at existentially quantified entries, i.e., where an agent has the (explicit) goal of making the formula true. Instead, at universally quantified entries we assume the agent chooses according to some unspecified goals. Not knowing these latter goals, we cannot characterize choises at universally quantified entries. We model this situation assuming that random choices are made at all entries containing an universal quantifier.

Given an instance $[A]$ for $[M]$, $[A](i,j)$ is said to *instantiate* variable x of $[M]$ if x occurs at $[M](i,j)$. Given a function f from $Const^*$ (the set of finite sequences of constants) to *Const*, we write $[M]_f$ for the set of instances $[A]$ of $[M]$ such that for any existentially quantified entry $[M](h,h')$, $f(h, < a_1, \ldots, a_{h'-1} >) = [A](h,h')$ with $< a_1, \ldots, a_{h'-1} >$ the empty string for $h' = 1$ and $a_r = [A](h,r)$ for all $1 \leq r < h'$. (In other words, f says how to move at existential entries: it takes as arguments the row-index and any initial part of that row of $[A]$ and outputs the next value in that row.)

Fix $\mathcal{M}$, s, and $\Im$ and assume $\models_{MBL}$ has been defined on any formula $[A]\varphi$ with $[A] \in cOP \cup vOP$. An i-*strategy* for $[M]\varphi$ in *MBL*, with $[M] \in qOP$, is a function $f_i : Const^* \rightarrow Const$ such that for all $[A] \in [M]_{f_i}$, $\mathcal{M}, s, \Im \models_{MBL} [A]\varphi^*$ where φ^* is obtained from φ as follows:
(a) any constant a of $[A]$ instantiating some variable x in row i of $[M]$ is substituted for all free occurrences of x in φ;
(b) if x occurs in $[M]$ at row j ($j \neq i$), then some constant a of $[A]$ instantiating a variable of $[M]$ not in row i is substituted for all free occurrences of x in φ.
Proviso: the overall number of variables for which a is substituted in φ cannot be higher than the total number of variables a itself instantiates in $[A]$.

The definition of i-strategy determines how an agent would play the k-game over $[M]\varphi$ if that very agent had to choose at *all* existential entries of $[M]$. (One can easily formulate a kind of k-game capturing this particular case.) Condition (a) forces any existentially quantified variable at row i of $[M]$ and its occurrences in φ to be correctly substituted by the corresponding value in $[A]$. Condition (b) ensures that the remaining variables are assigned some value that occurs in the output of the k-game. Each entry in the output $[A]$ can be associated with at most one variable in $[M]$ and, according to his information-matrix, our agent i does not know which constant instantiates which variable, thus the extra proviso guarantees a meaningful substitution.

Given k functions $f_h : Const^* \rightarrow Const$, $[M]_{\{f_1,\ldots,f_k\}}$ is the set of instances $[A]$ of $[M]$ such that for (i,j) existentially quantified entry of $[M]$, $f_i(i, < a_1,$

$\ldots, a_{j-1} >) = [A](i,j)$ with $< a_1, \ldots, a_{j-1} >$ as in the description of i-strategy. (In short, function f_i is used at all existential entries of row i only, $1 \leq i \leq k$.)

The semantics for *MBL* is as follows:

1. $\mathcal{M}, s, \Im \models_{MBL} p_i$ if $s \in [\![p_i]\!]$
2. $\mathcal{M}, s, \Im \models_{MBL} t_1 = t_2$ if $\hat{t_1} = \hat{t_2}$ where $\hat{t}$ is $[\![t]\!]$ if $t \in Const$, $\Im(t)$ otherwise
3. $\mathcal{M}, s, \Im \models_{MBL} \varphi \to \psi$ if $\mathcal{M}, s, \Im \models_{MBL} \varphi$ implies $\mathcal{M}, s, \Im \models_{MBL} \psi$
4. $\mathcal{M}, s, \Im \models_{MBL} \neg\varphi$ if not $\mathcal{M}, s, \Im \models_{MBL} \varphi$
5. Let $[M] \in cOP \cup vOP$. $\mathcal{M}, s, \Im \models_{MBL} [M]\varphi$ if $(s, s') \in R^{k,n}([\![\Im(M)]\!])$ implies $\mathcal{M}, s', \Im \models_{MBL} \varphi$
6. Let $[M] \in qOP$. $\mathcal{M}, s, \Im \models_{MBL} [M]\varphi$ if both the following conditions hold:
   a) for all i there is an i-strategy for the k-game over $\mathcal{M}$, s, $\Im$, and $[M]\varphi$;
   b) for all sets $\{f_1, \ldots, f_k\}$ where f_r is a r-strategy as in *a*), $[M]_{\{f_1,\ldots,f_k\}} \neq \emptyset$, and $[A] \in [M]_{\{f_1,\ldots,f_k\}}$ implies $\mathcal{M}, s, \Im \models_{MBL} [A]\psi$ with ψ like φ except that all the occurrences of the variable in $[M](h,j)$ are substituted by constant $A(h,j)$ and all the free occurrences of some variable x not bounded in $[M]$ are substituted by constant a with $\Im(x) = [\![a]\!]$.

4 Related Work

Researchers in multi-agent systems have been developing tools for concurrency and information issues for several decades but only recently features of information independence and information sharing with a broad prospective have been considered. This happens in particular in systems where logic and game theory merge. In this respect, we have been influenced mostly by Henkin [8] and Hintikka [9]. Nevertheless, our quantified modalities differ from branching quantifiers as used in linguistics and logic: on the one hand the notions of player and agent are quite different, on the other hand the gist of k-games lies in the operators themselves and not in the formulas to which these are applied.

Several approaches tackle issues relevant to our work. We mention a few.

R. Parikh introduced *Game Logic* in [10] to reason about program correctness. This system shows how to construct complex games out of simpler ones and its semantics does not use the formal notion of game. M. Pauly developed this work further and also presented a new system called *Coalition Logic* [11]. Although not shown in this paper, our approach generalizes the idea behind both Game Logic and Coalition Logic providing tools for a finer analysis of games.

An extensive study of the relationship between logic, game theory, and language has been carried out by Ahti Pietarinen. The analysis of epistemic issues is central to his work [12] where he introduces multi-agent systems to capture multi-person (or multi-self) games like two-agent games with imperfect memory. In comparison, our approach considers epistemic features only at the level of semantical parameters.

The logic *ATL* introduced by Alur et. al. in [1] can be captured in our language (dropping the temporal operators). *ATL* is a very interesting system with good deduction properties. Compared to our language, it cannot express multi-column operators with different quantifiers occurring in the same row.

[4] presents an extension of modal logic along the lines of Hintikka's work on IF-logic. The semantics is given through "local states". Our system has similar features and is applicable to a wider class of agents including agents acting on each other like biological substances. In [5], a system called *IF modal logic* is introduced and interpreted over runs of particular transition systems. The resulting language is quite interesting. The link between equivalences induced by the logic and those provided by the model is still unclear.

Finally, we conclude by pointing out that full *MBL* is a very rich language in expressive power. Although clearly undecidable, there are several ways to extract manageable subsets. Furthermore, a logic system with the quantified modalities of sect. 3 can be associated to different k-games, agent strategies, or semantic clauses; thus it may enjoy different properties and describe a variety of multi-agent systems. One can also express cooperation and communication among agents tuning the notion of k-game accordingly. In short, the semantics we have associated with this language should be considered as one possibility among many. In the future, we plan to expand these observations and to study the proof-theoretical properties of the interpreted languages obtained in this way.

References

1. R. Alur, T. Henzinger, and O. Kupferman. Alternating-time temporal logic. In de Roever W.-P., L. H., and P. A., editors, *Compositionality - The Significant Difference*, LNCS 1536, pages 23–60. Springer-Verlag, 1999.
2. B. Bennett, C. Dixon, M. Fisher, U. Hustadt, E. Franconi, I. Horrocks, and M. De Rijke. Combinations of modal logics. *Artificial Intelligence Review*, 17(1), 2002.
3. S. Borgo. Concurrency with partial information. In *CIMCA '03*, to appear, 2003.
4. J. C. Bradfield. Independence: logics and concurrency. In *CSL'00*, LNCS 1862, pages 247–261, 2000.
5. J. C. Bradfield and S. B. Fröschle. On logical and concurrent equivalences. *Electronic Notes in Theoretical Computer Science*, 52 (1), 2002.
6. R. Fagin, J. Y. Halpern, Y. Moses, and M. Y. Vardi. *Reasoning about Knowledge.* MIT Press, 1995.
7. D. Harel, D. Kozen, and J. Tiuryn. *Dynamic Logic.* MIT Press, 2000.
8. L. Henkin. Some remarks on infinitely long formulas. In *Infinitistic Methods*, Pergamon Press, pages 167–183, 1961.
9. J. Hintikka. *Principles of Mathematics Revisited.* Cambridge University Press, 1996.
10. R. Parikh. The logic of games and its applications. *Annals of Discrete Mathematics*, 24, 1985.
11. M. Pauly. A Modal Logic for Coalitional Power in Games. *J. of Logic and Computation*, 12(1):149–166, 2002.
12. A. Pietarinen. Reasoning about focussed knowledge in multi-agent systems. In *Workshop on Cognitive Agents and Multi-Agent Interaction*, 2001.
13. M. J. Wooldridge and N. R. Jennings. Intelligent agents: Theory and practice. *Know. Eng. Review*, 10(2):115–152, 1995.