World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:2, No:6, 2008

Modelling Multiagent Systems

Gilbert Ndjatou

Abstract**—** We propose a formal framework for the specification of the behavior of a system of agents, as well as those of the constituting agents. This framework allows us to model each agent's effectoric capability including its interactions with the other agents. We also provide an algorithm based on Milner's "observation equivalence" to derive an agent's perception of its task domain situations from its effectoric capability, and use "system computations" to model the coordinated efforts of the agents in the system . Formal definitions of the concept of "behavior equivalence" of two agents and that of system computations equivalence for an agent are also provided.

Keywords**—**Multiagent system, object system, observation equivalence, reactive systems.

I. Introduction

In [2], page 308, Genesereth and Nilsson have introduced a model of a "tropistic agent" that consists of the following 6-tuple:

(S, T, A, *see*, *do*, *action*)

where

S is the set of states where the agent's world can be in or task-domain situations,

T is a set of partitions of S that represents the sensory limitations of the agent: the agent can not distinguish states from the same partition,

A is the set of actions that the agent can perform

see is the "sensory function" *see* : $S \longrightarrow T$ that maps each state to its partition,

do is the "effectory function" *do* : $A \times S :\longrightarrow S$ that specifies the effects of the execution of each action, and

action is a function *action* : $T \longrightarrow A$ that maps each state partition into the action that the agent is to perform whenever it finds itself in a state in that partition.

The operation of a tropistic agent can be summarized as follows: on each cycle, the agent's world or task domain is in a state s. The agent observes the partition *see*(s) and uses the function *action* to determine the action $a =$ *action*(*see*(s)) that corresponds to the partition *see*(s). It then execute action a which transforms its task domain to state *do*(a, s), and the cycle repeats. However, in this paper, we are only concerned with the "effectoric capability" of an agent: that means what it can do or how it can be affected at each task domain situation.

The single agent model of Genesereth and Nilsson presumes a world in which there is only one agent and which therefore does not change except through the actions of this agent. However, as they claim themselves, "much work in AI concerns multiple agents and their interactions." Their model must therefore be extended to take into consideration the interactions among the agents. We do so by assuming that our system consists of autonomous agents: An agent has access to its task domain situations (or local states), but not to the task domain situations of the other agents in the system. The interactions among the agents are achieved by using "external actions" which correspond to Lamport's interface actions. An agent initiates an external action at one of its local states with respect to another agent in the system, which then executes it at one of its own local states with the resulting effect a transition to another local state.

Genesereth and Nilsson's agents also have sensory limitations (or perception of their task domain situations), and the system designer is supposed to know about them when he/she is specifying the "action function" to solve a problem or to perform a task. Although this is possible for some task domains such as their maze world [2], this information is not always available to the system designer. In our framework, an agent's perception of its task domain situations is derived from its "effectoric capability" by using an algorithm based on Milner's observation equivalence. In addition to specifying the "effectoric capability" of the agents in our system and to providing an algorithm to derive an agent's perception of its task domain situations, we also introduce the concept of "behavior equivalence" of agents, and the notion of system computations that describe the coordinated efforts of the agents in a system.

Our model to specify an agent's "effectoric capability" is provided in section 2. In section 3, we specify an algorithm to determine an agent's perception of its task domain situations, and also define the notion of 'behavior equivalence" of the agents. System computations and their properties are introduced in section 4.

II. Effectoric Capability of an Agent

We assume that our system consists of $m \geq 2$ agents, and we denote the set of possible *local states* of agent i by Q^i, with $Q^i \cap Q^j = \emptyset$ for $i \neq j$. An agent's local states represent its task domain situations, and correspond to the same concept as that of an agent 's external state in [2] or an environment state in [11] for a single agent system. For each agent i, we denote the union of the sets of local states of the other agents in the system by $Q^i_{\text{env}} = \bigcup_{j \neq i} Q^j$. We will refer to an agent's actions as its *internal actions*, and assume that the alphabet $\mathcal{A}$ of internal actions contains the null action ϵ.

For agent i, we replace the "effectory function" *do* with the relations $R^i_{\text{a}} \subseteq Q^i \times Q^i$, $a \in \mathcal{A}$, with $R^i_{\epsilon} = \{(s, s) | s \in Q^i \}$. R^i_{a} is the transition relation of internal action a for agent i. $(s, t) \in R^i_{\text{a}}$ if and only if agent i may perform internal action a in local state s with the resulting effect a transition to local state t.

We model the interactions among the agents by using *external actions* which correspond to Lamport's interface actions

G. Ndjatou is with the Department of Computer Science, William Paterson University, Wayne, NJ, 07470 USA, e-mail: ndjatoug@wpunj.edu.

[4]. An agent initiates an external action at one of its local states with respect to another agent in the system which then executes it at one of its own local states with the resulting effect a transition to another local state. For example, in a candy machine system where the agents are the user, the candy machine, and the candy bar container, "get coin" is an external action of the candy machine with the user as the possible initiator, and "release a candy bar" is an external action of the candy bar container with the candy machine as the possible initiator. Similarly to Lamport's interface actions [4], an agent may change its local state as a result of an external action initiation. We denote by $\mathcal{E}$ the alphabet of external actions such that $\mathcal{A} \cap \mathcal{E} = \emptyset$.

For external action $e \in \mathcal{E}$, the transition relation $S^i_{\rm e} \subseteq Q^i_{\rm env} \times Q^i_{\rm env} \times Q^i \times Q^i$ specifies how the other agents in the system may affect agent i using external action e, and $S^i_{\rm e}$- $\subseteq Q^i \times Q^i \times Q^i_{\rm env} \times Q^i_{\rm env}$ specifies how agent i may affect the other agents in the system using external action e. $(w, v, s, t) \in S^i_{\rm e}$ if and only if in state $w \in Q^i_{\rm env}$ an agent in the system may initiate external action e with respect to agent i and move to local state v, and if it does it, agent i may execute it in local state $s \in Q^i$ with the resulting effect the transition to local state $t \in Q^i$. Similarly, $(s, t, v, w) \in S^i_{\rm e}$- if and only if in local state $s \in Q^i$, agent i may initiate external action e with respect to another agent in the system and move to local state t, and if it does it, that agent may execute it in local state $v \in Q^i_{\rm env}$ with the resulting effect the transition to local state $w \in Q^i_{\rm env}$. For example, suppose that our candy machine system gives us a candy bar (but no change) if and only if the amount of money (nickels and dimes) deposited into the machine is 20 cents. If we identify the local states of the user with the value of the coin that it has selected from its pocket, and use the pairs (n, d) (where n and d are respectively the number of nickels and the number of dimes received so far by the candy machine) to represent its local states, then the transition relation that corresponds to the candy machine's external action "get coin" is specified as follows:

$(5, 0, (0, 0), (1, 0)), (5, 0, (1, 0), (2, 0)),$
$(5, 0, (2, 0), (3, 0)), (5, 0, (3, 0), (4, 0)),$
$(5, 0, (0, 1), (1, 1)), (5, 0, (1, 1), (2, 1)),$
$(10, 0, (0, 0), (0, 1)), (10, 0, (1, 0), (1, 1)),$
$(10, 0, (2, 0), (2, 1)), (10, 0, (0, 1), (0, 2)).$

After the user has deposited a coin into the candy machine, it moves to the local state 0 (no coin has been selected from its pocket) and after the candy machine has received a coin (from the user), it moves to the local state that corresponds to the increment of that coin counter by 1. The external action "release a candy bar" of the candy bar container (with the candy machine as the initiator) is specified as follows:

$((4, 0), (0, 0), 0, 0), ((2, 1), (0, 0), 0, 0), ((0, 2), (0, 0), 0, 0),$
and for $n \geq 1$,
$((4, 0), (0, 0), n, n - 1), ((2, 1), (0, 0), n, n - 1),$
$((0, 2), (0, 0), n, n - 1).$

The candy machine resets its nickels and dimes counters to zero after it has initiated the "release a candy bar" external action, and when there is no candy bar into the candy bar container, it does nothing when it is ordered (by the candy machine) to release a candy bar; otherwise, it decrements the candy bar counter by 1.

Note that if agent i interacts only with agent j using external action e and is the only agent that does it as in the case of the user, the candy machine, and the external action "get coin," then $S^j_{\rm e} = S^i_{\rm e}$-.

The *effectoric capability* of agent i is specified by the following model O^i that we refer to as *object System* [7, 8, 9].

$O^i = (Q^i, Q^i_{\rm env}, \{R^i_{\rm a}, a \in \mathcal{A}\}, \{S^i_{\rm e}, e \in \mathcal{E}\}, \{S^i_{\rm e}$-, $e \in \mathcal{E}\})$.

Note that there is a minor difference between this specification of an object system and the one introduced in [7, 8, 9]. In [7, 8, 9], an agent does not change state after the initiation of an external action.

An agent's external action (respectively its environment's external action) corresponds to an input action (respectively output action) of I/O automata [6], [10] that are used to model the behavior of a single agent that interacts with its environment. Object systems can be viewed as a generalization of I/O automata in the sense that they allow us to specify interactions between two or more agents instead of just one agent and its environment as do I/O automata.

III. Agents' Perceptions of the World and Behavior Equivalence

The action function [2] or policy [11] that an agent uses to solve a problem or to perform a task is based on its sensory limitations [2] (or perception of its task domain situations) and in [2], a system designer is supposed to know an agent's perception of its task domain situations in order to design these action functions. However, as we have stated in the introduction, this information is not always available to the system designer. In this section, we propose an algorithm to derive an agent's perception of its task domain situations that is based on Milner's [3] "observation equivalence." Intuitively, two local states of an agent are said to be "observation equivalent" if for an observer, their differences are irrelevant to the ways the agent performs its internal actions or interacts with the other agents at those states. In other words, the agent can only initiate the same external actions of the other agents (with identical effects) in both states, and the effects of the execution of internal and external actions from those states are identical in what it can later do.

Given an object system $O^i = (Q^i, Q^i_{\rm env}, \{R^i_{\rm a}, a \in \mathcal{A}\}, \{S^i_{\rm e}, e \in \mathcal{E}\}, \{S^i_{\rm e}$-, $e \in \mathcal{E}\})$, we define the decreasing sequence of equivalence relations $\sim_{i,m}$ over Q^i, $(m \geq 1)$ as follows:

- $\sim_{\rm i,0} = Q^i \times Q^i$
- $\forall m \geq 1, s \sim_{\rm i,m} t$ iff
  1) $\forall a \in \mathcal{A}$, if $\exists u \in Q^i$ such that $(s, u) \in R^i_{\rm a}$, then $\exists u' \in Q^i$ such that $u' \sim_{\rm i,m-1} u$ and $(t, u') \in R^i_{\rm a}$
  2) $\forall a \in \mathcal{A}$, if $\exists u \in Q^i$ such that $(t, u) \in R^i_{\rm a}$, then $\exists u' \in Q^i$ such that $u' \sim_{\rm i,m-1} u$ and $(s, u') \in R^i_{\rm a}$
  3) $\forall e \in \mathcal{E}$, $w, v \in Q^i_{\rm env}$, if $\exists u \in Q^i$ such that $(w, v, s, u) \in S^i_{\rm e}$, then $\exists u' \in Q^i$ such that $u' \sim_{\rm i,m-1} u$ and $(w, v, t, u') \in S^i_{\rm e}$

4) $\forall e \in \mathcal{E}$, $w, v \in Q^i_{\text{env}}$, if $\exists u \in Q^i$ such that $(w, v, t, u) \in S^i_{\text{e}}$, then $\exists u' \in Q^i$ such that $u' \sim_{\text{i,m}-1} u$ and $(w, v, s, u') \in S^i_{\text{e}}$
5) $\forall e \in \mathcal{E}$, $w, v \in Q^i_{\text{env}}$, if $\exists u \in Q^i$ such that $(s, u, w, v) \in S^i_{\text{e}}$-, then $\exists u' \in Q^i$ such that $u' \sim_{\text{i,m}-1} u$ and $(t, u', w, v) \in S^i_{\text{e}}$-
6) $\forall e \in \mathcal{E}$, $w, v \in Q^i_{\text{env}}$, if $\exists u \in Q^i$ such that $(t, u, w, v) \in S^i_{\text{e}}$-, then $\exists u' \in Q^i$ such that $u' \sim_{\text{i,m}-1} u$ and $(s, u', w, v) \in S^i_{\text{e}}$-

Since $\epsilon \in \mathcal{A}$, it follows from condition 1 or 2, that if $s \sim_{\text{i,k}} t$ then $\forall j \leq k, s \sim_{\text{i,j}} t$.

Local states s, t are *observation equivalent*, denoted by $s \sim_i t$ iff

$$\forall m \geq 0, s \sim_{\text{i,m}} t$$

The proof that the relation $\sim_i$ is well-defined and is an equivalence relation on Q^i is similar to that provided in [3]. The equivalence class of local state $s \in Q^i$ is denoted by $\bar{s}$ and the set of all equivalence classes by $\bar{Q}^i$. This set corresponds to agent i's perception of its task domain situations. An algorithm to determine the relation $\sim_i$ follows.

Algorithm to Determine the "Observation Equivalence" Relation

Step 1: Set $\Delta^i{}_0 = Q^i \times Q^i$ and j to 1.

step 2: Determine $\Delta^i_{\text{j}} \subseteq \Delta^i_{\text{j}-1}$ such that $(s, t) \in \Delta^i_{\text{j}}$ iff

- $\forall a \in \mathcal{A}$, if $\exists u \in Q^i$ such that $(s, u) \in R^i_{\text{a}}$, then $\exists u' \in Q^i$ such that $(u, u') \in \Delta^i_{\text{j}-1}$ and $(t, u') \in R^i_{\text{a}}$
- $\forall a \in \mathcal{A}$, if $\exists u \in Q^i$ such that $(t, u) \in R^i_{\text{a}}$, then $\exists u' \in Q^i$ such that $(u, u') \in \Delta^i_{\text{j}-1}$ and $(s, u') \in R^i_{\text{a}}$
- $\forall e \in \mathcal{E}$, $w, v \in Q^i_{\text{env}}$, if $\exists u \in Q^i$ such that $(w, v, s, u) \in S^i_{\text{e}}$, then $\exists u' \in Q^i$ such that $(u, u') \in \Delta^i_{\text{j}-1}$ and $(w, v, t, u') \in S^i_{\text{e}}$
- $\forall e \in \mathcal{E}$, $w, v \in Q^i_{\text{env}}$, if $\exists u \in Q^i$ such that $(w, v, t, u) \in S^i_{\text{e}}$, then $\exists u' \in Q^i$ such that $(u, u') \in \Delta^i_{\text{j}-1}$ and $(w, v, s, u') \in S^i_{\text{e}}$
- $\forall e \in \mathcal{E}$, $w, v \in Q^i_{\text{env}}$, if $\exists u \in Q^i$ such that $(s, u, w, v) \in S^i_{\text{e}}$-, then $\exists u' \in Q^i$ such that $(u, u') \in \Delta^i_{\text{j}-1}$ and $(t, u', w, v) \in S^i_{\text{e}}$-
- $\forall e \in \mathcal{E}$, $w, v \in Q^i_{\text{env}}$, if $\exists u \in Q^i$ such that $(t, u, w, v) \in S^i_{\text{e}}$-, then $\exists u' \in Q^i$ such that $(u, u') \in \Delta^i_{\text{j}-1}$ and $(s, u', w, v) \in S^i_{\text{e}}$-

Step 3: if $\Delta^i_{\text{j}} = \Delta^i_{\text{j}-1}$, set Δ^i to Δ^i_{j} and stop; otherwise increment j by 1 and go to Step 2.

By using this algorithm on the local states of the above candy machine, we obtain the following equivalence classes: $\{(0,0)\}$, $\{(1,0)\}$, $\{(2,0),(0,1)\}$, $\{(3,0),(1,1)$, and $\{(4,0),(2,1),(0,2)\}$.

Proposition 1: The equivalence classes are stable with respect to the relations R^i_{a}, $a \in \mathcal{A}$, S^i_{e}, $e \in \mathcal{E}$, and S^i_{e}-, $e \in \mathcal{E}$ in the sense that:

- $\forall a \in \mathcal{A}$, and $s, t \in Q^i$, if $(s, t) \in R^i_{\text{a}}$ then $\forall s' \in \bar{s}, \exists t' \in \bar{t}$ such that $(s', t') \in R^i_{\text{a}}$.
- $\forall e \in \mathcal{E}$, and $s, t \in Q^i$, if $(w, v, s, t) \in S^i_{\text{e}}$ for some $w, v \in Q^i_{\text{env}}$, then $\forall s' \in \bar{s}, \exists t' \in \bar{t}$ such that $(w, v, s', t') \in S^i_{\text{e}}$.
- $\forall e \in \mathcal{E}$, and $s, t \in Q^i$, if $(s, t, w, v) \in S^i_{\text{e}}$- for some $w, v \in Q^i_{\text{env}}$, then $\forall s' \in \bar{s}, \exists t' \in \bar{t}$ such that $(s', t', w, v) \in S^i_{\text{e}}$-.

This proposition follows immediately from the definition of the "observation equivalence" relation. An immediate consequence is that the relations R^i_{a}, $a \in \mathcal{A}$, S^i_{e} $e \in \mathcal{E}$, and S^i_{e}- $e \in \mathcal{E}$ can be extended to the relations $\bar{R}^i{}_{\text{a}} \subseteq \bar{Q}^i \times \bar{Q}^i$, $\bar{S}^i{}_{\text{e}} \subseteq Q^i_{\text{env}} \times Q^i_{\text{env}} \times \bar{Q}^i \times \bar{Q}^i$, and $\bar{S}^i{}_{\text{e}}$- $\subseteq \bar{Q}^i \times \bar{Q}^i \times Q^i_{\text{env}} \times Q^i_{\text{env}}$ respectively as follows:

- $(\bar{s}, \bar{t}) \in \bar{R}^i{}_{\text{a}}$ iff $\forall s' \in \bar{s}, \exists t' \in \bar{t}$ such that $(s', t') \in R^i_{\text{a}}$
- $(w, v, \bar{s}, \bar{t}) \in \bar{S}^i{}_{\text{e}}$ iff $\forall s' \in \bar{s}, \exists t' \in \bar{t}$ such that $(w, v, s', t') \in S^i_{\text{e}}$ and
- $(\bar{s}, \bar{t}, w, v) \in \bar{S}^i{}_{\text{e}}$- iff $\forall s' \in \bar{s}$, $\exists t' \in \bar{t}$ such that $(s', t', w, v) \in S^i_{\text{e}}$-.

$(\bar{s}, \bar{t}) \in \bar{R}^i{}_{\text{a}}$ iff at any local state $s' \in \bar{s}$, there is an execution of internal action a that terminates in some local state in $\bar{t}$, and all executions of a at a local state in $\bar{s}$ that terminate in some local state in $\bar{t}$ are indistinguishable.
$(w, v, \bar{s}, \bar{t}) \in \bar{S}^i{}_{\text{e}}$ iff an agent in the environment may initiate external action e (with respect to agent i) at local state w and move to local state v and if it does it, agent i may execute it in a local state $s' \in \bar{s}$ with the resulting effect the transition to a local state $t' \in \bar{t}$. Moreover, all such executions of e at a local state in $\bar{s}$ that terminate in some local state in $\bar{t}$ are indistinguishable.
$(\bar{s}, \bar{t}, w, v) \in \bar{S}^i{}_{\text{e}}$- iff in some local state $s' \in \bar{s}$, agent i may initiate external action e (with respect to another agent in the system) and move to a local state $t' \in \bar{t}$, and if it does it, that agent may execute it in local state w with the resulting effect the transition to local states v. Moreover, all such initiations of external action e at a local state in $s' \in \bar{s}$ with a move to a local state $t' \in \bar{t}$ are indistinguishable.

It follows from the above proposition that from object system $O^i = (Q^i, Q^i_{\text{env}}, \{R^i_{\text{a}}, a \in \mathcal{A}\}, \{S^i_{\text{e}}, e \in \mathcal{E}\}, \{S^i_{\text{e}}$-, $e \in \mathcal{E}\})$, we can derive the object system $\bar{O}^i = (\bar{Q}^i, Q^i_{\text{env}}, \{\bar{R}^i{}_{\text{a}}, a \in \mathcal{A}\}, \{\bar{S}^i{}_{\text{e}}, e \in \mathcal{E}\}, \{\bar{S}^i{}_{\text{e}}$-, $e \in \mathcal{E}\})$ with the same behavior as O^i and at most the same number of local states as O^i. We therefore have the following definition of "behavior equivalence" of object systems:

Definition 2 (Behavior Equivalence): Object systems O^1 and O^2 are behavior equivalent if and only if there exists a bijection $f : \bar{Q}^1 \longrightarrow \bar{Q}^2$ such that:

- $\forall a \in \mathcal{A}$, $\bar{s}, \bar{t} \in \bar{Q}^1$, $(\bar{s}, \bar{t}) \in \bar{R}^1{}_{\text{a}}$ iff $(f(\bar{s}), f(\bar{t})) \in \bar{R}^2{}_{\text{a}}$.
- $\forall e \in \mathcal{E}$, $\bar{s}, \bar{t} \in \bar{Q}^1$, $\exists w, v \in Q^1_{\text{env}}$ s.t. $(w, v, \bar{s}, \bar{t}) \in \bar{S}^1{}_{\text{e}}$ iff $\exists w', v' \in Q^2_{\text{env}}$ s.t. $(w', v', f(\bar{s}), f(\bar{t})) \in \bar{S}^2{}_{\text{e}}$.
- $\forall e \in \mathcal{E}$, $\bar{s}$, $\bar{t} \in \bar{Q}^1$, $\exists w, v \in Q^1_{\text{env}}$ s.t. $(\bar{s}, \bar{t}, w, v) \in \bar{S}^1{}_{\text{e}}$- iff $\exists w', v' \in Q^2_{\text{env}}$ s.t. $(f(\bar{s}), f(\bar{t}), w', v') \in \bar{S}^2{}_{\text{e}}$-.

It follows from the above definition that object systems O^1 and O^2 have the same behavior if and only if the object systems $\bar{Q}^1$ and $\bar{Q}^2$ are isomorphic.

IV. System Computations

In a single agent system, the ability of the agent to solve a problem or to perform a task is completely determined by its action and effectory functions. However, in a system with two or more agents, we must also take into consideration the interactions among the agents in the system: a problem is solved or a task is performed by the coordinated efforts of the agents in the system. These coordinated efforts of the agents in a system are modelled with system computations that we formally define along with the concept of "system computations equivalence" for an agent i in this section.

System computations are defined with respect to some initial local state q_0^i for each agent i, and are used to determine what can or cannot be achieved by the agents in the system when each one starts its computation at its initial local state. A *computation* of agent i is a sequence of events on agent i that starts in initial local state q_0^i. An *event* is either the execution of an internal action (*internal event*), the initiation of an external action (*send event*), or the execution of an external action initiated by another agent in the system (*receive event*). These events on agent i (i-events) may be specified as relations on agent i's local states as follows:

1) $s \xrightarrow{(i,a)} t$ iff $(s,t) \in R_{\mathrm{a}}^i$
2) $s \xrightarrow{S(i,\bar{u},e,j)} t$ iff $s \in \bar{u}$ and for some $w, v \in Q^j$, $(s,t,w,v) \in S_{\mathrm{e}}^i$-
3) $s \xrightarrow{R(i,\bar{u},e,j)} t$ iff for some $u' \in \bar{u}$, and $v \in Q^j$, $(u',v,s,t) \in S_{\mathrm{e}}^i$

Note that if $(s,t,w,v) \in S_{\mathrm{e}}^i$- for some $w, v \in Q_{env}^i$, then $\forall s' \in \bar{s}, \exists t' \in Q^i$ s.t. $(s',t',w,v) \in S_{\mathrm{e}}^i$-. Therefore, we can talk about an external action being initiated from a local state equivalence class, instead of a local state. Event 2 states that agent i initiates external action e of agent j in the local state equivalence class $\bar{u}$ and moves to local state t. Event 3 states that in local state s, agent i executes external action e initiated by agent j in local state equivalence class $\bar{u}$ with the resulting effect the transition to local state t. Note that the send event $S(i,\bar{u},e,j)$ corresponds to the receive event $R(j,\bar{u},e,i)$ and must occur in a system computation before it.

As in [1] and [4], we define a *system computation* to be an interleaving sequence of events on component agents such that the sequence of events on each component agent is a computation of that agent, and every receive event is preceded by the corresponding send event. For example, suppose that our system consists of two agents 1 and 2 and that $e_0^1 S(1,\bar{u},e,2) e_2^1 e_3^1 e_4^1$ and $e_0^2 e_1^2 e_2^2 R(2,\bar{u},e,1) e_4^2 e_5^2$ are computations on agents 1 and 2 respectively. Then the following are possible computations of the system: $e_0^1 S(1,\bar{u},e,2) e_2^1 e_3^1 e_4^1$; $e_0^1 e_0^2 e_1^2 S(1,\bar{u},e,2) e_2^1 e_3^1$; $e_0^1 e_0^2 S(1,\bar{u},e,2) e_1^2 e_2^1 R(2,\bar{u},e,1) e_3^1 e_4^2$. The first one consists of just the events on agent 1 and since there is no receive event without the corresponding send event in it, it is a system computation. But $e_0^1 e_0^2 e_1^2 R(2,\bar{u},e,1) e_3^1 e_4^2$ is not a system computation because it contains a receive event without the corresponding send event. Note also that if $e_0^1 S(1,\bar{u},e,2) e_3^1$ is not a computation on agent 1, then $e_0^1 e_0^2 e_1^2 S(1,\bar{u},e,2) e_3^1$ is not a system computation. We will denote by x^i the sequence of events on agent i in the sequence of events on component agents x. For example, if $x = e_0^1 e_0^2 S(1,\bar{u},e,2) e_1^2 e_2^1 R(2,\bar{u},e,1) e_3^1 e_4^2$ then $x^1 = e_0^1 S(1,\bar{u},e,2) e_2^1 e_3^1$ and $x^2 = e_0^2 e_1^2 R(2,\bar{u},e,1) e_4^2$. We now have the following definition of a system computation.

Definition 3: A sequence of events x on component agents is a system computation if:

- for each $i \geq 1$, x^i is a computation of agent i.
- for every receive event $R(j,\bar{u},e,i)$ of x, there is the corresponding send event $S(i,\bar{u},e,j)$ that occured earlier in x.

In order to define the concept of *system computation equivalence* for an agent i, we observe that the computation x^i of agent i in the system computation x starts at the initial local state q_0^i and terminates in some local state q_n^i with $n \geq 0$. We denote by $T^i(x)$ the set of local states of agent i where it can terminate after the execution of x^i. Intuitively, system computations x and y are *equivalent* for agent i (*i-equivalent*) if and only if the computations x^i and y^i of agent i can only terminate in local states of agent i such that it cannot distinguish one from the other; and this is the case when they belong to the same local state equivalence class or are "observation equivalent". We therefore have the following definition of "system computations equivalence" for an agent.

Definition 4: System computations x and y are equivalent for agent i (denoted by $x \equiv_i y$) iff

- $\forall s \in T^i(x), \exists s' \in T^i(y)$ such that $s \sim_i s'$
- $\forall t \in T^i(y), \exists t' \in T^i(x)$ such that $t \sim_i t'$

It is easily shown that the relation $\equiv_i$ is an equivalence relation on system computations.

If x and y are sequences of events on component agents, then we denote by xy the *concatenation* of x and y, and the fact that x is a prefix of y is denoted by $x \preceq y$. x is a proper prefix of y is denoted by $x \prec y$. Note that system computations are prefix-closed. That means, if y is a system computation and $x \preceq y$, then x is a system computation. The following properties of system computations are easy to derive.

Theorem 5 (Computation Extensions): 1) If x is a system computation and y is a sequence of events on component agents such that xy is a system computation, then we have the following:
   a) If there is no receive i-event in y, then xy^i is a system computation and $xy^i \equiv_i xy$.
   b) If every receive i-event in y has its corresponding send event in x, then xy^i is a system computation and $xy^i \equiv_i xy$.
2) If x is a system computation and y^i is a sequence of i-event such that xy^i is a system computation, then $\forall j \neq i$, $xy^i \equiv_j x$.
3) If x and z are system computations such that $x \equiv_i z$ and y^i is a sequence of i-events not containing a receive

event, then xy^i and zy^i are system computations, and $xy^i \equiv_i zy^i$.

V. Conclusion

We have proposed a formal framework for the specification of the behavior of a system of agents, as well as those of the constituting agents. In this framework, the effectoric capability of an agent is specified by using the single agent model of Genesereth and Nilsson to which we have added the possibility of interactions among the agents. The interactions among the agents are modelled by using Lamport's interface actions, and an agent's perception of its task domain situations is derived from its effectoric capability by using an algorithm based on Milner's observation equivalence. The coordinated efforts of the agents in a system are modelled using system computations, and we have also provided the formal definitions of the concepts of behavior equivalence of two agents and that of system computations equivalence for an agent. One possible application of this formal framework is the knowledge-based analysis of a system of agents if we assume that an agent's partition of its task domain situations corresponds to its local knowledge and that some goals of the interactions among the agents are to learn about this local knowledge and/or to provide information about it..

References

[1] K.M. Chandy and J. Misra. How processes learn. *Distributed Computing* 1(1), pp. 40-52. Springer-Verlag, 1986.
[2] M.R. Genesereth and N.J. Nilsson. *Logical Foundations of Artificial Intelligence*. Morgan Kaufmann, Palo Alto, CA, 1987.
[3] M. Hennesy and R. Milner. Algebraic laws for nondeterminism and concurrency. *Journal of the ACM* 32(1), pp. 137-161. 1985.
[4] L. Lamport. A simple approach to specifying concurrent systems. *Communications of the ACM* 32(1), pp. 32-45. 1989.
[5] B. van Linder, W. van der Hoek, and J.J. Ch. Meyer. Formalising abilities and opportunities of agents. *Fundamenta Informatica*, 34, (1,2), pp. 53-101, 1998.
[6] N. Lynch and M. Tuttle. An introduction to input/output automata. *CWI Quaterly* 2(3), pp. 219-246. 1989.
[7] G. Ndjatou. Minimizing agent specifications using a logic of knowledge and actions. *Journal of Logic and Computation*, vol. 11, No. 2, pp. 337-354, Oxford University Press, 2001.
[8] G. Ndjatou. *Modelling Objects, Knowledge and Learning in Distributed Object-Based Systems*. Ph.D thesis, Dept. of Computer Science, CUNY Graduate School, New York, NY, Feb. 93. Also appeared as Technical Report N0 TR-93-04-02, CUNY Graduate School, New York, NY, April 93.
[9] G. Ndjatou. Modelling objects and distributed object-based systems. In *proceedings of the 10th Israeli Symposium on AI and Computer Vision*, pp. 39-49, Ramat Gan, Israel, December 1993.
[10] N. Reingold, D.W. Wang and L.D. Zuck. Games I/O automata play. *Concur '92, Lecture Notes in Computer Science*, vol. 630, W.R Cleaveland, ed. pp. 325-339. Springer-verlag, 1992.
[11] G. Weiss, ed., *Multiagent Systems, A Modern Approach to Distributed Artificial Intelligence*. The MIT Press, 1999.