Agent Oriented Technology Based on the M-Agent Architecture.

Krzysztof CETNAROWICZ
Institute of Computer Science
University of Mining and Metallurgy - AGH
E-mail: cetnar@uci.agh.edu.pl

(presented: workshop at the Polish-Japan
Institute of Information Technology,
Warsaw, Poland 1999)

June 13, 2000

Abstract

The complexity of distributed and decentralized systems demands new tools for designing and programming processes. An idea of autonomous agents that arises as an extension of the object and process concepts may be applied to distributed and decentralized systems development ([9],[11]). In the paper the authors have undertaken an attempt to describe formally the architecture of multiagent systems (called M-agent architecture [4]) that may be considered as a starting point to develop decentralized multi-agent system technology such as Agent Oriented Analysis, Design and Programming.

keywords: Autonomous agent, agent architecture, multi-agent system, MAS, Agent Oriented Analysis, Agent Oriented Design.
1 Introduction

Although multi-agent systems have been studied for nearly ten years, a technology of such systems creation has not been proposed yet. There are some approaches ([7], [12], [16]), that make some attempts to a practical model of multi-agent systems definition, but the research has not been completed yet and the results are not always practically applicable.

An idea of M-agents that arises as an extension of the object, process ([4]) and multi-model control systems concepts ([1] [2]) may be applied to distributed and decentralized intelligent systems development. In the paper a proposal for a formal definition of the M-agent architecture, multi-agent world including a definition of agents, agents’ space, agents’ environment and the relations between agents and environment, and among agents themselves is presented.

The approach proposed, based on the concept of successive refinements of created constructions, gives tools that make possible to define multi-agent system for a given problem without using any programming languages [4], [15].

2 Introduction to agent architectures

The approaches proposed, (BDI architecture, Agent-0 etc.) ([12], [16]), ([7]) make some attempts to a practical model of multiagent system definition, but it seems to the author that these attempts are not always practically applicable, and particularly they present, among others, the following inconveniences:

• It takes as a base of considerations a low level of abstraction of the problem and therefore the proposed model of the agent architecture does not cover a wide range types of agents (or multiagent systems).

• The formalism proposed is not allways convenient in practical applications where the real problem with its characteristic features is to be interpreted in the formal model.

• The approches proposed do not take into consideration such problems as testing and debugging of the creating system.
The idea proposed of M-agents may be considered as an extension of the object, process and multi-model control systems concepts. It seems that it takes as a starting-point properly defined abstraction of the problem that may be a base for the Agent Oriented Analysis, design and even programming. In particular:

- A definition of the system is formulated the levels of specifications. It makes possible to apply the method to a wide range of MAS with more simple and more complicated agents.

- The approach proposed makes possible to use various (a choice of the most convenient) formalisms to make a specification of the system reasonable with such complicated and great systems as MAS.

- The approach proposed using abstraction levels for successive improvement of the system definition gives a tool to map a given real problem in the notion of the MAS. The concept of the M-agent architecture makes possible to define the multiagent system by successive refinements going step by step through abstraction levels. At a given level only some properties of the problem are considered and going down to the lower one, more details of the problem are incorporated into the model.

- The approach proposed makes possible to make a clear distinction between multiagent and non-multiagent system and keep a track on the development process to build desired multiagent systems.

- The concept of the analysis proposed is derived from the approach presented by Peter Coad in Object Analysis, Design and Programming ([5],[6]) and may be seen by a user as a logical extension or prolongation of that approach.

The paper presents an introduction to the M-agent architecture and its application to the description of the multiagent systems. The approach proposed, based on the concept of successive refinements of created constructions, gives tools that make possible to define a multi-agent system for a given problem without using any programming languages [4], [15].
3 Informal introduction to the concept of the M-agent architecture

We can consider a given problem for which a multiagent system (MAS) is to be created. To design a multiagent system and create a multiagent model of the problem we have to define:

- living space for agents,
- agents of several types,
- relations:
  - between the agents and the living space,
  - among the agents themselves.

![Diagram showing the principle of the M-agent architecture.]

Figure 1: Principle of the M-agent architecture. \(I\) - imagination operation, \(X\) - execution operation, \(m, m'\) - models of the environment, \(s\) - strategy, \(q\) - evaluation function, \(ev_i\) - events taking place in the environment, \(ev_x\) - event produced as the result of the realisation of the strategy \(s\).

The first undertaken step to put in order the multiagent world is to specify: what is an agent and what is not an agent. So the whole reality analyzed from the point of view of the multiagent system may be divided into two parts (fig. 1):
- environment - that may be observed by the agent and represented by a corresponding model;

- agent’s mind - an area in which the agent builds and processes a model of the environment;

So we can consider that a given agent $a$ remains and acts in an environment called $V$. For any created agent $a$ the following are defined:

- strategies $S$ ($s$ - strategy $s \in S$), goals $Q$ ($q$ - goal $q \in Q$), and models of the environment $M$ ($m$ - model of the environment, $m \in M$).

- operation of the observation $I$ and operation of the strategy execution $X$.

The main idea of the agent functioning is the following:

- The agent observes the environment around and builds a model $m$ of the environment in its mind. For this purpose it uses the observation (or imagination) operation $I$.

- The agent forecasts its possibilities using the strategy $s$. Applying the strategy $s$ to the model $m$ it obtains the model $m'$ of the modified environment. Then, the agent compares in its mind the models $m$ and $m'$ using the function $q$ that determines the goal of the agent functioning. It serves to select the best (from the point of view of the goal $g$) strategy $s^*$.

- If the evaluation of $m$ and $m'$ is satisfactory the agent realizes the selected strategy $s^*$ that is represented by operation $X$ - execution: $V' = X(s, V)$

So, in the rough, the first approach to the algorithm of a given agent may be expressed in three stages:

1. Observation - creation of the model $m$ of the environment $V$:
   \[ m = I(M, V) \]  
   (1)

2. Decision - selection of the optimal strategy $s^*$:
   \[ s^* = F(M, Q, S, V) \]  
   (2)

5
3. Action - realization of the optimal strategy \( s^* \) in the environment \( V \):

\[
v' = X(s, V)
\]  

(3)

where:

\( M, Q, S \) - sets of models, goals, strategies representing knowledge of the agent \( a \),

\( V \) - environment (state),

\( I \) - observation function,

\( F \) - decision function enabling selection of the optimal strategy \( s^* \),

\( X \) - function representing an operation of the realization (execution) of the agent’s strategies.

The whole decision process is carried out by the processing of the models of the environment in the area called agent’s mind.

The starting point to the agent oriented approach may be derived from the Object Oriented Analysis and Design and similarly expressed in the sentence: “you are an agent, you must get along by yourself in the environment” ([5], [6]). This is a useful starting point to define agents of the multiagent system with the M-agent architecture concept.

4 Formal description of the M-agent architecture

More formal definitions of the M-agent architecture are as follows:

\( a \) - agent (a given agent of the MAS),

\( A \) - set of agents that are in the MAS, called the configuration of agents, \( (a \in A) \),

\( \mathcal{A} \) - set of all configurations of agents, \( (A \in \mathcal{A}) \).

There may be several types of agents in the MAS.

\( a_i^g \) - a given agent \( i \) of type \( g \),

\( E \) - space of the MAS - living space for the agent of the system,

\( \mathcal{E} \) - set of all spaces of the system, \( (E \in \mathcal{E}) \).

The definition of the MAS environment is as follows:

**Def. 4.1** An Environment \( V \) is described by the triple:

\[
V = (E, A, C)
\]  

(4)
where:

\( C \) - connection between agents \( A \) and space \( E \). It may be defined as the location of agents in the space.

In some practical cases we may use a simplified version of the environment definition in which we consider only this part of the environment \( V \) that may be observed, modified and may interact with the considered agent \( a \). This environment is called "the environment from the point of view of the agent \( a \)" and is defined:

\[
V_a = (E, A_a, C_a)
\]

(5)

where: \( A_a \subset A \) - set of agents that may interact with the agent \( a \), \( C_a \) - connection between agents \( A_a \) and the space \( E \).

A set of all configurations of agents is denoted by \( A \ (A \in A) \); a set of all environments that take place in the MAS is denoted by: \( V \ (V \in V) \); a set of all connections that take place in the MAS is denoted by: \( C \ (C \in C) \).

The living space of agents may be defined in the following way:

- \( r \) - a resource of the MAS, \( R \) - set of resources (configuration of resources), \( r \in R \), \( R \) - set of all possible configurations of resources in the MAS, \( R \in R \),
- \( T \) - topology of the living space of the MAS; it may define a set of places \( t \ (t \in T) \) where agents can live and work, \( T \) - set of all possible topologies in the MAS, \( T \in T \),
- The structure of the space \( E \) may be defined as a couple: \( E = (R, T) \), and \( E = R \times T \), with: \( E \in E \)

We can consider more a complex model of M-agent architecture in which the agent is defined by the following way (fig. 2):

**Def. 4.2** Agent \( a \), \( (a \in A) \) may be described as 9-tuple:

\[
a = (M, Q, S, I, X, L, m, q, s)
\]

(6)

where:

- \( m \) - model of the agent’s environment, \( M \) - set of models of environments (configuration of models) - knowledge of the agent \( a \) about environments,
Figure 2: M-agent architecture

\( (m \in M), \mathcal{M} \) - set of agent’s configurations of models \( M, M \in \mathcal{M} \),

\( q \) - goal of the agent \( a \) defined as:

\[
q : M \times M \rightarrow \mathbb{R}, \quad q(m, m') \in \mathbb{R} \tag{7}
\]

where:

\( m \) - agent’s model of the environment, \( m' \) - model of the (predicted) new modified environment, \( \mathbb{R} \) real numbers,

\( Q \) - set (ordered) of agent’s goals called configuration of agent’s goals \( (q \in Q) \),

\( s \) - strategy of the agent \( a \) \( (s : M \rightarrow M, m' = s(m)) \), \( S \) - set of agent’s strategies called configuration of strategies \( (s \in S) \), \( \mathcal{S} \) - set of configuration of strategies, \( (S \in \mathcal{S}) \),

\( I \) - agent’s observation operation defined as:

\[
I : \mathcal{M} \times \mathcal{V} \rightarrow M \quad m = I(M, V), \quad m \in M \tag{8}
\]

\( X \) - agent’s strategy execution operation defined as:

\[
X : \mathcal{S} \times \mathcal{V} \rightarrow \mathcal{V} \quad V' = X(s, V), \quad s \in \mathcal{S}, \quad V, V' \in \mathcal{V} \tag{9}
\]
where: \( V' \) - new agent’s environment (environment with agent’s location) created due to the execution of the strategy \( s \) by the agent in the environment \( V = (E, A, C) \).

\( L \) - agent’s adaptation (learning) operation, \( L = \{L_M, L_S\} \)

\[
\begin{align*}
L_M : \mathcal{M} \times \mathcal{M} \times \mathcal{M} & \rightarrow \mathcal{M} \quad M' = L_M(M, m', m'') \\
L_S : \mathcal{S} \times \mathcal{M} \times \mathcal{M} & \rightarrow \mathcal{S} \quad S' = L_S(S, m', m'')
\end{align*}
\]  

Algorithm of the agent:

The agent acts according to the following algorithm (fig. 2):

1. Start - the agent’s creation: \((M, Q, S, I, X, L, m, q, s)\), and its location in the environment: \( V = (E, A, C) \), then go to 2.
2. The agent looks around and builds the model of its environment: \( m = I(M, V) \), then go to 3.
3. Selection of the best strategy \((s^*)\) that may be adopted and executed:

\[
DI(s, m) = q(m, s(m)) \quad DI^* = \max_{s \in \mathcal{S}} DI(s, m) \rightarrow s^* \tag{12}
\]

4. If there is \( s^* \) such that \( DI^* > 0 \) then go to 5. else go to 8.
5. Execute selected strategy \( s^* \): \( X(s^*, V) = V' \) then go to 6.
6. The agent looks around and builds the model of its new environment: \( m'' = I(M, V) \), then go to 7.
7. If there is considerable discrepancy between \( m' \) and \( m'' \) then adaptation to the environment (learning): \( M' = L_M(M, m', m''), S' = L_S(S, m', m'') \).

Then go to 3.
8. Go to point 2.

5 Agent’s Model with Multiple Profiles

In some case we may observe that beings quite alter their behaviour as if they were changing their point of view. It may be considered as a new M-agent model (described above). In such a case we may extend a model of an agent and considered it as multiprofile M-agent architecture. Each profile is a M-agent model. The agent observes the environment in each profile, and final decision is undertaken upon the results from each profile (fig. 3).
In a more complex case we can consider an agent with several profiles:

\[ a = (a_1, a_2, \ldots, a_i, \ldots, a_n) \]  \hfill (13)

where \( a_i \) is defined:

\[ a_i = (M_i, Q_i, S_i, I, X, L) \]  \hfill (14)

The operations \( I, X, L \) are common for all profiles, the configurations \( M_i, Q_i, S_i \) are different and characteristic of each profile. Each profile (eg \( i \)) has defined its own strategy configuration \( S_i \) but the agent must realize only one strategy \( s \) belonging to the set \( S \). This common configuration of strategies \( S \) is defined:

\[ S \subset S_1 \times S_2 \times \ldots \times S_n, \quad s = (s_1, s_2, \ldots, s_i, \ldots, s_n) \quad \forall 1 \leq i \leq n \quad s_i(m_i) = m'_i \]  \hfill (15)

The relation between the set of common goal \( Q \) and the goals in profiles \( Q_i \forall 1 \leq i \leq n \) is defined as follows:

\[ Q \subset Q_1 \times Q_2 \times \ldots \times Q_n, \quad q = (q_1, q_2, \ldots, q_i, \ldots, q_n) \quad \forall 1 \leq i \leq n \quad q_i(m_i, m'_i) \in \mathbb{R} \]  \hfill (16)

where \( \mathbb{R} \) - real numbers.

The observation function \( I \) of the agent is defined as:

\[ I : M_1 \times \ldots \times M_i \times \ldots \times M_n \times \mathcal{V} \rightarrow M_1 \times \ldots \times M_i \times \ldots \times M_n \]  \hfill (17)
\[(m_1, m_2, \ldots m_i, \ldots m_n) = I(M_1, M_2, \ldots, M_i, \ldots, M_n, V), \quad \forall 1 \leq i \leq n \quad m_i \in M_i \] (18)

The agent in every profile calculates the evaluation of a possible strategy \( s = (s_1, s_2, \ldots s_i, \ldots s_n) \) by the value \( DI_i(m_i, s) \).

Final decision undertaken by the agent to realize the strategy \( s \) is carried out using the decision function:

\[U(DI_1, DI_2, \ldots DI_n, agent - state)\] (19)

using an expression:

\[U^* = \max_{s \in S} U(DI_1, DI_2, \ldots DI_n, AS) \rightarrow s^* \quad S \subset S_1 \times S_2 \times \ldots \times S_n \] (20)

where \( AS \) - agent-state describes a general vital state of the agent and may be defined: \( AS = \{m_1, m_2, \ldots m_i, \ldots m_n\} \) (\( m_i \) - just created model in the profile \( a_i \) of the environment observed).

The selected strategy \( s \) is realized by the agent in the environment:

\[X : S_1 \times \ldots \times S_i \times \ldots \times S_n \times V \rightarrow V \] (21)

\[V' = X(s, V) = X((s_1, s_2, \ldots s_i, \ldots s_n), V), \quad \forall 1 \leq i \leq n \quad s_i \in S_i, \quad V, V' \in V \] (22)

6 Planning and Negotiating

6.1 planning

In the primary learning, sets \( S \) and \( M \) are not suitable for the agent \( a \) tasks. It has to enrich them. For this purpose it uses planning operation. Plan \( PL \) is defined as a sequence of strategies:

\[PL = \langle s_1, s_2, \ldots s_i, \ldots s_n \rangle \quad \forall 1 \leq i \leq n \quad s_i \in S \] (23)

Evaluation of the plan \( PL \) is realized as a composition of the strategies:

\[m_f = (s_n \circ s_{n-1} \circ \ldots \circ s_1)(m_b)\] (24)

where \( m_b \) - the issue model of the environment in its beginning state, \( m_f \) - forecasted model of the final state of the environment after an application of the adopted plan.
Using a given plan \( <s_1, s_2, ..., s_n> \), the agent can realize it in the environment:

\[
X(V, <s_1, s_2, ..., s_n>) = X(X(V, s_1), <s_2, ..., s_n>) = V_f
\]  

(25)

where \( V_f \) is the final state of the environment.

Using the observation function \( I \) the agent may build a model \( (m'_f) \) of the final environment \( V_f \) in its mind \( (m'_f = I(M, V_f)) \). If the discrepancy between a forecasted model \( m_f \) and an observed model \( m'_f \) is not considerable a new strategy \( s(m) = m'_f \) is created. It may be considered as an extension to the learning function of the agent.

The planning operation may be enriched with the milestones control system. In this case the agent observes the results of the plan not only as the final state of it in the environment.

We can consider that the plan i.e. sequence \( <s_1, ..., s_i, ..., s_n> \) is provided with inserted milestones. The role of the milestone is played by marked strategies (for instance \( i \)).

After execution of this "milestone strategies", the agent observes the environment and builds its models \( m'_i \). Then it compares the forecasted model \( m_i \) with the one obtained from the observation \( (m'_i) \). The discrepancy between these models serves to decide whether the plan is continued or stopped (i.e. modified or abandoned).

### 6.2 negotiating

Using M-agent architecture we may analyze a negotiating process from the agent’s point of view. We can distinguish between simple negotiations and more complex - planning negotiations.

In simple negotiations (fig. 4) one agent - the agent "b" would like to perform its own strategy \( s^b \) that is convenient for him (taking the goal \( g^b \) into consideration). It needs the agent’s "a" agreement for that.

- For this purpose the agent "b", sending appropriate information, presents the state of the current environment (as it was observed by him - \( m^b \)) and the forecasted model of the result environment \( (m'^b) \) to agent "a".

12
The agent "a" using acquired information, builds in its mind the models \(m^b_a\) and \(m^{b'}_a\) which are its own version of models \(m^b\) and \(m^{b'}\) of the agent "b".

- The agent "a" evaluates the correctness of the proposed issue model \(m^b_a\) and the changes in the environment using the forecasted model \((m^b)^*_a\).
  
  For this purpose it uses its own goal \(q^a\).

- If the changes are suitable for the goal of the agent "a" it agrees to the actions of the agent "b", if not the agent "a" may prepare a new proposal for the agent "b" in the same way.

- An agreement closing negotiations is achieved when both agents accept the changes in the environment convenient for their goals.

Agents may create a more complex solution using planning negotiations. The planning negotiations may be carried out in the following steps (fig. 5):

- agent "b" adopts strategy \(s^b\) that is convenient for the realization of its goal and sends information about it to agent "a".

- agent "a" considering the proposed strategy \(s^b\) tries to find its own strategy \(s^a\), which performed with the proposed strategy, is convenient from its goal \(q^a\) realization and informs the agent "b" about it.
agent "b" evaluates its own strategy $s^b$ and the proposed $s^a$ using its own goal $q^b$. If the evaluation is successful the negotiation are completed with a construction of a common plan $< s^b, s^a >$. If not, a dialogue is continued to find a new solution by means of new proposals given by the agents.

If the execution of both strategies (in a given order) is useful for the agents "a" and "b", the strategy is memorized by the agents as a common new strategy realized in cooperation and the cooperation may continue.

7 Conclusion

The model of the agent based on the concept of the M-agent architecture presented in the paper has the following advantages:

- Possibility to define multi-agent system independently of programming language to be applied.

- Description of the multi-agent system is general enough to give a free choice of particular formalism to describe different parts of the system, and then the way of its practical realization.
The approach proposed of the M-agent architecture enables a distinction between an agent approach to systems and a non-agent one. It gives the possibility of keeping a track upon realization if the designed system is really an agent system.

General methods of agent functioning in the environment defined in the proposed model make the whole agent algorithm specification simple to be mastered by designers and programmers.

The approach proposed may be considered as a continuation of general ideas of Object Oriented Analysis and Design.

The architecture proposed of the M-agent makes possible to create a knowledge representation method for the multi-agent system specification. It may be a useful approach to the technology of MAS creation - the Agent Oriented Analysis and Design processes.

References


