

# Metacognition in Complex Agent

N. K. Kamila<sup>1,\*</sup>, R. K. Swain<sup>2</sup>, Manas Ranjan Mohapatra<sup>1</sup>

<sup>1</sup>Department of Computer Science and Engineering, C V Raman College of Engineering, Bhubaneswar, India

<sup>2</sup>Department of Computer Science and Engineering, Bhubaneswar Institute of Technology, Bhubaneswar, India

---

**Abstract** This paper focuses role based architecture for complex agent and the idea towards the developmental approach of model for metacognition in complex agents with the principles of cognitive architecture. This model explains the mechanism of metacognition in a complex agent. However from architecture it is observed that complex agent under metacognition principle works well with less time in an environment. Refinement of the decisions is being made by the complex agent under metacognition environment. From the graph it is observed that the correctness and accuracy of decision gets increased in case of complex agent which is inevitable in nature.

**Keywords** Complex Agent, Metacognition, Architecture, Cognitive Task, Decision, Etc

---

## 1. Introduction

This paper will discuss role based architecture for complex agent systems and metacognition in complex agent. From the study of intelligent agents it presents that the agent architectures support the agent that are incapable of doing more than one cognitive task at a time like human being, in which different thought arises at a particular time, and also itself capable of thinking all of which that depends on its mind and putting the action/s to environment. The metacognition mechanism controls this aspect of complex agent.

So designing a model for metacognition in complex agent with this ability and role based architecture for complex agent is the current problem to the research on metacognition in artificial intelligence.

While metacognition has many definitions under the “thinking about thinking” umbrella, the interpretation of considering metacognition as the thinking, monitoring and regulation of cognitive tasks/processes. In computational agents that focus on decision making, this manifests itself in choosing between multiple reasoning strategies and determining the amount of resources (computation, memory, bandwidth, etc.) to use in pursuing each strategy. These choices can be interpreted as an agent choosing one or more modes of operation.

This mode is often based on environmental conditions. Internal environmental conditions could capture properties such as the hardware capabilities of the agent. This is particularly important for mobile agents who may wish to modify their processes based on the computation, memory

and communication available to them. External environmental conditions could include characteristics of the problem instantiation being faced such as the volatility/stability of the domain or the role required to be performed by the agent.

A single agent can monitor its environment and choose a decision process dynamically based on a metacognition strategy that was obtained *a priori*. One path to developing such a strategy is by learning optimal choices over multiple environments. This procedure is insufficient in a multi-agent team because system performance does not depend solely on environmental factors. Agents’ decisions are made based on the information available. In multi-agent systems, the information available and how often it is updated depends on the decision processes chosen by other agents. Thus the mode of operation of one agent chosen by its metacognition strategy affects and is affected by the modes of operation chosen by other agents in the team.

In multi-agent systems, the performance is determined by a *joint* set of modes. The metacognition goal should be to find a way to choose this joint set of modes such that the resulting behavior is good. This choice may be the result of a decentralized strategy that yields a consensus on a common mode or heterogeneous distribution of modes such that any penalties for heterogeneity are offset by its gains.

In this paper the concept and definition of complex agent, role based architecture for complex agent and the idea towards developmental approach of model for metacognition in complex agent with the principles of cognitive architectures, model of metacognition for single and multi agent described in the section and using principles of metacognition have been presented. This model explains the mechanism of metacognition in a complex agent. The developed algorithms have been described to explain the architecture for complex agent and its metacognition.

The remainder of the paper is now organized as follows: in

---

\* Corresponding author:

nkamila@yahoo.com (N. K. Kamila)

Published online at <http://journal.sapub.org/ajis>

Copyright © 2013 Scientific & Academic Publishing. All Rights Reserved

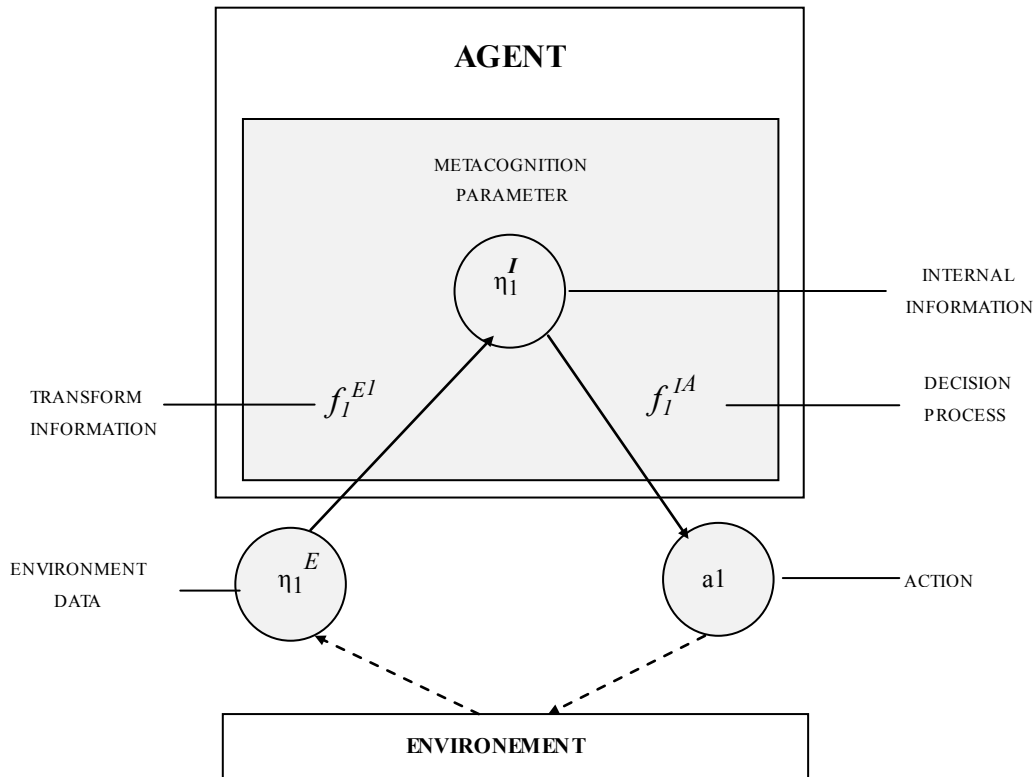
section 2 literature review has been illustrated to know about the research on this area. However, section 3 gives a short overview of complex agent and its functionality. In section 4 the examples of complex agent have been explained. Thereafter we present the role of complex agent in section 5 and followed by role based task execution through examples with different roles and the corresponding algorithm. In section 6 model for metacognition in complex agent has been discussed, followed by model description and corresponding algorithm whereas section 7 describes the empirical evaluation and results of the methods. However, section 8 summarizes the paper followed by references.

## 2. Literature Review

In this section we survey the main ideas on agent's performance under different roles holds by agent. In 2012, Corchado et al.[1] have presented a novel architecture which facilitates the integration of multi-agent systems, distributed services and applications to optimize the construction of ambient intelligence environments. Their architecture proposes a new and easier method to develop distributed intelligent ubiquitous systems, where applications and services can communicate in a distributed way with intelligent agents, even from mobile devices, independent of location restrictions. However, Kim et. al.[2] proposed a multi-agent system for autonomous microgrid operation. In order to build multi-agent system they have designed the functionalities of agents, interactions among agents and an effective agent protocol using an ADIPS/DASH framework.

Similarly, Kianifar[3] has used the intelligent agent in wireless sensor network for optimizing power consumption. But in 2008, Karsten et. al.[4] have analyzed and managed role-based access control policies using object constraint language and first order linear temporal logic because the role is assigned to user to perform different activities in all organizations. They have attempted to formally verify role-based access control policies with the help of a theorem prover and to validate policies with the UML based specification environment system, a validation tool for OCL constraints. Woo et. al.[5] have proposed a new access control approach for multi-agent system based on the role based access control model in which amount of context information collected from various sensors have been managed and accessed by authorized person. To effectively reflect the continuously changing environment and user's role, they have also proposed dynamic RBAC scheme that employs the notion of trust evaluated with the measure of satisfaction and reputation. It also updates the user's role and permission according to the context information. The effectiveness of the proposed approach has been verified by simulation study. Noor et. al.[6] have discussed the architecture for intelligent multi-agents paradigm in decision support system. But much attention has not given on complex agent under metacognition environment and its architecture to perform different roles based on the requirement and on its implementation.

## 3. Complex Agent



**Figure 1.** Metacognition Model and parameters for a Single Agent

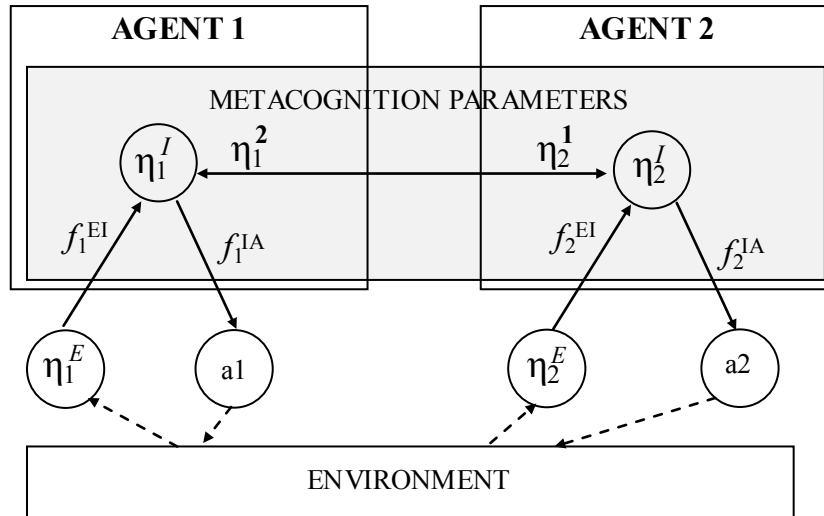


Figure 2. Metacognition Model and parameters for a Multi-Agent System

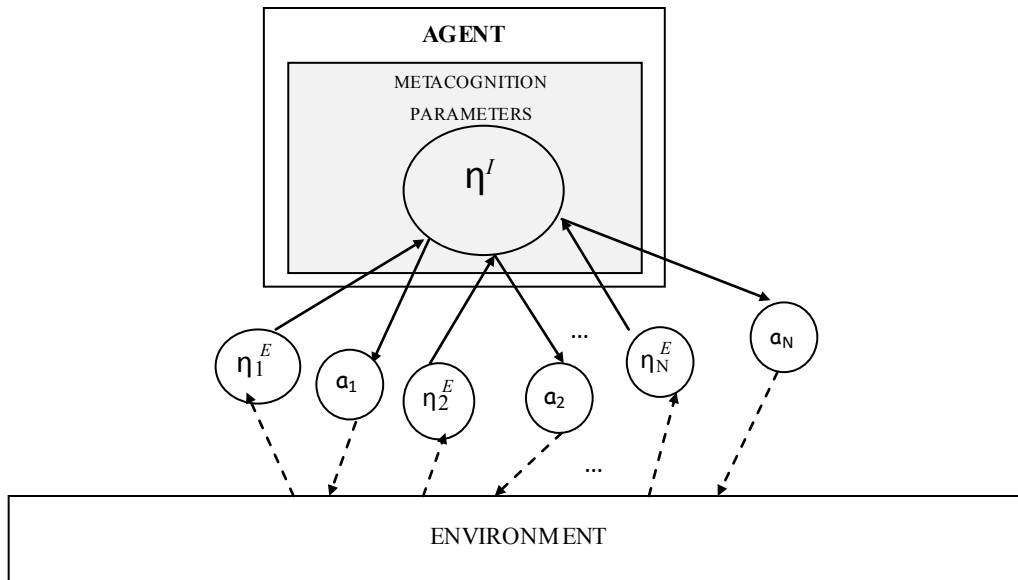


Figure 3. Metacognition Model and Parameters for a Complex Agent System

A single agent can monitor its environment and choose a decision process dynamically based on a metacognition strategy. In other words a single agent gets information from the environment ( $\eta^E$ ) which is transformed ( $f^{EI}$ ) into some internal information ( $\eta^I$ ). This internal information serves as the base input into a decision process ( $f^{IA}$ ) by which an action ( $a$ ) is chosen. The agent has a choice to determine the transformation of environmental information, the internal information and decision process from a larger set of possibilities:  $\eta^E \in H^E$ ,  $f^{EI} \in F^{EI}$ ,  $f^{IA} \in F^{IA}$ . Hence metacognition for a single agent is represented as choosing an element of  $M_1 := H^E \times F^{EI} \times F^{IA}$  based on information from the environment. Thus the resulting element is the mode of operation that an agent chooses which has been shown in figure 1.

Figure 2 reveals the multi-agent system. In multi-agent system, the agent performs multiple (two) tasks. In other words the internal information from an agent ( $\eta^I$ ) is also a function of the information that it receives from other agents

( $\eta^j, j \neq i$ ). This represents the problem in a new way. If agents come up with metacognition strategies (methods of picking modes) individually, then must consider the information from other agents ( $\{\eta^j\} j \neq i$ ) as additional environmental input. This would lead to uncoordinated metacognition strategies in multi-agent system. A better way would be to choose an element from  $M_1 \times M_2 \times H^I \times H^I$  that optimizes the team behavior based on joint environment ( $\eta^E, \eta^E$ ).

However, a complex agent is simply defined as “an intelligent agent capable of doing more than one (several) task/s simultaneously in an environment”. A multi-agent is a special case of complex agent. It perceives information from the environment with sensors, store in memory, find out the number of cognitive tasks/even same task of having different informations and the possible combination of tasks for execution for certain interval of time, pick out/select the particular combination for execution in an interval of time, and finally outputs the required action/s with actuators ( $a_1, a_2, \dots, a_n$ ) to the environment. It simultaneously correlates with

the mechanism of metacognition for these functions.

In other words, in a complex agent system, the internal information from an agent ( $\eta^I$ ) is also a function of different information that it receives from the environment. Here the agent is same throughout the execution period. But different activities/multiple tasks/same activity using different information are performed/executed. If the agent comes up with metacognition strategies (methods of picking modes), then it must consider different information from environment as additional input. This would again lead to uncoordinated metacognition strategies in the complex agent system. A better way would be to choose an element from  $M_1 \times M_2 \times \dots \times M_n \times H^1 \times H^2 \times \dots \times H^I$  that optimizes the team behavior based on the joint environment ( $\eta^{E_1}, \eta^{E_2}, \dots, \eta^{E_i}$ ). Really the challenge would be to achieve this joint strategy in a distributed manner. The entire description of the complex agent is depicted in figure 3.

## 4. Examples of Complex Agent

Human being itself is a real/natural example of complex agent. Human perceives different inputs from environment through sensory organs at a time such as eyes, ears and other parts of the body and simultaneously thinking all of them or that depends on status of the mind, then execute and performs required action/s to environment through actuators such as hands, mouth, legs and other body parts. These different thinking skills, execution, perception and actions of mind are controlled by metacognition mechanism.

Consider a situation where an agent is trying to coordinate its' activities over an extended period of time. The agent can be in one of the modes of operation such as *fast planning mode* or *long planning mode*. In the *fast planning mode* the agent has determined that either the environment is too volatile for extended analysis or it does not have the capability to do complex deliberation. Thus it sends small messages very often about its status and immediate plans. In *long planning mode*, an agent has determined that the environment and its capabilities can support deeper reasoning or it may only have periodic access to communication facilities. Under this condition the agent considers all the possible state evolutions and contingent actions over a longer time horizon. It sends larger messages with details of this information but not as often.

If the agent chooses its metacognition strategy based solely on environmental conditions, it may end up in different modes. Then most of the fast-planning agent will be ignored or provide insufficient information for the long-planning agent. The messages of the long-planning agent will be too cumbersome and infrequent for the fast-planning agent in multi-agent system.

As complex agent is being considered in metacognition activities, the agent itself gets information from the environment through different (n number of) ways. Thus the internal information of the complex agent gets updated.

## 5. Role of Complex Agent

Our versatile world can be outlined with three major aspects such as "A society is formulated with roles", "Roles are organized in structures" and "People play roles"[7],[8].

This section describes ideas of different roles of complex agent and its' corresponding role based architecture. Since a complex agent performs more than one task simultaneously, it can play different roles in an environment to perform these tasks.

Consider the complex agent that can play five roles such as  $R_1, R_2, R_3, R_4$  and  $R_5$  based upon which it performs more than one task at a time. Roles played by the complex agent mainly fall into two categories:

- Primary role ( $R_{PR}$ )
- Secondary role ( $R_{SR}$ )

$R_A$  = Active Roles to the Agent =  $\{R_1, R_2, R_3, R_4, R_5\}$   
 $R_{PR} = \{R_1\}, R_{SR} = \{R_2, R_3, R_4, R_5\}$

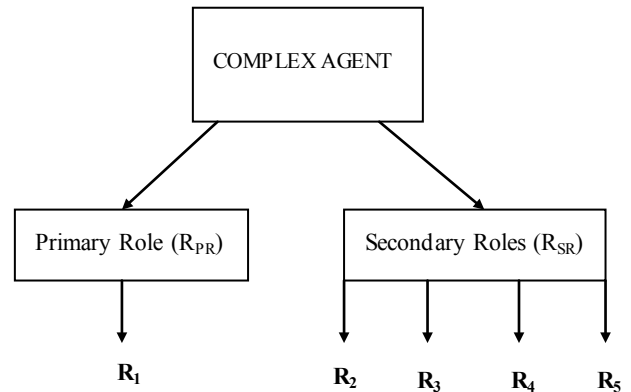


Figure 4. Type of roles played by complex agent

Figure 4 reveals the types of role played by the complex agent. The agent always has a single primary role that it should play under each instance.

### 5.1. Explanation of Role Based Task Execution through Example

Roles can be used to map the people's natural organizations, task distributions, system analysis, system design and system construction[9]. Further every agent is responsible for accomplishing a certain task. An agent can be considered as a self-contained object of some class which does not belong to the problem domain although it does involve itself with that domain or environment. To develop a role based system, the main tasks are specifying roles and the relationships among roles, specifying role players and assign roles to role players.

To understand role-based task execution, consider a person (complex agent) plays a CEO role of any IT organization again after playing a project manager role, should we create a new role instance or replay the original role instance? This is up to the actual requirement. If it is really a different one, we need to create a new role instance. If he/she really plays the same CEO role instance as before,

he/she should replay the original role instance. But it is the same complex agent which performs the tasks of different role instance. Consider the possible roles the agent that can play as follows:

Primary Role ( $R_{PR}$ )  $\rightarrow$  CEO ( $R_1$ )

Secondary Roles ( $R_{SR}$ )  $\rightarrow$  Project Manager ( $R_2$ ), Project Leader ( $R_3$ ), Software Developer ( $R_4$ ), Guest Faculty ( $R_5$ )

Complex agent having primary role CEO ( $R_1$ ) of any IT organization also performs the tasks of project manager by playing the role of a Project Manager ( $R_2$ ) i.e. secondary role, also at the same instance he can play the role of a Software developer ( $R_4$ ) being a secondary role if persons concerned is absent. Also it can perform the tasks of Project Leader ( $R_3$ ) and Guest Faculty ( $R_5$ ) simultaneously. This is the property of a complex agent.

That is, performing task in playing the primary role as a CEO the agent itself is able to play the tasks of two different roles that are task of project manager and task of a software developer. Thus complex agent performs simultaneous tasks of primary role and secondary roles with the position of primary role. This is described in the table - 1. Hence complex agent is said to be role based i.e. Role based complex agent.

## 5.2. Past Role, Active Role, Present Role and Future Role

To express an agent's dynamic and evolving aspects, we cannot avoid time. To completely express a live agent, we need to express the agent's past, present, active and future roles.

The dynamics of agents are based on their adaptabilities. Agents are required to adapt five aspects of an intelligent system: perception strategy, control mode, reasoning tasks, reasoning methods, and meta-control Strategy. To describe the dynamics of agents in an intelligent system and help agents' adaptability in a system, roles are a wonderful tool. Previous Roles, Active Roles, Present Roles and Future

Roles are enough and qualified to express the dynamics of agents.

### Past roles

The previous role of an agent is called past roles. To track an agent's status, a record of history is required. A history record helps answering such a question: What roles did an agent play in the past?

Suppose a person plays a developer role again after playing a CEO role, should we create a new role instance or replay the original role instance? This is up to the actual requirement. If it is really a different one, say, a senior developer, we need to create a new role instance. If he/she really plays the same developer role instance as before, he/she should replay the original role instance.

### Active Roles and Present Roles

An active role means the player responds to the messages relevant to these roles. An active role should be transferred to the present role to respond to the messages. However, present role means the agent is currently playing this role, i.e., it directly responds to the messages relevant to this role. These concepts are used to express the present state of an agent. They are used to answer such questions: What are its active roles? What is its present role?

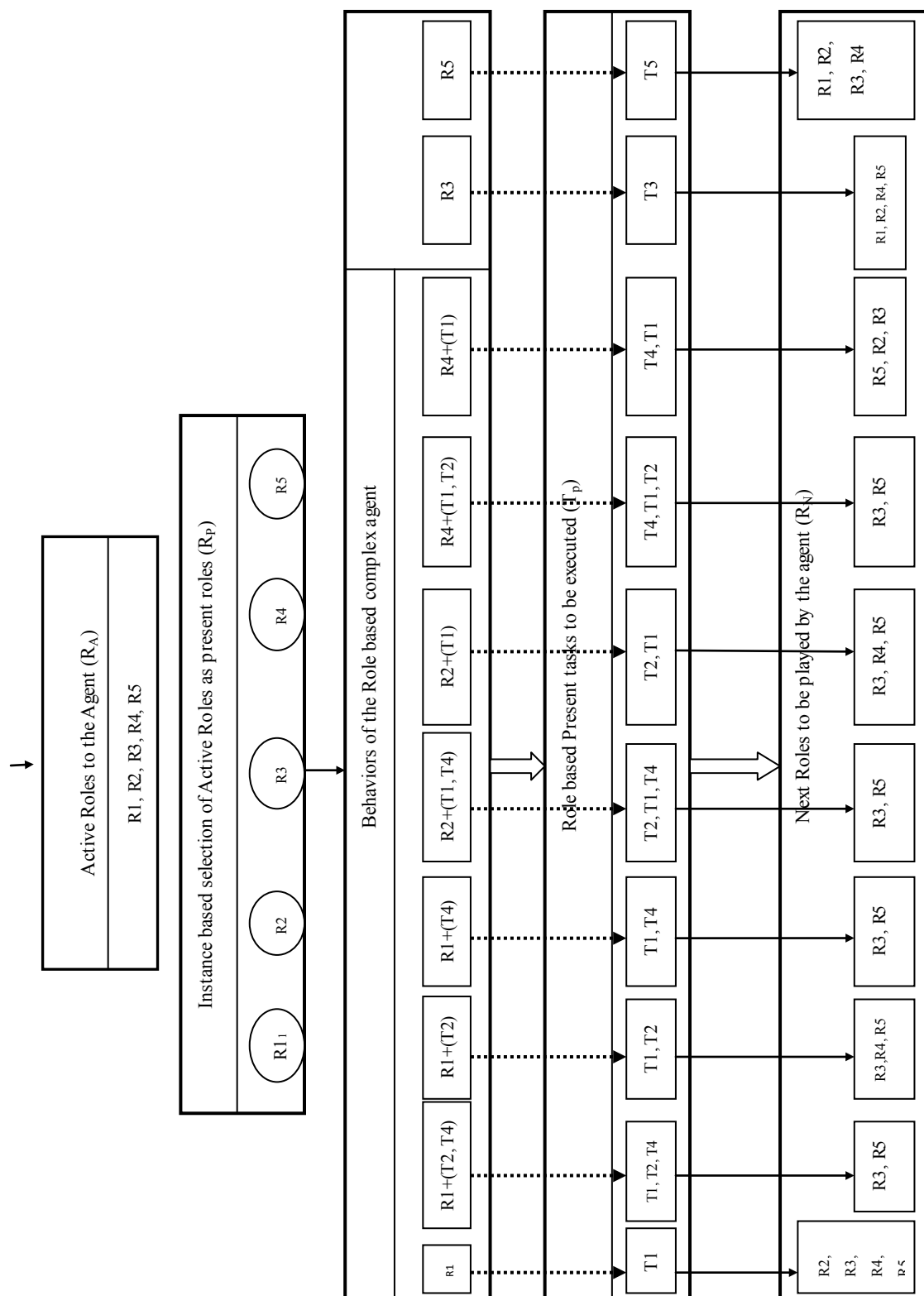
Active roles are the roles a player is holding and they are ready to respond to messages. The present role is the role that is directly responding to messages. That is to say, a role player can hold many active roles at the same time and it only holds one present role at a time but can perform the tasks of other roles. Active roles can also be used to express the meaningfulness of role transition, i.e., changing the present role from one active role to another.

### Future Role

Future role means the roles an agent hopes to play in the future is referred as its future role. In role-based agent system, roles can be taken as goals for an agent.

**Table 1.** Description of Different Roles

Past Roles	Active Roles( $R_A$ )	Present Roles( $R_P$ )	Present Tasks( $T_P$ )	Future Roles( $R_N$ )
R1.R2.R3, R4.R5	R1.R2.R3, R4.R5	R1	T1	R2,R3, R4,R5
R1.R2.R3, R4.R5	R1.R2.R3, R4.R5	R1+(Task of R2,R4	(T1,T2, T4)	R3,R5
R1.R2.R3, R4.R5	R1.R2.R3, R4.R5	R1+ tasks of R2	(T1,T2)	R3, R4,R5
R1.R2.R3, R4.R5	R1.R2.R3, R4.R5	R1+ tasks of R4	(T1,T4)	R3,R2,R5
R1.R2.R3, R4.R5	R1.R2.R3, R4.R5	R2+ tasks of R1,R4	T2,T1,T4	R3,R5
R1.R2.R3, R4.R5	R1.R2.R3, R4.R5	R2+ tasks of R1	(T2,T1)	R3, R4,R5
R1.R2.R3, R4.R5	R1.R2.R3, R4.R5	R4+ tasks of R1,R2	T4,T1,T2	R3,R5
R1.R2.R3, R4.R5	R1.R2.R3, R4.R5	R4+ tasks of R1	(T4,T1)	R3, R2,R5
R1.R2.R3, R4.R5	R1.R2.R3, R4.R5	R3	T3	R1,R2,R4,R5
R1.R2.R3, R4.R5	R1.R2.R3, R4.R5	R5	T5	R1,R2,R3,R4



**Figure 5.** Role Based Architecture for Complex Agent

From table 1, it is observed that ordering of future roles after doing the present roles depends on the agent. For example in the 1<sup>st</sup> row it may be  $R_2 \rightarrow R_5$  or  $R_5 \rightarrow R_2$ . For example in the 1<sup>st</sup> row it may be  $R_2 \rightarrow R_5$  or  $R_5 \rightarrow R_2$ . The columns of Present Roles and Present tasks from the 2<sup>nd</sup> row to 8<sup>th</sup> row of the table show the behavior of the role based complex agent. Figure 5 shows the role based architecture of complex agent.

### 5.3. Programming Logic

1. while  $R_A \leftarrow$  Active roles to the agent.
2.  $R_{PR} \leftarrow$  primary roles,  $R_{SR} \leftarrow$  secondary role.
3. select  $R_p$  out of  $R_A$  for an instance.  
where  $R_p =$  Present role.
4. find\_out ( $T_p$ ).  $T_p =$  role based task to be executed.
- Case-1:** 4.1. if  $R_{SR} \neq \text{NULL}$ .
  - 4.1.1  $R_p = R_{PR}$ .
  - 4.1.2  $T_p = T_{PR}$
- Case-2:** 4.2. if  $R_{SR} = \text{NULL}$ .
  - 4.2.1.  $R_{PR} \leftarrow T_{SR} + (T_{PR})$ .
  - 4.2.2.  $T_p = (T_{PR} + T_{SR})$ .
- Case-3:** 4.3. if  $R_p = R_{SR}$ .
  - 4.3.1.  $R_{SR} \leftarrow T_{SR} + (T_{PR} + T_{SR})$  (if  $R_{SR} = \text{NULL}$ )).
  - 4.3.2.  $T_p = (T_{PR} + T_{SR})$ .
5. execute\_task ( $T_p$ ).
6. do Next Role ( $R_N$ ).

## 6. Model for Metacognition in Complex Agent

This model for metacognition in complex agent has been derived from the different cognitive architectures and using the principles of metacognition. The model is depicted in the following figure 6. The complete description is presented in the following section. However, metacognition can be used as a generic concept for computational theories with respect to problem solving, reasoning and the decision making. Metacognition can also be applied on simple to complex cognitive architectures. The applicability of metacognition on problem solving through computational models, educational field and human problem solving, etc plays vital role in technical arena. The problem solving is a one area where a natural mind, complex agent (robots, animals, humans) fits for the artificial computational theories in artificial intelligence. The executive control and monitoring are important divisions of problem solving to manage problem complexity and to evaluate progress towards the goal. This architecture reveals how several informations are perceived from the environment and processed through different layers of metacognition to required actions. Also depicts the different functions of each layer in the architecture along with processing of many tasks.

### 6.1. Description of the Model

The above presented model for metacognition in complex

agent comprises of three main layers such as

- Reactive Layer
- Deliberation Layer
- Metacognition Layer

These layers support for different classes of emotions found in humans, animals and others. The above three layers consist of primary and secondary emotions. The function and importance of these layers can be explained in details during the description of the model

At first, the reactive layer reflexes to/from environment, thereby detects the objects in environment, executes, and then determines how to react. This layer interacts with the internal, external conditions. Then it produces internal or external state changes. The reactive system is very complex and powerful. This layer needs to store all the mechanisms of particular mind. Reactive layer includes a global alarm mechanism, which belongs to primary emotions.

With this mechanism, Complex agent percepts the correct required inputs from the environment, then store in the memory by initializing the memory. Information can be received from the environment through perception, and can be transferred to memory. The environment is connected to the different cognitions like perception, memory and other cognitive tasks such as image processing, problem solving, reasoning, decision making etc. These cognitive tasks are controlled by metacognition in Metacognition layer

The Metacognition layer supervises and controls the other layers of architecture, more efficiently. This layer can support and control the thoughts[10]. For example, human emotions such as infatuation, humiliation, thrill etc. The Metacognition layer further has three levels/stages which are:

- Metamemory
- Metacomprehension
- Metaregulation

The functions of these three levels in controlling the cognitive tasks are explained as follows:

Metacognition about memory, sometimes called metamemory, refers to the self-monitoring and self-control of one's own memory in the acquisition and retrieval of information.

For example, imagine a student who is studying for an examination that will occur tomorrow in French class, say on French-English vocabulary such as "chateau/castle" and "rouge/red". Let us keep that student in mind as we consider the monitoring and the control of the student's learning of the new vocabulary and his or her attempts to retrieve the answers during the test the next day.

Metamemory is a type of metacognition that focuses on one's memory and memory processes. This can include deciding whether one is confident that a memory one has retrieved is accurate. Metamemory can guide decisions. For example: When a student makes a decision about how to study material, decides how long to study it, or evaluates whether it has been committed to memory, he or she can be said to be engaging in metamemory. Metamemory is perhaps the most widely investigated category of Metacognition. It

- Identify memory strategies for different situations or content.

- Identify which particular strategies work for us.
- Predict future performance on a memory task.
- Estimate that retrieval is correct.

Metacognitive knowledge contains a database of knowing about an environment, the nature of the task, and strategies used for knowing the facts. Metacognitive knowledge contains three types of knowledge.

- (1) Declarative knowledge
- (2) Procedural knowledge and
- (3) Conditional knowledge

The declarative knowledge is the actual facts of the information. This contains the agent's or person's knowledge about formula, knowing the facts, places, etc.

The procedural knowledge is knowledge about execution of the given facts. For example, in solving a mathematical problem procedural knowledge is used to select which of the available formula in the declarative knowledge are appropriate to the problem.

The conditional knowledge is knowledge about particular skill and strategy used for conditions.

However, metacomprehension is used for detecting and rectifying the errors. This helps to improve the performance.

- Monitor our ability to comprehend instructions, explanations.
- Identify when we do not understand something.
- Know how to select and employ strategies to correct our misunderstandings when these occur.

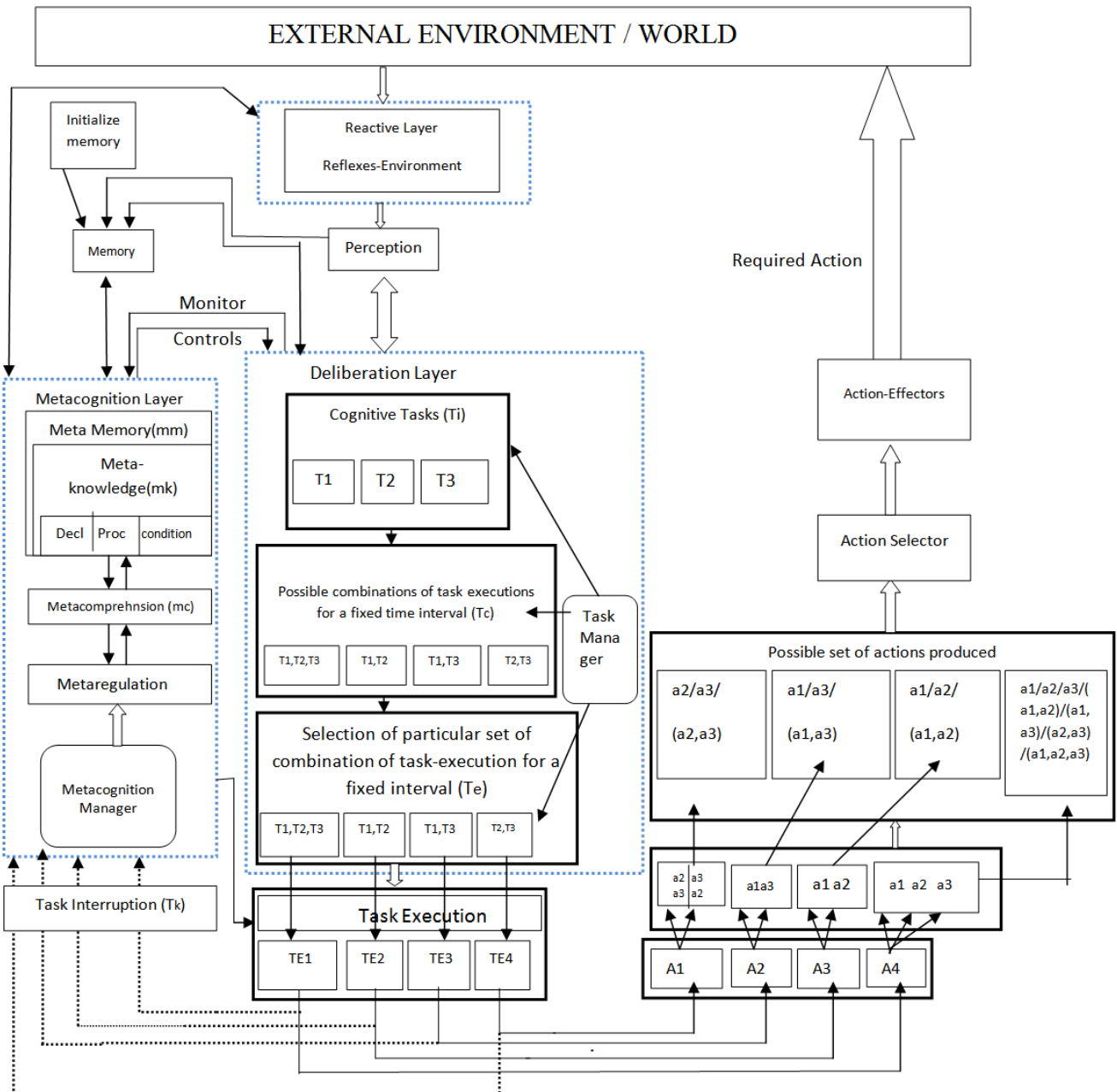


Figure 6. Architecture for Metacognition in Complex Agent



### Metaregulation

Controlling and monitoring a progress of cognitive task is termed as metacognitive regulation or metaregulation. This works for rectifying the errors in cognitive tasks, and thoughts are adjusted by giving feedback.

Various Metaregulation skills are:

- Use feedback to adjust the strategies
- Select the right strategy, apply it, & monitor its usefulness
- Need to know:
  - Which strategies are available?
  - How the strategies work?
  - When the strategies are useful?

The deliberation layer supports for secondary emotions. Secondary emotions are semantically rich emotions. This layer is responsible for perception, finding out no. of tasks, planning, finding possible set of combinations of tasks, evaluation, decision making (selection of task), allocation of resources and time for cognitive tasks. This layer can learn the generalizations and pass to the other layers.

In this layer the first function is carried out, that is finding out the number of cognitive tasks to be performed. In the above model it has taken three no's of cognitive tasks such as  $T_1$ ,  $T_2$  and  $T_3$ .

After finding out number of tasks the possible no. of combinations of tasks to be performed is to be found which are denoted by symbol ( $T_c$ ). The possible combinations are  $[(T_1, T_2, T_3), (T_1, T_2), (T_1, T_3), (T_2, T_3)]$ . Then the particular set of combination of task out of possible combination is to be selected and a fixed time interval, along with the resources required are allocated to selected set of task combination for task execution. Let it be denoted as symbol ( $T_e$ ). Here these are  $TE_1, TE_2, TE_3, TE_4$ . All these described processes are performed by a Task Manager in this deliberation layer by using the percept inputs from environment stored in memory and being controlled by the Metacognition process via Metacognition manager in the Metacognition layer. How it works is described as follows.

Cognitive processes can be split into an object-level and a meta-level. The meta-level contains information for controlling the object level. The information flow from object level to meta-level is referred to as the monitoring processes. Similarly, flow of information from meta-level to object-level is referred to as control processes.

As depicted in Figure 7, meta-level or metacognition includes the selection of tasks, allocation of time, termination of tasks, selection of task for execution and memory details etc. Metacognition can be constructed based on problem solving (e.g. planning) and metacomprehension (understanding) processes of the object level [11, 12]. "Meta-level is widely using in the reflective programming. Self-monitoring is a meta-level component and it controls and monitors the object level. This also changes its behavior if necessary. Meta-level is also called as self monitoring layer. This layer inspects and modifies the self-monitoring meta-level. If M is a meta-level and P is its object-level then

relationship as follows: (1) M is a program that interprets P, and it takes P as an argument; (2) P calls M to monitor, inspect, modify, correct or improve the P. This is called as reflective [13].

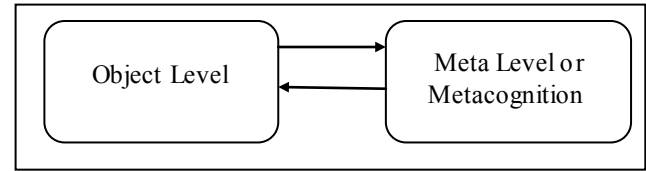


Figure 7. Cognition to Metacognition

After selecting combination of task executions ( $TE_1/TE_2/TE_3/TE_4$ ), the corresponding action sets ( $A_e$ ) i.e. ( $A_1/A_2/A_3/A_4$ ) are produced which will results their no. of actions ( $a_e$ ) corresponds to each task ( $T_e$ ) from which possible combinations of actions are to be found (Figure 6). Then finally, required/requested actions are passed to the environment through action-effectors/actuators.

If any interruption occurs to either task or more than one task in the selected set or to the all tasks in the set during the task execution, the execution of corresponding task/s is/are stopped and it will be carried out to the Metacognition layer, handled by metacognition manager and there will be further execution with the same mechanism as described above and the resulting action/s to the task/s are produced.

### 6.2. Algorithm for Explaining Metacognition in Complex Agent

Function `complex_agent` (percepts inputs from environment) return action/s.

1. *complex\_agent* reflexes to/from environment with sensors.

2. Initialize memory.

2.1. Store (memory)  $\leftarrow$  perceived inputs from environment.

For each  $i$  repeat from step-3 to 8 /\* $i$ =no. of cognitive task\*/

3. Perform(cognitive tasks( $T_i$ )) using percept inputs stored in memory controlled and monitored via Metacognitive process  $M_i$  by metacognition manager ( $i=1, 2, 3 \dots$ ).

$M_i = (mm_i + mr_i + mc_i)$

$mm_i$  = metamemory used to control memory, corresponds to each cognitive task ( $T_i$ ).

$mc_i$  = metacomprehension correspond to each cognitive task ( $T_i$ ).

$mr_i$  = metaregulation correspond to each cognitive task ( $T_i$ ).

3.1. while  $M_i$  interprets  $T_i$ .

3.2.  $T_i$  calls  $M_i$  i.e.  $M_i(T_i)$ , /\*  $T_i$  is taken as argument to  $M_i$ .\*/

3.2.1. metamemory ( $mm_i$ ) control memory using metaknowledge ( $mk_i$ ) to perform cognitive task.  $mm_i(\text{metaknowledge}(mk_i))$

/\* $mk_i$ =metaknowledge correspond to each

- cognitive task ( $T_i$ )\*/  
**3.2.2.**  $mc_i$  identifies error to the  $T_i$ .  
**3.2.3.**  $mr_i$  regulates the process of cognitive task ( $T_i$ ).  
**4.** function task manager (take cognitive task ( $T_i$ ) from memory) return selected task combination for execution for a time interval.  
**4.1.** find\_out (no. of tasks ( $T_i$ ))  
**4.2.** find\_out (possible set of combination of task-execution ( $T_e$ )). /\* $T_e$  is the combination of  $T_i$ \*/  
**4.3.** select (single set of combination of task execution ( $T_e$ )). /\* $T_e \in T_c$ \*/  
**4.4.** return  $T_e$ .  
**4.5.** allocate execution-time interval to  $T_e$ .  
**4.6.** execute\_task ( $T_e$ ).  
**5.** put\_actions ( $A_e$ ).  
**6.** select\_action (take input  $A_e$ ) returns request/required actions.  
**6.1.** select ( $a_e$ ) out of ( $A_e$ ).  
**6.2.** environment  $\leftarrow$  return ( $a_e$ ).  
**7.** while interruption (( $T_k$ ) of ( $T_e$ )).  $T_k = T_e$  or  $T_k = \text{single}$  or More than one task out of  $T_e$   
**7.1.** stop\_execution ( $T_k$ ).  
**7.2.** metacognition manager  $\leftarrow T_k$ .  
**7.2.1.** repeat step-3.  
**7.3.** execute\_task ( $T_k$ ).  
**7.4.** go to step-5.

Step-1 to Step-2 explains the function of reactive layer, perception from environment and memory storage. Step-3 explains the function of metacognition layer. Step-4 explains the process of deliberation layer. Step-5 to Step-6 describes the action processes to the environment. Step-7 explains the case of interruption.

## 7. Empirical Evaluations and Results

To examine the empirical efficacy of the complex agent comparing with single and multi-agent, we evaluated its performance along with metacognitive effect. In other words from empirical and extensive studies, it is observed that complex agent performs either one or more tasks alone based on the roles simultaneously where multi agent performs either one or more than one tasks of roles together at a time in coordination with each other. But in single agent system the agent performs single task of a role at a time which is depicted in figure 8. That means complex agent plays the primary role alone or along with secondary roles as per the designed architecture where as multi agent (several agents together) plays primary or secondary roles coordinately. This says that complex agent can replace several multi agents. However single agent can play one role at a time. How the roles are being played and executed has been explained in the algorithm. For example: in an financial institution the manager can play its own role along with the roles of other employee as per actual requirement like a complex agent

without creating new role instance. In figure 8 horizontal axis stands for agents and vertical axis for roles.

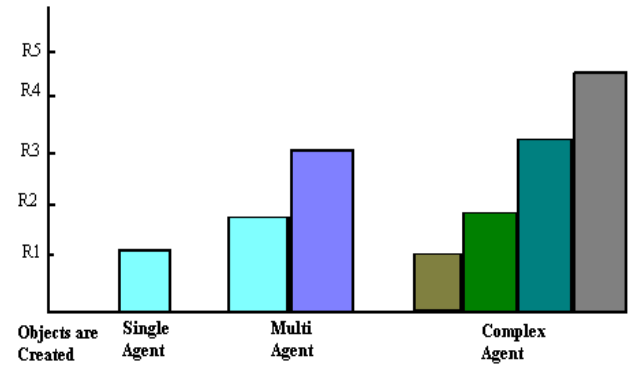


Figure 8. Agent with Roles

It has already been discussed in section 5, roles are played by agents. As time goes on roles may be added or discarded as per actual requirement. Figure 9 shows the role and the time segment for a single agent to play roles. From the figure it is seen that single agent play one role throughout.

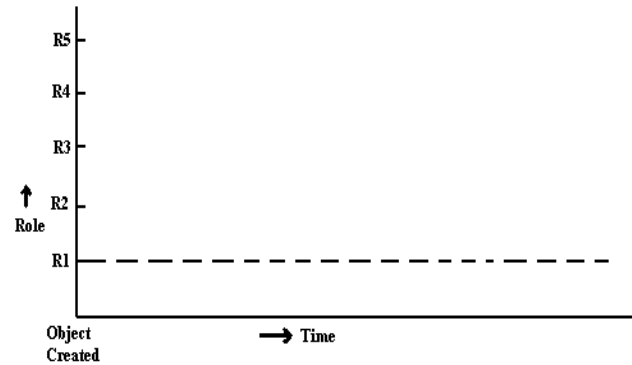


Figure 9. RolePlay Graph of Single Agent

However figure 10 depicts the role play graph of multi agent. Here several single agents are playing the role in coordinated manner. It is observed from the figure that multi agent performs the task of single role or more role but collaborative way. The decision for action is being taken by all agents constitute multi agent. But role is always with agent.

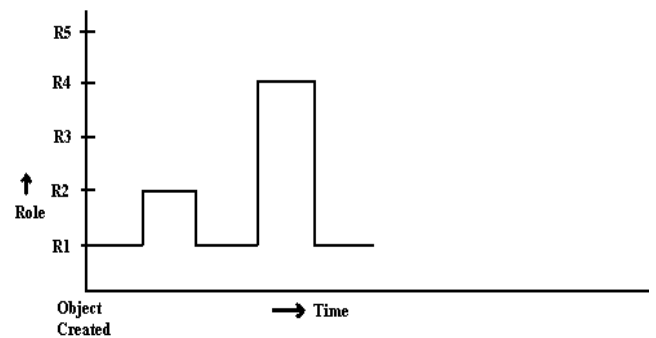


Figure 10. Role Play Graph of Multi Agent (Several Agents)

But figure 11 reveals the role play graph of complex agent. From the figure it is observed that complex agent has multi role always. When all secondary roles are discarded, it

comes back to its primary role. Again in course of time it posses several roles to play as per actual requirement. Hence it is concluded that complex agent replaces several multi agent as per the designed architecture.

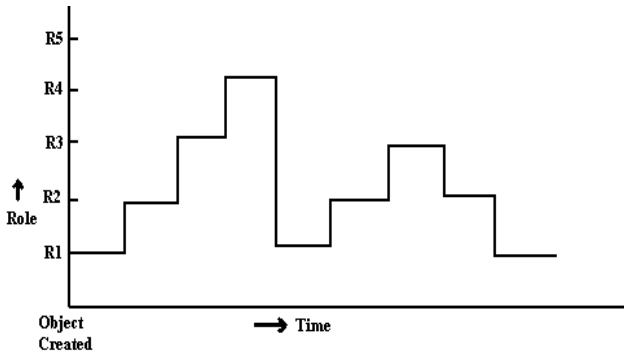


Figure 11. Role Play Graph of Complex Agent

Moreover to examine the efficiency of sequential approach to application of metacognition on complex agent, multi-agent and single agent, our research test bed was constructed to conduct experiments and to evaluate the performance of metacognition on above said agents. The idea behind it was to see the effect on actions to environment. The agent's basic task is to answer some intelligent question upon asking to rethink after answering once.

We began the experiments by constructing complex agent (executive officers, etc. who execute several roles with primary role), multi-agents (group of identical executives for one/more tasks) and single agents (that holds one role always) that solve the task in an extremely constrained manner. The tasks were intelligent questions to be answered based on presence of knowledge. Given the nature of tasks, we make the assumption that all agents have to perform the tasks by answering to environment.

When the agents were asked to perform different tasks/to answer different questions based on roles by leveraging knowledge, all agents individually answered the questions, but multi-agent answered in coordinated manner.

To examine the correctness and accuracy of the answer, next question was asked "Is the answer correct?". All agents reassessed the answer given by them for correctness. Subsequently they began applying metacognition strategies through different layers described in figure 6 specifically for complex agent. The correctness and accuracy of the obtained answer is much better than earlier answer in all the cases. In case of complex agent, accuracy gets increased up to more than 95%, but in case of multi-agent it is increased up to same level as earned by complex agent. However in case of single agent it gets improved, is depicted in figure 12. Hence it can be concluded that decision on answering questions gets refined followed by increased in accuracy after applying metacognition strategies.

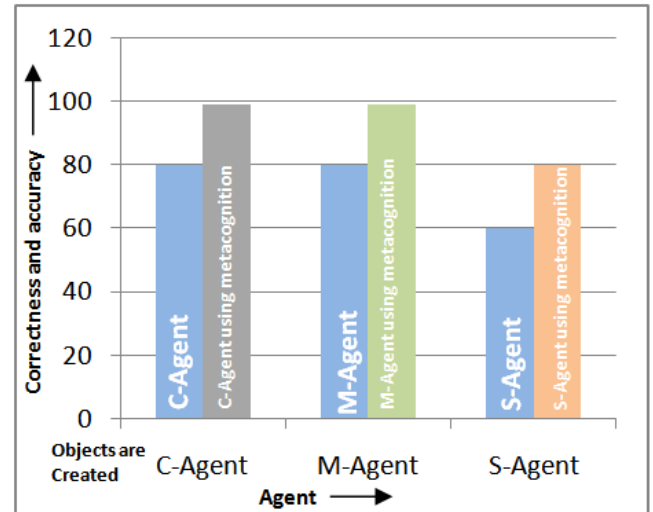


Figure 12. Performance of Agent

Where C-Agent: Complex Agent, M-Agent: Multi-Agent, S-Agent: Single Agent

## 8. Conclusions and Future Work

The paper described the concept of complex agent and discussed with examples. We also discussed how metacognition reasoning differs between a single and multi-agent system. The primary complexity is that the information an agent receives, in a multi agent system, depends on the information provided by other agents in the system. Thus the model of metacognition must address the coupling that exists between the choices of modes of operation. However in complex agent systems, the agent receives the different information's simultaneously from the environment and through the metacognition process the actions are performed to the environment. Thus the complex agent performs several tasks at a time and hence can replace many multi-agents. We have also discussed the role of complex agent while playing many roles with the help of corresponding architecture. Metacognition in complex agent has been discussed in details using its architecture and algorithm. It is observed from the architecture that complex agent under metacognition principle works well with less time in an environment. Graph shows that refinement of the decisions gets increased in complex agent using metacognition which increases the correctness and accuracy of the decision. In this paper we have considered only identical agents, later on different agents may also be taken into consideration as per the nature. The problem can also implemented by constructing authorization systems using UML and OCL.

## REFERENCES

- [1] Juan M. Corchado, Dante I. Tapia and Javier Bajo, "A multi-agent architecture for distributed services and applications", *International Journal of Innovative Computing, Information and Control*, Vol. 8, No. 4, April 2012.
- [2] Hak-Man Kim, Yujin Lim and Tetsuo Kinoshita, "An Intelligent Multiagent System for Autonomous Microgrid Operation", *Energies*, Vol. 5, pp.3347-3362, 2012.
- [3] Mohammadali Kianifar, "Using Intelligent Agents in Wireless Sensor Networks As Sinks for Optimizing Power Consumption", *Australian Journal of Basic and Applied Sciences*, Vol. 6, No. 8, pp. 36-42, 2012.
- [4] Karsten Sohr, Michael Drouineaud, Gail-Joon Ahn and Martin Gogolla, "Analyzing and Managing Role-Based Access Control Policies", *IEEE Transaction on Knowledge and Data Engineering*, Vol. 20, No. 7, pp.924-939, July 2008.
- [5] Jae Wook Woo, Myung Jin Hwang, Chun Gyeong Lee and Hee Yong Youn, "Dynamic Role-Based Access Control with Trust-Satisfaction and Reputation for Multi-Agent System", *Proceedings of 24<sup>th</sup> IEEE Conference on Advanced Information Networking and Applications*, pp.1121-1126, April 2010.
- [6] Noor Maizura Mohamad Noor and Rosmayati Mohamad, "New Architecture for Intelligent Multi-Agents Paradigm in Decision Support System", A book Chapter of Book entitled "Decision Support System", ISBN: 978-953-7619-64-0, January 2010.
- [7] Haibin Zhu and M. C. Zhou, "A Role-based collaboration and its kernel Mechanisms", *IEEE Transactions on Systems, Man and Cybernetics*, Part C, 2006.
- [8] Haibin Zhu, "Role Mechanisms in Collaborative Systems", *International Journal of Production Research*, vol. 41, no. 1, pp. 181-193, Jan 2006.
- [9] Haibin Zhu, "A Role-based Architecture for Intelligent Agent Systems", *Proceedings of IEEE Conference on Distributed Intelligent Systems: Collective Intelligence and Applications*, pp.354-362, June 2006.
- [10] Sloman A, "Varieties of Affect And The Cogaff Architecture Schema", University Of Birmingham. Available: <http://www.cs.bham.ac.uk/research/cogaff/>
- [11] Cox M T, "Metacognition, problem solving and aging", *Cognitive Science Tech. Rep.* No. 15, Atlanta, Georgia Institute of Technology, College of Computing, 1994.
- [12] Cox M T, "Metacognition in Computation: A selected research review", BBN Technologies, 10 Moulton St, Cambridge, 2005.
- [13] Kennedy C M, "Distributed Reflective Architectures for Anomaly Detection and Autonomous Recovery", PhD thesis, University of Birmingham, 2003.