

# A SOFTWARE AGENT FRAMEWORK TO OVERCOME MALICIOUS HOST THREATS AND UNCONTROLLED AGENT CLONES

Ms. G. Annie Sujitha<sup>1</sup> and Ms.T.Amudha<sup>2</sup>

<sup>1,2</sup>Department of Computer Applications , Bharathiar University,  
Coimbatore-641046, Tamilnadu, India

1annie.journals@gmail.com  
2amudha.swamynathan@gmail.com

## **ABSTRACT**

*An agent is anything that can be viewed as perceiving its environment through sensors and acting upon that environment through effectors. Enormous number of researches is going on by comparing the functional similarities of the Human Immune System for making the agents more adaptable in regard with security. In this research work, the functional similarities of Human Nervous system are given to the agents by proposing an agent-based framework where the agents can adapt themselves from one of the threats, the malicious host attack. The agents become aware of the malicious hosts' attack by learning and coordination is maintained by a Co-Agent to make this system work successfully. The concept of learning and coordination are taken from the Human Nervous system functionality. This system has shown a better functioning in maintaining the system performance by making the agents aware of malicious hosts and by producing limited number of clones.*

## **KEYWORDS**

*Nervous system, Learning, Coordination, Malicious Host, Agent Clones*

## **1. INTRODUCTION**

### **1.1 Intelligent Agents**

An agent is a piece of software that acts for a user or other program in a relationship of agency. Such "action on behalf of" implies the authority to decide which action is appropriate [63]. Intelligent agents may also learn or use knowledge to achieve their goals. They may be very simple or very complex: a reflex machine such as a thermostat is an intelligent agent, as is a human being, as is a community of human beings working together towards a goal [9]. The attributes of agents are autonomy, communication, coordination, responsiveness, pro-activeness, learning, goal oriented, reactivity, temporal, continuity, social ability, collaborative, inferential and personality [15, 5].

Coordination is the key functionality in agent systems. Agents need to coordinate their actions in order to reach their goals. Few approaches for agent coordination are Contract Net approach, Partial Global Planning (PGP) approach, Stochastic Coordination algorithm, Master Slave  
DOI : 10.5121/ijait.2012.2202

Coordination approach, Organisational approach, Domain-Independent coordination algorithm etc. [20, 35].

Learning ability is a crucial feature of intelligent agents. It is impossible to foresee all the potential situations an agent may encounter and specify agent behaviour optimally in advance. Agents therefore have to learn from, and adapt to, their environment. Learning research has been mostly independent of agent research and only recently has it received more attention in connection with agents and multi-agent systems. Few learning techniques are Logic-Based Learning, Reactive Learning, Q-Learning, Explanation-based Learning, Inductive Logic Programming, Reinforcement Learning, Supervised Learning, Unsupervised Learning, Hebbian Learning, Competitive Learning and Social Learning [21].

## 1.2 The Human Nervous System

The nervous system is the major controlling, regulatory, and communicating system in the body. It is the center of all mental activity including thought, learning, and memory. Every minute of every day, our nervous system is sending and receiving countless messages about what is happening inside and around our body. In short, our nervous system coordinates all the activities of our body.

The nervous system contains a network of specialized cells called neurons that coordinate the actions of human being and transmit signals between different parts of the body [65]. The primary function of the nervous system is Coordination. It does this by extracting information from the environment using sensory receptors, sending signals to the brain/ spinal cord, processing the information to determine an appropriate response, and sending output signals to muscles or glands to activate the response. All the messages which have been transferred are stored in the main organ- The Brain. The brain communicates with the rest of the body through the spinal cord and the nerves. The cerebral hemispheres form the largest part of the human brain and are situated above most other brain structures. The hemispheres are divided into two- The Right brain and the Left brain. The two cerebral hemispheres are connected by a very large nerve bundle called the corpus callosum [67].

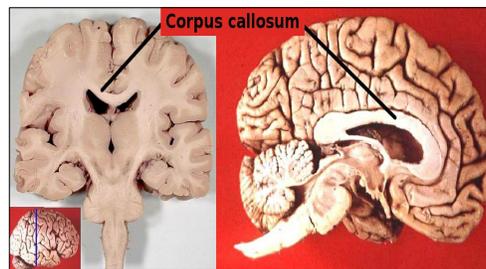


Fig 1: Corpus Callosum [67]

The coordination between the two hands, two eyes is maintained by corpus callosum. Any defect in callosum leads to Dyslexia or Split Brain disease.

## 1.3. Why Nervous System and Agents?

The first human system which was taken into account was the Human Immune System for agent security. The next focus was on nervous system where its functions help in developing the agents work in a coordinated way. The noticeable thing is that different processes are being controlled simultaneously by our nervous system in addition to processes that are going on at the sub-

conscious level and body regulation processes. Truly agents would require this very capability of the nervous system that is to operate many simultaneous foreground and background processes and bring background processes into the foreground immediately when necessary.

In this research work, a software agent framework with two knowledge bases has been designed to overcome malicious host threats and uncontrolled agent clones. These two knowledge bases are coordinated by an agent called Co-Agent. The jobs in one of the knowledge bases are allocated to the specialized agents by the Co-Agent. Those agents find the malicious hosts' information in the outside environment if any. The agents pass the hosts' information to other agents one by one because they are not sure whether they will return back to their home due to the attack by the hosts. The agents who return back to their home will inform the Co-Agent that they have returned back and confirm whether their allocated jobs were finished. Also they inform the other agents in the home about the host information. With the available number of agents in the home and the time taken to finish the jobs, the Co-Agent learns by itself in allocating the job for the next journey. Also the agent population is maintained by controlling the agent clones.

## **2. RELATED WORK**

Michael Luck, Peter McBurney and Chris Preist (2004) have proposed a paper about agent technologies in which they have added the background to agent technology, the state-of-the-art in agent technology, a long-term vision for the field, the technology gaps between the state-of-the-art and the long-term vision and finally they have ended up with the trends and challenges that will need to be addressed over the next ten years so as to progress the field and realize the benefits [39].

Lily Chang, Junhua Ding et al., (2007) have provided a formal net-within-net paradigm and demonstrated how to apply it to model the coordination of multi-agent systems. They have designed a formal model which enables us to better understand system requirements and critical design issues [35].

M. Anissimov (2007) says "The human nervous system is the most complex object known to science, as it includes the intricate CNS and a brain with 10 billion neurons and many times more interneural connection. For some people, this brain and the increase in intelligence it signifies is what separates human beings from other animals. As the difference is mainly a quantitative one, the gulf between humans and the great apes is seen as small, to the point that there is some advocacy to have chimpanzees placed in the same genus as human beings and great apes to be considered "persons" that should be accorded "human rights." However, for adherents of many religions, human beings are separated qualitatively from the rest of the animal world by having a spiritual nature and the presence of a soul or spirit, which exists in harmony with the body and CNS, but also transcends the physical aspect. For such, it is this spiritual aspect that defines the nature of humans more than the complex central nervous system [7].

Charles Sherrington (1906) developed the concept of stimulus-response mechanisms in much more detail, and Behaviorism, the school of thought that dominated Psychology through the middle of the 20th century, attempted to explain every aspect of human behaviour in stimulus-response terms [13].

James S. Albus and Anthony J.Barbera (2005) have designed a cognitive architecture named Real-Time Control System (RCS) to enable any level of intelligent behavior including human levels of performance. RCS was inspired 30 years ago by a theoretical model of the cerebellum, the portion of the brain responsible for fine motor coordination

and control of conscious motions. It was originally designed for sensory-interactive goal-directed control of laboratory manipulators. Over three decades, it has evolved into real-time control architecture for intelligent machine tools, factory automation systems, and intelligent autonomous vehicles [31].

Thi-Minh-Luan Nguyen and Christophe Lecerf (2004) have proposed a paper in which they have described the use of an agent-based method for studying cerebellar cortex's behavior which relies on three important notions: S-propagator, Discrete Event System Specification (DEVS) and multi-agent systems. They have also solved the questions related to the representation of system behavioral dynamics by DEVS on one hand and its dynamics simulation by agent-based techniques on the other hand [57].

Qingping Lin, Liang Zhang, Sun Ding, Guorui Feng and Guangbin Huang (2008) have proposed an intelligent mobile agent framework to address the scalability issue in LCVE. The autonomous migration/cloning of the mobile agents decentralized the system workload and optimized the resource utilization. The intelligent mobile agents can autonomously migrate or clone at any suitable participating host dynamically at run-time, the system workloads can be distributed more pervasively to avoid potential bottleneck. With these qualities, they have improved the scalability of LCVE [47].

### **3. PROBLEM FORMULATION AND PROPOSED FRAMEWORK**

#### **3.1. Objectives**

- To develop an agent framework functionally similar to the Human Nervous System to overcome malicious host threats.
- To enable agent learning and adaptation using the acquired knowledge to control the agent clones.
- To design two Knowledge Bases (like the two lobes of the Human Brains) and to design a Co-Agent (like Corpus Callosum) for knowledge base coordination.

#### **3.2. Problem Formulation**

The main spotlight of the agent framework is coordination and learning. Coordination is done by Corpus Callosum. In the absence of the corpus callosum, the human body will function but not in an efficient way such that it leads to malfunctioning and syndromes. Here, the role of corpus callosum is taken by the Co-Agent. It coordinates between the two knowledge bases and makes the system to function effectively. In the absence of the Co-Agent, the system will function but not in an efficient way such that it leads to the continuous attack of the same malicious hosts and lack of agents in the system due to agents' death.

The Brain stores and updates the information during each experience that happens in day to day life and behaves accordingly with the past experience. This controls in repeating the same mistakes. This is the process of learning. The knowledge base has been updated by the information of the malicious hosts. The malicious hosts' attack has to be faced for the first time because the agents do not know them and they cannot learn from the knowledge base also. But they are very tactic for the next time in skipping from the attack of the malicious hosts. This is done by the learning process of the agents from the knowledge base. The Co-Agent also learns and decides on selecting the number of agents from the past travel/experience. Even the number

of clones is also produced only in a needed number to avoid the overflow of clones. Too many unwanted clone agents will reduce the performance of the system.

### 3.3. Proposed Framework

The agent-based framework for agent coordination and learning included two Knowledge Bases, few agents and a coordinating agent known as the Co-Agent. Fig 2 shows the Proposed Framework to overcome malicious host threats and uncontrolled agent clones.

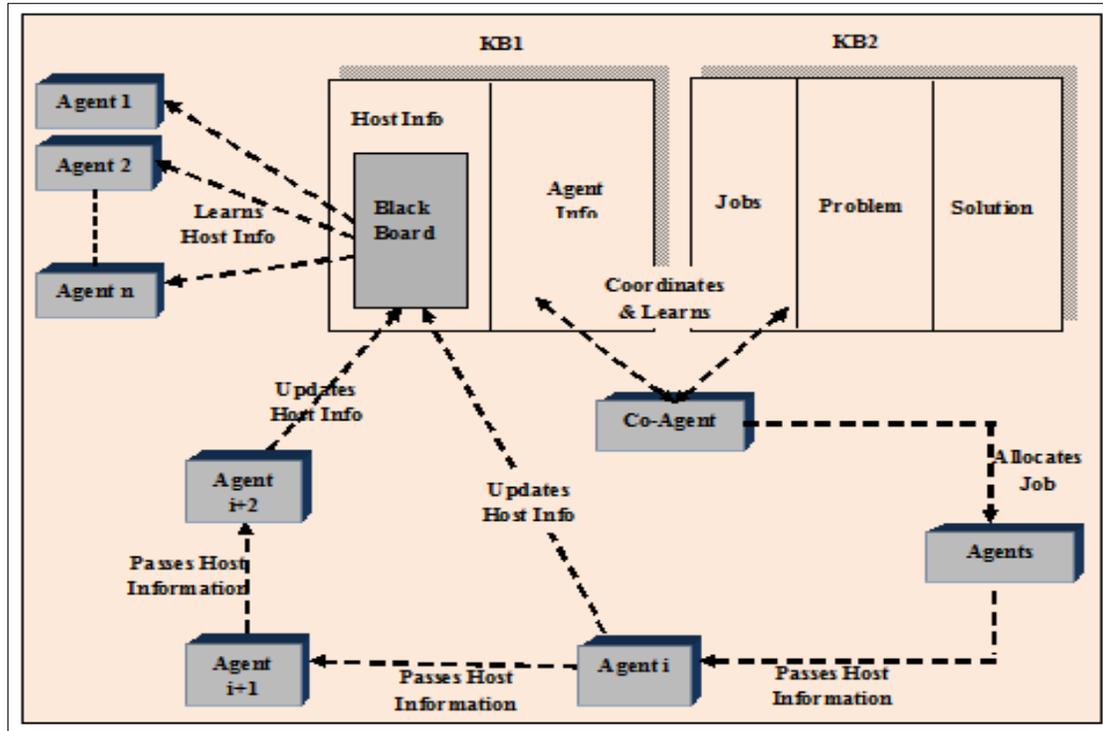


Fig 2. Proposed framework to overcome malicious host threats and uncontrolled agent clones

Two Knowledge Bases (KB) are used in which one Knowledge Base known as KB2 is the usual Knowledge Base that is used in any agent developing shells. The other one known as KB1 is the additional KB which is attached in the framework for solving the above stated problems. KB1 has the details of known malicious host information and the information of the agents. KB2 has the problems and its corresponding solutions that are to be read and used by the agents for problem solving when they are in need. KB2 also has the list of jobs that are to be finished by the agents within the given time and the jobs have their own priorities and the agents have their own capabilities.

When the jobs are allotted to the agents by the Co-Agent, they come in a queue according to its priority. The agents' details and the host details are stored in KB1 and the host details will be nil during the first cycle. Each agent has unique capability which is eligible to do more than one job. The Co-Agent selects a minimum number of agents and they are given a minimum time to finish the job because the Co-Agent does not know the job size, the number of needed agents and the needed time.

Before starting their job, the agents learn the malicious host information in KB1 and the jobs' corresponding problems and solutions from KB2 to make sure that they update themselves with the problems and the solutions which will be helpful for them to reach their goal. The agents are sent into the outside world for finishing their jobs where they face so many threats which are vulnerable for them. For this framework, the focus is only on the malicious hosts' attack. If any agent is attacked by any malicious host, they may die immediately or they may live for a shorter period or may be alive for a longer period. When the agents are getting attacked by any malicious host, the agents start scratching the host's IP address which must reach the home so that the agents avoid the future attacks by the same host. This social ability of making other agents becoming aware is one of the attributes of agents.

If any agent gets attacked by a malicious host, it will communicate with the other agent and pass the found IP address because the affected agent is not sure whether it will reach its home safely. The received agent will also communicate with the other agent and pass the same IP address. So whether they finish their jobs or not, the agents try reaching their home with the jobs they have finished so far and the IP address if they had had any because they have to report the Co-Agent within the given time.

After reaching the home, they report the Co-Agent that they have come back and inform whether the job is finished. The Co-Agent makes a note in KB1 of how many agents have returned back and how many agents have the jobs incomplete. It also notes their reporting time. The agents are called *Healthy agents* when they return back within the time, which will be given more preference for doing the job in future. The agents who come after the given time are known as *Delayed agents* and they will be given less preference than the healthy agents. Some agents may not return back due to several reasons, and these agents are noted as *Missed agent*. If any agent has brought any new solution, then they can update the solution for the corresponding problem which is in KB2.

The IP addresses which are brought to the home are informed to all the other agents through Blackboard approach. The agents having the IP address write the address in the blackboard but avoids writing the same if the IP address is already written. The Co-Agent updates the total details in both KB1 and KB2. The jobs which are not yet completed have to be completed. The Co-Agent learns from the previous experience and decides whether the agent number has to be increased or decreased compared with the previous allocation. This learning is done by Q-Learning.

When the agents come through the same IP address during the next travel, they skip the IP address and make them prevent from getting attacked. If the needed number of agents is less than the available agents, then the Co-Agent makes a clone and uses it for the function. Note that only the needed of clones will be produced. The cycle goes on till the jobs get completed.

### **3.3.1. Knowledge Base**

A knowledge base is a special kind of database for knowledge management. It provides the means for the computerized collection, organization, and retrieval of knowledge. DoyleSoft Knowledge Base has been used in developing the framework. It can rapidly search through megabytes of information, easily annotate articles, track the use of articles, and Distribute information to others in our company or to other organizations [72]. Fig 3 shows the screenshot of the knowledge base.

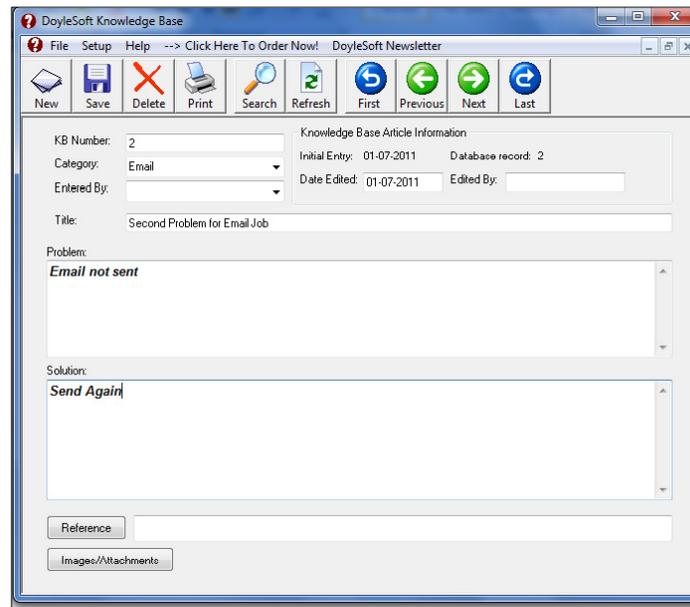


Fig 3: Knowledge Base

### 3.3.2. Blackboard Approach

This approach is an indirect communication approach. A blackboard is an excellent framework for combining several separately established diagnostic systems. The development of an application involves numerous developers. The modularity and independence provided by large-grained Knowledge Sources (KS) in blackboard systems allows each KS to be developed and tested separately. The software-engineering benefits of this approach apply during design, implementation, testing, and maintenance of the application. The scope of the problem to be solved was the prime factor in selecting a blackboard approach.

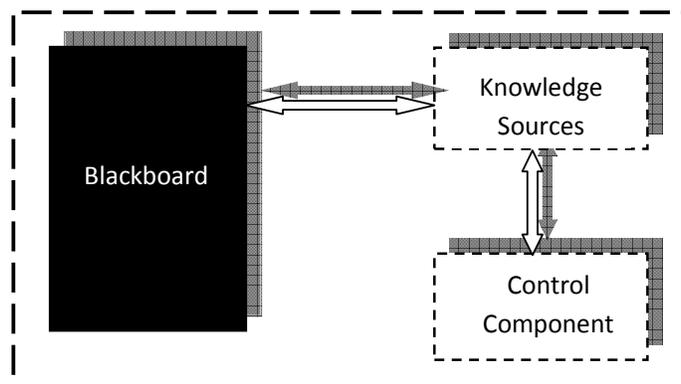


Fig 4: Components of the Blackboard Model [18]

*Knowledge Sources* are independent modules that contain the knowledge needed to solve the problem. KSs can be widely diverse in representation and inference techniques. *The Blackboard* is a global database containing input data, partial solutions, and other data that are in various problem-solving states. A *Control Component* makes runtime decisions about the course of

problem solving and the expenditure of problem-solving resources. The control component is separate from the individual KSSs. In some blackboard systems, the control component itself is implemented using a blackboard approach (involving control KSSs and blackboard areas devoted to control) [18].

### 3.3.3. Q-Learning

Q-learning was first introduced by Watkins in 1989. The convergence proof was presented later by Watkins and Dayan in 1992. Q-learning is a reinforcement learning technique that works by learning an action-value function that gives the expected utility of taking a given action in a given state and following a fixed policy thereafter. One of the strengths of Q-learning is that it is able to compare the expected utility of the available actions without requiring a model of the environment.

The problem model consists of an agent, states  $S$  and a number of actions per state  $A$ . By performing an action  $a$ , where  $s \xrightarrow{a} s'$ , the agent can move from state to state. Each state provides the agent a reward (a real or natural number) or punishment (a negative reward). The goal of the agent is to maximize its total reward. It does this by learning which action is optimal for each state.

The algorithm therefore has a function which calculates the Quality of a state-action combination:

$$Q: S \times A \longrightarrow R$$

Before learning has started,  $Q$  returns a fixed value, chosen by the designer. Then, each time the agent is given a reward (the state has changed) new values are calculated for each combination of a state  $s$  from  $S$ , and action  $a$  from  $A$ . The core of the algorithm is a simple value iteration update. It assumes the old value and makes a correction based on the new information [73].

The procedural form of the algorithm is:

```

Initialize  $Q(s, a)$  arbitrarily
Repeat (for each episode):
  Initialize  $s$ 
  Repeat (for each step of episode):
    Choose  $a$  from  $S$  using policy derived from  $Q$ 
      (e.g.,  $\epsilon$ -greedy)
    Take action  $a$ , observe  $r, s'$ 
     $Q(s, a) \leftarrow Q(s, a) + \alpha [r + \gamma \max_{a'} Q(s', a') - Q(s, a)]$ 
     $s \leftarrow s'$ ;
  until  $s$  is terminal

```

The parameters used in the Q-value update process are:

$\alpha$  - The learning rate, set between 0 and 1. Setting it to 0 means that the Q-values are never updated, hence nothing is learned. Setting a high value such as 0.9 means that learning can occur quickly.

$\gamma$  - Discount factor, also set between 0 and 1. This models the fact that future rewards are worth less than immediate rewards. Mathematically, the discount factor needs to be set less than 1 for the algorithm to converge.

$\max_{a'} R$  - the maximum reward that is attainable in the state following the current one. i.e the reward for taking the optimal action thereafter.

This procedural approach can be translated into plain English steps as follows:

- Initialize the Q-values table,  $Q(s, a)$ .
- Observe the current state,  $s$ .
- Choose an action,  $a$ , for that state based on one of the action selection policies
- Take the action, and observe the reward,  $r$ , as well as the new state,  $s'$ .
- Update the Q-value for the state using the observed reward and the maximum reward possible for the next state. The updating is done according to the formula and parameters described above.
- Set the state to the new state, and repeat the process until a terminal state is reached.

### 3.3.4. Why Knowledge Base and not Database

The system needed two knowledge bases where Doyle Soft Knowledge Base is used. The system should have a knowledge base where the information of agents, hosts and jobs are to be kept. The information of the jobs are read and learnt by the agents directly from one of the knowledge bases. The agents also read and learnt the information of the malicious hosts' from the other knowledge base which is the most important work in the system. The Co-Agent wanted the information of the available agents which is mainly needed for the whole system's processing. A Database cannot be used in the system as the agents wanted only the information directly to be learnt by them but not the database from which the raw data has to be processed again and then learnt by the agents. This is the major reason for the usage of the Knowledge Base in the system.

## 4. SYSTEM IMPLEMENTATION AND RESULTS

The proposed framework was implemented with two knowledge bases, fifteen agents and a Co-Agent. The time allotted for every agent to finish its job was 60 seconds. The maximum wait time for a delayed agent was fixed as 70 seconds. The host information in KB1 was kept empty when the cycle started. The Agent information contained the list of agent names and the capabilities of doing their jobs. The given host path for the agents was 34.65.100.2, 22.45.120.12, 32.4.101.78, 43.12.1.1, 5.5.5.5, 55.66.11.20, 45.101.9.76, 67.22.6.6, 99.9.9.89, 71.20.20.6, 4.4.4.100. Table 1 shows the list of available jobs and capable agents.

Table 1: List of available jobs and capable agents

<b>Name of the Jobs</b>	<b>Capable Agents</b>
Email	Email Sender, Email Receiver, Email Processor, Email Checker, Email Cleaner
Scheduling Traffic	Info Reader, Info Passer, Info Searcher, Info Fetcher, Info Sender
Information Network	Job Checker, Time Allocator, Job Measurer, Job Priority Giver, Job Allocator

The jobs were finished with 9 agents with one duplicate agent with the time limit of 60 seconds. Let us have an overall view of the job cycle in Table 2.

Table 2: Final status of the agents and the jobs

Agent Names	Cycle 1		Cycle 2		Cycle 3		Cycle 4		Cycle 5		Final Status
	Selected agents	Job	Selected agents	Job	Selected agents	Job	Selected agents	Job	Selected agents	Job	
Email Sender	Y	IC D	N	N	N	N	Y	X	---	---	D
Email Receiver	Y	X	---	---	---	-	---	---	---	---	D
Email Processor	N	N	Y	X	---	-	---	---	---	---	D
Email Checker	N	N	Y	C	Y	C	Y	X	DY	C	CL
Email Cleaner	N	N	N	N	Y	C	Y	C	Y	C	H
Info Reader	Y	C	N	N	Y	C	Y	X	---	---	D
Info Passer	Y	IC	N	N	N	N	Y	IC	Y	C	H
Info Searcher	N	N	Y	IC	Y	C	N	N	Y	C	H
Info Fetcher	N	N	Y	ICD	N	N	N	N	Y	C	OD
Info Sender	N	N	N	N	Y	IC	Y	IC	Y	C	H
Job Checker	Y	IC	N	N	N	N	N	N	Y	C	H
Time Allocator	Y	C	N	N	Y	IC	N	N	Y	C	H
Job Measurer	N	N	Y	C	N	N	Y	C	Y	C	H
Job Priority Giver	N	N	Y	C	N	N	Y	C	N	N	H
Job Allocator	N	N	N	N	Y	IC	N	N	N	N	H

Note: Y- Agent Selected Complete

C- Job Complete Incomplete

IC- Job Incomplete

H-Healthy Agent

DY- Cloned Agent

D-Dead Agent

X- Agent Missed

OD-Once

C(D)- Delayed and Job

ICD- Delayed and Job

CL-Cloned Agent

Delayed

The list of the nodes which were specified for the agents to travel and the list of the hosts which were found to be malicious are given in Table 3.

Table 3: List of specified host path and the malicious hosts

Host Path	Malicious Hosts
34.65.100.2	Yes
22.45.120.12	No
32.4.101.78	Yes
43.12.1.1	No
5.5.5.5	Yes
55.66.11.20	No
45.101.9.76	Yes
67.22.6.6	Yes
99.9.9.89	No
71.20.20.6	No
4.4.4.100	No

The proposed system can be proved to be better in performance compared with few of the already proposed research works. The merits of the proposed system are given in Table 4.

Table 5: Comparison of other systems with the agent based framework

Author Name	Year	Approach/ Technique Used	Demerits of the Technique	Merits of the Proposed Approach in this research work
Francesco Aiello, Giancarlo Fortino, Antonio Guerrieri and Raffaele Gravina [23]	2011	MAPS	The number of agent production is unknown thereby it leads to the unlimited number of agent clones.	The number of agent clones are always known as it is taken care of by the Co-Agent.
Qingping Lin, Liang Zhang, Sun Ding, Guorui Feng and Guangbin Huang [47]	2008	IA Framework	The number of uncontrolled migrating clones are produced	Only the required number of clones is produced.
Senol Zafer Erdogan, E.Murat Esin and Erdem Ucar [52]	2008	Self Cloning Ant Colony Approach	The ants self clones and then destroys themselves after a period of time.	Cloning happens after the destruction or missing of agents where the missing process happens in the outside environment.
Junhua Ding, Dianxiang Xu, Xudong He, and Yi Deng [33]	2005	SPIN	The coordination between the agents become tedious when the mobile agent system becomes complex	Coordination is always maintained among the agents as the role is played by the Co-Agent
Fuyuki Ishikawa, Nobukazu Yoshioka, Yasuyuki Tahara and Shinichi Honiden [24]	2004	BEPL	The agents are very rigid in Web Service applications to adapt to various environments.	The agents are very flexible in Web Services and they can adapt to various environments.

Gheorghe Tecuci and Michael R. Hieb [26]	2002	Disciple Approach	Three independent knowledge bases are used to train the agents where the KBs have to be updated by the user.	Dependent Knowledge bases shows the coordination between them and makes the system perform better compared with the independent KBs.
Costas Tsatsoulis and Leen-Kiat Soh [16]	2000	Agents in Telecommunication Network.	Uncontrolled agents are produced more in number which reduced system performance	Reduces the production of uncontrolled agents which maintains the system performance.
Curtis A. Carver, Jr., John M.D. Hill and John R. Surdu [17]	2000	Adaptive Intrusion Response	When complexity of computer attacks increases, more robust Intrusion response systems were necessary.	As the agents are aware of the malicious hosts, how much ever attacks increase, they can protect themselves.

## 5. CONCLUSION AND FUTURE WORKS

Agents are still in an emerging state. There are many threats for the agents on their voyage to fulfil their goals. This research work has concentrated on one threat which is the malicious host attack. The result of the malicious host threat is loss of agents. So far, the problem of loss of agent was solved by cloning the agents which leads to uncontrolled production of agents thereby reducing the system performance. Researchers are keen in making the agents adaptive to escape from the malicious hosts' attack through many processes like encryption.

We have designed and developed an agent based framework in which agents learn the host information and adapt themselves to escape from the malicious hosts by the process of learning. Our system has got records of present agents and the malicious host information which will solve the above stated problems. Controlled number of clones is produced to maintain the system performance. Coordination is maintained between two knowledge bases by the Co-Agent in solving the problem and run the system effectively. The functions of the agent based system are inspired by the functions of the Human Nervous System.

The present work can be extended by taking the time allotment for the agents into consideration by giving more time for the agents if the job size is more. The cloned agents are produced from the available agents, which will affect cloning if all available agents are missing. The system can be tested using real time resources. The other vulnerabilities of the agents and other cloning techniques can be taken into account to develop the system.

## REFERENCES

- [1] Alan Kay, Computer Software. Scientific American 251(3): 53–59,1984.
- [2] Alexander Serenko and Brian Detlor, Agent Toolkits: A General Overview of the Market and an Assessment of Instructor Satisfaction with Utilizing Toolkits in the Classroom, 2002.
- [3] Alexander Serenko and Brian Detlor, Intelligent Agents as Innovations, Published in Artificial Intelligence and Society 18(4), Pages: 364-381, 2004.
- [4] Alexander Serenko, Umar Ruhi and Mihail Cocosila, Unplanned effects of Intelligent Agents on Internet Use: A Social Informatics Approach, Published in Artificial Intelligence and Society 21(1-2), Pages: 141-166, 2007.
- [5] Anderer and Bernd, Marketing Strategies for Business-to-Consumer Electronic Commerce and the strategic implications of software agents and XML, Diplomarbeit, Karlsruhe 1998.
- [6] Andrew Secker, Alex A. Freitas and Jon Timmis, AISEC: An Artificial Immune System for E-mail Classification, The 2003 Congress on Evolutionary Computation (CEC'03), Pages: 131- 138, Vol.1, 2004.
- [7] Anissimov, M., How does the nervous system work? 2007.
- [8] Bob van der Vecht, Frank Dignum, John-Jules Ch.Meyer and Martijn Neef, A Dynamic Coordination Mechanism using Adjustable Autonomy, Autonomous Agents & Multiagent Systems/Agent Theories, Architecture and Languages – AAMAS(ATAL), 2007.
- [9] Bjorn Hermans, Intelligent Software Agents on the Internet, Chapter 3.
- [10] Blinkov S. M., Glezer I.I., The Human Brain, Published in In Figures and Tables, A Quantitative Handbook, 1968.
- [11] Carlos A. Coello Coello and Nareli Cruz Cortes, Solving Multiobjective Optimization Problems using an Artificial Immune System, Genetic Programming and Evolvable Machines – GPEM, 2005.
- [12] Chamberlin, S. L., and Narins B., The Gale Encyclopedia of Neurological Disorders. Detroit: Thomson Gale. ISBN 078769150X, 2005.
- [13] Charles Sherrington, The Integrative Action of the Nervous System, 1906.
- [14] Chauvet G., On the Mathematical Integration of the Nervous Tissue Based on the S-propagator Formalismk, I. Theory . Journal of Integrative Neuroscience, vol 1(1), p 31-68, 2002.
- [15] Cheikhrouhou M., Conti, Labetoulle J. and Marcus P.K., Intelligent Agents for Network Management: Fault Detection Experiment, Published in Sixth IFIP/IEEE International Symposium on Integrated Network Management, 1998.
- [16] Costas Tsatsoulis and Leen-Kiat Soh, Intelligent Agents in Telecommunication Network, Computational Intelligence in Telecommunication Networks, 2000.
- [17] Curtis A. Carver, Jr., John M.D. Hill and John R. Surdu, A Methodology for Using Intelligent Agents to provide Automated Intrusion Response, Proceedings of the 2000 IEEE Workshop on Cybernetics Information Assurance and Security, 2000.
- [18] Daniel D.Corkill, Blackboard Systems, Blackboard Technolgy Group, Inc., Original, Unabridged version of the article appeared in AI Expert 6(9):40-47, September 1991.
- [19] Descartes R (1989). Passions of the Soul. Voss S. Hackett. ISBN 9780872200357.Deugo D., Weiss M., and Kendall E., Reusable Patterns for Agent Coordination, Coordination of Internet Agents: Models, Technologies and Applications, pp.347-368, 2001.
- [21] Eduardo Alonso, Mark d' Inverno, Daniel Kudenko, Michael Luck and Jason Noble, Learning in Multi-Agent Systems, 2001.
- [22] Fernando Fernandez, Daniel Borrajo and Lynne E. Parker, A Reinforcement Learning Algorithm in Cooperative Multi-Robot Domains, Published in Journal of Intelligent and Robotic Systems, Pages:161-174, 2005.
- [23] Francesco Aiello, Giancarlo Fortino, Antonio Guerrieri and Raffaele Gravina, A Java-Based Agent Platform for Programming Wireless Sensor Networks, The Computer Journal, 54(3), pp:439-454, 2011.
- [24] Fuyuki Ishikawa, Nobukazu Yoshioka, Yasuyuki Tahara and Shinichi Honiden, Mobile Agent System for Web Services Integration in Pervasive Networks, International Workshop on Ubiquitous Computing, pp:38-47, 2004.
- [25] Gheorghe Tecuci and Michael R. Hieb, Teaching Intelligent Agents: the Disciple Approach,International Journal of Human Computer Interaction, Vol.8, No.3, pp.259-285, 1996.

- [26] Gheorghe Tecuci, Mihai Boicu, Dorin Marcu, Bogdan Stanescu, Cristina Boicu and Jerome Comello, Training and Using Disciple Agents: A Case Study in the Military Center of Gravity Analysis Domain, In AI Magazine, 24, 4, pp.51-68, 2002.
- [27] Hare G.M.P.O., Grady M.J.O., Keegan S., Tynan R. and Marsh D., Intelligent Agile Agents: Active Enablers for Ambient Intelligence, 2004.
- [28] Huhns M., and Weiss G., Special Issue on Multiagent Learning, Machine Learning Journal, 33(2-3), 1998.
- [29] Hyacinth S. Nwana, Software Agents: An Overview, Published in Knowledge Engineering Review, Vol.11, No3, pp: 205-244, 1996.
- [30] Imam I.F., Intelligent Adaptive Agents, Papers from 1996 AAAI Workshop, Technical Report WS-96-04, AAAI Press, 1996.
- [31] James S. Albus and Anthony J.Barbera, RCS: A Cognitive Architecture for Intelligent Multi-Agent Systems, Annual Reviews in Control, 2005.
- [32] Jennings, N.R. and Wooldridge M., Applications of Intelligent Agents. In N. Jennings and M.J. Wooldridge (eds.), Agent Technology: Foundations, Applications, and Markets. Springer-Verlag. pp. 3-28, 1998.
- [33] Junhua Ding, Dianxiang Xu, Xudong He and Yi Deng, Modeling and Analyzing a Mobile Agent-Based Clinical Information System, International Journal of Intelligent Control and Systems, Vol:10, No:2, pp:143-151, 2005.
- [34] Kandel ER, Schwartz JH, Jessel TM, ed. "Ch. 2: Nerve cells and behavior". Principles of Neural Science. McGraw-Hill Professional. ISBN 9780838577011, 2000.Lily Chang, Junhua Ding, Xudong He and Sol M. Shatz, A Formal Approach for Modeling Software Agents Coordination, 2007.
- [36] Luck M., McBurney P., Shehory O. and Willmott Steve, Agent Technology Roadmap, 2005.
- [37] Maton, Anthea; Jean Hopkins, Charles William McLaughlin, Susan Johnson, Maryanna Quon Warner, David LaHart, Jill D. Wright (1993). Human Biology and Health. Englewood Cliffs, New Jersey, USA: Prentice Hall. pp. 132–144. ISBN 0-13-981176-1.Michael Luck, Peter McBurney and Chris Preist, Agent Technology:Enabling Next Generation Computing; A Roadmap for Agent-Based Computing, 2003.
- [39] Michael Luck, Peter McBurney and Chris Preist, A Manifesto for Agent Technology: Towards Next Generation Computing, Autonomous Agents and Multi-Agent Systems, 9, pp:203-252, 2004.
- [40] Na Tang and Rao Vemuri V., An Artificial Immune System Approach to Document Clustering, Proceedings of the 2005 ACM Symposium on Applied Computing, Pages:918-922, 2005.
- [41] Nieuwenhuys R, Ten Donkelaar HJ and Nicholson C., The Central Nervous System of Vertebrates, 1998.
- [42] Nikola Kasabov, ECOS: Evolving Connectionist Systems and the Eco Learning Paradigm, Proceedings of The Fifth International Conference on Neural Information Processing(ICONIP), 1998.
- [43] Onn Shehory, Katia Sycara, Prasad Chalasani and Somesh Jha, Agent Cloning, 1998.
- [44] Oren Etzioni and Daniel S. Weld,"Intelligent Agents on the Internet: Fact, Fiction and Forecast", IEEE Expert: Intelligent Systems and their Applications, Vol:10, pp:44-49, 1995.
- [45] Pedro Cuesta-Morales, Zahia Guessoum, Juan-Carlos Gonzalez-Moreno and Juan Pavon-Mestras Agent Technologies at Work UPGRADE Vol. V, No.4, August 2004.
- [46] Predrag Tomic, "Distributed Coalition Formation for Collaborative Large-Scale Multi-Agent System", Phd Thesis, University of Illinois at Urbana-Champaign, 2006.
- [47] Qingping Lin, Liang Zhang, Sun Ding, Guorui Feng and Guangbin Huang, Intelligent Mobile Agents for Large-Scale Collaborative Virtual Environment, The International Journal of Virtual Reality, 7(2), pp: 63-72, 2008.
- [48] Roger L. King, Aric B. Lambert, Samuel H. Russ, and Donna S. Reese, The Biological Basis of the Immune System as a Model for Intelligent Agents, Second Workshop on Bio-Inspired Solutions to Parallel Processing Problems, Lecture Notes in Computer Science 1586, 1999.
- [49] Russell, S. and Norvig P., Intelligent agents, Artificial Intelligence: A Modern Approach. (Prentice Hall, 2003) second edition [ISBN 0137903952]. Chapter 2, 2003.
- [50] Sen S., Coevolution and Learning in Multiagent Systems, Papers from 1996 Springer Symposium, Technical Report SS-96-01, AAAI Press, 1996.
- [51] Sen S., Special Issue on Evolution and Learning in Multiagent Systems, International Journal of Human-Computer Studies, 48(1), 1998.
- [52] Senol Zafer Erdogan, Murat Esin E. and Erdem Ucar, Refinding of the Optimal Path When the Network Changes, ISSN 1305-6468, Trakya Univ J Sci, 9(1), pp:25-29, 2008.

- [53] Slavisa Sarafijanovic and Jean-Yves Le Boudec, An Artificial Immune System Approach with Secondary Response for Misbehaviour Detection in Mobile Ad-Hoc Networks, IEEE Transactions on Neural Networks, Vol.16, Pages: 1076-1087, 2005.
- [54] Srividhya Sathyanath and Ferat Sahin, AISIMAM- An Artificial Immune System Based Intelligent Multi Agent Model and its Application to a Mine Detection Problem, International Conference on Systems, Man and Cybernetics, Vol.3, ISSN: 1062-922X, 2003.
- [55] Stephen Haag, Management Information Systems for the Information Age, Pages 224-228, 2006.
- [56] Tecuci G., Wright K., Lee S.W., Boicu M. and Bowman M., A Learning Agent Shell and Methodology for Developing Intelligent Agents, The AAAI-98 Workshop on Software Tools for Developing Agents, 1988.
- [57] Thi-Minh-Luan Nguyen and Christophe Lecercf, An Agent-Based Simulation Method for Studying Nervous System, Published in International Conference of RIVF, 2004.
- [58] Wei Chen and Keith Decker, Applying Coordination Mechanisms for Dependency Relationships under Various Environments, Workshop on MAS Problem Spaces and Their Implications to Achieving Globally Coherent Behaviour, AAMAS, 2002.
- [59] Weiss G., Sen S., Adaption and Learning in Multiagent Systems, Lecture Notes in Artificial Intelligence, Vol.1042, Springer-Verlag, 1996.
- [60] Weiss G. Distributed Artificial Intelligence Meets Machine Learning, Lectures Notes in Artificial Intelligence, Vol. 1221, Springer-Verlag, 1997.
- [61] Weiss G., Special Issue on Learning in Distributed Artificial Intelligence Systems, Journal of Experimental and Theoretical Artificial Intelligence, 10(3), 1998.
- [62] Weiss G., Learning in Multiagent Systems, In Weiss G., editor, Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence, Pages 559-598, 1999.
- [63] M. Wooldridge and N. R. Jennings, Intelligent agents: theory and practice, Knowledge Engineering Review., vol. 10(2), pp. 115–152, 1995.
- [64] Zbigniew Skolicki and Tomasz Arciszewski, Intelligent Agents in Design, ASME Proceedings of Design Engineering Technical Conferences and Computers and Information in Engineering (DETC), 2003.
- [65] <http://library.thinkquest.org/26618/en-2.1.1=The%20human%20nervous%20system.htm>
- [66] <http://library.thinkquest.org/5777/ner1.htm>
- [67] [http://en.wikipedia.org/wiki/Human\\_brain](http://en.wikipedia.org/wiki/Human_brain)
- [68] [http://en.wikipedia.org/wiki/Software\\_agent](http://en.wikipedia.org/wiki/Software_agent)
- [69] [http://en.wikipedia.org/wiki/Intelligent\\_agent](http://en.wikipedia.org/wiki/Intelligent_agent)
- [70] <http://alumnus.caltech.edu/~croft/research/agent/definition/>
- [71] [http://en.wikipedia.org/wiki/Human\\_brain](http://en.wikipedia.org/wiki/Human_brain)
- [72] <http://www.answers.com/topic/knowledge-base>
- [73] <http://en.wikipedia.org/wiki/Q-learning>
- [74] <http://www.giveawayoftheday.com/doylesoft-knowledge-base/>
- [75] <http://doylesoft.com/>

## Authors

**Annie Sujitha** studied Master of Science - Information Technology in Stella Maris College, Chennai. She received her M.Phil degree in the field of Agent Based Computing in Bharathiar University, Coimbatore. Her area of interests are Software Agents and Artificial Intelligence.



**Mrs.T.Amudha** is serving as Assistant Professor in the Department of Computer Applications, Bharathiar University and has more than 12 years of academic experience. She is specialized in the fields of Agent based Computing, Bio-inspired Computing, Object Technology and Distributed Computing. She has more than 20 research publications for her credit in International/ National Journals and Conferences. She is a member in Computer Society of India[CSI], International Association of Computer Science and Information Technology[IACSIT] and International Association for Engineers[IAENG].

