# SELF ORGANIZATION MECHANISM IN AN AGENT NETWORK BY DECENTRALIZED APPROACH

**K.Rajasri[1], Pulaka Gupta[2], Rahul.S[3]**

*[1]Senior Assistant Professor, Department of Computer Science and Engineering, Christ College Of Engineering and Technology, Moolakulam, Puducherry, India*
*[2]Student of M.Tech, Department of Computer Science and Engineering, Christ College Of Engineering and Technology, Moolakulam, Puducherry, India*
*[3]Student of M.Tech Department of Computer Science and Engineering, Christ College Of Engineering and Technology, Moolakulam, Puducherry, India*

**Abstract**

*Self-organization provides a suitable model for developing self-managed complex distributed systems, such as grid computing and sensor networks. Unlike current related studies, which propose only a single principle of self-organization, this mechanism synthesizes the three principles of self-organization: cloning/ spawning, resource exchange and relation adaptation. Based on this mechanism, an agent can autonomously generate new agents when it is overloaded, exchange resources with other agents if necessary, and modify relations with other agents to achieve a better agent network structure. In this way, agents can adapt to dynamic environments. The proposed mechanism is evaluated through a comparison with three other approaches, each of which represents state-of-the-art research in each of the three self-organization principles. Experimental results demonstrate that the proposed mechanism outperforms the three approaches in terms of the profit of individual agents and the entire agent network, the load-balancing among agents, and the time consumption to finish a simulation run. In addition, in a dynamic environment, it is nearly impossible to use a static, design time generated system structure for efficient problem solving. Instead, the system needs to be able to self-organize at runtime, which means that the components of the system are responsible for adapting themselves to suit the dynamic environment. Self-organization is usually defined as "the mechanism or the process enabling the system to change its organization without explicit external command during its execution time.*

***Keywords:*** *Self Organizations, Cloning/Spawning, Resource exchange, Relation adaptation*

-----------------------------------------------------------------------***-----------------------------------------------------------------------

## 1. INTRODUCTION

COMPLEX distributed systems are becoming more and more common. With their evergrowing exponential complexity in deployment, operation, management and maintenance, it is highly desirable that these systems should be autonomous and be capable of self-management. In a dynamic environment, it is nearly impossible to use a static, design time generated system structure for efficient problem solving. Instead, the system needs to be able to self-organize at runtime, which means that the components of the system are responsible for adapting themselves to suit the dynamic environment. Currently, self-organization has been employed in many complex distributed systems, such as supply network management, computer networks security management , and sensor and communication networks management

## 2. SYSTEM ANALYSIS

### 2.1 Existing System

In this project it focuses on self organization in multi agent systems [1], we only consider the three basic principles they are cloning/spawning, resource exchange and relation adaptation.Despite this, to the best of our knowledge, there has been no attempt to combine the three principles together in order to achieve better performance than possible with those self organization approaches which consider only one of the three principles. Towards this end, in this project, we present an integrative self organization mechanism, which combines the three basic principles together. It is difficult to combine three existing approaches to realize the self organization mechanism [6], because existing approaches have been devised based on different purposes, different models and different environmental settings. Therefore, for each principle,we develop an innovative and efficient approach, and these three approaches can effectively work together as a whole.

### 2.2 Proposed System

The proposed self-organization mechanism is devised in an agentnetwork, where each agent is directly connected with some other agents, called neighbors, and each agent can communicate only with its neighbors. Aiming at a general problem, instead of a particular problem, makes the proposed selforganizationmechanism potentially applicable in a wide variety of applications.This project focuses on a cooperative agent network, where agents cooperatively maximize their overall profits, rather than a selfish agent network, where each agent tries to maximize only its own profit. This focus is reasonable, because many applications are in a cooperative environment. Therefore, for each principle, we develop an innovative and efficient approach, and these three approaches can effectively work together as a whole

## 3. SYSTEM REQUIREMENTS

### 3.1 Hardware Requirements:

| | |
|---|---|
| PROCESSOR | : PENTIUM IV 2.6 GHz, Intel Core 2 Duo. |
| RAM | : 512MBDD RAM |
| MONITOR | : 15" COLOR |
| HARD DISK | : 40 GB |

### 3.2 Software Requirements:

| | |
|---|---|
| Front End | : Java (Swings) |
| Back End | : M SQL 5.5 |
| Operating System | : Windows XP/07 |
| IDE | : Eclipse |

## 4. SYSTEM DESIGN

### 4.1 Modules

This proposed scheme is divided into four modules. The modules are listed below,

1. User Interface Design
2. Cloning/Spawning.
3. Resource Exchange.
4. Relation Adaptation.
5. Self Organization Analysis.

### 4.2 Design Overview

#### 4.2.1 User Interface Design

In this module we design the windows for the project. These windows are used for secure login for all users. To connect with server user must give their username and password then only they can able to connect the server. If the user already exits directly can login into the server else user must register their details such as username, password and Email id, into the server. Server will create the account for the entire user to maintain upload and download rate. Name will be set as user id. . Logging in is usually used to enter a specific page.

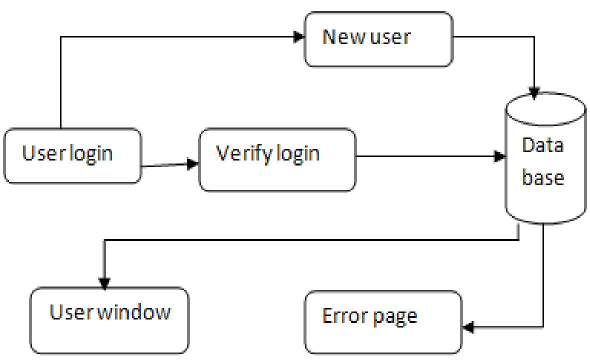


**Fig 4.2.1.1** User Interface Design

#### 4.2.2 Cloning/Spawning

When an agent is overloaded, it will create a new agent to take part of its load. Specifically, for an individual agent, spawning is triggered when it cannot finish the assigned tasks on time. If a task or several tasks in its list cannot be completed before the expiry time, an agent will spawn one or several apprentice agent(s), each of which has a corresponding resource to complete a task. The original agent then assigns tasks to these spawned apprentice agents. A spawned agent is a subordinate of the original agent, and a spawned agent cannot establish relations with other agents. On the other hand, cloning happens when an agent has too many neighbours (set as double as the average number of neighbours of each agent in the network), which means that the agent has a heavy overhead for managing relations with other agents. In this situation, to avoid possible communication congestion, the agent clones a new agent, and assigns half of its neighbours to the cloned agent. The cloned agent has the same resources as the original agent has, and maintains a peer relation with the original agent.

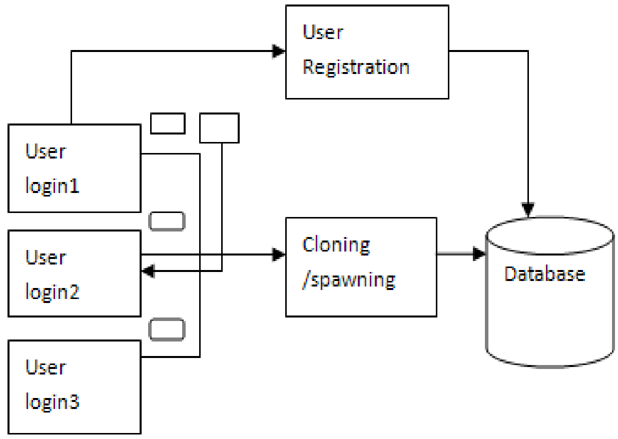


**Fig 4.2.2.1** Cloning/Spawning

#### 4.2.3 Resource Exchange

For a single agent, when a resource has not been used for a long time, the agent will transfer the resource to a neighboring agent, which needs this resource. Here, the "long time" duration is set to 20 time steps, obtained through experimental attempts to achieve best results. In this paper, for simplicity, we use an integer to represent a resource: resource No. 1, resource No. 2 and so on. In real applications, such as human social networks, such a resource might be a tool, e.g., a hammer. If a person has a hammer but he/she has not used it for a long time, he/she will give (or sell) the hammer to another person who needs it. Since this paper considers a cooperative agent network, an agent directly gives its unused resource to another agent.

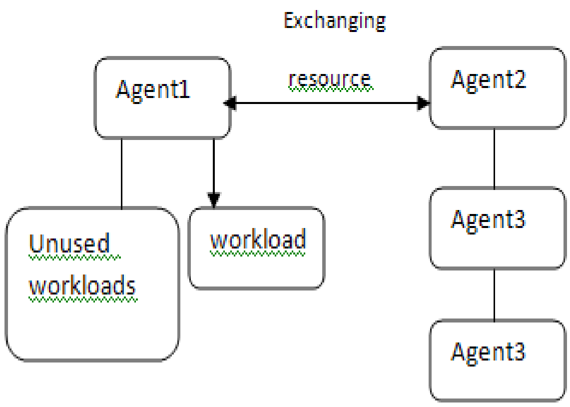


**Fig 4.2.3.1** Resource Exchange

### 4.2.4 Relation Adaptation

The relation adaptation algorithm is based on historical information of individual agents. Specifically, agents use the information regarding previous task allocation processes to evaluate their relations with other agents.This generally deals with add , modify, update, delete.

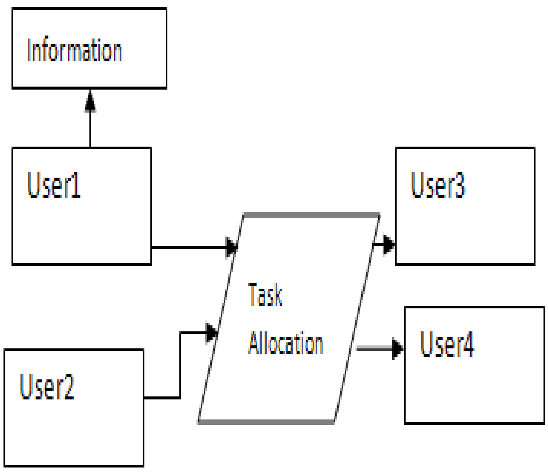


**Fig 4.2.4.1** Relation Adaptation

### 4.2.5 Self Organisation Analysis

In order to objectively evaluate the performance of our mechanism, we ran an experiment to compare our mechanism, written as Self-organization. For relation adaptation, we opt for the latest and the most efficient approach, K-Adapt. If an agent is overloaded, it spawns a new agent to handle part of its load. Then, if the task is too big and cannot be completed by a single agent, the task has to be divided into small tasks, one or some of which are assigned to the newly spawned agent. This approach is referred to as breakup.

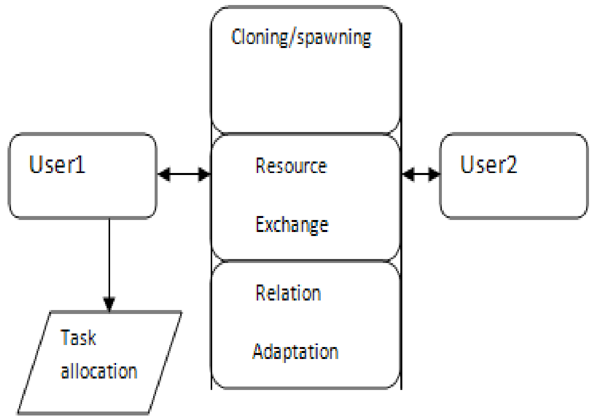


**Fig 4.2.5.1** Self Organisation Analysis

## 5. CONCLUSION

This project introduced an integrative self-organization mechanism, which combines the three principles of self organization, cloning/spawning, resource exchange and relation adaptation. Through combining the benefits of the three principles, our mechanism outperforms state-of-the art approaches, each of which focused on only a single principle. .

## REFERENCES

[1]. D. Ye, M. Zhang, and D. Sutanto, "Cloning, Resource Exchange and Relation Adaptation: A Self-Organising Multi-Agent Framework," Proc. 11th Int'l Conf. Autonomous Agents and Multiagent Systems (AAMAS '12), pp. 1431-1432, 2012.

[2]. J.O. Kephart and D.M. Chess, "The Vision of Autonomic Computing," Computer, vol. 36, no. 1, pp. 41-50, Jan. 2003.

[3]. The Mabe Middleware: Extending Multi-Agent Sixth IFIP Int'l Conf. Information Technology for Balanced Automation Systems in Manufacturing and Services, 2004.

[4]. N. Foukia, "IDReAM: Intrusion Detection and Response Executed with Agent Mobility," Proc. Int'l Conf. Autonomous Agents and Multiagent Systems (AAMAS '05), pp. 264-270, 2005.

[5]. M. Prokopenko, P. Wang, and P. Valencia, "Self-Organizing Hierarchies in Sensor and Communication Networks," Artificial Life, vol. 11, pp. 1-20, 2005.

[6]. R. Frei and G.D.M. Serugendo, "Self-Organizing Assembly Systems," IEEE Trans. Systems, Man and Cybernetics - Part C: Applications and Rev., vol. 41, pp. 885-897, Nov. 2011.Systems to Enable Open Business Collaboration," Proc.