



Coordinating a Self Organisation Mechanism in a Distributed Agent Network

K. Rekha

PG Student–CSE,

Krishnasamy College of Engg. and Technology,
Tamil Nadu, India**K. Priya** (B. E., M.tech)

Assistant Professor- CSE,

Krishnasamy College of Engg. and Technology
Tamil Nadu, India

Abstract- *Self-organization provides a suitable paradigm for developing self-managed complex distributed systems. This mechanism synthesizes the three principles of self-organization: cloning, 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, and modify relations with other agents to achieve a better agent network structure. 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. The agents can adapt to dynamic environments.*

Index terms: *Distributed multi-agent system, self-organisation, reinforcement learning*

I. INTRODUCTION

self-organisation has been employed in many complex distributed systems, such as supply network management[3], computer networks security management[4], sensor and communication networks management[5], and evolvable manufacturing assembly systems development[6], in order to improve their autonomy and efficiency[1], [2]. 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[7]. The self-organising distributed systems can continuously and autonomously arrange and rearrange their organisational structures, without any external control, so as to adapt to environmental changes. In addition, the self-organisation process is performed in a decentralised manner, so that the autonomous systems are robust against failures of any components in the system. This research is limited to self-organising multi-agent systems. Although some other research problems will also be mentioned, such as task allocation and load-balancing, they are used only to assist the discussion of self-organising multi-agent systems. An intelligent agent is an entity that can perceive its environment through sensors and act upon that environment via effectors. A multi-agent system is composed of several intelligent agents, and individual agents may perform different roles. An intelligent agent is able to make rational decisions autonomously in a dynamic environment by blending pro-activeness and re-activeness, showing rational commitments to decision making, and exhibiting flexibility when facing an uncertain and changing environment[10]. In this paper, an agent simply represents a node in an agent network. For the design of self-organising multi-agent systems[11], three basic principles of self-organising multi-agent systems are

Cloning/Spawning: Agents within the system will generate new agents to take part of their work load once they are overloaded.

Resource exchange: Agents can exchange skills or resources, if necessary, between each other to increase autonomy.

Relation adaptation: Agents should be able to create new specific relations between agents in order to remove the middle-agents.

In addition to the three basic principles, real-world self-organising system includes additional principles[12], [6] such as self (re)configuration, self repair and self adaptation.

II. MODEL DESCRIPTION

The self-organisation mechanism is designed in an agent network. Thus, it is necessary to define a model to describe the features of a network. The features include the number of agents in the network, the neighbours of each agent, the task and resource types in the network, and the resources possessed by each agent. In order to describe these features, four elements, A; N; T; G, are defined.

A is a set of collaborative agents in the network,

N is the set of agent a_i 's neighbours.

T is a set of task types.

G is a set of resource types.

The neighbour set, N_i , of agent a_i consists of three different types of neighbours: peer, subordinate and superior, which are formed by two relations, given in Definition 1 and Definition 2 respectively.

Definition 1 (peer-to-peer) A peer-to-peer relation is a Compatible Relation, which is reflexive and symmetric,

Definition 2 (subordinate superior) A subordinate superior relation is a Strict Partial Ordering Relation, which is irreflexive, asymmetric and transitive,

Definition 3 (relation strength) Relation strength indicates how strong a relation is between agents a_i and a_j . Larger value means a stronger relation.

III. EVALUATION INDICES OF THE MODEL

After describing the model, it is necessary to introduce an evaluation method to estimate the model. Towards this goal, three evaluation indices are introduced. These include cost, benefit and profit of an agent and the agent network. The cost, benefit and profit of the network are calculated after allocating a predefined number of tasks.

3.1 Cost Evaluation

The cost of the network consists of six parts: communication cost, computation cost consumed by agents to complete the assigned tasks, management cost for maintaining tasks and resources, management cost for maintaining relations with other agents, resource transfer cost and relation adaptation cost.

3.1.1 Communication cost

Communication cost relies on how many tokens are transmitted during previous task allocation processes.

3.1.2 Computation cost

Computation cost is measured as the number of time steps used by the agents for computing the assigned task.

3.1.3 Management cost

Management cost is management computation cost for maintaining tasks and resources. First management cost is that one list stores the tasks, owned by the agent and another list records the tasks assigned by other agents. The second management cost for an individual agent is the management relation cost generated by maintaining relations with other agents.

3.1.4 Resource Transfer Cost and Relation Adaptation Cost

The resource transfer cost occurs when a resource is transferred between two agents, and it is set to a constant denoted by $cost^{RT}$. The relation adaptation cost occurs once a relation adaptation happens between two agents. The relation adaptation cost is also represented as a constant, $cost^{RA}$.

3.2 Benefit Evaluation

The benefit obtained by an agent depends on how many tasks are completed by that agent. As each task i has a corresponding benefit. When the task is successfully completed by an agent, this agent can obtain this benefit. The benefit of the network is the sum of the benefits that all the agents obtain in the network.

3.3 Profit Evaluation

The aim of this mechanism is to maximize the profit of the agent network. of course, it is unreasonable to sum up indices with different units. Profit_{NET} is used to demonstrate the general situation of an agent network.

IV. THE INTEGRATIVE SELF-ORGANISATION MECHANISM

This mechanism consists of three components: cloning/ spawning, resource exchange and relation adaptation.

4.1 CLONING/SPAWNING

When an agent is overloaded, it will create a new agent to take part of its load. The agent has two options: cloning or spawning a new agent. For an individual agent, having tasks in the waiting list incurs management cost. In addition, each task has an expiry time, before which the task has to be finished. Therefore, when an agent has too many tasks in the waiting list, these tasks not only incur management cost but also face the risk of not being completed on time. When this happens, the agent is considered "overloaded." Likewise, having neighbours also incurs management cost. Thus, when an agent has too many neighbours, the management cost is high and the agent is considered "overloaded".

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. When a spawned agent finishes the assigned task, it is in an idle status. When the spawned agent remains in an idle status for a pre-defined period (set to 20 time steps) and no more such tasks need to be completed, it will destroy itself to save the original agent's relation management cost.

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. Unlike the apprentice agents, the cloned agent will not destroy itself even if it remains in an idle status. Instead, the original and cloned agents will recombine together, once the total number of their neighbours is less than a predefined threshold (set as 1:5 times as the average number of neighbours of each agent in the network). It should be noted that the cloning and the recombination can happen recursively.

4.2 RESOURCE EXCHANGE

Agents incur management cost for maintaining their resources. Hence, for a single agent, when a resource has not been used for a long time, the agent will transfer the resource to a neighbouring 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.

4.3 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. We develop a multi-agent Q-learning algorithm to tackle the relation adaptation problem. Before describing our relation adaptation algorithm, we first describe the actions, which are used to modify relations between agents. An action is defined as a decision made by an agent to adapt a relation with another agent.

V. EXPERIMENTAL ANALYSIS

In order to objectively evaluate the performance of our mechanism, we ran an experiment to compare our mechanism, written as Self-org., with three other approaches.

Because, to the best of our knowledge, there does not exist a mechanism, which combines the three self-organisation principles together: cloning/spawning, resource exchange and relation adaptation, select the most efficient approach from each of the three research areas for comparison.

(1) Central. This is an ideal centralised task allocation approach, in which there is an external omniscient central manager that maintains information about all the agents and tasks in the network. The central manager is able to interact with all the agents in the network without cost. This method is neither practical nor robust, but it can be used as an upper bound of the performance in our experiment.

(2) Hybrid Model. For cloning/spawning, select a famous model, developed by Kamboj [15]. Kamboj’s hybridMmodel consists of spawning and cloning. 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. On the other hand, if a task cannot be broken up into smaller parts, the entire task will be assigned to the new agent. This approach is referred to as cloning.

(3) MAL-Allocation. For resource exchange, choose an efficient approach from the area of resource allocation research. The selected resource allocation approach was devised by Zhang et al. [13], which includes two multiagent learning functions for task allocation and transfer. Although Zhang et al.’s work is not the latest work in the resource allocation research area, based on our investigation, it is the best one suitable for resource exchange, as their work was developed in a cooperative and distributed environment and the transferred tasks did not need to be returned to the original agents.

(4) K-Adapt. For relation adaptation, choose opt for the latest and the most efficient approach, K-Adapt, proposed by Kota et al. [9], which utilises a meta-reasoning approach to adapt the relation between two agents.

VI. CONCLUSION

This paper introduced an integrative self-organisation mechanism, which combines the three principles of self organisation: 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.

In the future, the analysis of the type of system generated by the proposed self-organisation mechanism will be an interesting research area. In this paper, it is assumed that each agent has an identical capacity to complete tasks. This assumption could be relaxed to let different agents have different capacities. Agents with higher capacities could complete tasks faster than those with lower capacities. This is another aspect of our future work. Also intend to improve and evaluate our mechanism in an open environment where agents can join and leave freely and new types of resources may be dynamically introduced into the environment.

REFERENCES

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

- [2] H. Czap, R. Unland, and C. Branki, *Self-Organization and Autonomic Informatics*. IOS Press, 2005.
- [3] A. Reitbauer, A. Battino, A. Karageorgos, N. Mehandjiev, P. Valckenaers, and B. Saint-Germain, "The Mabe Middleware: Extending Multi-Agent Systems to Enable Open Business Collaboration," *Proc. 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.
- [7] G.D.M. Serugendo, M.-P. Gleizes, and A. Karageorgos, "Self-Organization in Multi-Agent Systems," *The Knowledge Eng. Rev.*, vol. 20, no. 2, pp. 165-189, 2005.
- [8] J.P. Jamont, M. Occello, and A. Lagreze, "A Multiagent Approach to Manage Communication in Wireless Instrumentation Systems," *Measurement*, vol. 43, pp. 489-503, 2010.
- [9] R. Kota, N. Gibbins, and N. Jennings, "Decentralized Approaches for Self-Adaptation in Agent Organizations," *ACM Trans. Autonomous and Adaptive Systems*, vol. 7, pp. 1-28, Apr. 2012.
- [10] S.J. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*. Prentice Hall, 2003.
- [11] P. Mathieu, J.-C. Routier, and Y. Secq, "Principles for Dynamic Multi-Agent Organizations," *Proc. Fifth Pacific Rim Int'l Workshop Multi Agents: Intelligent Agents and Multi-Agent Systems (PRIMA '02)*, pp. 109-122, Aug. 2002.
- [12] R. Frei, B. Ferreira, G.D.M. Serugendo, and J. Barata, "An Architecture for Self-Managing Evolvable Assembly Systems," *Proc. IEEE Int'l Conf. Systems, Man and Cybernetics*, pp. 2707-2712, 2009.
- [13] C. Zhang, V. Lesser, and P. Shenoy, "A Multi-Agent Learning Approach to Online Distributed Resource Allocation," *Proc. 21st Int'l Joint Conf. Artificial Intelligence (IJCAI '09)*, pp. 361-366, July 2009.
- [14] E.R. Gomes and R. Kowalczyk, "Dynamic Analysis of Multiagent Q-Learning with E-Greedy Exploration," *Proc. 26th Ann. Int'l Conf. Machine Learning (ICML '09)*, 2009.
- [15] S. Kamboj, "Analyzing the Tradeoffs between Breakup and Cloning in the Context of Organizational Self-Design," *Proc. Eighth Int'l Conf. Autonomous Agents and Multiagent Systems (AAMAS '09)*, pp. 829-836, May 2009.

ABOUT AUTHOR



K. Rekha, Completed her B.Tech. (IT) degree in the year 2012. Currently she is pursuing M.E (CSE) at Krishnasamy College of Engineering & Technology, Cuddalore, Tamil Nadu, India. Her research areas are Mobile Computing, Cloud Computing, and Networking, especially in parallel and Distributed Systems. She had attended many workshops also interested in attending seminars and conferences in various technologies.



K. Priya, Completed her B.E. degree from Madras University, M. Tech degree from SRM University Chennai. Currently she is working as a Assistant professor in Computer Science and Engineering at Krishnasamy College of Engineering & Technology, Cuddalore, Tamil Nadu, India. Her area of interest include Image Processing, Network Security, and Compression Techniques. She had presented research papers in National/ International conferences