Stroke economy approach for constructing complex expert system using multi agent architecture.

U.Chandrasekhar[1], S. Sahithya[2]

[1,2]School of Information Technology and Engineering
VIT University, Vellore -632014

Tamil Nadu, India

***Abstract*— In a highly competitive and ever evolving dynamic environment the need for an effective and a highly efficient work force that can reduce various overheads, is massive. This is where a distributed, multi agent architecture comes into play. In this paper, we propose a decentralized, multi agent architecture that works in a distributed and dynamic environment. It uses stroke economy approach to optimize itself and evolve into an expert system of experts. The aim is to construct a unified mind from several minds, each expert in a sub domain. The goal is to achieve variety in task handling approaches and quality output, while optimizing task scheduling process. We also present a learning mechanism which helps individual agents to lift themselves up to the level of more experienced agents, in their sub domain. We also aim to minimize the size of system, initially commissioned for that application, optimize transfer of data and schedule tasks effectively, promising a high quality performance.**

***Keywords*— Multi agent architecture, Distributed AI, Subsumption architecture, Reinforcement learning**

I. Introduction

Artificial Intelligence is a branch of science that targets at modelling life. Researchers have concentrated on working on subsections of AI problems which give rise to domain specific architectures. However, in real world scenarios, mind does not build itself as a complete application [1]. It builds in pieces and evolves along with time [Brooks, 1991]. In Animats approach [Dennett, 1987], life starts with simple self sufficient creatures, which slowly work up to grow into complex, whole creatures. Evolutionary approach is another field designed using Genetic algorithms. It believes complex systems evolve in subsequent generations [Harvey et al., 1992]

1.1 Distributed Multi agent expert systems

There is active work going on, to build expert systems working in distributed fashion [4]. Decentralized multi agent expert systems are useful in areas like medical diagnosis, surveillance etc. This paper presents subsumption architecture as shown in Fig.1, to construct a complex domain expert using multiple, decentralized, expert agents.

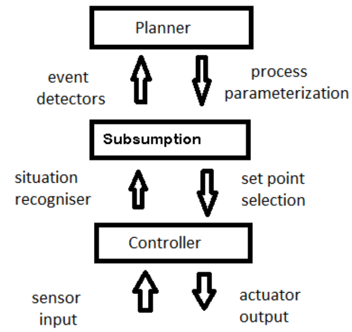

Fig. 1 A sample diagram to visualize subsumption architecture

Each agent is self sufficient in itself, to solve a basic task related to the domain. An expert system evolves into a higher mind, through continuous learning and gaining experience. A multi agent approach serves tasks that are distributed and also maximizes performance, through task scheduling among agents of varying expertise. We have various architectures such as centralized, hierarchical to decentralized and lateral. Here we discuss a decentralized approach to ensure robustness of whole system. Individual agents communicate with each other only within clusters of domain or sub domain expertise, as shown in Fig. 6.

1.2 The Subsumption Architecture

Brooks referred to this as “The world is its own best model”. This architecture has following properties.

1. Behaviour is distributed
2. Response to stimuli is not modulated by cognition. It is Reflexive.
3. The agents are organized in bottom up fashion. Complex behaviours or highest levels of expertise are fashioned by combining or reinforcing underlying lower ones.
4. So the set of lower levels still work even if the higher experts are removed. However some expertise gained is lost, if the knowledge was not yet passed to any other agent
5. Similarly removing lower levels also does not affect quality of output. However the agents better equipped with solving high end tasks may be kept busy with low end tasks.
6. It is parallel and asynchronous.

II. Stroke Economy

A stroke is defined as a unit of "Recognition". Famous phycologist Eric Bern coined this term to describe how human emotional system is boosted through stroke mechanism. In our work we define three categories of stroking.

A. Positive stroke

Multi agent architecture is comparable to human society, in terms of communication, sharing knowledge, co-operation, scheduling tasks and optimizing performance. An agent receiving a feed back from a client is called reinforcement [5]. However, based on the level of expertise and the amount of learning, an agent is bound to receive importance from its colleagues that would complement its performance. This is called *Positive Stroke.* This expert agent, if idle, is given priority amongst the other agents, in its expertise cluster to handle most recent task arriving. This will ensure that the expert agent expands its KB with additional learning and client gets the best quality output possible. Each expert is in constant endeavour to attain tasks that will help in increasing its sub domain expertise. With every task allocation from its colleagues, an expert will increment its rank, as shown in Fig.2. Hence every agent is associated with a rank, which specifies its expertise in the sub domain.

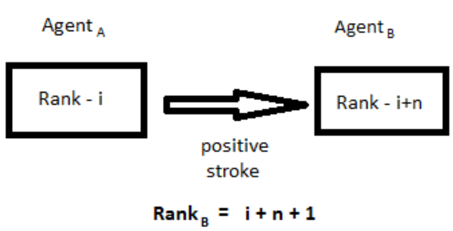

Fig. 2 Rank of Agent B improves from a positive stroke

B. Negative stroke

When a high end expert in a sub domain cluster is busy handling another task, the new incoming task is allotted to an expert with lower rank. This agent, after completing the task submits its output for review, to other agents of higher rank in the sub cluster. During the peer review process, it may receive a few negative strokes, as shown in Fig.3. This internal review ensures that external client gets highest possible satisfaction. Depending on the worth of the negative strokes and positive strokes received, the rank of an agent could drop to lower value. However the agent gets a chance to improve, by enhancing its rule set.

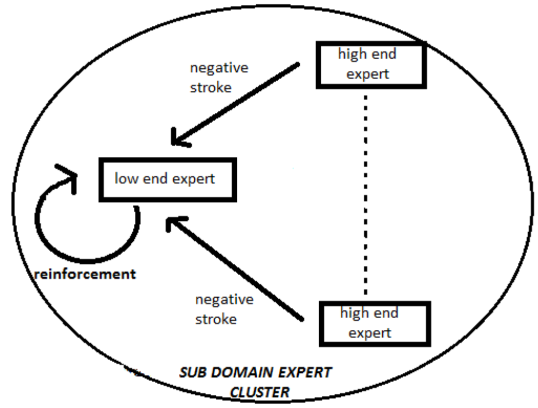

Fig. 3 Lower Rank expert receives negative strokes from its seniors. It reinforces itself with feedback received.

C. No Stroke

The agents are spread in a vast problem space. So it is possible that few agents rarely or never get a chance to work on any task. Such agents are called stroke deprived. There could be various reasons why certain agents do not receive sufficient work. Ex: Due to low stimuli in the region where the agent is deployed, due to lower work load, due to insufficient expertise to handle a task or communication overhead to transfer tasks to this agent from other agents etc. In such scenarios stroke deprived agents may either ask for strokes from other agents through a teaching-learning process, as shown in Fig. 4, or decommission themselves from the network to save energy and communication overhead. Decommission process has to happen after a prolonged wait, since work load can be expected sometime in the future. Hence this is a self optimizing architecture, which will ensure that only optimum number of agents are commissioned in a given task environment, through a process of right sizing.

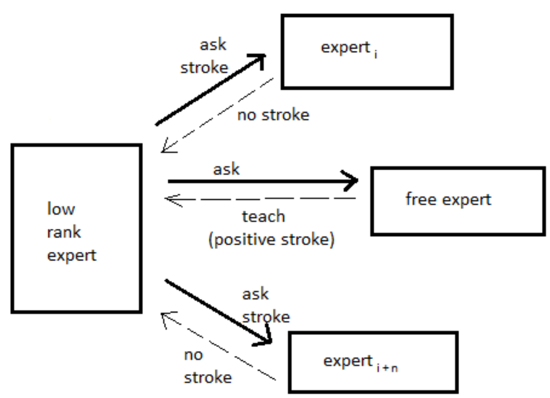

Fig. 4 Idle, stroke deprived agents ask from strokes from their idle colleges. At times they may receive a positive stroke from some agent.

III. LEARNING PROCESS

A basic level of learning happens from reinforcement received from client. Apart from that there is knowledge sharing happening among agents of a sub cluster first and later among agents of inter disciplinary domains. Two approaches are found for this. Sharing knowledge by using a distributed black board or by using multiple black boards (one per agent or sub environment) [3]. In out model each agent will contain the knowledge needed to analyse an event. Robustness is guaranteed by having a local black board copy with each agent.

In the beginning, all the agents look alike when deployed in the task environment. Depending on opportunities and positive strokes they receive, each agent is modelled into an expert and in the process creates a sub domain expert cluster. *The Aim* of each expert agent hence forth is, to attain further expertise in that sub domain. All the agents who have worked on similar tasks become part of that sub cluster. The goal of this model is to provide opportunities to its experts and groom them further. As the time goes on, whole model constructs itself in to a dependable expert system, able to handle variety of tasks, ensuring maximum collective happiness.

There is a constant process of learning happening between two idle agents of differing ranks, as shown in Fig 5. A lower rank agent, when idle, waits for any of higher rank agent in the sub domain expertise cluster, to become idle. There follows a mutual sharing of KB among these two agents. This learning process is similar to learning happening in humans. Knowledge is never replaced with knowledge from better ranking agent. It's always supplemented and is mutual. Though an agent is of lower rank, it cannot be compared with a higher ranking agent, since rank is only a measure of experience and quality of its past success. The inspiration is to see if competition be resolved between agents that had little in common. After sharing knowledge, the lower rank agent improves its rank to that of its senior.

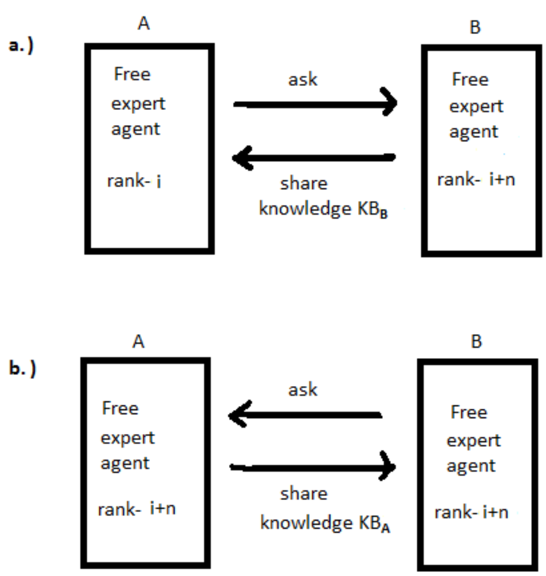

Fig. 5 Both agents share the additional Knowledge they acquired.

Agents deprived of strokes wait for any higher rank agent in their vicinity to become idle. Such agents have fewer choices, to find place in a sub cluster of their interest. Ex: An agent may have some expertise in handling a particular sub domain task. However if it is unemployed or does not find a find a relevant teacher, it will have to compromise and learn from who soever is willing to share their Knowledge with it. We termed it as positive stroke. A stroke, positive or negative, ensures a longer life for an agent, since it could be engaged with a higher end task, any time in near future.

IV. MULTI AGENT ARCHITECTURE

The multi agent architecture uses hierarchical clustering. It gives a multi modal functionality, since each agent is vying to grow into an autonomous expert in a sub domain of problem space. Agents update their expertise levels just the way humans do in a society. Each mind is mutually exclusive of another mind. Yet they are ranked in their area of expertise. Teaching - learning process takes place first among similar expert group and only later, inter disciplinary knowledge sharing gets priority. This method will ensure that there is no overhead of memory transfers and at the same time the whole system is ever growing into a robust, reliable and complex expert system. In other cases, data transfers are only limited to message passing, task scheduling among sub clusters and communicating with clients for reinforcement.

Fig 6. Shows agents form sub clusters of specific domain expertise and how communication takes place between them. A task from client will be accepted by any agent closest to it. The task is passed to a mediator agent inside its sub cluster, which takes care of task scheduling.

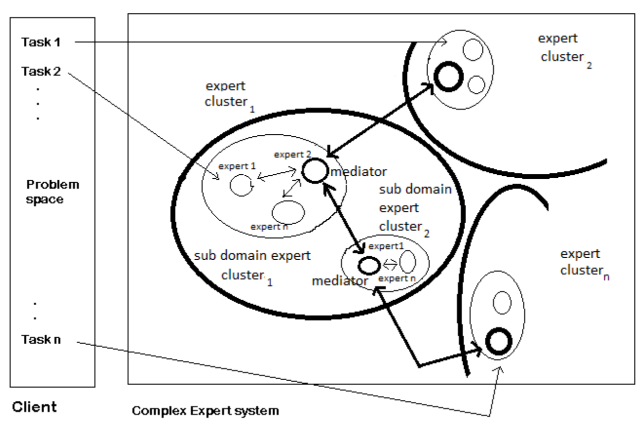

Fig. 6 Expertise based hierarchical clustering among agents.

Since this is a decentralized architecture, the agents elect local leader or mediator for jobs like message passing, task scheduling within cluster, communicating with mediators of other sub clusters, maintaining knowledge of domain expertise of other clusters and information about any idle, stroke deprived agents etc. Idle agents are waiting outside clusters or sub clusters, and will be given a task when no relevant agent is currently idle.

The paper aims at presenting high level design of stroke economy approach. Hence lower level design details like electing other kinds of agents like control agents, transport agents etc [2] are abstracted as mediator agent. Following are the functions of mediator.

1. It receives details of tasks sensed by its colleagues.
2. It assesses sub domain expertise required for a task and passes the task to relevant mediator of its sibling cluster.
3. It receives a task from its sibling mediator and assesses to which sub cluster or agent, this task needs to be allotted.
4. Task is prioritised and allotted such a way so as not to keep any higher rank agents idle. (Design of this paper does not go deeper into issues of look ahead [6], where highest rank experts may need to be temporarily kept idle, waiting to be allotted a task fitting its expertise. Jobs requiring lesser expertise may be scheduled to idle agents, possessing sufficient expertise).
5. Sense requests for positive strokes (for learning) from idle agents and identify idle experts good enough to teach them.
6. Identify need to split or merge clusters and monitor that process.
7. Coordinate peer review process when a high end task was allotted to a lower rank expert, due to non availability of higher rank experts.
8. Keep highest rank experts busy with challenging tasks. At the same time ensure a balance to provide an opportunity for lower rank agents to grow.
9. Maintain an updated record of ranks of every agent in its sub cluster.
10. Maximize global happiness through effective and efficient task – agent selection strategy.

V. Conclusions

This paper has described need to decentralise work in AI domain. This will ensure agents to specialise on different aspects, and a complex mind could be constructed from these multiple specialist agents. As we see in a human society, no single person is equipped with expertise in every field. He is constantly looking forward for challenges that will improve his domain knowledge. In this paper we discussed how the above said model could be evolved through stroke economy. Positive strokes ensure emotional satisfaction and there by improve expertise. Negative strokes, though sometimes deteriorate ranks of an individual, still play a crucial role in uplifting him by proper reinforcement. Stroke deprivation may kill an agent. Yet it is useful in right sizing the system population. Also stroke deprived minds are in constant endeavour to request knowledge sharing from their peers and indirectly struggle to uplift their quality. Hence the system is ready for any task overhead in the future.

References

[1] Mark Humphrys and Ciaran O'Leary, "Constructing complex minds through multiple authors", 7tn International conference, SAB'02, August -2002

[2] Munir Merdan, Vedran Kordic, Alois Zoitl & Aleksandar Lazinica,"Knowledge-based Multi-agent Architecture", IEEE (CIMCA-IAWTIC'06).

[3] D. Vallejo, J.Albusac, J.J.Castro-Schez, C.Glez-Morcillo, L.Jime´nez, "A multi-agent architecture for supporting distributed normality-based Intelligent surveillance", Engineering Applications of Artificial Intelligence 24(2011)325–340.

[4] Yuanda Cao,Nutao Jiang, Qing Zhou, "A Distributed Multi-agent Cooperative Expert System Tool",Proceedings of ICSP '98.

[5] Ming Tan, "Multi-Agent Reinforcement Learning: Independent vs. Cooperative Agents", Proceedings of 10th International Conference on Machine Learning. 330-337.

[6] Jyi-Shane Liu, Katia P.Sycara, "Multiagent Coordination in Tightly Coupled Task Scheduling", Proceedings of 1st International conference on Multi Agent systems, 1996. American Association for AI ,181-188