Multi Sensor Information Fusion: Distributed Agents-Based Paradigm

V.K.Murthy¹, E.V.Krishnamurthy²

Abstract- This paper describes a distributed agent-based paradigm, for hybrid (soft and hard) computation involved in multisensor information (data) fusion. The computations are interpreted as the outcome arising out of deterministic, nondeterministic or stochastic interaction among the agents, in an environment containing multiple sensors. These interactions are like chemical reactions and the evolution of the multiset of agents can mimic the data fusion arising in a complex sensory system to achieve a required outcome e.g. nature inspired computations and active walker (swarm - intelligence, battlefield tactics) models. Since the reaction rules are inherently parallel, any number of actions can be performed cooperatively or competitively among the subsets of the agents. We also describe how scale-free and small world networks can arise in the connectivity structure of agents during the information fusion.

Index Terms-Agents, Assortative- Dissassortative mixing, Conscious systems, Information fusion, Multi-agent tool kits, Random - Scale-free- Small world network.

I. INTRODUCTION

Fusion of information obtained from hard and soft computing , Krishnamurthy and Murthy [9] plays an important role in multisensor applications, Ovaska and Sick [12], Ovaska [13],Hall and Llinas [7] and in the design of Swarm-Bot,Dorigo et al.[4]. In this paper we describe a distributed agent -paradigm for realising the soft and hard computations involved in multi sensor information fusion.

The multi-agent paradigm for Information (data) fusion (MAIF) proposed in this paper consists of the following features:

(i) A multiset that contains agents (called the agent-space) whose information is structured in an appropriate way to suit the problem at hand

(ii) A set of interaction rules that prescribes the context for the applicability of the rules to the agents. Each rule consists of a left-hand side (a pattern or property or attribute) describing the conditions under which the agents can communicate and interact, and a right hand side describes the actions to be performed by the agents, if the rule becomes applicable, based on some deterministic or probabilistic criteria.

(iii) A control strategy that specifies the manner in which the agents will be chosen and interaction rules will be applied, the kinetics of the rule- interference (inhibition, activation, diffusion, chemotaxis) and a way of resolving conflicts that may arise when several rules match at once.

¹ Colleges of Applied Sciences, Ministry of Higher Education, Sultanate of Oman, and Australian National University, Canberra, ACT 0200, Australia edayathuk@gmail.com and ariyalurk@gmail.com.

²Australian National University, Canberra, ACT Australia,

Evk.Krishnamurthy@anu.edu.au

(iv) A coordinating agent evaluates the performance of the agents to determine the effectiveness of rule application.

This agent ensures that the contract among the different agents hold; if the contract fails, the coordinator can rescue, abort or restart as in i-Contract or in Eiffel.

The MAIF is applicable to physical, chemical, biological and agent-based computational problems, since it has the following computational features:

(i) Interaction -Based: The computations are interpreted as the outcome of interacting agents to produce new agents (or same agents with modified attributes) according to specific rules. Hence the intrinsic (genotype) and acquired properties due to interaction (phenotype) can both be incorporated in the agent space. Since the interaction rules are inherently parallel, any number of actions can be performed *cooperatively or competitively* among the subsets of agents, so that the new agents evolve toward an equilibrium or unstable or chaotic state.

(ii) Content-based activation of rules: The next set of rules to be invoked is determined solely by the contents of the agent-space, as in the context of chemical reactions.

(iii) Pattern matching: Search takes place to bind the variables in such a way to satisfy the left hand side of the rule. It is this characteristic of pattern (or attribute) matching that gives the agent-based paradigm its distinctive capabilities for innovative computing.

(iv) Suitable for deterministic, non-deterministic, fuzzy and probabilistic evolutionary modes:

This paper is organized as follows: In Sections II and III, general properties multi-agent systems are described. Section IV deals with the computational aspects MAIF. Section V we describe the connectivity structure that can arise among many sensory agents during information fusion process. Section VI deals with swarm dynamics- a simple and efficient information fusion strategy used by nature. Section VII briefly describes the currently available software-agent tools. Section VIII is the conclusion.

II. MULTI-AGENT SYSTEM

The AOIS (agent oriented information system community) defines an agent as a system that is capable of perceiving events in its environment, or representing information about the current state of affairs and of acting in its environment guided by perceptions and stored information. The multiagent system consists of the following single agent-system, Fig.1. Thus whenever several agents N are involved $i = 1,2,3, \dots$ N then each of the agents will be denoted with a label (i), Woolridge [16].



Figure 1

(1) Worldly states or environment U:

Those states which completely describe the universe containing all the agents.

(2) Percept: Depending upon the sensory capabilities (input interface to the universe or environment) an agent can receive from U an input T (a standard set of messages), using a sensory function Perception (PERCEPT): PERCEPT : $U \rightarrow T$. PERCEPT can involve various types of sensory perception: see, read, hear, smell. The messages are assumed to be of standard types based on an interaction language that is interpreted identically by all agents. Since U includes both the environment and other agents the input can be either from the agents directly or from the environment that has been modified by other agents. Thus we can deal with agents that can communicate directly, as well as, indirectly through the environment as in active walker model; e.g., "stigmergy" in Ant colony where each ant can modify the environment and affect the behaviour of another ant (see also EFFECT).

(3) Epistemic states or Mind M:

We assume that the agent has a mind M (that is essentially a problem domain knowledge consisting of an internal database for the problem domain data and a set of problem domain rules) that can be clearly understood by the agent without involving any sensory function. The database D sentences are in first order predicate calculus (also known as extensional database) and agents mental actions are viewed as inferences arising from the associated rules that result in an intentional database, that changes (revises or updates) D.

An ordered pair of elements (D, P) represents the agent's state of belief at a certain time. Here, D is a set of beliefs about objects, their attributes and relationships stored as an internal database and P is a set of rules expressed as preconditions and consequences (conditions and actions). When T is input, if the conditions given in the left-hand side of P match T the elements from D that correspond to the right-hand side are taken from D and suitable actions are carried out locally (in M) as well as on the environment. The nature of internal production rules P, mode of application and the action set determines whether an agent is deterministic, nondeterministic, probabilistic or fuzzy. Rule application policy in a production system P can be modified by:

(1) Assigning probabilities/fuzziness to apply the rule

(2) Assigning strength to a rule using a measure of its past success

(3) Introducing a support for a rule by using a measure of its likely relevance to the current situation.

The above three factors provide for competition and cooperation among the different rules. Such a model is useful for many applications, including emergence and self organization among many agents. If the choice of the rules and corresponding actions are deterministic we, have a

deterministic system suitable for hard computation. If, however, there are several competing choices, a nondeterministic choice or a probabilistic choice of the rules is made and the corresponding actions are carried out; then the agent is nondeterministic or stochastic and is suitable for soft computation in Complex Systems (e.g., Active-walker, Selforganization and swarm models). Also, we assume that each agent can carry out basic computations- having memory, simple addition, comparison, control rules and the generation of random numbers. Such mechanisms can help us simulate tumbling, as well as running of organisms for foraging.

(4) Organizational Knowledge (O): Since each agent needs to communicate with the external world or other agents, we assume that O contains all the information about the relationships among the different agents. For example, the connectivity relationship for communication, the data dependencies between agents, interference among agents with respect to rules. Information about the location of different domain rules are in O.

(5) INTRAN:

On the receipt of T, the action in the agent M is suitably revised or updated by the function called Internal transaction (INTRAN).

Revision: means acquisition of new information about the environment; it requires a change in the rule system P. This may result in changes in the database D.

Example: Inclusion of a new tax-rule in Tax system.

Update: means adding new entries to the database D; the rules P **are not changed**.

Example: Inclusion of a new tax -payer in Tax system.

Both revision and update can be denoted in set-theoretic notation by:INTRAN: M X $T \rightarrow M(D,P)$

T can be interpreted as a transaction for updating or revising a set of database instances. Hence, if one or several interaction conditions hold for several non- disjoint subsets of objects in the agent at the same time, the choice made among them can be nondeterministic or probabilistic. This leads to competitive parallelism. The actions on the chosen subset are executed atomically and committed. In other words, the chosen subset undergoes an 'asynchronous atomic update'. This ensures that the process of matching and the follow-up actions satisfy the four important properties used in Transaction Processing, namely, ACID properties: Atomicity (indivisibility and either all or no actions or carried out), Consistency (before and after the execution of a transaction), Isolation (no interference among the actions), Durability (no failure). Once all the actions are carried out and committed the next set of conditions are considered.

As a result of the actions followed by commitment, we may revise or update and obtain a new database for each agent; this may satisfy new conditions of the text and the actions are repeated by initiating a new set of transactions. These set of transformations halt when there are no more transactions executable or the databases does not undergo a change for two consecutive steps indicating a new consistent state of the databases.

However, if the interaction condition holds for several disjoint subsets of elements in the database at the same time, the actions can take place independently and simultaneously. This leads to *cooperative parallelism*.

(6) **EXTRAN:** External action is defined through a function called global or external transaction (EXTRAN) that maps an

epistemic state and a partition from an external state into an (i) Autonomy: Make decisions on actions they want to do action performed by the agent. That is: EXTRAN:

M X T \rightarrow A. This means that the current state of mind and a (ii) *Reactive*:Respond appropriately depending upon the new input activates an external action from the action set A.

(7) EFFECT: The agent also can affect U by performing an (iii) Proactive: Act in anticipation of future goals to meet the action from a set of actions A (ask, tell, hear, read, write, specified objectives. speak, send, smell, taste, receive, silent), or more complex actions. Such actions are carried out according to a particular of events, signals and conditions that are often distributed agent's role and governed by an etiquette called protocols. The effect of these actions is defined by a function EFFECT that modifies the world states through the actions of an agent: EFFECT: A X U \rightarrow U; EFFECT can involve additions, deletions and modifications to U. Thus the agent paradigm created and destroyed. provides for simulating the active walker model. Thus an The fusion process is sensitive to the order of events. In order agent is defined by a set of nine entities, called a 9-tuple:

(U,T,M(P,D),O,A,PERCEPT,INTRAN,EXTRAN, EFFECT). The interpreter repeatedly executes selected rules in P, until concurrently. This offers the possibility of carrying out parts no rule can be fired.

Finite state machine or a Turing machine) and parallel possibilities would require the analysis of the rules as to how computational models as subclasses of the agents, by suitably formulating the definitions.

III. MULTIAGENT COMPUTATION

A multi-agent system can be defined as a loosely coupled network of agents that interact among them and through the environment to solve a problem. Operationally, the multiagent system carries out distributed computation by sending, receiving, handshaking and acknowledging messages and performing some local computations and has the following features:

1. An agent has the structure as described in Figure 1.

2. There is a seeding agent who initiates the computation process.

3. Each agent can be active or inactive.

4. Initially all agents are inactive except for a specified seeding agent that initiates the computation.

5. An active agent can do local computation, send and receive messages and can spontaneously become inactive.

6. An inactive agent becomes active if and only if it receives a message.

7. Each agent may retain its current belief or revise its belief as a result of receiving a new message by performing a local computation. If it revises its belief, it communicates its revised state of belief to other concerned agents; else it does not revise its solution and remains silent.

Hence the basic agent model can realise:

(i) Reactive agent that make decisions at run time with a limited amount of information,

(ii) Deliberating agent that has an internal representation of the environment and has a logical inference mechanism for decision making and planning and

(iii) Interacting agent that is capable of coordinating the activities with other agents through communication and negotiation.

IV. INFORMATION FUSION

Three crucial properties of Agents make them suitable for the multi sensor information fusion:

without explicit control from the user,

context, and

In reactive fusion, the system has to react to various kinds and concurrent. Also they can be time critical exhibiting both digital and analog (or hybrid) behaviour. In addition the reactive system, as in cell biological system can contain components that signal each other and also repeatedly

to speed up the use of the multi-agent fusion paradigm we need to consider how to permit multiple agent execution or all of computations in parallel on distinct processors or We can interpret all the abstract machine models (such as a performing multiple-sensory functions simultaneously. Such the rules interfere with There are four ways in which such interference can take place, Murthy and Krishnamurthy [10].

> 1. Enabling dependence (ED): Agent A(i) and agent A(j) are called enable dependent (or dataflow dependent) if the messages from A (i) creates the required precondition in A(j) to carry out a specific action .

> 2. Inhibit dependence (ID): Agents A (i) and A (j) are called inhibit dependent, if the actions of A (i) creates the required precondition in A(j) to prevent it from executing a specific action.

> 3. INTRAN Conflict (IC) : Agents A (i) and A (j) are opposition dependent (also called data-output dependent) through A(k), if the order in which A (i) and A (j) enable A(k) and update A(k) produce different results in A(k); that is the objects A(i) and A (j) perform operations on A(k) that are not order reversible. That is, local serializability is not ensured in the INTRAN within A(k), if the actions are carried out within an agent in different order.

> 4. EXTRAN Conflict (EC): Agents A (i) and A(j) are data antidependent through A(k) if the order in which A(i) enables (inhibits) A(k), and A(j) enables (inhibits) A(k) result in different external actions (EXTRAN) by A(k) on the environment. That is the order in which information arrives from the environment and other agents affects the global serializability of the actions of an agent.

Remark: ED and ID:

The two properties ED and ID are crucial for modelling any sensory system which requires both positive and negative regulation These rules permit an agent to enable itself and also an agent A(i) to enable A(j) and A (j) to enable A(i) cyclically.

For example, A(i) can create the required precondition in A(k), so that A(j) can enable A(k). Also, A(i) can inhibit the required precondition in A(k) so that A(j) is prevented from enabling A(k).

A. Concurrency and Conflicts

In distributed computing and transaction processing: we require that the following two conditions are satisfied for global serialization when concurrent operations take place.

1. At each agent the actions in local actions are performed in the non-conflicting order (Local serializability).

2. At each agent the serialization order of the tasks dictated by every other agent is not violated. That is, for each pair of conflicting actions among transactions p and q, an action of p precedes an action of q in any local schedule, if and only if, the preconditions required for p do not conflict with those preconditions required for execution of the action q in the required ordering of **all tasks in all agents (Global serializability)**.

The above two conditions require that the preconditions for actions in different agents A(i) and A(j) do not interfere or cause conflicts. These conditions are necessary for the stabilization of the multi-agent systems so that the computations are locally and globally consistent.

Termination: For the termination of agent –based program, the interaction among the agents must come to a halt. of agents. Then we have an equilibrium state (or a fixed point). **Non-termination, instability, multiple equilibria and chaos:** These cases arise when the agents continue to interact indefinitely as in chemical oscillations, biological reactions, and sensory signal processing. Then the multiagent-system is sensitive to initial conditions leading to chaos having strange attractors and self-organization.

Conflicts: Resolution or compromise?

The conflicts arising in INTRAN and EXTRAN require resolution or compromise. e.g, the actions, may need a compromise, or a blending of the behaviour of actions if the quantitative parameters can be suitably averaged over. These rules should be based on the context.

Vector, Pipeline and Data Parallelism

1. Vector parallelism: If all the agents are independent then we can apply all the rules simultaneously, e.g., a vector addition.

2. Pipeline parallelism: Here multiple agents enable each other and passing data in a pipeline fashion - e.g. multienzyme reactions, where at each membrane an "imprisoned" enzyme performs a given operation and then sends it on to the next stage.

3. Data parallelism: Multiple identical agents are activated in parallel based on distinct data- e.g., fusing multiple sensory infOrmation.

B. Advantages

(i) **Multi Agent Information Fusion** (MAIF), starts with simple rules of interaction among the individual components that drive the system to the complex behaviour observed. It works bottom up by examining what low-level rules and what kind of heterogeneous, autonomous agents are required to synthesize the required higher level behaviour. Thus, MAIF is useful for realising, Evolutionary algorithms, Genetic algorithms, Genetic Programming , Immuno-computing, Self-organized criticality, Active walker models where each walker (e.g., ants with scent) can influence (repel or attract) other walkers using a shared landscape.

(ii) Agent-based fusion enables us to make predictions about the processes occurring at the intermediate mesoscopic scale due to the interplay between the microscopic dynamics and the macroscopic environment.

(iii) The fusion of deterministic, chaotic/ stochastic systems are possible.

(iv) Different models where iterated application of simple rules can generate complex forms can be studied.

(v) Agent-based hybrid fusion (for soft and hard computation) can be used to study the global behaviour of a complex adaptive system from the local interactive behaviour of its components.

C. Special Purpose Agents in Fusion

Agents can be designed to realize special purpose functionalities that depend on the problem domain. For example, collaborative mobile agents that migrate among hosts to enhance efficiency of computation and improve the network throughput; information agents that manage, manipulate and collate information from many distributed sources; reactive agents that respond to stimulus and respond in an environment where they are embedded; smart agents that learn from their actions, skill agents that can build composable, blendable behaviours.

V. CONNECTIVITY PATTERNS IN FUSION

In multi sensor fusion the information arrival is nondeterministic, fuzzy or probabilistic. The communication or interconnection patterns among the agents play a key role for applications to various fusion aspects. The fusion agents therefore modify the pattern of their communication pathways, namely, the topology and geometry at will. Here we need to study the Graph model to analyse the connectivity structure among the agents in a network arising from cooperative and competitive interactions.

Three important statistical properties of the networks, namely average degree, characteristic path length and cluster coefficient, to be defined below are used as measures to distinguish the disordered networks from regular networks. These are:(i) Random networks (ii) Power-law scaling networks, and (iii) Small World Networks, Watt [15]. For a survey of complex networks, Newman [11], Chung and Lu [3].

Let us consider a finite graph G(V,E) where V is the set of n nodes and E the set of edges. Let us assume that the graph is represented as an adjacency matrix A with elements A(i,j) = 1, if there is an edge from node i to node j; and A(i,j)=0, otherwise. We assume A(i,i) = 0, that is no self loops.

The following parameters are derived from adjacency matrix:

(i) Average degree:
$$K = 1/n \sum_{i} k(i)$$
, and $k(i) = \sum_{i}^{n} A(i,j)$, or $k(i)$ is the degree of node, $0 \le K \le (n-1)$

(ii)The *Characteristic path length* L measures the global property, namely, the average path length of the network. Given L(i,j) the shortest distance between nodes i and j, L is defined by:

L= 2/n(n-1)
$$\sum_{i=1}^{n-1} \sum_{j=i+1}^{n} L(i, j)$$
; $1 \le L \le (n-1)$

This is referred to as "global connectivity" in the context of lattice percolation, if we need to only infer whether they are connected or not. Thus the notion of percolation in lattice grid is closely related to the small-world and scale-free networks.

(iii) *The cluster coefficient* C is the average of C(i), where C(i) is defined by:

i=1

$$C(i) = 2\sum_{j=1}^{n}\sum_{k=1}^{n}A(i,j)A(i,k)A(k,j)/[k(i)k(i) - 1] =$$

Number of E(i) existing between k(i) neighbours of node i /Total number of possible edges [k(i)(k(i)-1]/2 .or, C=1/n $\sum C(i)$ Note that $0 \le C \le 1$.

(i) Random Network:

In Random network the degree distribution is a binomial or Poisson distribution in the limit of a large graph. Most of the nodes have the average degree and few nodes have more than average and few nodes have less than the average degree. Also L and C are small in random graphs.

(ii)Scale -free Network:

In this network, many nodes have a low degree (few links) and a few nodes have a high degree (many links). The distribution of the degree of the nodes has an unusual fattailed form or a power-law scaling property: namely the P(k) the degree distribution of a network is given by: $P(k) = k^{-g}$ where 2 < g < 3.

This power-law degree distribution or scale-invariant property arises from two kinds of operations on a random graph, Barabasi et al. [1]. It has been experimentally observed that Chung and Lu [2006] the biological networks have a range 1 \leq g < 2.5 and the social networks have a range 2 \leq g<3:

1. *Creating new nodes*: Growth of the graph by adding new nodes into an initial group of nodes as time progresses and

2. Preferential attachment of Links: The new nodes created are linked to old nodes, with a probability based on certain dominant properties the old nodes possess, e.g. the nodes having a higher degree (or attractiveness), chemical or physical interaction strength. In each case, the neighbourhood is appropriately defined as a conceptual graph. As the network grows the ratio of well-connected nodes to the number of nodes in the rest of the network remains nearly a constant, Dorogovtsev et al. [5] prove that the range 2 < g < 3 is crucial to have the following properties:

(a) Self-organization and (b) Resilience against random damage. Also g is related to the fractal dimension; it has been shown that for networks such as: www, Actor, E *coli*, 2 < g < 3.

(iii)Small-world graphs:

A graph is called a small -world graph, by Watts [15], if it exhibits the following two properties (compared to a random graph of same number of nodes and average degree):

1. *Higher clustering coefficient C closer to unity:* this implies that two nodes are more likely to be adjacent, if they share a common neighbour and

2. Smaller average distance L between any two nodes:

L scales logarithmically with the number of nodes. This measures a global property.

This is called the small world effect .In agent based systems, where a very large number of agents are interconnected, small-world network, permits distant neighbours to interact.

A. Assortative / Disassortative Mixing

In some networks, the high degree nodes are connected to high degree nodes. These are called assortative or homophilic networks. In disassortative networks, high degree nodes avoid being connected to high degree nodes. These two types of networks are distingushed by using a degree-correlation coefficient that is positive for assortative networks and negative for disassortative networks, Newman [11]. The assortative mixing results in larger positive Lyapunov exponents (eigenvalues) of the interacting matrix of the dynamical system. This means the system can quickly become unstable resulting in the formation of giant components in graph networks or the phenomenon of percolation in a lattice.

In disassortative mixing high degree nodes avoid being connected to high degree nodes and result in a smaller positive Lyapunov exponent (or positive eigenvalues) and hence the dynamical fluctuation are not amplified and the system can reach stability more quickly.

Assortative mixing is more prevalent among social networks, while disassortative networks is common in biological networks, Newman [11]. Assortative networks are less stable to random fluctuations leading to percolation like phenomena, while disassortative networks are more stable to fluctuations. Biological systems seem to self-organize themselves into assortative or disassortative networks according to their need to adapt themselves to their environment.

VI. SWARMING STRATEGY

Swarming strategy refers to a kind of problem solving ability that is inspired by Nature. The swarm phenomenon is exhibited by ants , bees , flocking of birds, and a school of fish, where a large group of agents carry out a desired task (e.g. foraging) that cannot otherwise be done individually, by communicating and interacting among them and their environment through various means; Edwards [2000], Bonabeau et al [2], Dorigo et al [4], Kennedy and Eberhart [8]. Swarming is a self-organization process that enables a large group of agents to carry out a task that cannot be done individually by autonomously switching connections among them, Murthy and Krishnamurthy [10].

Swarming for foreaging requires four basic steps: *Locate* a target through sensing, *Reach* the target through coordinated motion and path formation (Self -assembly), *Carry* the food through cooperative action, *Disperse* from the target.

In using swarming as a foreaging or battlefield tactic, or to realise swarm-robots, Dorigo et al.[4], we need to consider three aspects: logistics, command and organization, degree of autonomy of agents, nature (tacit or explicit) and amount of communication, and sensor and communication technology used between the agents. In the military context three factors contribute to the success of swarming -Elusiveness either thro mobility or concealment, A longer range fire power-stand-off capability, and Superior situational awareness (having more information about the location, activity and intent of the enemy).

A multiset of agents that use sensory perception and computation can evolve into self-organizing swarms. We can use two different forms of communication to enable (connect) or inhibit (disconnect) agents to form interactive networks, exhibiting the properties of small world graphs or scale-free property or their combined properties. 1. Tacit (Indirect) communication: Use of markings similar to a chemical gradient or diffusion mechanism or a communication field (Agents with simple intelligence, e.g., ants). This provides a common spatial resource, where each agent can leave a mark that can be perceived by other agents.

2.Explicit (Direct) communication: Use of voice, signals, radio resulting in a positive feed-back or nonlinear response to the information available from knowledge other agents may possess (by connecting or disconnecting with other agents at random):

This would require that *each agent knows what other agents know, and how much they know measured in a taxonomic scale* (Agents with more complex intelligence) so that each agent can have a score about its neighbours to link, de-link and form clusters. This would result in a nonlinear iterative scheme among the agents. Here, individual agents are points in space, and change over time is represented as movement of points, representing particles with different properties and the system dynamics is formulated using the rules:

(1) Stepping (or local coupling) rule:

The state of each individual agent is updated or revised in many dimensions, in parallel, so that the new state reflects each agent's previous best success.

(2) Landscaping (or global coupling) rule:

Each agent assumes a new best value of its state that depends on its past best value and a suitable function of the best values of its interacting neighbours, with a suitably defined neighbourhood topology and geometry.

All agents in the universe or selected chunks are updated using rules (1) and (2).

The above two rules permit us to model Markovian random walks which is independent of the past history of the walk and non-Markovian random walks, dependent upon past history- such as self-avoiding, self-repelling, communicating, and active random-walker models. This can result in various kinds of attractors having fractal dimensions presenting a swarm-like, flock-like, bacterial colony-like appearances depending upon the Jacobian of the mapping.

Simulation results show that the swarm network topology is very sensitive to the nature of interaction and threshold values, cost and aging of nodes. The swarms also exhibit features of both the small world graphs and scale-free graphs and can tune themselves into one class or another. This emphasizes the fact that the system is self-organizing exhibiting both the small-world and scale-free properties, or changing from one class to another, yet preserving self similarity. Thus the agent-fusion networks can alter their fractal dimensions and from assortativity to disassortativity and conversely, depending upon the environmental influence.

Presently, swarm-intelligence is widely used to solve a variety of problems in multi-objective optimisation, selforganization, swarming robots, battle- field simulation, and multi-sensor data fusion. The rapidly advancing softwareagent technology and animation tools provide the necessary support to simulate such swarms for a desired application, and study them in detail with the aid of visual displays.

VII. MULTI-AGENT TOOLKITS

Shakshuki et al. [14], evaluate multiagent tool kits, such as: Java Agent development framework (JADE), Zeus Agent building toolkit and JACK Intelligent Systems. They consider Java support, and performance evaluation and the Java based architecture Cougaar. The paradigm described here is well-suited for implementing in Cougaar, a Java based agent architecture, since Cougaar is based on human reasoning. A Cougaar agent consists of a blackboard that facilitates communication and operational modules called plug-in that communicate with one another through the blackboard and contain the logic for the agent's operations. The use of blackboard and direct communication are useful to simulate the information fusion.

VIII. CONCLUSION

This paper described a multiagent Information fusion (MAIF) methodology .We described the important features of agents, the cooperative and competitive parallelism, nondeterministic and probabilistic connectivity structure that can arise among the agents resulting in scale-free and small world networks during the multi-sensory information fusion process. We also illustrated these aspects through the "Swarming strategy". Also we briefly described the software tools available for agent -based information fusion.

REFERENCES

[1] A.Barabasi and R. Albert, *Emergence of Scaling in Random Networks*, *Science*, Vol.286, pp.509-512, 1999.

[2] E. Bonabeau, and M Dorigo, and Theraulaz, G. Swarm Intelligence: From natural to artificial systems, Oxford University Press, U.K, 1999.

[3]F.Chung and L.Lu, Complex Graphs and Networks, American Mathematical Society, CBMS, Vol.107, Providence, Rhode Island, 2006.

[4]M.Dorigo et al, *Swarm-Bot: design and implementation of colonies of self-assembling robots*, pp.103-136, in G.Yen and D.B.Fogel (Eds), IEEE Computational Intelligence Society, IEEE Press, New York,2006

[5] S.N.Dorogovtsev and J.F.F.Mendes, *Evolution of Networks*, Oxford University Press, Oxford, 2003.

[6] S. J. Edwards, *Swarming on the Battlefield*, National Defence Research Institute, RAND, U.S.A, 2000

[7] D.L.Hall and J.Llinas (Eds), *Handbook of Multisensor data fusion*, CRC Press, Boca Raton, Fl.,2001

[8] J.Kennedy and R.C.Eberhart, *Swarm Intelligence*, Morgan Kauffman, London,2001

[9]E.V.Krishnamurthy and V.K.Murthy, *Distributed agent paradigm for soft and Hard computation*, Journal of Networks and computer Applications, Vol.29,pp.124-146,2006.

[10] V.K.Murthy and E.V.Krishnamurthy, Interacting Agents in a Network for in silico Modeling of Nature - Inspired Smart

Systems, Chapter 7, pp.177-231, in Studies in Computational Intelligence (SCI) Vol.72, Eds: R.S.T.Lee and V.Loia, Springer Verlag, New York, 2007.

[11] M.E.J.Newman, *The structure and Function of complex Networks*, Santa Fe Institute Publication, 2004.

[12] S.J Ovaska, (Ed), *Computationally Intelligent Hybrid systems: The fusion of Soft and hard computing*, Wiley, IEEE Press, Hoboken,N.J,2004.

[13] S.J.Ovaska and B.Sick, *Fusion of soft and hard computing, in Computational Intelligence*,pp.47-72, G.Yen and D.B.Fogel (Eds),IEEE Computational Intelligence Society, IEEE Press, New York, 2006

[14] E.Shakshuki and Y. Jun, *Multi-agent development toolkits: An Evaluation*, Lecture Notes in Artificial intelligence, 3029, Springer Verlag, New York, pp.209-218,2004.

[15] D.Watts, *Small Worlds*, Princeton University Press, Princeton, 1999.

[16]M.Woolridge, Introduction to Multi-Agent systems, John Wiley, New York, 2002.