International Journal of Computer and Communication Technology

Volume 2 | Issue 3 Article 7

July 2011

A Review on Intelligent Agent Systems

Lokanath Sarangi
lokanathsarangi@yahoo.com

Dr. Chittaranjan Panda
panda_chitaranjan@rediffmail.com

Recommended Citation

Sarangi, Lokanath and Panda, Dr. Chittaranjan (2011) "A Review on Intelligent Agent Systems," *International Journal of Computer and Communication Technology*: Vol. 2 : Iss. 3 , Article 7.
DOI: 10.47893/IJCCT.2011.1092
Available at: https://www.interscience.in/ijcct/vol2/iss3/7

A Review on Intelligent Agent Systems

Lokanath Sarangi *

E-Mail: lokanathsarangi@yahoo.com
*Corresponding author

Dr. Chittaranjan Panda*

E-Mail: panda_chitaranjan@rediffmail.com
*Corresponding author

Abstract: Multi-agent system (MAS) is a common way of exploiting the potential power of agent by combining many agents in one system. Each agent in a multivalent system has incomplete information and is in capable of solving entire problem on its own. Multi-agent system offers modularity. If a problem domain is particularly complex, large and contain uncertainty, then the one way to address, it to develop a number of functional specific and modular agent that are specialized at solving various problems individually. It also consists of heterogeneous agents implemented by different tool and techniques. MAS can be defining as loosely coupled network of problem solvers that interact to solve problems that are beyond the individual capabilities or knowledge of each problem solver. These problem solvers, often ailed agent are autonomous and can be heterogeneous in nature. MAS is followed by characteristics, Future application, What to be change, problem solving agent, tools and techniques used, various architecture, multi agent applications and finally future Direction and conclusion. Various Characteristics are limited viewpoint, effectively, decentralized; computation is asynchronous, use of genetic algorithms. It has some drawbacks which must be change to make MAS more effective. In the session of problem solving of MAS, the agent performance measure contains many factors to improve it like formulation of problems, task allocation, organizations. In planning of multivalent this paper cover self-interested multivalent interactions, modeling of other agents, managing communication, effective allocation of limited resources to multiple agents with managing resources. Using of tool, to make the agent more efficient in task that are often used. The architecture o MAS followed by three layers, explore, wander, avoid obstacles respectively. Further different and task decomposition can yield various architecture like BDI (Belief Desire Intension), RETSINA. Various applications of multi agent system exist today, to solve the real-life problems, new systems are being developed two distinct categories and also many others like process control, telecommunication, air traffic control, transportation systems, commercial management, electronic commerce, entertainment applications, medical applications. The future aspect of MAS to solve problems that are too large, to allow interconnection and interoperation of multiple existing legacy systems etc.

1. Introduction

An agent is an entity that is able to carry out tasks, to help the human beings. It is also a set of conflicting tasks where one can be active simultaneously. This is a high level behavioral sequence as opposed to the low level actions performed direct by actuations. An agent can be a biologic entity or robotic or computational. The discussion is primarily concerned with software agents, reactive agent [1], interface agent, information agent and various multi agent systems (MAS) [1]. A software agent is a computer program designed to carry out some task on behalf of a user. There are a number of ways in which software agents can be built and a number of properties that they can have. One property of an agent which is most important is its intelligence. Multi-agent systems are a common way of exploiting the potential power of agents by combining many agents in one system. Each agent in a multivalent system has incomplete information and is incapable of solving the entire problem on its own. But combining them together, the agent forms a system that has sufficient information and ability to solve the problem. The system does not have a centralized control mechanism for solving the problem. Which is defined as multi agent system?

The multivalent systems have been studied for the last 10 years by a group of AI community. Some of the critical behavior of MAS and the associated work has been addressed here. AI endeavors to address more complex, realistic, and large-scale problems. Such problems are beyond the capabilities of an individual agent. The capacity of an intelligent agent is limited by its knowledge, its computing resources, and its perspective.

The most powerful tools for handling complexity are modularity and abstraction. Multi agent systems (MAS) offer modularity. If a problem domain is particularly complex, large, and contain uncertainty, then the one way to address, it to develop a number of functionally specific and modular agents that are specialized at solving various problems individually. This decomposition allows each agent to use the most appropriate steps for solving its particular problem. When interdependent problems arise, the agents in the system must cooperative with one another to ensure that interdependencies are properly managed.

An open system is one in which the structure of the system itself is capable of dynamically changing. The characteristics of such a system are that its components are not known previously and can change over time. Secondly it also consist of heterogeneous agents implemented by different people, with different tools and techniques such as internet, which can be viewed as a large, distributed information resource, with nodes on the network designed and implemented by different organizations and individuals. In an open environment, information sources, communication links, and agents could appear and disappear unexpectedly. The next generation of agent technology can perform information gathering and sophisticated reasoning in support of solving various problems. For this cooperation and coordination among the agent is essential [14]. In addition, these capabilities will allow agents to increase the problem solving scope of single agents. Such functions will require techniques based on negotiation or Cooperation, which lie firmly in the domain of MAS [18].

MAS can be defined as a loosely coupled network of problem solvers that interact to solve problems that are beyond the individual capabilities or knowledge of each problem solver. These problem solvers, often ailed agent, are autonomous and can be heterogeneous in nature [19].

1.1 Characteristics:

The various characteristics of MAS [3, 2, 5] have been enumerated as follows,

(i) Each agent has incomplete information or capabilities for solving the problem and, thus, has a limited viewpoint.
(ii) MAS have more knowledge or understanding of their environment than simple agents and are able to use this intelligence to carry out their tasks more effectively.
(iii) Data are decentralized.
(iv) Computation is asynchronous.
(v) MAS are able to carry out their task without direct input from a human.
(vi) It can give the ability to learn, to solve new problem using genetic algorithms.
(vii) It is collaborative or competitive.
(viii) These agents will competing with each other.

1.2 What to be change:

MAS are rapidly expanding in many human-computer environments. Although MAS provide many potential advantages, they also present many difficult challenges, which have been enumerated below:

(i) To formulate, describe, decompose, and allocate problems and synthesize results among a group of intelligent agents.
(ii) To enable agents to communicate and interact. Selection of communication languages and protocols.
(iii) To inter operate the heterogeneous agents interoperate [39].
(iv) To ensure the agents act coherently in making decisions or taking action, accommodating the non-local effects.
(v) To ensure that MAS does not be resource bounded.
(vi) To avoid unstable system behavior.

(vii) To enable individual agents to represent and reason about the actions, plans, and knowledge of other agents to coordinate with them.
(viii) To recognize and reconcile disparate viewpoints and conflicting intentions among a collection various architecture of agents trying to coordinate their actions.
(ix) To engineer and constrain practical distributed artificial intelligence systems.

2. Problem Solving Agents:

Intelligent agents are enhancing, their performance measure by adopting a goal. The agent's performance measure contains many factors like to improve its sustain, improve its dynamics and avoid hangovers. The dictions problem is a complex one involving tradeoffs [13]. The goal is a set of world states in which it is satisfied. The agent's task is to find out which sequence of actions will get it to a goal state. Before this, it needs to decide the various input problem an agent to be consider. Problem formulation is the process of deciding what actions and states to consider, for a given goal. Generally, an agent with several immediate options of unknown value can decide the, process of examining different possible sequence of actions which lead to state of known value, and then choosing the best sequence. The process of looking for such a sequence is called search. The search algorithm put a problem as input and returns a solution in the form of an action sequence. This solution recommends an action to be carried out and the phase doing this is called the execution phase. A simple problem –solving agent [4], formulates a goal and a problem, then searches for a sequence of actions that would solve and executes the actions for the problem at same time. When this is complete, it formulates another goal and complete up to the execution phase. During the period of executing the sequence it ignores its percepts like:

(i) It assumes that the solution it has found will always work.
(ii) The solution design by above assumption in a static environment, does not give attention to any changes that might be occurring in the environment.
(iii) The agent design also assumes that the initial state of the problem is known to the system.
(iv) The idea of enumerating "alternate courses of action" assumes that the environment can be viewed as discrete.
(iv) Finally the agent design assumes that the environment is deterministic.

2.1 Solution to Problem:

Solutions to problems are single sequence of action, so they cannot handle any unexpected event but the solutions can be executed without giving attention to the percepts. An agent that carries out its plan with closed eye must be quite certain that what is going on and it is a part of open loop control theory. The well defined problems and its solutions are as follows.

(i) The initial state that the agent starts with.
(ii) A description of the possible actions available to the agent .The most common formulation uses a successor function [23].
(iii) The goal test, which determines whether a given state is a goal state.
(iv) A path cost function that assigns a numeric cost to each path.

2.2 Formulating Problems:

The Formulation of problems may occur in terms of the initial state of successor function, goal test, and path cost .This formulating seems reasonable, and omits great aspects of the real world. The descriptions of state are essential for finding a root in details from the representation, is called abstraction. The abstraction is useful if carrying out each of the actions in the solutions, is easier than the original problem .The choice of a good abstraction thus involves removing as much detail as possible while retaining validity and ensuring that the actions are easy to carry out .The problem–

solving approach has been applied to a vast array of task environments. The list from some of the best known here,

Distinguishing between toy and real

(i) Would problems. A toy problem is intended to illustrate of exercise various problem – solving methods. It can be performance of algorithms.
(ii) A real world problem is one whose solutions is discussed actually by the people .They do not have a single agreement – upon description of their formulation.

Before 30 years a group of AI researchers held the first DAI [41] workshop at MIT to discuss issues concerning intelligent problem solving with systems consisting of multiple problem solvers. It was decided that Distributed AI [2] was not concerned with low level parallelism issues, and its application of processing over different machines, to parallelize centralized algorithms, without issues to intelligent problem solvers for coordinating effectively to solve problems [25, 26].This is the beginning, of the DAI [6] field and then interred into a major international area.

Actors are the one of the first model in multi agent problem solving system. Actors were proposed as universal primitives of concurrent computation .They are self – contained, interactive autonomous components for a computing the system that communicate by asynchronous message passing. The basic actor primitives are:

(i) Create: Create an actor from a behavior description and a set of parameter, possibly including existing actors.
(ii) Send: Sending a message to an actor.
(iii) Become: Changing an actor's local state.

Actor models are used of concurrent computation .But it is noted that actor models, along with other DAI [7] models, face the issue of coherence .The low – level granularity of actors also poses issues relating to the composition of actor behaviors in larger communities and achievement of higher level performance goals with only local knowledge. These issues were address in an overview of open loop systems and its challenges are presented, to an origination architecture called ORG [10].

2.3 Task Allocation:

The issue of flexible allocation of taxes to multiple problems solvers received attention early on in the history of DAI .Davis and Smith's work resulted in the well known contract Net protocol. In which the agents can dynamically take two roles:

Manager or contractor.

(i) Given a task to perform, an agent first determines whether it can break it into subtasks that could be performed concurrently. It employs the contract Net Protocol to announce the tasks that could be transferred, and requests bids from nodes that could perform any of these tasks .A node receives a task announcement replies with a bid task, and indicate its degree thinking for performing the task.

(ii) The contractor collects the bids and awards the task to the best bidder. According to the contract to be negotiation technique, it is really a coordination method for task allocation .The protocol land provides load balancing its limitations are that it does not detect or resolve conflicts. There is no pre-option in task execution and is communication intensive .To rectifying some of its short coming, a number of extensions to the basic protocol have been proposed. Those are grouped together under the heading of the task environment .For the agronomical minded, this is called as PEAS (Performance, Environment, Actuators, Sensors) description [17].

The performance measure in a automated driver system aspire the desirable qualities which include the getting of the correct destination, minimizing fuel consumption, wear and tear, minimizing the trip time and cost, minimizing violations of traffic laws and disturbances to other drivers, maximizing safety and passenger comport and profits. Since the driving environment deals with a variety of roads, traffic, stray animal's police car, puddles. The actuators available to an automated taxi will be more or less the same as those available to a human driver. To achieve its goals in the driving environment the taxi will need to known its position the hurdles on the road and its speed of driving. Its basic sensors should therefore include one or more controllable TV Cameras speedometer and odometer.

2.4 Organizations:

An organization provides a framework for agent interactions through the definition of roles, behavior, characteristic, expectations, and authority relations. Organizations are, in general, conceptualized in terms of their structure. In cooperative problem solving, a structure gives each agent a high level view to solve the problems. The structure should also indicate the connectivity information to the agents so they can distribute sub problems to individual agents.

This is a challenging, task especially in environments which include large numbers of agents and that have information sources, communication links, and/or agents that might be appearing and disappearing. An agent shows its strength to a Middle agent that is looking for finding another which possesses a particular strength to query a middle agent In the RETSINA infrastructure; there could be multiple middle agents, not only in type but also in number. The other perspective in DAI defines organization less in terms of structure and more in terms of current organization theory. Theory says an organization as a "particular set of settled and unsettled questions about beliefs and actions through which agents view to other agents." So an organization is defined as a set of agents with mutual commitments, global commitments, and mutual beliefs. An organization consists of a group of agents; agents are formed by the set of activities. The structure imposes constraints on the ways the agent's authority communicates and coordinates. The decision-making and control is concentrated in a single problem solver at each level in the hierarchy. The Interaction occurs between them through vertical communication from superior to subordinate agent and vice versa. Superior agents exercise control over resources and decision making. The Community of experts says that organization is flat, where each problem solver is a specialist in some particular area. Agents coordinate though mutual adjustment of their solutions so that overall coherence can be achieved. Agents coordinate through mutual adjustment of prices is specified by a pluralistic community could operate to each other.

Solutions to problems are locally constructed, then they are communicated to other problem solvers that can test, challenge, and refine the solution .In dynamic environments, the issue of organizational adaptively is critical in dynamic environments RETSINA exhibits organizational adaptively through cooperation mediated by middle agents. RETSINA agents find their collaborators. dynamically based on the requirements of the task and on which agents are parts of the society at any given time .Hence organization in general conceptualized in terms of their structure, that is, the pa tern of information and control relations that exist among agents and the distribution of problem solving capabilities among them.

3. Tools And Techinques:

The mechanism used for learning tool, to make the agent more efficient in tasks that are often used. This architecture has been used in the MANTA system to simulate the behavior of ants. Reactive agents [1] do not have representations of their environment and act using a stimulus response type of behavior. They do not take history into account or plan for the future. Through simple interactions with other agents, complex global behavior can emerge. The two properties of reactive are robustness and fault tolerance. A group of agents can complete a task even if one of them fails. But pure reactive systems suffer from two main limitations:

(i) Reactive agents make decisions based on local information, so they cannot take into consideration the non-local information or predict the effect of their decisions on global behavior.

(ii) In reactive systems, the relationship between individual behaviors, environment, and overall behavior is not understandable. So some of the specific tasks can not be fulfill.

Most of the architectures of Multi Agent system use three layers. At the first level, there is typically a reactive layer, which makes decisions about what to do base on raw sensor input. The second level takes abstracts away the raw sensor input and deals with a knowledge-level view of the agent's environment. The third level tends to deal with the social aspects of the environment and the way the layers interact with each other. The area of agent architectures, particularly layered architectures, continues to be an area of on going work to investigate the appropriateness of various architectures for different environment. It turns out to be hard to evaluate one agent architecture against another, especially within the context of MAS.

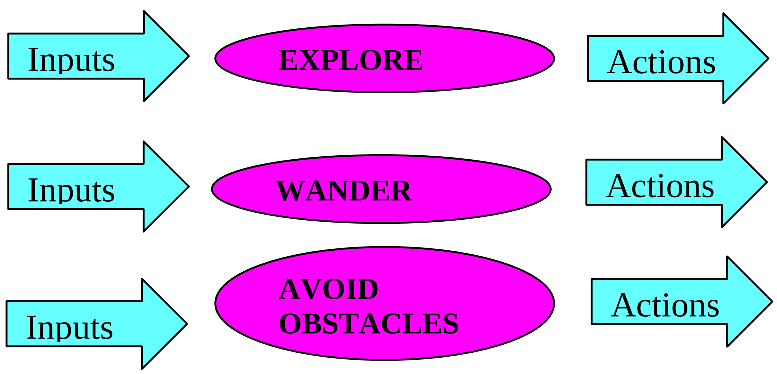

Figure – 1 Three layered MAS

4. Multi Agent Planning:

In single agent environments the agent are alone. If other agents in the environment are inducted then they are simply included in the model of environment, without changing its basic algorithms .In many cases, this lead to poor performances .The interaction other agent is not same as with the nature. In particular nature is different to the agent's intention .Whereas other agents are not .This section introduces multi agent planning to handle these issues.

The multi agent environments are cooperative and competitive. A simple cooperative example is team planning in double tennis. The Plane ground is constructed that specified actions for both players of the teams. The techniques for constructing such ground. This requires some form of coordination, which is achieved by communication. Agents can improve coherence by planning their actions. Planning for a single agent is a process of constructing a sequence of actions. But the planning in a MAS environment also considers the difficulties that the other agents' activities place on an agent's choice of actions. Most early work in DAI has dealt with groups of agents pursuing common goals. Most work on multi-agent Planning assumes an individual sophisticated agent architecture that enables them to do rather in complex environment. The distributed planning was the approaching the complete planning before action. To produce a coherent plan, the agents must be able to recognize sub goal interactions and avoid them or resolve them. The agents develop their plan; examined them for critical situation. If contention for resources could cause them to fail. The agent then inserted synchronization messages so that one agent would wait till the resource was released by another agent. In air-traffic control the synchronizing agent was dynamically assigned according to different criteria, and it could alter its p1an to remove the interaction. In the PA/C model, agents do not need to have all the necessary information locally to solve their sub problems. With the FA/C model, a series of sophisticated distributed control schemes for agent coordination were developed, such as the use of static Meta level information specified by an organizational structure and the use of dynamic Meta level information developed in partial global planning (PGP) [16]. The interactions between agents communicate the plans and goals at an appropriate level of abstraction. These communications enable a receiving agent to be associated with the future behavior of a sending agent. The cooperation, among the recipient agent uses the information in the plan to adjust its own local planning appropriately, so that the common planning goals are met. TAUV45 is a framework for evaluation of coordination algorithms between agents secondly in cooperative multi agent planning the focus is based modeling teamwork explicitly. Explicit modeling of teamwork is particularly helpful in dynamic environments where team members might fail or be presented with new opportunities. Joint-intention framework is characterizing a team's mental state. A joint commitment is defined as a joint persistent goal. The commitment protocol synchronizes the team such that all members simultaneously enter into a joint commitment toward a team task .All team members must consent, using confirmation, to the establishment of a joint commitment goal. The SHAREDPLAN [20] model is not based on joint mental attitude but rather on a new mental attitude intending that an action be done. COLLAGEN [21] is a Prototype toolkit, which has its origins in SHAREDPLAN, and is applied to building a Collaborative interface among agents that helps with Air-travel arrangements. Responsibility based on a joint commitment to a team's joint goal and a joint recipe commitment to a common recipe. Team work model based (a shell for TEAMWORK), steam is a on enhancements to the SOAR architecture, plus a set of about 300 domain-independent SOAR rules. STEAM uses a hybrid approach that combines joint intentions [8] but also uses partial SHAREDPLANS .So the emphasis of multi-agent planning has been on flexible communication and action execution in complex, dynamic environments.

4.1 Self –interested Multi Agent Interactions:

The nature of interactions among self-interested agents has been centered and around negotiation. Negotiation is seen as a method for coordination and conflict resolution e.g., resolving goal disparities in planning, resolving constrains in resource allocation, resolving task inconsistencies in determining organizational structure [34, 35]. Negotiation has also been used as a metaphor for communication of plan changes, task allocation, or centralized resolution of constraint violations. Hence, negotiation is almost as ill – defined as the notion of "agent".

The main characteristics of negotiation that are necessary for developing application in the real world are:

- The conflicts are resolved in a decentralized manner,
- Self – interested agents under conditions of
- Bounded rationality, and
- Incomplete information.

Furthermore, the agents communicate and iteratively exchange proposals and counter- proposals. The persuader system by Sycara [35] and work by Rosenschein represent the first work by Dai researchers on negotiation among self-interested agents. The two approaches differ in their assumptions, motivations, and oprationalisation. The work of Rosenschein was based on game theory. Utility values for alternative outcomes are represented in a payoff matrix that is common knowledge to both parties in the negotiation. Each party reasons about and chooses the alternative that will maximize its utility. Despite the mathematical elegance of game theory .game theoretic models suffer from restrictive assumptions that limit realistic problems. Real world negotiations are conducted under uncertainty, involve multiple criteria rather than a single utility dimension, the utility of the agent are not common knowledge but are instead private, and the agents are not omniscient. Work by Kraus [22], focuses on the role of time in negotiation .Using a distributed mechanism, agent negotiate and can reach efficient agreements without delays .It is also shown that the individual approach of each agent towards the negotiation time affects (and may even determine) the final agreement that is reached. In [22] the Bazaar negotiation model was presented, which included multi-agent learning through agent interactions. The benefits to learning, if any, on the individual utilities of agents, as well as the overall joint system utility were examined and the experimental results suggest that:

(i) When all agents learn, the joint system utility is near optimal and agent' individual utilities are very similar;

(ii) when no agent learns the agents ` individual utilities are almost equal but the joint utility is very low (much lower than in the " all agents learn " condition); and (iii) When only one agent learns its individual utilities increases at the expense of both the individual utility of the other agents as well as the overall joint utility of the system (I .e, only one agent learning has a harmful overall effect)[22].

4.2 Modeling Other Agents:

The most effective way to handle partial absorbability is for the agent to keep track of the part of the world it cannot see now. That is the agent should maintain some sort of internal state that depends percept history and thereby reflects at lest some of the unobserved aspects of the current state. The knowledge about "how the world works" whether implemented in simple Boolean circuits or in complete scientific theories – is called a model of the world. An agent that uses such a model is called a model based agent. Agents can increase the accuracy and efficiency of their problem solving if they are given knowledge about other agents or the ability to model and reason about others.

The ability to model others increases an agent's flexibility. Instead of operating on the basis of a fixed protocol of interaction, modeling others allows the agent to change the pattern of interaction. We saw already how explicit commitments to joint activity are the cornerstone of models of team. In the RETSINA multi agent Infrastructure, commitment of an agent to performing a task is not communicated explicitly to others but is implicit in its advertisement to a middle agent. An agent's advertisement describes its capabilities, that the set of services it can provide to others. For example, the advertisement of an information agent expresses the set of queries that the agent is capable of answering. In this way, an agent that needs a service can, through asking a middle agent, find out who is capable of providing the service and can contact the service provider(s). Thus, for example, Instead of agent A either having a model of agent B built into It at design time or having to infer it, agent B makes its capability model available to agent A through an advertisement to a middle agent. These two ways would be infeasible and extremely time consuming for agents in large-scale and open environments such as the internet. If the agent goes down, it "unadvertised"; that is, it lets it be known that it is no longer a member of the society.

*4.3 Managing Communic*ation

Agents can improve the coherence of their problem solving by planning the content, amount, type, and timing of the communication they exchange. It has been noted that [9] using abstraction and multilevel information is helpful because they help decrease communication overhead, in dynamic and open environments. The most prominent among them is agent interoperability. Agents is populating the internet at a rapid pace. These agents have different functions. Agents can increase their problem-solving scope by cooperation [28]. In an open environment, heterogeneous agents that would like to coordinate with each other (either cooperate or negotiate).The two major challenges:

(i) They must be able to find each other (in an open environment, agents might appear and disappear unpredictable,

(ii) They must be able to inter-operate. To address the issue of finding agents in an open environment such as the internet, middle agents [24] have been proposed. Different agent types were identified and implemented [26]. In preliminary experiments [27] it was seen that the behaviors of each type of middle agent have certain performance characteristics; although brokered systems are more vulnerable to certain failures, but are cooperative with a rapidly fluctuating agent workforce. Middle agents are advantageous because they allow a system to operate robustly in the face of agent appearance and disappearance and intermittent communications. To allow agents to interoperate, communication languages, the performance is based on message types, efficient languages. Message content that allows agents to understand each other have not been demonstrated effectively. The ontology problem that is, how agents can share meaning, is still open.

4.4 Managing Resources:

Another critical issue is effective allocation of limited resources to multiple agents. Like internet network congestion. Different methods developed for effective resource allocation to multiple agents and some of them hail from operations research based techniques for single-agent scheduling, and others use market-oriented approaches. In the first category, Distributed constraint heuristic search [28] are mentioned, which combines decentralized - search with constraint satisfaction and optimization. The key points of these methods are

- A set of variable and value-ordering heuristics that quantify several characteristics of the space being searched.
- A communication protocol that allows the agents to coordinate in an effective manner. The distributed scheduling model uses a large number of simple agents by partitioning problem constraints.

This methodology was applied to solve job-shop-scheduling constraint-satisfaction and constraint-optimization problems with good results. Economic-based mechanisms have been utilized in MAS to address problems of resource allocation (the central theme of economic research). In economics-based approaches, agents are assumed to be self-interested utility maximizes. In markers, agents that control scarce resources (labor, raw materials, goods, money) agree to share by exchanging some of their respective resources to achieve some common goal. Resources are changed with or without explicit prices. Market mechanisms have been used for resource allocation Markets assume that the exchange prices are publicly known. Hence, the agents exchange minimal amounts of information for price allocation. It is probable that In the future, most agents will be self-interested. A self-interested agent simply chooses a course of action that maximizes its own utility. In a society of self-interested agents it is desired that if each agent maximizes its local utility, the whole society exhibits good behavior. Economics-based approaches, such as market mechanisms, are becoming increasingly attractive to MAS researchers because of the ready availability of underlying formal models and their potential applicability in internet-based commerce. The various characteristics of market based resource are

(i) when no agent behavior will be acceptable, which is observable, and the action taken by learns, the agents' individual utilities called mechanism design. Many probes the learning agent can strongly bias is almost equal.

(ii) when overuse and, hence, congest a shared agent's belief process is characterized only one agent learns, its individual resource, such as a communications in terms of conjectures about the utility increases at the expense of both network. This problem is called the effect of its actions. A conjectural equips the individual utility of the other tragedy of the commons. Librium is then defined where all agents as well as the overall joint utility are less congestive. Generally, the problem of tragedy of agent's expectations is realized, and utility of the system is solved by pricing or tax- each agent responds optimally to its agent learning has a harmful expectations. MAS are presented effect. Second, a society of self-interest. A survey of multi agent learning can be computational agents can exhibit response of others. Oscillatory or chaotic behavior mental results show that depending [29] on the starting point, an agent might highly effective.

5. Various Architecture:

The different modeling schemes of individual agents can constrain the range of effective coordination regimes; communication protocol and behaviors. Further different problem and task decompositions can yield various interactions.

5.1 Belief Desire-Intention (BDI):

Various works in AI research has been formulated through logical axioms for rational agents. This is accomplished by formalizing the agent behavior in terms of beliefs, desires and intension .These works are known as belief desire-intention (BDI) [30] systems. The agent who has BDI architecture also called a deliberative. As agent orientation are very broad field topics concerning inter alias agent organizations. Agent behavior as well as messaging it becomes obvious that most of these platforms focus on specific objectives and therefore cannot address all important aspects of agent technology equally well. Two important categories of platforms are middleware and reasoning-orient system. Most middleware platforms intentionally leave open the issue of internal agent architecture and employ a simple task oriented approach [2, 11 and 17].In contrast, reasoning-centered platforms focus on the behavior model of a single agent trying to achieve rationality and goal-directedness. Most successful behavior models are based on adapted theories coming from disciplines such as philosophy, psychology or biology. Depending on the level of detail of the theory the behavior models tend to become complicated and can result in architectures and implementations that are difficult to use.

5.2 Retsina:

In multi-agent infrastructure, agents coordinate to collect information for the of user problem-solving tasks. RETSINA [31] – Reusable Environment for Task Structured Intelligent Network Agents are BDI-type that integrates planning, scheduling, execution of information [29]. It is an open multi agent system that supports communities of heterogeneous agents. The RETSINA system has been implemented on the premise that agents in a system should form a community of peers that engage in peer to peer interactions. Any coordination structure in the community of agents should emerge from the relations between agents, rather than as a result of the improved constraints of the infrastructure itself. In accordance with this premise, RETSIANA does not employ centralized control within the MAS rather it implements distributed infrastructural services that facilitate the interactions between agents, as opposed to manage them. The RETSINA multi-agent infrastructure consists of system reusable agents types that can be adapted to address a variety of different domain specific problems. Each RETSINA agent draws upon a sophisticated reasoning architecture that consists of four reusable modules for communication, planning scheduling and execution monitoring. The communication and coordination module accepts and interprets messages to an agent that can increase MAS coherence. The individual agent can cause non local effects of local actions, to the expectations of other agent's behavior. These requests are the goals of the recipient agent. The scheduling module priorities various steps of the plant. The scheduling process use the input, as the agents current set of plan and the set of all executable actions which is to be executed next. Agent-reactivity considerations are handled by the execution-monitoring process; further this bifid to the next intended action as its input and prepares, monitors, and completes its execution. The execution module monitor prepares the action for execution by setting up a process failed actions that is handled by the exception-handling process. The agent has a domain independent library of plans execution that are indexed by goals as well as a domain-specific library of the current input parameters. According to Brook Reactive agents have the following properties roots in Brook's system. The deliberative agents and his assertions are

(i) Intelligence to interact with other agents and its environment and
(ii) Intelligence behavior that emerges from the interaction of various simple behaviors organized in a layered way through a master-slave relationship of inhibition.

6. Multiagent System Applications:

Agent's technology is rapidly breaking out of universities and research labs, and is beginning to be used to solve real-world problems in a range of industrial and commercial applications. Various applications of multi agent system exist today, and new systems are being developed mainly in two distinct categories. They are,

(i) It aims to identify the main areas where agent-based approaches are currently being used and to provide pointers to some exemplar system.

(ii) It aims anticipate likely future directions applied agent work and to highlight open issues which need to be addressed if this technology is to fulfill its full potential results show that imperfect knowledge attempted to learn a model of the other suppresser oscillatory behavior at the agents. Such equivocal results are the first MAS applications appeared in, expense of reducing performance [33]. In Thomas out central control and compete for agreement.

6.1 Process Control:

Process control is a natural application for agents, since process controllers [43] are themselves autonomous reactive system [5]. The best known of these is ARCHON, a software platform for building multi-agent system and an associated methodology for building applications with this platform [34]. ARCCHON also has the distinction of being one of the worlds earliest field tested multi agent system. It is applied in electricity transportation management, particle accelerator control, spacecraft control and climate control and steel coil processing control [38]. Multi-agent systems are powerful and flexible tools for controlling complex phenomena. The complexity of a phenomenon can be tackled in such a way that each agent embeds the controller for a portion of the phenomenon. In this perspective, the interaction among the agents results in a complex controller for the whole phenomenon. The actions the agents undertake to control their portions of the phenomenon may conflict, as a result of the "overlapping" of the controlled portions; hence a mediated interaction is needed. A class of complex phenomena that present several difficulties in their satisfactory modeling and controlling is the class of physiological processes. We illustrate the negotiation [36, 45 and 46] paradigm of a general regulating multi-agent architecture called anthropoid agency.

6.2 Telecommunication:

Telecommunication systems are large, distributed networks of interconnected components which need to be monitored and managed in real-time. Features in a telecommunication [44] system provide added functionality on top of the basic communication. The traditional approach of analyzing services at design time and hard wiring in solutions for all possible interaction permutations is doomed to failure. Given this situation, Griffith and Velthuijsen decided to adopt a different strategy and tackle the problem on an as needed basis at run time .They did this by employing negotiating agents to represent the different entities who are interested in the set up of a call. When conflicts are detected, the agents negotiate with one another to resolve them so that an acceptable call configuration is established. It also include network control, transmission and switching, service management and network management. Direct communication is typically associated with cognitive agents, where the information encoded in the messages is related to a mental state. This generally assumed view on communication however, does not fit the approach of situated, behavior-based agents. We propose a protocol-based communication model for situated agents. Communication specified in terms of protocols, i.e. well-defined sequences of messages, shifts the focus of communication from the reasoning upon messages towards the relationship between the exchanged messages. The model decomposes communication into three functional modules: message decoding, communicating and message encoding. The core of the model, the communicating module

(i) Interprets decoded messages and reacts to them in accordance with the applicable protocol, and
(ii) Initiates or continues conversations when the conditions imposed by the applicable protocol are satisfied.

6.3 Air Traffic Control:

Ljunberg and Lucas describe a sophisticated agent realized air traffic control system known as OASIS [32]. The agent metaphor thus provides a useful and natural way of modeling real world autonomous components. As an aircraft enters into airspace, an agent is allocated for it, and the agent is instantiated with the information and goals corresponding to the real world aircraft and some control agents are responsible for managing the system. OASIS is implemented using a belief desire intention system called DMARS. Since critical socio-technical systems include people interacting with equipments in workplaces, their intrinsic reliability problems have been concerned with both these two "actors" Air Traffic Control (ATC) is going to be such a system in which controllers use a large number of distributed software tools to provide safety ATC services. The reliability of these services relies on the availability of the various tools. Indeed, a partial failure of a tool in use can have tragic consequences. This paper presents a multi-agent approach to this problem. We propose an agent-based decision-aided system that helps controllers in using their multiple software tools in situations where some tools are not available due to technical incidents. We build and test our system in an ATC simulation environment, thus develop an Agent-Based Simulation (ABS). Experimental work has demonstrated the significance of our system to air traffic controllers. Air Traffic Management (ATM) of the future allows for the possibility of free flight, in which aircraft choose their own optimal routes, altitudes, and velocities. The safe resolution of trajectory conflicts between aircraft is necessary to the success of such a distributed control system. In this paper, we present a method to synthesize provably safe conflict resolution maneuvers. The method models the aircraft and the maneuver as a hybrid control system, and calculates the maximal set of safe initial conditions for each aircraft, so that separation is assured in the presence of uncertainties in the actions of the other aircraft.

6.4 Transportation Systems:

This management is well suited to an agent based approach of its geographically distributed nature. Here two agents, one representing to the customer and other are representing to the station where the customers congregate in order to be picked up. Customer agents inform relevant stations of their requirements and the station agent determines whether their requests can be accommodated and, if so, which car they should be booked into. To improve the situation of wasting natural resources, the existing transportation systems have to be optimized. This means that we should not only think about new technologies for saving energy but also about better use of the existing traffic ways. The most efficient way to achieve these objectives is to automate the existing means of communication and to improve transportation management. Since most of the communication channels and technologies for automation in transportation management are Web-based, we want to describe how to improve Web-based transportation management. Talking about Web-based environments means talking about the Internet and services it offers. These Web-based environments build up a good basis for an agent-based approach, because all aspects for communication and information processing are also used in agent systems. In this approach, Web servers build the agents by themselves and an agent-based interaction works with the support of Web services. Thus, we can build an agent-based structure for transportation control that is similar to the structure of the Internet. Agent-based transportation management is a possible contribution to make transportation management more effective in regard to saving energy (fuel) and protecting our environment by stopping the increase of existing traffic ways.

6.5 Commercial Management:

It deals with information management towards mass market. The lack of effective information management tools has given rise to what is collo-quality known as the information overload problem. There are many reasons for this. Both human factor and organizational factor conspire against users attempting to use the resource in a systematic way. The two overload problems are

(i) Information filtering
(ii) Information gathering

Maxims is an electronic mail filtering agent which “learns to priorities, delete, forward, sort and achieve mail messages on behalf of user. Maxims constantly makes internal predictions about what a user will do with a message. If these predictions turn out to be inaccurate, then Maxims keeps them to itself, else it suggests to the user about what to do. The WARREN financial portfolio management system is a multi-agent system that integrates information finding and filtering in the context of supporting a user to manage her financial portfolio. The system consists of agents that self organize to monitor and tack stock quotes, financial news and company earnings reports to appraise the portfolio owner of the evolving financial picture.

6.6 Electronic Commerce:

Commerce is entirely driven by human interactions. But some of the commerce can not be automated that is what to buy, when to buy, how much to pay etc. So some commercial decision making can be placed in the hands of the agents. Although widespread electronic commerce is likely far away, an increasing amount of trade is being undertaken by agents. A simple “electronic marketplace” called Kasbah [37]. This system realizes the marketplace by creating buying and selling agents for each good to be purchased or sold respectively. Other commerce applications include Bargain Finder [38] an agent which discovers the cheapest CDs a personal shopping assistant able to search on line stores. MAGMA [38] is a virtual marketplace for electronic commerce and several agent- based interactive catalogues [38, 39].

6.7 Entertainment Applications:

The leisure industry is frequently seen as somehow peripheral to the “serious” applications of computers. Agents have an obvious role in computer games, interactive theater, and cinema and related virtual reality applications. Such systems tend to be full of semi autonomous animated characters, which can naturally be implemented as agent. In interactive theatre and cinema the system allows a user to play out a role analogous to those played by real, human actors in plays or films, interacting with artificial, computer characters that have the behavioral characteristics of real people. Agents that play the part of human in theatre style applications are often know as believable agents.

6.8 Medical Applications:

Medical informatics is a major growth area in computer science. Two of the major applications are (i) Patient monitoring; (ii) Health care. In patient monitoring GUARDIAN system [40] is intended to help manage patient care in the surgical Intensive Care (SICU). The GUARDIAN system. Distributes the SICU patient monitoring function among a number of agents like perception agent, reasoning agent and

control agent. The agents are organized into hierarchies and the system as a whole is closely based on the black board model of control. In health care a prototypical agent based distributed medical care system is described in [41]. This system is designed to integrate the patient management process, typically involves many individuals. The system allows a natural representation of this process, with agents mapped on to the individuals and, potentially, organizations involved the patient care process.

7. Future Direction And Conclusion:

The aforementioned systems can be considered as the first wave of a agent – based application. In addition to providing solutions to meet real – world they demonstrate that agent based systems are a useful and powerful solution technology. That is, the conception of (multiple) autonomous problem solvers interacting in various ways to achieve individual and system goals is a useful software engineering abstraction (just as objects and abstract data types are). The abstraction is useful to the extent that it enables software engineers to do more or to things more cheaply.

However, these developments also show that designing and building agent system is difficult. They have all the problems associated with building traditional distributed, concurrent systems, and have the traditional difficulties which arise from having flexible and sophisticated interaction between autonomous problem solving components. For these reasons most extant agent system applications are built by, or in consultation with, designers and developers who are themselves active in the agents research community .Whilst this may suffice for a niche software technology, we feel agents have the potential to be far more present every where than this. In fact, agent technology has the potential to enter the mainstream of software engineering solutions. However for this to occur, it must be possible for professional software engineers to design and build multi – agent systems. The big question then becomes one of how this is achieved.

The other major impediment to the widespread adoption of agent technology has a social as well as technical aspect. For individuals to be comfortable with the idea of delegating tasks to agents, they must first trust them. Both individuals and organizations will thus need to become more accustomed and confident with the notion of autonomous software components, if they are to become widely used. Users have to gain confidence in the agents must strike balance between continually seeking guidance (and needlessly distasting the user) and never seeking guidance and exceeding its authority. Put crudely, an agent must know its limitations.

7.1 Future Application:

The motivations for the MAS research include the ability of MAS to do the following in future.

(i) To solve problems that is too large for a centralized agent.
(ii) To allow for the interconnection and interoperation of multiple existing legacy systems. Therefore the legacy systems can remain useful to incorporate them into a cooperating agent community in which they can be exploited by other pieces of software.
(iii) To provide solutions to problems that can naturally be regarded as a society of autonomous interacting components agents. Such agents can be customized to reflect the preferences and constraint.
(iv) To provide solutions that efficiently use information sources that are spatially distributed.
(v) To provide solutions in situations where expertise is distributed.
(vi) To enhance performance along the dimensions of Computational efficiency, reliability and maintainability [42, 12].
(vii) Responsiveness to handle anomalies locally, not propagate them to the whole system;
(viii) Flexibility to *various* abilities for adaptively organing.
(ix) Reuse functionally for specific agents in different agent teams to solve various problems.

Finally with some assumptions, the field of autonomous agents and multi-agent systems is a vibrant and rapidly expanding area of research and development. It represents a melting pot of ideas originating from such areas as distributed computing, object oriented systems, software engineering, artificial intelligence, economics, sociology, and organizational science. At its core is the concept of autonomous agents interacting with one another for their individual and collective good. This basic conceptual framework has become common currency in a range of closely related disciplines, and offers a natural and powerful means of analyzing, designing, and implementing a diverse range of software solutions.

Over the past two decades, a number of significant conceptual advances have been made in both the design and implementation of individual autonomous agents, and in the way in which they interact with one another. Moreover, these technologies are now beginning to find their way into commercial products and real world software solutions. However, despite the obvious potential, there are a number of fundamental research and development issues which remain. As we have indicated, these issues cover the whole gamut of the agent field, and only when robust and scalable solutions have found will the full potential of agent based system be realized.

References:

[1] Bonasso, IL R; Kortenkamp, D. Miller, D.P; and Slack, M. "Experiences with an **Architecture** for Intelligent, Reactive Agents". In Intelligent Agents II, eds.1996

[2] Bond, A. H., and Gasser, I.*Readingsin "Distributed Artificial intelligence"*. San Francisco, Calif.: Morgan Kaufmann, 1998.

[3] Brooks, R. A. "Intelligence without Representation. Artificial Intelligence" vol. 47, no 3, pp 139-159,1991.

[4] Cammarata. S.; McArthur, D.; and Steeb, II. "Strategies of Cooperation in Distributed Problem Solving". In *Proceedings of the Eighth International Joint Conference on Artificial Intelligence (IJCAI-83)* Menlo Park, Calif.: International Joint Conferences on Artificial Intelligence, pp 767-770, 1983.

[5] Casteifranchi. C.; Micell, M.; and Cesra," *A*. Dependence Relations among Autonomous Agents". In *Decentralized Artificial Intelligence,* eds. S. Werner and Y. Demazeau. Amsterdam, The Netherlands: Elsevier/North-Holland, pp 215-231, 1992.

[6] Chaib-draa, B.; Moulin, B.; Mandiau, R.; and Millot, P. Trends in "Distributed Artificial Intelligence". *Artificial Intelligence Review vol. 6, no 1, pp 35-66, 1992*.

[7] Conry, S. L; Meyer, Lesser, V. R. "Multistage Negotiation in Distributed Planning". In Readings in Distributed Artificial Intelligence, Eds, 1988.

[8] Cohen, P. *P.,* and Levesque, H. J. "Intention Is Choice with Commitment". *Artificial Intelligence* 42(2-3): 213-261vol 42, no 2-3: PP 213-261, 1990.

[9] Decker, K, and Lesser, V. "Designing a Family of Coordination Algorithms". In Proceedingsof the First International Conference on Multi-agent Systems, 73-80. Menlo Park, Calif.: AAAI Press.PP 73-80, 1995.

[10] Decker, K; Williamson. M.; and Sycara, K. "Intelligent Adaptive Information Agents". Paper presented at the AAAI-96 Workshop on Intelligent Adaptive Agents, Portland. Oregon.4-7 August, 1996(a).

[11] Decker, K.; Williamson, M.; and Sycara, K. Matchmaking and Brokering. In Proceedings of the Second International Conference on "Multiagent Systems". Menlo Park, Calif.: AAAI Press, PP 432, 1996(b).

[12] Drogul, A., and Ferber, J.From Tom Thumb to the Dockers: "Some Experiments with Foraging Robots. In From Animals to Animats": Second Conference on Simulation of Adaptive Behavior, eds.1992.

[13] H. K J. Meyer and S. Wilson. Cambridge, Mass.: MIT Press. Durfee, E. H. "Coordination of Distributed Problem Solvers".1988.

[14] Boston: Kluwer Academic. Durfee, E.H. "A Unified Approach to Dynamic Coordination: Planning Actions and Interactions in a Distributed Problem Solving Network", Ph.D. dissertation, Department of Computer and Information Science, University of Massachusetts, 1987.

[15] Ferguson. I. A. "Integrated Control and Coordinated Behaviour": A Case for Agent Models. In *Intelligent Agents: Theories, Architectures, and Languages,* eds. M. Wooldridge and N. K Jennings. Lecture Notes in Artificial Intelligence, Volume 890. New York.Springer-Verlag, PP 203-218, 1995.

[16] Gasser, L. "The Integration of Computing and Routine Work. ACM Transactions on Office Information Systems": Vol 4, No 3,PP 205-225,1986..

[17] Huberman, B, and Clearwater, S. H. "A Multi-agent System for Controlling Building Environments". In Proceedings of the First International Conference on Multi-agent Systems, Menlo Park, Calif.: AAAI Press. PP 171-176, 1995.

[18] Jennings, N. "Controlling Cooperative Problem Solving in Industrial Multi-agent Systems Using Joint Intention". Artificial Intelligence: Menlo Park, Calif.: AAAI Press.Vol 75, No 2, PP 195-240, 1995.

[19] Jennings, N.; Sycara, K; and Wooldridge, M. **"**A Roadmap for Agent Research and Development. Autonomous Agents and Multi-agent Systems", 1998

[20] Jha, S.; Chalasani, Pr; Shehory, 0. And Sycara, K. "A Formal Treatment of Distributed Matchmaking". In Proceedings of the Second International Conference on Autonomous Agents (Agents '98). New York: Association of Computing Machinery.1998.

[21] Jordan, J S. "The Exponential Convergence of Bayesian Learning in Normal Form Games". Games and Economic Behavior: Vol. 4, No 2, PP 202-217, 1992.

[22] Kraus, S.; Wilkenfeld, J.; and Zlotkin. G. "Multi-agent Negotiation under rime Constraints". *Artificial Intelligence*. Lander, S.; Lesser. V. R.; and Connell, M.E. Conflict-Resolution Strategies for Cooperating Expert Agents. In *CKBS-90, Proceedings of the International Working Conference on Cooperating Knowledge-Based Systems*, ed. S. M. Deen, 183-200. New York: Springer-Verlag. Vol. 75, No 2, PP 297-345, 1995.

[23] Lesser, V. R. 1991. "A Retrospective View of FA/C Distributed Problem Solving". *IEEE Transactions Systems, Man, and Cybernet*ics. Lesser, V. K., and Corkill, D. D. Functionally Accurate, Cooperative Distributed Systems. IEEE *Transactions on Systems, Man, and Cybernetics.Vo.l 21, No 6, PP 1347-1363, 1981.*

[24] Lesser, V. K., and Corkill, D. D. "Functionally Accurate, Cooperative Distributed Systems". IEEE *Transactions on Systems, Man, and Cybernetics* Vol. 11, No 1, PP 81-96, 1981.

[25] Lesser, V. R**;** Durfee, E. H. and Corkill, D. "Trends in Cooperative Distributed Problem Solving". IEEE Transactions on Knowledge and Data Engineering.Vol 1, No 1, PP 63-83, 1989

[26] Liu, J., and Sycara, K P. "Exploiting Problem Structure for Distributed Comataint Optimization". In Proceedings of the International Conference on Multi-agent Systems. Menlo Park, Calif.: AAAI Press.PP 246-253, 1995(b).

[27] Liu, J., and Sycara, K P. "Collective Problem Solving through Coordinated Reaction". In Proceedings of the IEEE International Conference on Evolutionary Computation. Washington. D.C.: Institute of Electrical and Electronics Engineers.PP 575-578, 1993.

[28] Schwuttke, U. M., and Quan, A. G. “Enhancing Performance of Cooperating Agents in Real-Time Diagnostic Systems”. In Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence (IJCAI-93), Menlo Park, Calif.: International Joint Conferences on Artificial Intelligence. PP 332-337.1993.

[29] Shehory, 0. Jha. S.; and Sycara, K. “Multi-agent Coordination through Coalition Formation”. In Agent IV Agent Theories, Architectures, and Languages. eds. 1997.

[30] M. Singh, A. Rao, and M. Wooldridge, “Lecture Notes in Artificial Intelligence”. New York Springer-Verlag.PP 143-154.

[31] Smith, I... And Cohen, P. “Toward Semantics for an Agent Communication Language Based on Speech Acts”. In Proceedings of the Fourteenth National Conference on Artificial Intelligence (AAAI-96). 1996.

[32] Thorndyke. P.; and Wesson, R. “Distributed Intelligence for Air Fleet Control”. In Readings in Distributed Artificial Intelligence, eds. 1988.

[33] Stone, P., and Veloso, M. “Multi-agent Systems: A Survey from the Machine-Learning Perspective”, Technical report, CMU-CS-, School of Computer Science, Carnegie Mellon University. PP 97-193, 1997.

[34] Sycara, K... “Negotiation Planning”: An AI Approach. European Journal of Operational Research Vol 46 No 2: PP 216-234. 1990(a).

[35] Sycara. K. “Persuasive Argumentation in Negotiation”. Theory. And Decisions 2&PP 203-242. 1990(b).

[36] Sycara. K. P. “Resolving Goal Conflicts via Negotiation”. In Proceedings of the Seventh National Conference on Artificial Intelligence (AAAI-88). Menlo Park, Calif.: American Association for Artificial Intelligence. 1988.

[37] Sycara. K.; Decker, K.; and Zeng, D. “Intelligent Agents in Portfolio Management”. In Agent Technology Foundations, Applications, and Markets. eds. M. Wooldridge and N. R. Jennings, Berlin: Springer. PP 267-283.1998

[38] Tambe, M. “Toward flexible Teamwork”. Journal of Artificial Intelligence Research Vol 7, No 1: PP 83-124. 1997.

[39] Thomas, J., and Sycara, K. “Stability and Heterogeneity in Multi-agent Systems”. In Proceedings of the Third International Conference on Multi-agent Systems. Menlo Park, Calif: AAAI Press. 1998.

[40] Travers, M. “Animal Construction Kit In Artificial Life”, ed. C. G. Langton, Reading, Mass.: Addison-Wesley. Van Dyke.PP 421-442. 1988.

[41] Parunak, H. “Manufacturing Experience with the Contract Net”. In Distributed Artificial Intelligence, eds. 1987.

[42] M. Huhns, London: Pitman. “In Proceedings of the First International Workshop on ROBOCUP”**.** San Francisco, Calif.: Morgan Kaufmann. PP 285-310.

[43] Wang, H., and Wang, C. “Intelligent Agents in the Nuclear Industry”. IEEE Computer Vol 30, No 11: PP 28-34. 1997.

[44] Weihmayer. R., and Velthuijsen, “H. Application of Distributed AI and Cooperative Problem Solving to Telecommunications”. In AI Approaches to Telecommunications and Network Management, eds. 1994.

[45] Zeng, D., and Sycara, K. “Bayesian Learning in Negotiation”. International Journal of Human- Computer StudiesVol 48: PP 125-141. 1998.

[46] Zeng, D., and Sycara, K. “Benefits of Learning in Negotiation. Cooperation and Conflict Resolution via Negotiation among Autonomous Agents”. In Non cooperative Domains. IEEE Transactions on Systems, Man, and Cybernetics (Special Issue on Distributed Artificial Intelligence) Vol 21 No 6: PP 1317-1324. 1997.