CHAPTER 2

LITERATURE SURVEY

The research work in distributed AI has concentrated on many different aspects and level of processing. The goal of this chapter is to give a glimpse of the variety of work in this field and present it in a broader perspective. For years, distributed AI researchers have borrowed ideas from different fields and examined the capabilities of each idea essentially in isolation. The fact that distributed AI has interested researchers over the years is revealed by the number of books brought out over the years. All these books are essentially collections of research papers discussing various distributed AI issues. The chapter starts by discussing the various concepts covered in each book, that is, distributed AI in the context of distributed problem solving models including Blackboard model, Negotiation model, and Functionally accurate and Cooperative models are discussed. The chapter then goes on to discuss theme papers on important topics of distributed AI, such as communication, modeling of distributed AI, reasoning, negotiation, decomposition of problems, and learning. In this thesis, distributed AI system is viewed as an organization with a group of intelligent systems. Papers dealing with group characteristics, software agents in distributed AI systems, and applications of distributed AI are also included.

2.1 DISTRIBUTED AI

Current research in distributed AI can be characterized and understood in the context of following paradigm: there is a collection of agents (logically distinct processing elements) which are attempting to solve
a problem. Although there are identifiable global states and problem solving goals, each agent has only a partial and inexact view of these. Each agent attempts to recognize, predict, and influence the global state such that the local view of the goals is satisfied.

Davis (1980) presented a summary report on the workshop on distributed AI in SIGART Newsletter. This report on the first workshop conducted in this area discusses differences between distributed AI and distributed processing. A number of ideas are presented which include: organizational issues, planning and multiagent planning, contract-net systems, message passing systems, and some applications of distributed AI. Fehling and Erman (1983) presented a report on the third workshop on distributed AI. The themes in the informal discussions were: the value of distributed AI in promoting modular design, communication/computation tradeoffs, representation of knowledge sources, control issues, coherence and focusing in control, the question of whether individually dumb agents collectively can be intelligent and, finally, a realization that most conventional AI issues remain relevant in a distributed setting.

Huhns (1987) as the editor of book on distributed AI, presented a collection of papers describing the state of research in distributed AI at that time. According to him, distributed AI is concerned with the cooperative solution of problems by a decentralized group of agents (Intelligent System). The agents may range from simple processing elements to complex entities exhibiting rational behavior. The group of agents is decentralized and both control and data are logically and often geographically distributed. The papers are organized into three parts: Theoretical Issues, Architectures and Languages, and Applications and Examples. The readings in distributed AI by Bond (1988), gives a collection of papers, where he divides the world of distributed AI into two primary areas. They are, Distributed Problem Solving
(DPS) and Multi-Agent systems (MA). He also discusses the impact of distributed AI in cognitive science (e.g., mental models, social cognition), distributed systems (reasoning about knowledge and actions in distributed systems), architectural language support for distributed AI, human-computer interaction (task allocation, intelligent interface, dialogue coherence), and engineering of AI systems.

Durfee (1991) was the Guest editor of a special issue on "Distributed AI Melting Pot" of IEEE Transactions on Systems, Man, and Cybernetics, that played a critical role in defining the field of distributed AI. That special issue collected together articles, which discussed seminal ideas in the field and put forth many of the questions that have driven distributed AI research. It discusses articles on distributed reasoning, on negotiation and coordination, the critical role that distributed AI should play in bridging the gap between what mainstream AI technology can provide for enhancing personal productivity and what open systems science provides for supporting dynamic, networked organizations. The author also pointed out that distributed AI must embrace social and anthropological concepts if it is to bring AI ideas into human organizations.

Avouris and Gasser (1992) came out with a book, which again concentrated on distributed AI issues. They present an overview of distributed AI theory and practices, with special attention to defining the key issues of distributed AI, setting the stage for assimilating information, and introducing several newly-emerging foci for distributed AI, including social-level knowledge and action, emergent behavior, structure, and control in communities, and theories of reactive problem solving. The first section of this book gives an overview of distributed AI, and paradigm of reactive and situational multiagent systems. The second section addresses development tools and techniques for multiagent and distributed AI systems. It includes
object-oriented concurrent computation, the application of software
engineering principles to the design and construction of distributed AI
systems. The third section of the book treats several new application areas for
distributed AI.

Uma and Sarma (1996a) are the guest editors of the IETE special
issue on "Parallel and Distributed AI". The editor Sarma (1996b) given a brief
description about intelligent agents, in which he explains different types of
agents such as autonomous agents, adaptive agent, agent oriented
programming etc. This issue consists papers on decomposition, sub problem
allocation and agents in induction process. Uma (1996b) presented a new
decomposition method in her paper. This method decomposes the given
problem either spatially, temporally or by abstraction, and the results are
compared and tabulated. Nalini Kumari et. al. (1996), proposed an algorithm
for sub problem allocation. It uses heuristics for allocation of knowledge to
for knowledge partition and allocation. They propose heuristic solution for
knowledge base partitioning and allocation. Patra and Mohanty (1996)
presents a frame work for engineering an induction process. The authors
explain the issue of induction, where an agent tries to influence another
agent in a given context.

2.2 PARALLELISM IN AI

Distributed problem solving was first initiated by adopting
techniques to incorporate parallelism in AI. Fine grained problem
decomposition characterizes parallel AI. Fennel and Lesser (1977), use
parallel techniques to improve the speed of problem solving in AI, which they
explained with a case study. They analyze HEARSAY-II with respect to
parallelism and multiprocessor implementation of the system. Simulation
experiments include tests on the overhead of multiprocessing, parallelism and processor utilization, inter process communication, and different system organizations. These performance results help in gaining insight and providing guidance to the development of blackboard architecture systems. Kornfeld (1981), presents the salient features of scientific communities. He states that, parallelism is fundamental to the design and implementation of expert systems in many domains. A language called ETHER was designed to create highly parallel problem solving systems. The authors of the paper (Kibler 1985) also argue that, contrary to the attitude then prevalent, the use of parallel processing would provide significant impact on AI program processing. However research moved from parallel AI to distributed AI, where heterogeneous systems solved global problem in an individualized manner with intelligent coordination and cooperation between the distributed systems (nodes).

2.3 DISTRIBUTED PROBLEM SOLVING

Distributed AI was initially considered in the context of distributed problem solving. Decker (1987) in his paper discusses taxonomy of distributed problem solving systems. He also tries to differentiate between distributed AI and distributed problem solving. The following sections bring out various distributed problem solving models.

2.3.1 Blackboard Model

The Blackboard (Shapiro 1987) is one important distributed problem solving model. It is a highly structured, special case of opportunistic problem solving. This model arose from abstracting features of the HEARSAY-II: Speech understanding system (Erman 1980) developed between 1971 and 1976. Nii (1986) traces the history of blackboard systems, and the early applications which resulted in the evolution of the blackboard framework,
and its use in the paradigm of distributed problem solving. Hayes-Roth (1984) proposed BB1: a new blackboard architecture in his paper. BB1 is a system that has all the features of blackboard systems, but explicitly incorporates the treatment of controlled problem solving. Jagannathan et al. (1986), investigate the representation and reasoning with constraints in a blackboard environment. Some of the issues in constraint satisfaction are raised, and a comparison is made to similar work using AI techniques. Harmon et al. (1986) describes a distributed blackboard environment for information exchange and structuring in a single robot and also in a collection of cooperating robots. Since communication bandwidth is a significant limiting factor compared to the processing capabilities, a distributed blackboard system offers a variety of features, which deal with the inherent complexity of sophisticated robot systems. As the blackboard system is a centralized system, it is not considered in the proposed work.

2.3.2 Cooperative Distributed Problem Solving

Cooperation is an important part of distributed problem solving, but the amount of cooperation between problem solvers in a distributed problem solving environment can range from fully cooperative to noncooperative. Cammarata et al. (1983), discusses four different ways in which distributed problem solving can work for a particular application. The application they have considered is distributed air traffic control. Some of the modalities used are: i) one agent (problem solver) arbitrarily selected to replan its activity to avoid conflict with others, ii) the least constrained agent is selected, iii) the most knowledgeable one is selected, and iv) most knowledgeable agent replans other agents for global coherence. Measurements are taken, that show the behavioral properties of complex cooperative strategies under different organizations. Smith and Davis (1981) presents framework for cooperation in distributed problem solving. They explore two important
elements of distributed problem solving, namely task-sharing and result-sharing. It begins by giving the differences between traditional distributed processing and distributed problem solving. Then, they explain the task-sharing paradigm with reference to the contract net protocol and give the major features of this form of cooperation. Result-sharing is explained with reference to examples from the blocks world and distributed interpretation in HEARSAY-II. Khan and Jain, (1985), complete distribution is taken to mean the following: In a hierarchically structured set of knowledge sources, there is cooperation at any given level as well as between levels, accomplished with the help of a blackboard. The problem domain considered is the recognition of an object.

2.3.2.1 Negotiation Model

Negotiation is a complex and environment dependent phenomenon, and to date cooperative distributed problem solving researchers have studied only some of its specific forms (Shapiro 1987) (Litjer 1996) (Huhns 1987) (Hewitt 1991). Negotiation is important in cooperative distributed problem solving research, because it is a natural way for systems to coordinate decisions, to achieve several cooperative goals, including assigning tasks to increase parallelism and to effectively use network resources. Davis and Smith (1983) discuss the utility of negotiation as a fundamental mechanism for interaction. Here, the task distribution is viewed as an interactive process, a discussion carried on between a node with a task to be executed and a group of nodes that may be able to execute the task. Task distribution is viewed as a form of contract negotiation. This protocol is basically modeled after government contract awarding setup. Zlotkin (1991), discusses the coordination issue among agents. He developed a theoretical negotiation model. Here, the interaction between the agents is characterized as requirement of cooperation, semi-cooperation, of conflicts. The approach takes
a decision-theoretic, probabilistic view of decision making in a multiagent environment. Hewitt, (1977) introduces the concept of actors and message passing, and how this metaphor can be used in problem solving. This theory lends itself well to communication and parallelism in a general problem-solving environment. He has developed an actor language called PLASMA. Feber (1991), point out that other Actor languages have violated aspects of the Actor model in order to make them easier to use and implement, but argue that such violations can be made unnecessary by introducing reflection into the language called MERING IV.

The above approaches use a single-stage negotiation. Multistage negotiation is another way of achieving cooperation. This method is suggested by (Conry 1988), and developed a multistage negotiation protocol. Further, she (Conry 1991), applied this method for solving distributed constraint satisfaction problems. Multistage Negotiation provides agent with sufficient information about the impact of local actions on non local state. Sheng Li (1994), presented a practical scheme for coordinating mobile robots based on inter robot communication. Here, each robot is treated as an agent, and they work cooperatively to operate and/or process tasks. They use negotiation technique to solve the conflicts. The aim of the negotiation is to make a common agreement between involved robot, so that each of them can re-schedule (for instance delay or advance) its prior planned motion accordingly. Ozaki et al. (1994) emphasize on an organizational method for a collaborating team in a multiagent distributed robotic system. Organization strategy using negotiation and learning mechanisms are also discussed. The above negotiation models do not support past experiences of the experts (collaborators) however, the negotiation protocol proposed in this thesis incorporates trust factors and the knowledge gained by past experience.
2.3.2.2 Functionally Accurate Cooperation

A recurrent problem in cooperative distributed problem solving research is how to get nodes with inconsistent views and information to cooperate effectively, if consistencies arise because nodes might have incomplete or out-of-data views of the states of other nodes. Due to the increased demand for distributed problem solving capability in AI systems, the authors Lesser and Corkill (1981, 1991), proposed the concept of Functionally Accurate, Cooperative distributed system (FA/C). It distinguishes from the traditional distributed systems, in that each node is capable of producing some results, even in the presence of incomplete and possibly inconsistent information towards a single system goal. Some of these results may be incomplete, inconsistent, and even incorrect. Cooperation between nodes is critical in the synthesis of intermediate solutions. Hence, these systems have the inherent capability to deal with uncertainty. Experiments are presented in FA/C systems including HEARSAY-II type of hypothesis and test systems, distributed planning systems, and interactive refinement control systems. This work is compared with organizational theory and relevant issues in distributed problem solving are discussed.

Conry et al. (1991) report on techniques for solving a type of distributed constraint satisfaction in their paper. The approach involves a tentative exchange (in a Functionally Accurate/Cooperative manner) of resource assignments among managers. The authors describe techniques for detecting and recovering from over constrained situations (where all connections cannot be established). However, the model proposed in this thesis uses incremental learning method, which is based on FA/C model.

2.3.2.3 Communication

Now the chapter discusses some of the basic and broader issues that have interested researchers over the years. Constructing of multiagent system requires that developers choose the right architecture. This includes
how to distribute the system responsibilities among its component agents and how agents are to interact to achieve these responsibilities. At a minimum, a simple communication scheme is necessary to coordinate the activities among the different agents to exchange information.

Traditional protocol like Remote Procedure Call (RPC) (Birrell 1984) is usually synchronous, the client spends itself after sending a request to the server, waiting for the results of the call. SSL (Secure Socket Layer) protocol (Rubin 1998) is a stream based protocol, which has an initial handshake phase in which secure communications are established, and is designed for Web security. User level network interface protocols such as TCP/IP (Oppliger 1998), deals with data transfer mechanism, protection mechanism, and reliability and multicast. These protocols form a low-level base for problem solving communication and emphasis has been on establishing reliable and efficient communication, but, however they do not consider the semantics of the information being passed.

On the other hand high-level protocols assist the agents in deciding what to say to each other, rather than how to say it. High-level protocols are also called as coordination protocols (Lejter 1996), which include standard AI coordination protocols such as Voting protocol (Garcia-Molina 1982, Tanenbaum 1995), Contract protocol (Smith 1988), and Negotiation protocol (Davis 1988, Conry 1991). In order to achieve good cooperation among the agents, Smith and Davis (1988) described a tentative approach to cooperation using contract bidding (Contract Net Protocol) mechanism through negotiation. They define a high-level protocol (Contract-Net Protocol) for contracting (through negotiation) among modules (sub tasks or sub problems). In this approach, both collaborators with tasks to be executed and collaborators ready to execute task proceed simultaneously. The contracting process involves 4 basic steps: (1) task announcements on the net, (2) task announcement processing, (3) bidding, the interested collaborators submits
bids on announced task, and (4) bid processing. Once the bidding is successful, they are now contractors for a task through an announced award message.

The Unified Negotiation Protocol (UNP) has been developed by (Zlotkin 1991), to resolve conflicts under cooperative, semi-cooperation conflict situations. He presented a theoretical work for negotiation among the autonomous agents. Stirling (1993), proposed a epistemic utility theory for coordination control among agents. Here agents are independent and are capable of learning by observing (through sensor devices) the behavior of other agents. Klein (1991) discusses the general principles of existing conflict resolution approaches in his paper. His work separates generic strategies for conflict resolution from domain knowledge about the design problem. Pan and Tanenbaum (Pan 1991), put an effort in building computer tools that support human collaboration. The authors developed an infrastructure in which a number of fairly simple agent work together to perform important support tasks, particularly the more routine well-defined tasks in an organization. When two or more computing agents work on interacting tasks, their activities should be coordinated so that they cooperate coherently.

Durfee et al. (1987) discuss the cooperation issue among problem solvers. They describe three mechanisms that improve network coherence: i) organizational structuring, ii) planner at each node, and iii) meta-level communication utility about current state of local problem solving. Hewitt (1977) presents an approach to modeling intelligent agent in terms of a society of communicating knowledge-based problem-solving experts. He investigates the nature of the communication mechanism needed for problem solving by a society of experts. The high level protocols discussed above may fail to provide good interaction between the experts (or collaborators) as they do not support trust and belief issues. The protocol proposed in this thesis mainly concentrates on the above issues. The proposed model also utilizes background knowledge, which is not used in the protocols discussed above.
2.3.2.4 Modeling

Hewitt (1977), investigates the problem solving model of a society of experts to implement the model of a single very intelligent human. The paper also concentrates on communication issues. Yonezawa (1977), a description is presented for multiprocessing, distributed problem solving based on the concept of "actors". Communication and cooperation between actors is by messaging passing with flexible control structures. This model of computation has been implemented as a programming language called PLASMA. The concept tested with an example of an airline reservation system. This approach to problem solving and its concept of actors and messages is a forerunner to object-oriented programming. Kornfeld and Hewitt (1981), discuss modeling of distributed problem solving on the scientific community, specially the idea of resources constraints modeled after proposal and sponsors, knowledge sources (agents) are called sprites and are triggered when certain fact(s) are asserted on a global data base (not called blackboard). The work was based on ACT1 and is a forerunner ACT2 system design. Keen (1981), discusses causes of social inertia in organizations in relation to information systems. He also discusses the problems of implementation that result in systems being a technical success but organizational failures.

Fox (1981), compares the distributed computer systems and human organizations. Here, Organization's theoretic concepts are introduced, and these ideas are compared to the concept of the HEARSAY-II system. He further elaborates the organizational structure and control, starting from single-person, to group, to a hierarchy, to a multi-division hierarchy. These evolve with the complexity of the task and are explained with reference to the HEARSAY structure for speech understanding. He also elucidates factors which influence the choice of a particular organization, with emphasize on the complexity and uncertainty of reduction techniques. Hammond et al.
(1987) compares the rationality of a person's intuitive judgment under uncertainty with analytically derived answers produced by a formal model such as Baye's theorem, a multiple regression equation, or other rules from the conventional probability calculus. That is, they compare a person's intuitive process and judgments with those of an analytically derived rule or equation put forward as a standard of rationality. Huber (1983), gives another view in his paper. He analyzes individual characteristic (e.g. cognitive style) as predictors of human behavior.

Researchers have not been able to predict consistently behavior/performance on the basis of individual personality characteristics. Rather behavior appears to be (to a very large degree) determined by the characteristics of the task in which the individual is involved (Robbins 1996). Hudlicka et al. (1987), presents a new model in which the distributed system is able to function effectively, even though processing nodes have inconsistent and incomplete views of the databases necessary for their computations. Hewitt (1991), describes how distributed AI deals with issues of large-scale open systems. This open system is called open information systems since they receive information from outside themselves at any time. The author tries to have one framework for commitment, conflict, negotiation, cooperation, and distributed problem solving. The author also explains the limitations with respect to distributed AI.

Rouse and Hammer (1991), discusses the impact of modeling limitation on intelligent systems. He shows, how limitations of the physical, operational, and behavioral models that underlay intelligent systems can lead to undesirable consequences in terms of the behaviors of intelligent systems. Models of physical, behavioral, and operational process often form the basis of developing intelligent systems. These models are used to automate, offer expert advice and to provide explanations. Ten types of modeling limits are identified and are discussed by him. Demazeau and
Rocha Costla (1996), defines the multiagent systems as an Open Multiagent Systems, in which agents enter and leave freely. They introduced population-organization model, which uses a toolbox approach to agent oriented programming.

Jones and Marsh (1997), proposed a theoretical model called trust model in Computer Supported Cooperative Work (CSCW). CSCW is concerned with supporting the activities of work group through the use of computer technology. It emphasize technological issues of support at the expense of social issues such as relationship, roles, and social protocols. The trust model used in CSCW is characterized by eight parameters. Fukuda and Iritani (1994), have designed the intention model for collective behavior in group robotic system and addresses its application. They use self-recognition concept to organize the behavior in dynamic environment for the decision making of the behavior in a robotic group. In addition, the authors incorporated the evolution group behavior with the intention model. Zhi-Dong (1994), describe a multiple vehicle robot system for cooperative object manipulation. This system consists of a host and several distributed behavior-based vehicle robot agents. A strategy on cooperatively organizing the behavior based robots having only limited ability in manipulation. Simulation results are tabulated in this paper. The models discussed above concentrate on various issues of management and a theoretical model has been proposed for trust factors. However, in the proposed work a model called Behavior Model has been designed to cater to the above factors. The model computes the above factors based on the available knowledge and knowledge gained by the past experience.

2.3.2.5 Reasoning

Computer reasoning programs usually construct computational models of situations. To keep these models consistent with new information and
changes in the situations being modeled, the reasoning programs frequently need to remove or change portions of their models. These changes sometimes lead to further changes, for the reasoned often constructs some parts of the model by making inferences from other parts of the model. Research in distributed AI concentrate on understanding the knowledge and reasoning techniques needed for intelligent coordination.

Kleer (1986a) presented a new view problem solving motivated be a new kind of truth maintenance system. It is different from the TMS (Doyle 1979) discussed above, in that it is in addition based on manipulating assumption sets, and is called Assumption-based TMS (ATMS). Conventional TMS (discussed above) insist on consistency, but ATMS, it allowed one to prove all assertions even in presence of an inconsistency. Further, Kleer (1986b), extends this ATMS. Here, the ATMS is designed to function in tandem with a problem solver as part of an overall reasoning system. The problem solver constructs records of all the inferences it makes (justifications) and hypothesis it introduces (assumptions). The task of the ATMS is to determine, given the inferences that have been made so far, all possible context and their contents. One reason for model updating is the detection of contradictory information.

YI Deng (1990), discussed a new model called G-Net Model for knowledge representation and reasoning. It has several important features: i) all reasoning algorithms are based upon Petri net firing rules. ii) supports different reasoning process in a mixed-type knowledge hierarchy. iii) the model can be implemented with ease using the knowledge table representation.

Reasoning plays an important role in DAI systems, as the system consist of group of intelligent computational agents. Huhns and Brigland (1991) developed a justification based TMS (JTMS). In JTMS, every datum
has a set of justification and associated status of IN(believed) and OUT(disbelieved). The authors also discussed the distributed TMS (DTMS) algorithm. Further, the idea is extended from the truth maintenance for use in multiagent domains. Mazer (1991), proposes an interdisciplinary approach to understanding distributed AI systems. He integrates some theoretical ideas about knowledge with experimental ideas about action, negotiation, and problem solving in a multiagent context.

MacIntosh et al. (1991), presented a distributed automated reasoning system (DARES), for distributed environments that gives an agent the ability to reason beyond the limitations of its local knowledge. It emphasizes on task coordination and agent cooperation. Here, each theorem proving agent begins with a subset of the initial axioms, and no agent begins with sufficient information to prove a target theorem alone. The authors describe heuristics for determining whether progress toward proving a theorem is being made, and methods for formulating requests for more information. Experimental results are also presented to illustrate the performance of the DARES system. Munindar (1993), presented a declarative representation model for Multiagent systems. The declarative model at each agent specifies what the agents know and what capabilities they have. The model is demonstrated with an example. Probabilistic reasoning is another approach. Bayesian reasoning is on such method. As the reasoning is beyond the scope of this proposed work, it is not considered.

2.3.2.6 Decomposition

In distributed AI, most of the approaches proceed with the assumption that the problem is already decomposed and concentrate on other issues (Gasser 1992, Uma 1996b). However decomposition, which is partitioning of the problem to be solved so that it can be distributed to individual problem solvers, is an important issue of distributed AI.
Decomposition methods can be broadly classified as (i) general methods and (ii) domain specific methods. The general methods including algorithmic/hierarchical decompositions, give only the broad steps (hierarchical) however and as these steps split further the dependency relationship among them also increases. These methods do not make use of background and heuristic knowledge to decompose the space intelligently. As these methods use top-down approach, they are well suited only if the problems are structured problems (Myers 1978, Corkill 1979). Domain specific methods on the other hand make use of application specific knowledge. One such method is the ad hoc procedure. Here the problem is decomposed in an ad hoc manner, which is very specific to the application (Winston 1993, May 1989). Another way of task decomposition is functional decomposition. It refers to grouping the classes of generic task, that is, based on the behavior of the agent. It enforces precedence relationships between the subtasks (Uma 1996a, Malone 1988). Lastly, Object-Oriented decomposition (Booch 1994, Gasser 1992) is done on the basis of the key abstracts in the problem domain. The method is viewed as a set of autonomous agents that collaborate to perform some high level behavior. But it is very difficult to find the key abstraction of objects. The above decomposition methods do not support background knowledge and the knowledge gained during each steps fully. The proposed decomposition method completely utilizes the above two types of knowledge in order to improve the performance of the whole system.

2.3.2.7 Learning

Learning is one of the most important characteristics of human and machine intelligence. Machine learning is to help acquire new knowledge new skill, new ways of organization of existing knowledge (Winston 1993). Learning takes place as a result of the interaction between the agent and the world, and from observation by the agent of its own decision-making process (Russell 1995). Several learning methods have been evolved, like learning
from examples (Russel 1995), construction of decision trees (Russell 1995),
learning by analogy (Winston 1993), learning by discovery (Frawley 1991),
etc. In this work, the collaborators learn from experience.

2.4 COLLABORATIVE WORK

Many of the real world problems are either too large or complex in
nature. Since such problems cannot be handled by an expert due to inherent
vagueness. Hence there will be a group of collaborators collaborating
together to solve a problem (Ernest 1994). In addition to technical challenges,
GroupWare possess this fundamental problem for product developers,
because of new challenges arises from individuals direct involvement in the
group processes. Jonathan (1994) briefly outlines the GroupWare. He
describes eight specific problem areas and finally examines GroupWare
successes in search of better approaches to supporting work in group settings.
Commercial Computer Supported Cooperative Work (CSCW) products such
as the Coordinator and the PC-based software are often referred to as
eamples of GroupWare. The CSCW looks at how groups work and seeks to
discover, and how technology can help the work. Krant and Streeler (1995),
make a survey on cooperation in software development, and they discuss the
features of the projects and the coordination practices that influence the
sharing of information and goals.

But the Web changes these assumptions through geographically
distributed members, low software distribution costs, new medium of
software distribution, and Internet based collaboration tools etc., (Oriezy
1997). Fielding and Kaiser (1997), discusses collaborative work. Here, a small
group of Webmasters (called Volunteers) gathered together via the Internet
to coordinate changes and produce a common distribution. Fujitsu (Jerry
1999) has taken the first step in constructing an enterprise-wide,
Internet-based infrastructure that allow teams around the world to
 colaborate on every phase in the life cycle of a global software product. The
primary goal is to establish a software environment that has a configurable system infrastructure and flexible information repositories that support a set of collaborative software tools on a distributed network. As the proposed work concentrates on distributed environment, the internet based collaborative environment is considered and the collaborators are assumed to be distributed over the internet.

2.5 SOFTWARE AGENTS

Several researchers have worked in the area of intelligent agents. Most of the agents are designed for World Wide Web. O'leary (1997, Huhns 1991, Maes 1997) discusses the role of AI in making WWW useful. At present there are a number of Web based AI applications: intelligent search engines and browser, learning agents, and knowledge sharing agents etc. He classified the agents as general-purpose agent and special-purpose agents. Riecken (1994) was the guest editor of the special issue on intelligent agents, Communications of the ACM, discussed various types of agents.

Norman (1994) describes human-agent interaction in his paper. He discusses two social issues: one deals with the way people feel about agents (feel of control), the other issue deals with comfort and acceptance of automatic autonomous actions. Etzioni and Weld (1994) introduced on Internet based intelligent agent call softbot (software robot) in their paper. It is an AI agent developed in University of Washington, which uses a Unix shell and the World Wide Web to interact with a wide range of Internet resources. Effecting which include ftp, telnet, mail, and file manipulation commands; and Internet facilities such as archie, gopher, netfind etc. The agent can backtrack from one facility to other. Mitchell et al. (1994) described the design of one particular learning assistant; a calendar manager, called CAP (Calendar Apprentice), that learning its users scheduling preferences from experience. It provides interactive access to an on-line calendar and to e-mail. Users can edit the calendar by adding, deleting, removing, copying
and annotating meetings and they can mark various calendar events as either tentative or confirmed. COACH (Selker 1994) (Cognitive Adaptive Computer Help), is a system designed in the paper (Selker 1994). It watches the user actions to build an adaptive user model that selects appropriate advice. It is an advisory system that does not interfere with the users actions, but comments at certain times to help the user. There are several industry trends that have influenced our thinking about agents. Kautz (1994), discusses a bottom-up design method design software agents. The agent may be User Interface (UI) to system that uses planning algorithm to generate shell scripts. The agents will assist users in a range of activities such as setting up meeting, sending out messages, locating information in multiple databases etc.

Hendler (1996) was the guest editor for the special issue on intelligent agents in IEEE Expert Intelligent System and Applications. The papers in this issue discusses intelligent agents, with an emphasis on the relationship between AI and Information Technology. The News agent (O'Leary 1997) probably one of the most widely used agent, is an agent that helps the user to select article from a continuous stream of news. The meeting scheduling agent, assist a user with the scheduling of meetings (accept/reject, schedule, reschedule, negotiate meeting time etc.). Agents are also designed as reusable agents. Sycara et al. (1996), have developed a reusable, multi-agent system that self-organized and cooperate in response to task requirements, called as RETSINA, trust address the issues of distributed information gathering in an open world environment. Lander and Lesser (1997), designed a reusable multi-agent system in which expert agents are dynamically selected from a library and integrated with minimal customized information.

Further, the agents where designed as mobile agents. (Kiniry 1997), presents JAVA mobile agent. Java based mobile system help in providing an agent server. These agent can migrate from server to server carrying their
state with them. Agents can load their code from variety of sources. They will run with all computers compatible with Java 1.1 runtime. In order to avoid the overhead of interactions between the collaborators and to reduce internal workload, agent aspect is incorporated into the collaborators as an integral part.

2.6 APPLICATIONS OF DISTRIBUTED AI

Distributed AI is characterized by the interaction of many agents trying to solve a variety of problems in cooperative fashion. Durfee and Lesser (1988), considered a distributed vehicle monitoring application. They describe the experimentation of distributed problem solving techniques using the vehicle-monitoring task with information collected by a geographically distributed network of sensor nodes. This paper details the concept of distributed processing versus distributed problem solving, and is pilot experiment to further illustrate the features and potential of such techniques to AI research in general. The paper (Hogg 1991) describes a simple and robust procedure for chaotic behavior in systems composed of interacting agents making decisions based on imperfect and delayed information. ARCHON (Architecture for Cooperative Heterogeneous On-line system) (Jose 1996, Corera 1996) addresses both of these issues. ARCHON provides a decentralized platform that offers necessary control and level of integration to help the sub components work together.

2.7 CONCLUSION

In this chapter, various approaches to solve problems in distributed AI systems have been highlighted. Summaries of problems that are faced by collaborative work is given in this chapter. Different types of agents and their approaches are also brought out in the chapter. Behavior modeling which has received comparatively less attention from distributed AI perspective, has been considered in this work. The following chapter gives the overall view of the work presented in this thesis.