Coordination, Cooperation and Conflict Resolution in Multi-Agent Systems

W. Alshabi[*], S. Ramaswamy[+], M. Itmi[*], H. Abdulrab[*]

Abstract - In this paper, we present a condensed survey of multi-agent systems, with special emphasis on cooperation coordination, conflict resolution and closely related issues; issues that are critical for the development of large-scale, distributed complex software systems. We then present three different cooperative MAS architecture types, discuss their drawbacks and propose the need for a service driven framework for the development of cooperative multi agent systems.

Keywords: Multi-agent systems, cooperative systems, service computing, coordination, conflict resolution.

1. Introduction

The development of "stand-alone systems" that solve problems with minimal help from the outside environment have traditionally been brittle in nature. Predominantly AI has encountered such brittleness by injecting more knowledge into the system, including common-sense knowledge, to enlarge the system's range of capabilities. However, attempts to develop larger and more complex and intelligent systems have revealed the shortcomings and problems of centralized, single-agent architectures and current agent based development practices. Such strategies are very shortsighted in general and the ability to flexibly team-up and coordinate group activities toward individual and collective goals is a hallmark of natural intelligence [25]. Research in distributed artificial intelligence (DAI) therefore concentrates on understanding the knowledge and reasoning techniques needed for intelligent coordination, and on embodying and evaluating this understanding in computer systems. Multi-agent systems (MAS), may be regarded as a group of entities called agents, interacting with one another to achieve their individual as well as collective goals. Decentralization causes other serious problems, such as conflicts among the agents and their respective goals. This is because the knowledge contained in each agent might be incomplete, and goals of agents might be in conflict. Therefore, conflict resolution is a critical and implicit problem in MAS. Thus, this paper is an attempt to integrate the various issues and flavors of MAS and propose an enhanced MAS framework for MAS that allows large-scale cooperative behaviors. Our recent work on several related issues are reported in [56-62].

In MAS, agent interaction [80] is generally governed by various needs such as cooperation, competition or coexistence [73, 74, 91] in order to jointly carry out a required task or to achieve a particular goal. Agent interaction may be via direct communication, by means of an intermediary agent or indirectly by actions carried out in the environment. This definition indicates that there are three dimensions that characterize an agent: its goals, its capacities to carry out certain tasks and its available resources. Interactions among agents in MAS are justified by their interdependence accordingly along these three dimensions [75]: *(i)* Goals Compatibility, i.e. the MAS problem is to determine whether or nor the respective goals of the various agents in the system are compatible. *(ii)* Agent Capacity, i.e., the MAS problem is task accomplishment through agent interaction. *(iii)* Resource Relationships, i.e., the MAS problem is the identification and resolution of agent conflicts.

This paper is organized as below: Section 2 different aspects of MAS environments with particular focus on conflict resolution. Section 3 deals with architectures and frameworks for establishing agent cooperation. Section 4 introduces and motivates a service based framework to enable agent cooperation & coordination. Section 5 concludes the paper.

2. Multi Agent Systems, Conflict Resolution and Agent Cooperation

MAS may be comprised of homogeneous or heterogeneous agents, MAS is considered as crucial technology for the effective exploitation of the increasing availability of diverse of heterogeneous and distributed on-line information sources. MAS can also be a framework for building large, complex, and robust distributed information processing systems which exploit the efficiencies of organized behaviour. [2, 3, 13, 15, 31, 37]. Teamwork and communication are two important processes within multi-agent systems designed to act in a coherent and coordinated manner. The need for responsive, flexible agents is pervasive in many application domains due to their complex, dynamic, and uncertain nature of the environment. Sensible Agents [13, 15, 31] are MAS systems designed for domains with a high level of dynamism and uncertainty. Autonomous and interactive characteristics of agents do allow widespread applications for agent-based applications [1, 4, 11, 36]. The immediate application of planning and scheduling to real-world problems has been a motivational factor in using MAS as proof-of-concept for application designs [5, 6, 9, 10].

---

* LITIS Laboratory, INSA – Rouen, 76131 Mt. St. Aignan Cedex, Rouen, France. This work was supported, in part, by INSA-Rouen – for supporting Dr. Ramaswamy as a visiting professor at INSA during June 2006. Email: waledhaz@hotmail.com, {itmi, habib.abdulrab}@insa-rouen.fr

+ Knowledge Enterprises for Scalable Resilient Infrastructures (KESRI), Computer Science Department, Univ. of Arkansas at Little Rock, AR 72204 USA.. This work was supported, in part, by an NSF (#CNS-0619069) and the Graduate Institute of Technology at UALR. Email: srini@acm.org

Conflict resolution (CR) includes conflict detection, search for solutions, and communication among agents to reach an agreement with regard to the CR solution to be pursued. Due to the basic characteristics of multi-agent systems, conflict resolution is a common phenomenon in multi-agent systems [35, 37]. Application domains in which multi-agent system technology is appropriate typically have a naturally spatial, functional or temporal decomposition of knowledge and expertise.

2.A. MAS and Conflict Resolution

Research in conflict resolution of multi-agent systems has been approached from three distinct perspectives: organization autonomy [43, 44, 48, 51, 56, 58, 59]., non-cooperative domains [30, 46] and cooperative multi-agent systems [4, 8, 11, 35, 36, 46, 49, 54, 55]. Research in conflict resolution of cooperative multi-agent systems has been approached from three distinct perspectives: distributed decision-making[52], model description [12, 16, 64-65], and applications[7, 14, 17-19, 20-24, 26, 27, 29, 32-34, 38-42, 45, 47, 50, 53, 63, 66-69].

2.B. Agent Cooperation

Cooperation is a key MAS concept [72, 77, 79, 80]. Durfee and colleagues [78] have proposed four generic goals for agent cooperation: *(i)* Increase the rate of task completion through parallelism; *(ii)* Increase the number of concurrent tasks by sharing resources (information, expertise, devices, etc); *(iii)* Increase the chances for task completion by duplication and possibility using different modes of realization; *(iv)* Decrease the interferences between tasks by avoiding the negative interactions. However, cooperation in agent-based systems is at best unclear and at worst highly inconsistent [49]. Researchers like Galliers [82, 83] and Conte [76] underlined the importance of adopting a common goal for agent of cooperation which they consider as an essential element of the social activity. We can characterize a MAS system by its type of cooperation it implements which can range from total cooperation to the total antagonism [84]. Completely cooperative agents can change their goals to meet the needs of other agents. Antagonistic agents, on the other hand, will not cooperate and, their respective goals may be blocked.

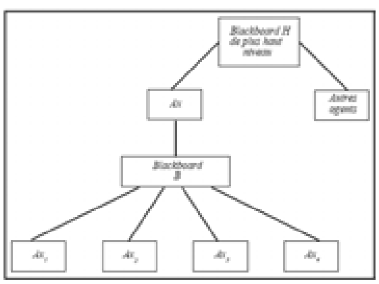

Figure 1: HOPES Model

3. Architectures, Frameworks for Cooperative MAS

Bond [86] describes the existence of two types of MAS architectures: *(i) Horizontal*: This structure is useful in some contexts, for example, a situation where a group of agents having different (non-overlapping) capabilities and hence can work towards the goal without needing any conflict resolution. Here all the agents are on the same level with equal importance without a Master / slave relationship. *(ii) Vertical*: In a vertical architecture, the agents are structured in some hierarchical order. Agents at the same sub-level may share the characteristics of a horizontal structure. The 'horizontally structured' MAS model has several issues –a critical issue is that it quickly becomes too complex and unwieldy for practical applications, wherein agents in the MAS may share some common capabilities. Hence most current frameworks have adopted a hierarchical MAS model (vertical) by organizing the agents in some *organizational structure*.

3.A. MAS Architectures

Here, we compare three widely used models for agent cooperation in MAS: HOPES, HECODES and MAGIC.

HOPES [70] (Hierarchically Organized Parallel Expert System) is well adapted to the needs of problem resolution at a given level. The interaction between agents is carried out by means of blackboard (blackboard) divided between agents at several levels. For example, inFigure 2, agent *Ax* will be responsible for the centralization of the results of the cooperation between the agents *Ax1, …, Ax4* using *blackboard B*. Results are posted on the blackboard to be used by the agent *Ax*, but *B* is also used by the agents of the lower level which, in their turn, interact for the collective resolution of a task. Agents use *B* like a channel during their intra-group communications. Other agents on the same level as *Ax* can want to use this result or to collaborate with *Ax* in another way. If so, they would use another *blackboard H* of higher level. This stratification can be repeated for blackboards at multiple levels.

HECODES (HEterogeneous COoperating Distributed Expert System) [70] is a suitable environment for a horizontal, hierarchical and recursive co-operation. It is also a blackboard-based system and presents a system for centralized control. The cooperating agents can be heterogeneous with respect to their control strategy, their methods for knowledge representation and the programming language used. In Figure 4, the expert system, the control subsystem and the blackboard subsystem represent the agent network interconnected by communication channels. Each expert system can provide solutions and solve local problems in an autonomous way by using its own domain-specific knowledge. However, each one would need the assistance of the other agents or can provide services to other agents through its control subsystem. There are interfaces between the expert system and the control subsystem, which are used as communication interfaces or man-machine interfaces for the management of heterogeneous expert system. The blackboard subsystem centralizes the information shared by the expert systems. The

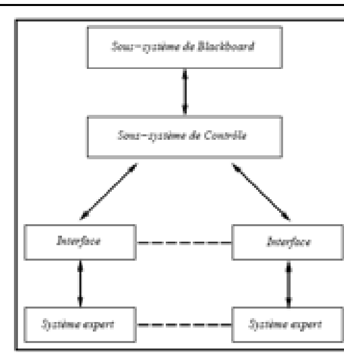

Figure 2: HECODES

control subsystem is responsible for the management of the cooperation and the communication between the agents and monitoring execution times. In practice, the subsystems of blackboard and control are gathered at a central location to minimize communication delays.

A model of hierarchical multi-agents architecture (MAGIC) is another model of hierarchical agent organization [87-88]. In this structure it is possible to set up *powers of delegation* between agents, thus facilitating the development. In MAGIC, an agent is an entity having a certain number of competences. These competences make it possible for the agent to hold a role in a MAS application. The competences of an agent can be evolved/moved dynamically (by exchanges between agents) during its existence, which implies that agents can play different roles (thus improving their stature) within the MAS. An agent is dynamically built from an elementary agent “vacuum”, by enriching itself by competency acquisition. MAGIC uses an agent termed supervisor that is charged with connecting various qualified agents and in allowing an invocation of method (for competence). *The invocation of competences* implies that an agent has to achieve a task which requires the exploitation of a certain number of competences that it may (or may not have) by itself. This means that the mode of invocation of competences is established by a hierarchical organization. As a consequence, when an agent is seeking to realize a competence it does not necessarily have or know explicitly, the agent that will carry out a search for this competence. The supervisor is given the responsibility to find the competence necessary to address this need. The invocations are not named and one agent will not have to know “who makes what”. The principle interaction mechanism is therefore, as follows:

1. If the agent has the required competence, it calls upon this directly. *(i)* If the existence of the competency at another agent is known, an appropriate support request is initiated. *(ii)* Otherwise, help of the supervisor is sought. If the supervisor knows of the existence of the competency at another agent, an appropriate support request is initiated. *(iii)* If not, the superior recursively tries to find another qualified supervisory member from higher up in the hierarchy, and then the same mechanism is reapplied.

In MAGIC, the reorganization of the agents’ structure can be to some extent dynamic. Bonds between agents can be created when relations appear between two agents of the hierarchy (Figure 3). That causes to remove the recurring communications along the tree structure and this allows direct agent communication. The decision of creating such links is a prerogative of the agents themselves.

3.B. Shortcoming of Above Frameworks

HOPES and HECODES are based on a blackboarding technique [89]. The agent is reduced to a knowledge source and the system is made up of the entire rules that form the base of knowledge. Blackboard allows a centralized control and hence can be more effective. But on the other hand, it represents a classic bottleneck and the sources of knowledge are not locally available. Agents do not have any mechanism for direct communication for information exchange and to cooperate transparently. The centralization by means of a blackboard allows for indirect communication via a shared structure and hence has a rigid centralized structure. MAGIC is more appropriate for better resolution of distributed problems with autonomous agents. Nevertheless, MAGIC suffers from several weaknesses for real world applications. *(i)* **Acquaintances link:** Specific links when established between agents, allows for direct communications. This means that if new resources are added, or if other agents which are better suited to a need arrive within a group, this may be unknown to the agent. This may cause performance degradation over time. *(ii)* **Zone Restriction**: When there is a task competency required that an agent does not possess, we need to traverse the hierarchy in order to seek the necessary competence. MAGIC places no bounds on the search scope. This can be a time-consuming activity and not very appropriate for applications that may require ‘time-sensitive’ answers. Selectively exploring availability of necessary competencies in restricted ‘search zones’ is impossible. In the worst case, if the required competency is non-existent, the request can reach a standstill at the root until a qualified agent joins the structure. *(iii)* **Redundant Competencies:** If there are several agents in the system that possess the same competences, there is no mechanism that allows for choice of the agent that will carry out the needed task. The first agent that is identified is the one that is used. In most practical applications, is imperative to know which agent will carry out the task effectively and a choice/selection mechanism for using quite precise comparative criteria needs to exist. *(iv)* **Essentiality of a Supervisor:** Similar to a blackboarding system, a supervisor makes it possible to coordinate agent interactions. However there is no mechanism to detect the breakdown of a supervisor agent; hence some agents can become isolated.

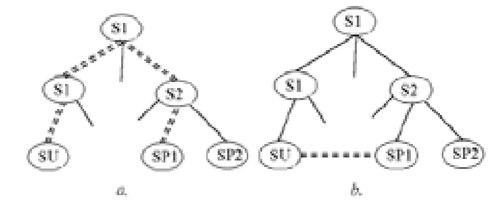

Figure 3. MAGIC

4. Our Framework

The co-operation between agents, to generate a coherent total behaviour of the MAS, requires an elaboration mechanism of coordination so to avoid potential conflicts and to support the synergy of the activities of agents while enabling them to profit from their respective capacities, and to benefit from the actions of the ones and others.

For this reason, we introduce the concept of coordinator agent (CA) into our model. It plays a central function in our approach. It constantly holds the total state of its group but has a partial and limited vision system. Its role is well to manage the interactions of its group of agents (the under-hierarchy) and to ensure their coordination like guiding each sending of request for resource towards the suitable local staff. It makes it possible to gather and put in connection several local staff. Each CA has a table made up of the various local staff of its group indexed by the type of resource that it manages.

We mentioned earlier that Durfee and colleagues [78] have proposed four generic goals to establish agents' cooperation. However, an important issue to consider with respect to supporting tasks *(ii-iii)* is that agents need to be designed with the inherent ability to 'share' pertinent information with other agents in the environment then the resulting efforts by another agent in establishing successful agent cooperation can be vastly improved. Hence, we propose an alternative network of agents architected using a service oriented framework. **Error! Reference source not found.** presents a service based agent architecture model in which each agent can be an autonomous entity having control over its own resources and associated skills which enable the agent to cooperate, to communicate and interact with the other agents; agents can be divided into many groups, and they cooperate in different levels; each group of agents consists of several cooperative agent members and a public superior member, this last detains the global state of its group but possesses a partial and limited vision of the system.

Our model comes over the limitations of hierarchical agent architectures, also allows for intelligent information transfer among multiple agents that provide appropriate services. Identification of such information is the key to orchestrating cooperative interactions between agents. Each individual agent in the community of cooperating agents can then appropriately and contextually define its choice of balance (one of the gray triangles in **Error! Reference source not found.**) between cooperation and autonomy. **Error! Reference source not found.** illustrates the two competing goals in MAS namely agent autonomy versus cooperation in MAS. Each agent then is capable providing services autonomously. An application over such an agent network is then represented by a dynamic tree-based data structure of agents that receive and provide various services to accomplish a task / goal. The roots of the tree are coordinator agents. The agents' connections of bonds, which can be between local agents or coordinators agents, form the group. Intermediate and roots nodes are appropriate coordinator agents, which has a pseudo- global view of the system. A bidirectional link represents the existence of a bidirectional channel of communication between two agents.

4.A. Advantages of Proposed Framework

Our proposed model vastly differs from the HOPES and HECODES models in that all communication between the agents of the group are directed by dynamic coordinator agent and not by a static blackboard, which completely eliminates the disadvantages of the blackboard. Coordinator agents have a principal role - requests routed through these agents offer the best possible solution. Nevertheless, since the links are created at run time, a bottleneck at the coordinator agent is avoided. An agent will be regarded as 'unavailable' if it does not answer a request over a certain time. At the time of research of a solution, the request can contain a delimitation of the zone of research. It is an important point in the case of transport. In case, we have several agents that have the same types of resources, a mechanism of decision based on the negotiation is used to define the optimal solution. The division of the agents in several groups, in order to have an indirect communication, aims to reduce the number of communication and to have a less complex structure interns.

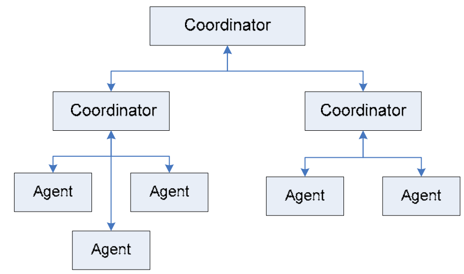

Figure 5. The dynamic hierarchical model

They allow for a coherent coordinated behavior of the MAS. Since the interactions are at run time, there needs to be appropriate load balancing mechanisms available. Agents may be able to self-augment a request with appropriate work zone / time delimitations**.** When several agents with similar resources can provide a needed service, a costing mechanism can be used. The self-aggregation of agents in several groups is nicely supported, communications are also less complex.

In case of conflict, the agent must enter into a negotiation with the conflict group. Various protocols for negotiation exist in the MAS literature. In [59], three different techniques for complex negotiations are presented. These include: *(i)* negotiation through an arbitrary leader - in this method an arbitrary leader is selected for arbitrating the conflict (interference) resolution process between the agents, *(ii)* negotiation through chaining – in this method a ranked order assigned to each agent based on when they join the group is used for conflict resolution, and *(iii)* negotiation through cloning - in this method each agent creates a 'restricted' clone (agent for negotiation) and passes them to every other agent in the group.

5. Conclusions

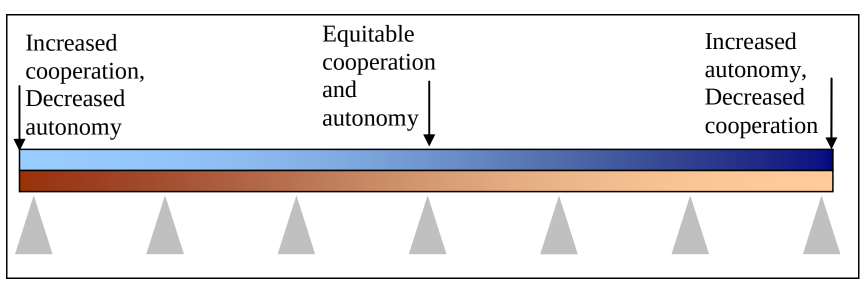

Figure 4. Agent Cooperation versus Autonomy

Cooperation is a key process for multi-agents systems research. In this paper, we have presented a hierarchical multi-agents model which aims to overcome the limitations of hierarchical agent architectures, to allow for intelligent information transfer among multiple agents and to answering for other applications which wasn't concerned by the previous models.

6. References

1. S. I Morisbak and B. Tessem, "Agents For Case-Based Software Reuse," Applied Artificial Intel., 15(3): 297-332, 2001.
2. K. E. Biggers and T. R. Ioerger, "Automatic Generation of Communication and Teamwork within Multi-Agent Teams," Applied Artificial Intelligence, 15(10): 875-916, 2001.
3. G. Nitschke, "Cooperating Air Traffic Control Agents," Applied Artificial Intelligence, 15(2): 209-235, 2001.
4. P. Stone, M. L. Littman, S. Singh, M. Kearns, "ATTac-2000: An Adaptive Autonomous Bidding Agent," Jour. of AI Res., 15: 189-206, 2001.
5. Garrido and F. Barber, "Integrating Planning and Scheduling," Applied Artificial Intelligence, 15(3): 471-491, 2001.
6. P. Priore, D. De La Fuente, R. Pino and J. Puente, "Learning-Based Scheduling Of Flexible Manufacturing Systems Using Case-Based Reasoning," Applied Artificial Intelligence, 15(10): 949-963, 2001
7. K. S. Barber, T. H. Liu, and S. Ramaswamy, "Conflict Detection During Plan Integration for Multi-Agent Systems," IEEE Trans. on Systems, Man, and Cybernetics, 31(4): 616-628, August 2001
8. R. Liu, M. J. Shih, and Y. Kao, "Adaptive Exception Monitoring Agents for Management By Exceptions," App. Art. Intel., 15(4): 397-418, 2001.
9. J.L. Ambite, and C. A. Knoblock, "Planning by Rewriting," Jour. of Art. Intel. Res., 15: 207-261, July-Dec. 2001.
10. I. Refanidis and I. Vlahavas, "The GRT Planning System: Backward Heuristic Construction in Forward State-Space Planning," Jour. of Artif. Intel. Res., 15: 115-161, July-Dec. 2001.
11. B. Zhang and Y. Seo, "Personalized Web-Document Filtering Using Reinforcement Learning," Appl. Artif. Intel., 15(7): 665-685, 2001.
12. J. Broersen, M. Dastani, J. Hulstijn, Z. Huang, and L. V. Torre, "The BOID Architecture: Conflicts Between Beliefs, Obligations, Intentions and Desires," 5th Intl. Conf. on Autonomous Agents 2001, pp. 9-16.
13. K. S. Barber, R. McKay, M. MacMahon, C. E. Martin, D. N. Lam, A. Goel, D. C. Han, and J. Kim, "Sensible Agents: An Implemented Multi-Agent System and Testbed," Proceedings of the Fifth Inter. Conf. on Autonomous Agents 2001, pp. 92-99
14. K. S. Barber, T. H. Liu, A. Goel, and C. E. Martin, "Conflict Representation and Classification in a Domain-Independent Conflict Management Framework," Proc. of the Third Inter.Conference on Autonomous Agents 1999, pp. 346-347.
15. K. S. Barber, J. Kim, "Constructing and Dynamically Maintaining Perspective-based Agent Models in a Multi-Agent Environment," Proc. of the 3rd Intnl Conf. on Auto. Agents 1999, pp. 416-417.
16. H. Jung, M. Tambe, S. Kulkarni, "Argumentation as Distributed Constraint Satisfaction: Applications and Results," Proc. of the Fifth Intnl Conf on Autonomous Agents 2001, pp. 324-331.
17. C. Huang, J. A. Ceroni, and S. Y. Nof, "Agility of Networked Enterprises - Parallelism, Error Recovery and Conflict Resolution," Computers in Industry, 42(2-3): 275-287, 2000.
18. Y. Yan, T. Kuphal, and J. Bode, "Application of Multi-agent Systems in Project Management," International Journal of Production Economics, 68(2): 185-197, 2000.
19. G. W. Tan, C. C. Hayes, and M. Shaw, "An Intelligent-Agent Framework for Concurrent Product Design and Planning," IEEE Tran. on Engineering Management, 43(3): 297-306, 1996.
20. E. C. Lupu, and M. Sloman, "Conflicts in Policy-Based Distributed Systems Management," IEEE Transactions on Software Engineering, 25(6): 852-869, November / December 1999.
21. G. J. Pappas, C. Tomlin and S. Sastry, "Conflict Resolution for Multi-Agent Hybrid Systems," Proceedings of the IEEE Conference on Decision and Control v2 1996, pp. 1184-1189.
22. J. K. Kuchar and L. C. Yang, "Survey of Conflict Detection and Resolution Modeling Methods," AIAA-97-3732, AIAA Guidance, Navigation, and Control Conf., Aug. 11-13, 1997, pp. 1388-1397.
23. J. K. Kuchar and L. C. Yang, "A Review of Conflict Detection and Resolution Modeling Methods," IEEE Transactions on Intelligent Transportation Systems, 1(4): 179-189, December 2000.
24. S. Cooper and A. Taleb-Bendiab, "CONCENSUS: Multi-Party Negotiation Support for Conflict Resolution in Concurrent Engineering Design," Jour. of Intel. Manuf., 9: 155-159, 1998
25. E. H. Durfee, "The Distributed Artificial Intelligence Melting Pot," IEEE Trans. on SMC, 21(6): 1301-1306, Nov/Dec 1991
26. M. A. Vilaplana, C. Goodchild, "Application of Distributed Artificial Intelligence in Autonomous Aircraft Operations," Proc. of AIAA/IEEE Digital Avionics Sys. Conf. 2001, pp. 7B31-7B314.
27. M. Klein, "Supporting Conflict Resolution in Cooperative Design Systems," IEEE Trans. on SMC, 21(6): 1379-1390, Dec.1991.
28. R. S. Sutton and A. G. Barto, "Reinforcement Learning: An Introduction," The MIT Press, Cambridge, Massachusetts, 1998.
29. A. Wallis, Z. Haag and R. Foley, "A Multi-Agent Framework for Distributed Collaborative Design," Proc. of the 7th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises, WET ICE 1998, pp. 282-287.
30. G. Zlotkin and J. S. Rosenschein, "Cooperation and Conflict Resolution via Negotiation Among Autonomous Agents in Non-cooperative Domains," IEEE Trans. on SMC, 21(6): 1317-1324, Nov./Dec. 1991.
31. K. S. Barber, R. M. McKay, A. Goel, D. C. Han, J. Kim, T. Liu, and Cheryl E. Martin, "Sensible Agents: The Distributed Architecture and Testbed," IEICE Trans. on Comm., E83-B(5): 951-960, May 2000.
32. X. Huang and J.Hallam, "Spring-based Negotiation for Conflict Resolution in AGV Scheduling," IEEE Intl Conf. on SMC, v1 1995, pp. 789-794.
33. E. Uchibe, T. Kato, M. Asada, K. Hosoda, "Dynamic Task Assignment in a Multi-agent/Multitask Environment based on Module Conflict Resolution," 2001 IEEE Intnl Conf. on Robotics & Auto., May 21-26, 2001, pp. 3987-3992.
34. H. Yongtong, L. Ping, Y. Yuhong, Z. Danian, M. Changchao, J. Bode, and R. Shouju, "A Multi-agent System For the Support of Concurrent Engineering," IEEE Conf. on SMC, 1996, pp. 959-964.
35. S. Arai, K. Miyazaki, and S. Kobayashi, "Controlling Multiple Cranes Using Multi-agent Reinforcement Learning: Emerging Coordination among Competitive Agents," IEICE Tran. on Comm., E83-B(5): 1039-1047, May 2000.
36. H. Venkataramani Johar, "SoftCord: An Intelligent Agent For Coordination In Software Development Projects," Decision Support Systems, 20: 65-81, 1997.
37. J. K. Lee and M. W. Jeong, "Intelligent Audit Planning System for Multiple Auditors: IAPS," Expert Sys. w/ App., 9(4): 579-589, 1995.
38. D. C. Brown, B. V. Dunskus, D. L. Grecu and I. Berker, "SINE: Support For Single Function Agents," Proceedings of the 1995 10th Intnl. Conf. on Apps. of AI in Engineering, Computational Mechanics Inc, Billerica, MA, USA., pp. 525-532.
39. J. Chu-Carroll, "Conflict resolution in collaborative planning dialogs," Int.. J. of Hum. Comp. Stud., 53(6): 969-1015, Dec 2000.
40. J. Chu-Carroll and S. Carberry, "Communication for Conflict Resolution in Multi-Agent Collaborative Planning," Intnl.Conf. on Multi-Agent Systems (ICMAS-95) 1995, pp. 49-56.
41. P. McDonnell, and S, Joshi, "Intelligent hierarchy: A framework for distributed shop floor control," Proc. of 4th Ind. Engg. Res. Conf. 1995. IIE, GA, USA., pp. 808-816.
42. K. S. Barber, D. C. Han, and T. H. Liu, "Strategy Selection-Based Meta-level Reasoning for Multi-agent Problem-Solving," P. Ciancarini, M.Wooldridge (Eds.): Agent-Oriented Software Engineering, First Intl Workshop, AOSE, June 2000, LNCS 1957, Springer 2001, pp. 269-283.
43. K. S. Barber, A. Goel, and C. E. Martin, "Dynamic Adaptive Autonomy in Multi-agent Systems," Jour. of Experimental and Theoretical AI, 12(2): 129-147, April 2000.
44. K. Suzanne Barber, Tse-Hsin Liu, and David C. Han, "Agent-Oriented Design," F. J. Garijo, M. Boman (Eds.): Multi-agent System Engineering, 9th European Workshop on Modelling Autonomous Agents in a Multi-Agent World, MAAMAW '99, Valencia, Spain, LNCS 1647, Springer 1999, pp. 28-40.
45. K. S. Barber, T. H. Liu, A. Goel, and S. Ramaswamy, "Flexible Reasoning Using Sensible Agent-based Systems: A Case Study in Job Flow Scheduling," Jour. of PPC, 10(7): 606-615, 1999.
46. William Kilmer, "Reaching Agreement Among Randomly Interacting Agents: Brain Models, Schemas, and Unitary Responses," Journal of Intelligent Systems, 9(3-4): 203-217, 1999.

47. J. Sun, Y. F. Zhang, and A. Y. C. Nee, “Agent-Based Product Design and Planning for Distributed Concurrent Engineering,” IEEE Int. Conf. on R&A, v 4 2000, pp. 3101-3106.
48. K. S. Barber and C. E. Martin, “Dynamic Reorganization of Decision-Making Groups”, Proceedings of the Fifth Internl. Conf. on Autonomous Agents 2001, AGENTS’01, May 28-June 1, 2001, Canada, pp. 513-520.
49. J. E. Doran, S. Franklin, N. R. Jennings and T. J. Norman, “On Cooperation in Multi-Agent Systems”, The Knowledge Engineering Review, 12(3): 309-314, 1997.
50. T. H. Liu, C. Chuter, and K. S. Barber, “Virtual Environment Simulation for Visualizing Conflict Resolution Strategies in Multiple Robot Systems,” Proceedings of the Fifth IASTED International Conference Robotics and Manufacturing, May 29-31, 1997, Cancún, Mexico, pp. 154-158.
51. K. Suzanne Barber and Cheryl E. Martin, “Dynamic Adaptive Autonomy in Multi-agent Systems: Representation and Justification,” Int. J. of Pat. Recog. and AI, 15(3): 405-433, 2001.
52. K. S. Barber, D. C, Han, and T. H. Liu, “Coordinating Distributed Design Making Using Reusable Interaction Specifications,” C. Zhang and V. – W. Soo (Eds.): PRIMA 2000, LNAI 1881, Springer Berlin 2000, pp. 1-15.
53. L. Lin-sen, Y. Hai-xun, L. Ji-xun, and T. Ming-an, “Intelligent Resolution of Cooperative Conflict,” Chinese Jour. of Aeronautics, 13(1): 24-29, February 2000.
54. M. Adler, E. Durfee, M. Huhns, W. Punch, and E. Simoudis,“ AAAI Workshop on Cooperation Among Heterogeneous Intelligent Agents”, AI Magazine, 13: 39-42, Summer 1992.
55. V. R. Lesser, “Cooperative Multi-agent Systems: A Personal View of the State of the Art,” IEEE Tran. on Know. and Data Engineering, 11(1): 133-142, January/February 1999
56. S. Ramaswamy , K. Srinivasan , P. K. Rajan , R. MacFadzean , S. Krishnamurthy, "A Distributed Agent-based Simulation Environment for Interference Detection and Resolution", Spl. Issue on Software Agents and Simulation, SIMULATION, June 2001.
57. R. Donath, A. Dean, D. Girardi and S. Ramaswamy, " Distributed Simulation of an Automated Highway System with Intelligent Vehicles", Sum. Comp. Sim. Conf., 2001.
58. S. Krishnamurthy, S. Ramaswamy, P.K. Rajan, " Implementation and Evaluation of a Dynamic Switching Protocol for Interference Resolution in Naval Radar Units", Sum. Comp. Sim. Conf., 2001.
59. V. Ramachandran, S. Ramaswamy, P.K. Rajan, Complex Negotiation Protocols for a Distributed Simulation Environment, Sum. Comp. Sim. Conf., 2001.
60. S. Ramaswamy, Y. Yan, "Interactive Modeling and Simulation of Virtual Manufacturing Assemblies: An Agent-Based Approach", Spl. Issue on "Computer Integrated Manufacturing Systems: Recent Developments and Applications", Jour. of Intelligent Manufacturing, Vol. 10, No. 6, December 1999. pp.503-518.
61. R. Maarfi, A. Dean, E. Brown, S. Ramaswamy, “A Three-tier Communication and Control Structure for the Distributed Simulation of an Automated Highway Systems”, 2002 PERMIS Workshop, NIST, MD. August 13 - 15, 2002.
62. S. Ramaswamy, R. Neelakantan, “Software Design and Testing Using Petri Nets: A Case Study Using a Distributed Simulation Software System”, 2002 PERMIS Workshop, NIST, MD, 2002.
63. R. Thomson, “Deontic Login as Founded on Tense Logic, In. R. Hilpinen, Editor, New Series in Deontic Logic: Norms, Actions and the Foundations of Ethics, D. Riedel, pp. 165-176, 1981
64. P. Cohen and H. Levesque. Intention is choice with commitment. Artificial Intelligence, 42:213–261, 1990
65. A. Rao and M. Georgeff. Modeling rational agents within a BDI architecture. In R. Fikes and E. Sandewall, editors, Proc. of the 2nd Conf. on Know. Rep. and Reas., 473–484. Morgan Kaufman, 1991.
66. J. B. H. Kwa, “Tolerant Planning and Negotiation in Multiagent Environments”, *Applied Artificial Intelligence,* No. 2, Hemisphere Publishing Corporation, 1988
67. J. B. H. Kwa, “Planning Automated Guided Vehicle Movements in a Factory”, PhD Thesis, Univ. of Ediburgh, 1988
68. E. Brunswik, “Perception and the representative design of psychological experiments”, (2nd ed.). Univ. of CA Press. 1956.
69. C. Sherif, M. Sherif, & R. Nebergall, “Attitude and attitude change: The social judgment-involvement approach”, Philadelphia: Saunders. 1965.
70. D. Bell and J. Grimson, “*Distributed Database Systems*”, Addison-Wesley, 1992.
71. A.H. Bond, “Distributed Decision Making in Organisation”, *IEEE Systems, Man and Cybernetics Conference*, Nov. 1990.
72. S. Cammarata, D. McArhur, and R.Steeb, “Strategies of Cooperation in Distributed Problem Solving”, In Karlsruhe, editor, *Proc. of the 8th Intnl Joint Conf. on AI*, v 2, pp 767-770, Aug.1983.
73. B. Chaib-draa, “Distributed Artificial Intelligence: An overview”, A. Ken, J. G. Williams, C. M. Hall, and R.Kent, eds, Encyclopedia Of Comp. Sc. & Tech., v. 31, 215-243. Marcel Dekker, Inc, 1994.
74. B. Chaib-draa, “Industrial applications of distributed AI”, Communications of the ACM, 38(11):49-53, 1995.
75. B. Chaib-draa, B. Moulin, I. Jarras, "Agent et Systèmes Multiagents", In Principles et architecture des systèmes multi-agents, JP. Briot et Y Demazeau (eds) (Hermes, Lavoisier), 2001.
76. R. Conte, M. Miceli, and C. Castelfranchi. Limits and levels of cooperation. In Y. Demazeau and J.-P. Müller, editors, *Decentralized AI 2 – Proceedings of the Second European Workshop on Modelling Autonomous Agents and Multi-Agent Worlds (MAAMAW-90),* pages 147-160. Elsevier Science Publishers, 1991.
77. Y. Demazeau and J.-P. Müller, editors. Decentralized AI 2 - Proceedings of the Second European Workshop on Modelling Autonomous Agents and Multi-Agent Worlds (MAAMAW-90). Elsevier Science Publishers, 1991.
78. E. H. Durfee and V. Lesser. Negotiating task decomposition and allocation using partial global planning. In L. Gasser and M. Huhns, editors, Distributed Artificial Intelligence Volume II, pages 229-244. Morgan Kaufmann: San Mateo, CA, 1989.
79. E. H. Durfee, V. R. Lesser, and D. D. Corkill. Cooperative Distributed Problem Solving. Vol. 4, page 83-147,Addison Wesley,1989.
80. J. Ferber. Les systèmes multi-agents, vers une intelligence collective. InterEditions, 1995.
81. K. Fischer, J. P. Müler, and M. Pischel. A Model for Cooperative Transportation Scheduling. *Proceedings of the 1st Internaional Conference on Multiagent Systems (IC-MAS’95),* San Francisco, June 1995.
82. J. R. Galliers. *A Theoretical Framework for Computer Models of Cooperative Dialogue, Acknowledging Multi-Agent Conflict*. PhD thesis, Open University, UK, 1988.
83. GALLIERS J.R. (1991) Modeling autonomous belief revision in dialogue. In Yves Demazeau et Pierre Müller (éd.) Decentralized Artificial Intelligence 2: Proceedings of the Second European Workshop on Autonomous Agents in a Multi-Agents World, Elsevier Science Pub./North Holland.
84. B. Moulin and B. Chaib-draa, “An overview of distributed artificial intelligence” In G. M. P. O'Hare and N. R. Jennings, editors, *Foundations of Distributed AI*, pages 3-54. J. Wiley & Sons: England, 1996.
85. R. G. Smith, “The contract net protocol: High-level communication and control in a distributed problem solver”, IEEE Transactions on Computers, C-29(12):1104-1113, December 1980.
86. A. H. Bond, “Distributed Decision Making in Organisation” *IEEE Transactions on SMC Conference*, November 1990.
87. N. Bensaid and P. Mathieu, “A Framework for Cooperation in Hierarchical Multi-Agent Systems”, Mathematical Modeling and Scientific Computing. Vol 8. 1997.
88. N. Bensaid and P. Mathieu, “An Autonomous Agent System to Simulate a Set of Robots Exploring a Labyrinth”, Proceedings of the 11th Intl. FLAIRS Conference, FLAIRS'98. AAAI Press
89. P. Nii “Blackboard systems“, In *Handbook of Artificial Intelligence.* Volume IV. Addison-Wesley: Reading, MA, 1989.
90. A.H. Bond, L. GASSER L , “Readings in Distributed Artificial Intelligence”, Morgan Kaufmann, San Mateo, CA, 1988.