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. Then we 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.**

I. 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.

II. 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].

Conflict resolution (CR) includes conflict detection, search for solutions, and communication among agents to reach an

---

* 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

T. Sobh (ed.), Innovations and Advanced Techniques in Computer and Information Sciences and Engineering, 495–500.
© 2007 *Springer*.

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.

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].

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 the type of implemented cooperation 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.

III. 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.

A. Frameworks for Cooperation in MAS

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

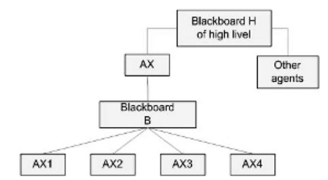

Fig. 1: HOPES Model

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 divided between agents at several levels. For example, in Fig.1, 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 used programming language. In Fig. 2, 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 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

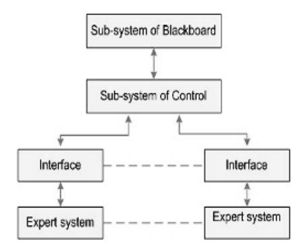

Fig. 2: HECODES