Deepthi Sehrawat al. International Journal of Recent Research Aspects ISSN: 2349-7688, Vol. 1, Issue 2, September 2014, pp. 202-205

Developing enhanced coordination between agents of MAS

Deepthi Sehrawat

Assistant Professor, CSE Dept. ASET, Amity University Haryana, India

***Abstract*— We provide a framework for planning systems, constituting a rich landscape of possible configurations, where the centralized and fully decentralized approaches are two extremes. We define and discuss agent based systems and in particular multi agent systems (MAS). We emphasize the issue of the role of MAS coordination architectures, and then explain that transportation is, next to production, an important domain in which MAS can and actually are applied. However, implementation is not widespread and some implementation issues are explored. In this manner, we conclude that planning problems in transportation have characteristics that comply with the specific capabilities of agent systems. In particular, these systems are capable to deal with inter-organizational and event-driven planning settings, hence meeting today's requirements in supply chain planning and execution**

***Keywords*— Case-based reasoning, Case retrieval, Clustering, Cure algorithm**.

I. INTRODUCTION

Agent theory first began to appear in the computer science and artificial intelligence (AI) literature in the mid- to late-1980s as an outgrowth of objected oriented and distributed AI fields1. Despite almost twenty years of history, a definition for the term agent still remains debated. Schleiffer (2005) states that: "intelligent agent technology is the articulation of human decision making behavior in the form of a computer program". While this definition is particularly elegant it is lacking in that it does not explicitly specify the characteristics of human behavior that agents seek to emulate. One of the most cited agent definitions was published by Wooldridge and Jennings in 1995. They put forth four distinct characteristics known as the weak notion of agency. These four characteristics are: autonomy, social ability, reactivity, and pro-activeness. These characteristics are widely accepted as they are at the heart of what agents represent – human decision making processes. This set of four properties has been expanded on significantly over the years and across multiple fields.

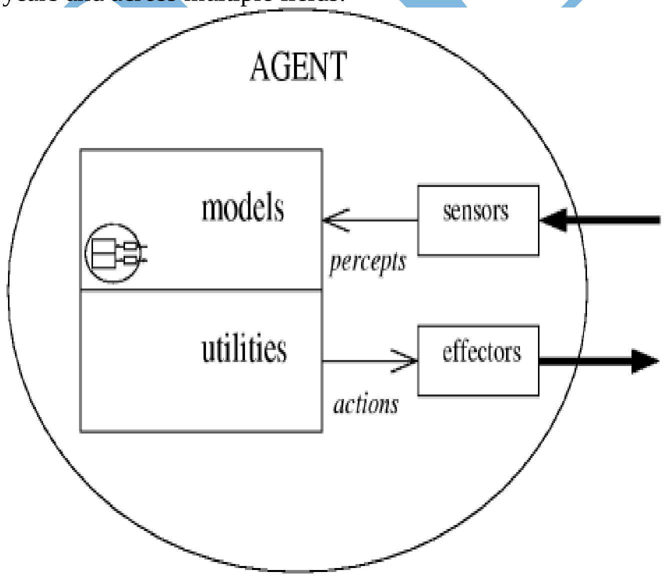

Figure 1 Intelligent Agent

This list of agent characteristics may at first seem to comprise only terms that are ambiguous in their application as part of a software system. This ambiguity can in part be clarified via a review of agent architectures. Agent architectures provide a formalized description of how an agent software entity perceives its environment and subsequently transforms this information into decisions (Wooldridge, 1999). In 1999, Wooldridge identified four primary types of agents and their corresponding architectures – logic based agents, reactive agents, belief-desire-intention agents, and layered architectures. In the first case, decision making is performed via logical deduction or theorem proving. The second case, reactive agents operate based on a maintained library mapping situations (or perceived situations) to actions. Belief-desire-intention (BDI) agents, developed by Rao and Georgeff (1991), are premised on practical reasoning – the process of manipulating data structures in an effort to decide what goals should be achieved and how those goals should be achieved. Finally, layered architectures as described by Wooldridge (1999) are architectures.

based on the premise that agents must be capable of both reactive and proactive behavior. As such, the architecture is layered with one layer handling the reactive behavior and a second layer handling the pro-active behavior. Luck and d'Inverno (2003), add to this list of four architectures with a description of autonomous agents, memory agents, planning agents, and sociological agents.

II. MAS CHARACTERSTICS

Systems consisting of multiple agents interacting with each other and the environment in which they are situated are known as multi-agent systems. In such a setting the agent construct becomes more than just an entity performing local optimization tasks – the agent must also possess the ability to communicate and coordinate. The important characteristics of a multi-agent system are, according to Rudowsky (2004): (1) Each agent's information or capabilities for solving the problem is incomplete; (2) No global control system; (3) Data are decentralized; and (4) Computation is asynchronous.

The methodologies implemented to achieve this communication and coordination are among the defining features of a MAS; as Odell (2002) put it: “designing an agent based system is not just about designing the agents, it is also about designing the agent environment.” The agent environment does not only include the different agents but also the principles and processes under which the agents exist and communicate. Agent communication is described by both the language and the method by which they exchange messages in that language. Agent coordination (sometimes referred to as “interaction”) refers to the mechanism by which agents. organize themselves to work on the problem of the full system. The following two subsections address, in turn, MAS communication and coordination.

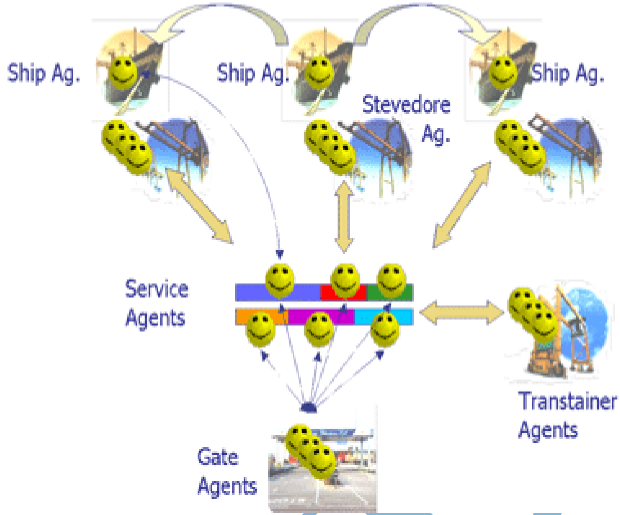

Figure 2 MAS

Agent communication is a field of study unto itself; situated at the crossroads of linguistics, cognitive science, artificial intelligence, formal logic, and computer science. This field of communication is dominated by both language semantics and dialogue protocol. Language semantics refer to the meaning that is expressed in a language or code [www.wikipedia.org].

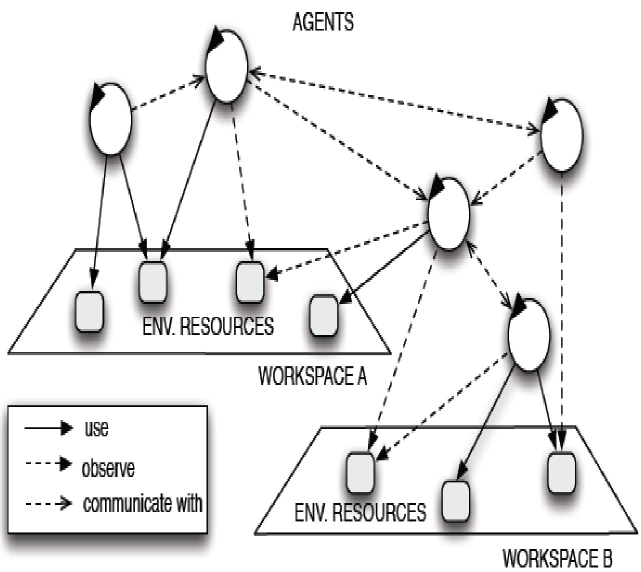

Figure 3 MAS Communication

There are a multitude of pre-cursors to formalized agent communication languages2. These languages arose on a predominately ad hoc basis and afforded a low level of inter-operability across systems. One largely used language pioneered by the United States Defense Advanced Research Programs Agency (DARPA) was Knowledge Query and Manipulation Language (KQML) – a language premised on restricted message sets and types (Finin et al, 1994).

A dialogue protocol, additionally, specifies a set of rules that regulate the dialogue between two or more communicating agents (Endriss et al, 2003). The remainder of this section presents a brief (and by no means comprehensive) review of the multi-agent work being carried out in both areas of communication.

Throughout the 1990s, as interest in agent technology grew with the rise of the internet, efforts were made to formalize these early and fragmented languages into one Agent Communication Language (Singh, 1998).

III. RELATED WORK

Baldoni et. al (2010) tackled the relation between declarative languages and multi-agent systems by following the dictates of the five Ws (and one H) that characterized investigations. The aim was to present this research field, which had a long-term tradition, and discussed about its future. The first question to answer is *“What? What are declarative agents and multi-agent systems?”*. Therefore, they introduced the history of declarative agent systems up to the state of the art by answering the question *“When? When did research on them begin?”*. We will, then, move to the question *“Where? Where can it take place?”*: in which kind of real applications and for which kind of problems declarative agents and MAS have already proven useful? Connected to where is *“Why? Why should it happen?”*. We will discuss the benefits of adopting the abstractions offered by declarative approaches for developing communication, interaction, cooperation mechanisms.We will compare with other technologies, mainly service-based and object-oriented ones. *“Who? Who can be involved?”*: in order to exploit this kind of technology what sort of background does a specialist have to acquire? We address this question by looking at the Italian landscape of Computer Science
research and education. Finally, with the question *“How? How can it happen?”* we will shortly report some examples of existing declarative languages and frameworks for the specification, verification, implementation and prototyping of agents and MAS.

Ismail et. al (2012) conducted interview survey on respondents from service, manufacturing and education industries in Malaysia, to understand the processes of personal knowledge management (PKM) among knowledge workers. The findings showed that personal knowledge network was enhanced when recommendations from associates outside the organisation were relied upon to identify the required knowledge experts. Thus the reputation of knowledge experts was known by some people in the network since it was the basis for assessing and deciding the reliability of the expertise required. This paper proposed a framework for a multi-agent system to search an existing network, analyse and manage reputation points in the process of identifying knowledge experts to fulfil the need of connecting to knowledge experts in managing personal knowledge. Recommendation on future work includes the technical possibility of expanding this multi-agent system to be implemented in the Semantic Web.

Holmgren et. al (2013) presented the Transportation And Production Agent-based Simulator (TAPAS), which was an agent-based model for simulation of transport chains that can be used, e.g., for analysis of transport-related policy and infrastructure measures. TAPAS was more powerful than traditional approaches to freight transport analysis, as it explicitly models production and customer demand, and it captures the interaction between individual transport chain actors, their heterogeneity and decision making processes, as well as time aspects. Whereas traditional approaches rely on assumed statistical correlation, TAPAS relied on causality, i.e., the focus is on the decisions and negotiations that lead to activities. TAPAS was composed of two connected layers, one that simulates the physical activities, e.g., production and transportation, andone that simulates the decision making and interaction between actors. They illustrated TAPAS with a scenario in which the consequences of three transport policy and infrastructure measures were studied..

IV. COORDINATION ISSUE IN MAS

The problem of coordination in multiagent systems (MASs) is of crucial importance in AI and game theory. Given a collection of agents charged with the achievement of various objectives, often the optimal course of action for one agent depends on that selected by another. If the agents fail to *coordinate* the outcome could be disastrous. Consider, for instance, two agents that each want to cross a bridge that can support the weight of only one of them. If they both start to cross, the bridge will collapse; coordination requires that they *each* agree" which one of them should go first. Coordination problems often arise in *fully cooperative MASs*, in which each agent shares the same utility function or *common interests*. This type of system is appropriate for modeling a team of agents acting on behalf of a single individual (each tries to maximize that individual's utility). In the bridge example above, it may be that neither agent cares whether it crosses first, so long as they both cross and pursue their objectives. In such a setting, coordination problems generally arise in situations where there is some flexibility regarding the "roles" into which agents fall. If the abilities of the agents are such that it makes little difference if agent pursues objective and pursues , or vice versa, the agents run the risk of both pursuing the same objective—with consequences ranging from simple delay in goal achievement to more drastic outcomes—unless they coordinate. This issue arises in many team activities ranging from logistics planning to robotic soccer.

An obvious way to ensure coordination is to have the agents' decision policies constructed by a central controller (thus defining each agent's role) and imparted to the agents. This is often infeasible. Approaches to dealing with "independent" decision makers include: (a) the design of conventions or social laws that restrict agents to selecting coordinated actions [9, 15]; (b) allowing communication among agents before action selection [16]; and (c) the use of learning methods, whereby agents learn to coordinate through repeated interaction [5, 6, 8, 11].

Unfortunately, none of these approaches explicitly considers the impact of coordination problems in the context of larger sequential decision problems. If the agents run the risk of miscoordination at a certain state in a decision problem, how should this impact their policy decisions at *other states*? Specifically, what is the long-term(or sequential) *value* of being in a state at which coordination is a potential problem? Such a valuation is needed in order for agents to make rational decisions aboutwhether to even put themselves in the position to face a coordination problem.

Unfortunately, there are no clear-cut definitions of sequential optimality for multiagent sequential decision processes in the general case. Most theoretical work on coordination problems assumes that a simple repeated game is being played and studies methods for attaining equilibrium in the stage game. In this paper, we argue that optimal sequential decision making requires that agents be able to reason about the specific coordination mechanisms they adopt to resolve coordination problems. With this ability, they can make optimal decisions by considering the tradeoffs involving probability of (eventual) coordination, the consequences of miscoordination, the benefits of coordination, the alternative courses of action available, and so on. We develop a dynamic programming algorithm for computing optimal policies that accounts not only for the underlying system state, but also the *state of the coordination mechanism being adopted*. Specifically, we show how the underlying state space can be expanded minimally and dynamically to account for specific coordination protocol being used.

V. CCORDINATION ALGORITHM IN MAS

The role of the coordination mechanisms is to provide information to the local scheduler that allows the local scheduler to construct better schedules. This information can be in the form of modifications to portions of the subjective task structure of the episode or in the form of local and non-local commitments to tasks in the task structure. The five mechanisms we will describe in this paper form a basic set that provides similar functionality to the original Partial Global Planning algorithm as shown in [5]. Mechanism 1 exchanges useful private views of task structures;Mechanism 2 communicates results;Mechanism 3 handles redundant methods; Mechanisms 4 and 5 handle hard and soft coordination relationships. More mechanisms can be added, such as one to update utilities across agents as discussed in the next section, or to balance the load better between agents. The mechanisms are independent in the sense that they can be used in any combination. If inconsistent constraints are introduced, the local scheduler will return at least one violated constraint in all its schedules. Since the local scheduler typically satisfices instead of optimizes, it may do this even if constraints are not inconsistent (i.e. it does not search exhaustively). The next section describes how a schedule is chosen by the coordination module substrate.

Soft coordination relationships are handled analogously to hard coordination relationships except that they start out with high negotiability. In the current implementation the predecessor of a relationship is the only one that triggers commitments across agents, although relationships are present. The positive relationship indicates that executing before decreases the duration of by a 'power' factor related to and increases the maximum quality possible by a 'power' factor related to for the details). A more situation-specific version of this coordination mechanism might ignore relationships with very low 'power'. The relationship is negative and indicates an increase in the duration of and a decrease in maximum possible quality. A coordination mechanism could be designed for (and similar negative relationships) and added to the family. To be pro-active like the existing mechanisms, a mechanism would work from the successors of the relationship, try to schedule them late, and commit to an earliest start time on the successor. Figure 4 shows Agent B making a commitment to do method which in turn allows Agent A to take advantage of the relationship, causing method to take only half the time and produce 1.5 times the quality.

VI. CONCLUSION & FUTURE SCOPE

In this paper we studied some aspects of the problems of coordination and cooperation of multiple agents that have individual goals and operate in the same environment. We formalized the coordination problem in a general way, and modelled a special case of multi-agent cooperation called assistance. For both problems we presented algorithms that rely on a classical planner and generate optimal solutions. The multi-agent assistance algorithm we propose is an innovative approach to dealing with the problem of mutual assistance among agents with complementary capabilities, whereas our coordination procedure presents certain advantages over previous approaches.

There are several lines for future work. The proposed algorithms must be evaluated experimentally, and possible computational inefficiencies must be addressed. There seems to be room for improvement by combining the classical planning based algorithms we presented with techniques from multi-agent plan merging as those presented eg. In [16] and [5]. Another direction of future study concerns the extension of the proposed framework to other problems from multi-agent planning, such as action synchronization and interleaved planning and execution. As a concluding remark, we reiterate that this work can be seen as a first step towards establishing a framework where different multi-agent planning problems can be studied in the light of recent advances in classical planning.

REFERENCES

[1] R. Alami and S. da Costa Bothelho. Plan-based multi-robot cooperation. In Plan-Based Control of Robotic Agents, volume 2466, pages 1–20. Springer-Verlag, 2002.

[2] A. Blum andM. Furst. Fast planning through planning graph analysis. Artificial Intelligence Journal, 90(1-2), 1997.

[3] G. Boutilier and R. Brafman. Partial-order planning with concurrent interacting actions. Journal of Artificial Intelligence Research, 14, 2001.

[4] J. Cox and E. Durfee. Discovering and exploiting synergy between hierarchical planning agents. In AAMAS03, 2003.

[5] J. Cox and E. Durfee. An efficient algorithm for multiagent plan coordination. In AAMAS05, 2005.

[6] E. Ephrati and J. Rosenschein. Divide and conquer in multiagent planning. In AAAI94, 1994.

[7] A. E. Fallah-Seghrouchni and S. Haddad. A recursive model for distributed planning. In ICMAS96, 1996.

[8] A. Gerevini. 5th international planning competition: Results of the deterministic truck. Available from http://zeus.ing.unibs.it/ipc-5/.

[9] S. Kambhampati, M. Tenenbaum, and S. Lee. Combining specialized reasoners and general purpose planners: A case study. In AAAI99, 1999.

[10] M. J. Katz and J. S. Rosenschein. The generation and execution of plans for multiple agents. Computers and Artificial Intelligence, 12(1):5–35, 1993.

[11] H. Kautz, D. McAllester, and B. Selman. Encoding plans in propositional logic. In Principles of Knowledge Representation and Reasoning, KR, 1996.

[12] H. Kautz and B. Selman. Unifying sat-based and graphbased planning. In IJCAI'99, 1999.

[13] H. Kautz, B. Selman, and J. Hoffmann. SATPLAN: Planning as satisfiability. In Booklet of the 2006 International Planning Competition, 2006. Available from http://zeus.ing.unibs.it/ipc-5/publication.html.

[14] Surjeet Dalal, Vivek Jaglan and Dr. S. Srinivasan, Enhancing security of agent program code using jar files in International Journal on Computer Science and Engineering ISSN: 0975-3397, Vol. 3 No. 4, April 2011 pp. 1627-1632

[15] G. R. Beddoe and S. Petrovic, "Selecting and weighting features using a genetic algorithm in a case-based reasoning approach to personnel rostering," Eur. J. Oper. Res., vol. 175, no. 2, pp. 649–671, 2006.

[16] K. Bradley and B. Smyth, "Personalized information ordering: A case study in online recruitment," Knowl.-Based Syst., vol. 16, nos. 5–6, pp. 269–275, 2003.

[15] C. M. Vong, P. K. Wong, and W. F. Ip, "Case-based classification system with clustering for automotive engine spark ignition diagnosis," in Proc. 9th Int. Conf. Comput. Inf. Sci., Aug. 2010, pp. 17–22.

[16] F. Azuaje, W. Dubitzky, N. Black, and K. Adamson, "Discovering relevance knowledge in data: A growing cell structures approach," IEEE Trans. Syst., Man, Cybern. B, Cybern., vol. 30, no. 3, pp. 448–460, Jun. 2000.

[17] Surjeet Dalal, Dr. Vivek Jaglan, Dr. Kamal Kumar Sharma, Integrating Multi-case-base-reasoning with Distributed case-based reasoning, International Journal of Advanced Research in IT and Engineering ISSN: 2278-6244, Vol. 3, No. 1, January 2014, pp. 29-38

[18] Z. Y. Zhuang, L. Churilov, F. Burstein, and K. Sikaris, "Combining data mining and case-based reasoning for intelligent decision support for pathology ordering by general practitioners," Eur. J. Oper. Res., vol. 195, no. 3, pp. 662–675, 2009.

