NATIONAL CONFERENCE ON ICT EMPOWERED TEACHING, LEARNING AND EVALUATION (NCICT-2016)

International Journal of Advanced Scientific Technologies in Engineering and Management Sciences (IJASTEMS-ISSN: 2454-356X) Volume.2,Special Issue.1Dec.2016

LOGICAL FOUNDATIONS OF MULTI AGENT SYSTEMS

K. Balaji Suneel Chandra
Asst Prof. of Computer Science & Engineering, S.V. Institute of Technology, Anantapur-515001
Email: kbsc@rediffmail.com

Abstract*—The aim of the paper was both to facilitate dissemination of recent research within the multi-agent systems community and also to promote discussion within this often dive diverse area. Again, the two-day workshop was based around a mixture of invited presentations from Keith Decker of the University of Delaware, USA, and Moshe Tennenholtz of the Technion, Israel, paper presentations and panel discussions. Generously supported by the UK's Engineering and Physical Sciences Research Council (EPSRC), the European Agent Link Network of Excellence for agent-based computing and Hewlett Packard Laboratories, the aim was to provide an opportunity for promoting and supporting activity in the research and development of multi-agent systems across academia and industry. There were three panel sessions, which addressed the pragmatic issue of making money from agents, the nature of argumentation and negotiation, and the possibility and merit of transferring models of agents between disciplines. All three provided engaging discussions, and summaries of them follow separately. In this report, we summaries the other contributions to the workshop through paper presentations and invited talks, which cover a wide range of relevant topics On the basis of their theory and the current window on the environment, agents draw conclusions by means of some consequence operation that characterizes their reasoning pattern. As the field of agent-based systems continues to expand, and the diversity of research grows (Howe and Parsons, 1998), the value of well-focused and directed, yet informal, workshops like UKMAS'98 becomes more pronounced. Indeed the way in which it has engaged communities from both academia and industry is demonstrated by the location and organization of the next workshop in the series, which will be held in Bristol in December 1999, chaired by Chris Priest of Hewlett Packard Labs, who have generously supported the previous workshops.

***Index Terms*—Logic, Agent-based systems**

I. INTRODUCTION

The 1998 Workshop of the UK Special Interest Group on Multi-Agent Systems was held in Manchester in December, chaired and organized by Michael Fisher of Manchester Metropolitan University, continuing the series of focused and constructive meetings in this field. After two very successful workshops on the Foundations of Multi-Agent Systems at the University of Warwick in 1996 (Luck, 1997; Doran *et al*., 1997; d'Inverno *et al*., 1997; Fisher *et al*., 1997) and 1997 (Luck *et al*., 1998; Aylett *et al*., 1998; Binmore *et al*, 1998), the scope was broadened for 1998 to a wider range of issues concerning all aspects of multi-agent systems. About 50 people attended, representing both industry and academia, and from a variety of relevant disciplines. The aim of the workshop was both to facilitate dissemination of recent research within the multi-agent systems community and also to promote discussion within this often diverse area. Again, the two-day workshop was based around a mixture of invited presentations from Keith Decker of the University of Delaware, USA, and Moshe Tennenholtz of the Technion, Israel, paper presentations and panel discussions. Generously supported by the UK's Engineering and Physical Sciences Research Council (EPSRC), the European Agent Link Network of Excellence for agent-based computing and Hewlett Packard Laboratories, the aim was to provide an opportunity for promoting and supporting activity in the research and development of multi-agent systems across academia and industry.

There were three panel sessions, which addressed the pragmatic issue of making money from agents, the nature of argumentation and negotiation, and the possibility and merit of transferring models of agents between disciplines. All three provided engaging discussions, and summaries of them follow separately. In this report, we summaries the other contributions to the workshop through paper presentations and invited talks, which cover a wide range of relevant topics. The structure of the report reflects the organization of the workshop.

The first day of the workshop began with an invited talk from Moshe Tennenholtz of the Technion, Israel, who discussed the relation between economics and artificial

International Journal of Advanced Scientific Technologies in Engineering and Management Sciences (IJASTEMS-ISSN: 2454-356X) Volume.2,Special Issue.1Dec.2016

intelligence, which have overlap-ping interests in some important fundamental issues. While economic models typically deal with the behavior and interaction of rational agents, artificial intelligence deals with the construction of such agents. In spite of these fundamental connections, there still seems to be a considerable distance between works in artificial intelligence and work in economics. There are at least two major challenges one has to address in order to bridge between the related theories:

1. We need to re-consider the theory of (economic) mechanism design in view of its use in computational settings; and

2. We wish to incorporate distributed systems features into game-theoretic models, and study these new models.

In his talk, Tennenholtz presented two papers that deal with these two issues respectively entitled "Internet Auctions "and "Distributed Games" (both co-authored with Dov Monderer, Economics, Technion). In the former paper, *Internet Auctions*, several new features of such auctions are discussed, focusing on two features in particular: the high-level of risk for the participants and the competitive environment for the sellers. For auction organizers they recommend (with some reservations) conducting third-price auctions. In the latter paper, they present a new model — *distributed games*. In such a model, each player controls a number of agents (for example, software agents) which participate in asynchronous parallel multi-agent interactions (for example, auctions). The agents jointly and strategically control the level of information monitoring by broadcasting messages. As an application of this work, they show that the cooperative outcome of the Prisoner's Dilemma game can be obtained in equilibrium in such a setting, and generalize this result to other multi-agent interactions.

Much additional work has been carried out, both on the study of economic mechanisms in view of their use in computational settings, as well as on bridging the gap between distributed computing and mechanism design in economics. In particular, Tennenholtz and Monderer are able to show that a second-price auction leads to close to optimal revenue when there are many agents that participate in the auction (as is the case in Internet auctions), when these agents behave according to classical economics assumptions. In addition, they have shown that economic mechanisms can be transformed into working protocols in a variety of communication networks. Together, these results initiate new unified theories that may serve as foundations for a theory of electronic commerce.

In the first of the paper presentations, Beer described his work with Bench-Capon and Six smiths at the University of Liverpool on the issues involved in managing dialogues between information agents. In particular, they focus on the problems associated with conversation classes derived from an agent- based distance learning application. This 'Virtual College' system involves the use of mediators to provide intelligent management of information flow between multiple agents, which can be people, databases and expert systems. As a result of this organization, there can be comparatively complex conversation classes, sometimes involving very large numbers of particular agents, with communication between them specified by means of performatives. Conversation classes define performatives to meet overall requirements so that in different conversations the same performative may have different conditions associated with it. By the adoption of appropriate *conversation classes*, however, it is possible to arrange for a wide range of services to be provided robustly and securely.

Next, Ghidini of Manchester Metropolitan University and the University of Trento described her work with Serafini on information integration for electronic commerce. In agent-mediated electronic commerce, agents need to exchange information with other agents and to integrate the information obtained from other agents in their own information. Integration is a very complex task as: information is *distributed* among different agents; each agent autonomously represents and manages a piece of information; information might be partial, as an agent cannot wait to have complete information before acting; and information might be redundant, as the same information might be represented by two different agents. The goal of the work is to provide a formal semantics for information integration

Able to cope with distributed autonomous, partial, and redundant information. Two examples from an electronic commerce scenario which emphasize critical problems in the integration of information were introduced to illustrate the issues, and semantics for information integration was defined with its adequacy tested by formalizing the examples.

In the second of the paper sessions two papers concerned with *rights* were presented. Alonso of the University of York began by describing preliminary work on rights and coordination in multi-agent systems. He introduced some intuitive ideas about basic rights or liberties: how they are understood, their functions and their relations with unconstrained actions in a general model of coordination. Norms and conventions have previously been used to constrain agent behavior but lead to problems with a lack of autonomy and stability respectively. Instead, Alonso proposes the notion of rights that guide but do not control the behavior of autonomous agents, as restrictions of actions that allow them enough freedom but still constrain them. At the macro-level, as a consequence, he argues, systems perform much more efficiently.

In contrast to Alonso's view of rights as liberties, Norman's work at Queen Mary and Westfield College with Sierra and Jennings offers a view of rights by which they are a means of defining flexible agreements between agents in order for them to act in collaboration. He presented a language in which agents are constrained to act to uphold the rights of others and act in accordance with an agreement to which they are bound. Various properties (morality, delegation and persistence) can then be introduced and it can be shown how they may be used as as axioms of a theory of agency. Norman argues that the intuitions captured by his model provide a flexible way of describing agreements between agents, while retaining a notion of joint commitment, which is widely recognized as necessary to ensure that agents act on their agreements.

The first day of the workshop ended with an effort to understand the relationship between different disciplines contributing to the field of agent-based systems. Edmonds of Manchester Metropolitan University introduced the notion of *social embeddedness* as a way to distinguish between the engineering perspective on agents as constructing systems that meet certain performance criteria in a reliable way, and the social simulation perspective in acting as models of social agents to increase our understanding of them. An agent is socially embedded in a collection of other agents if it is more appropriate to model the agent as a part of the total system of agents and their interactions, as opposed to modeling it as a single agent interacting with a unitary environment. Edmonds argues that social embeddedness will need to be a feature of many social simulation models since it has practical consequences for agents within them, but that it may not be practically possible with the engineering perspective. The claim is that it may not be possible to engineer truly social agents because a critical aspect of sociality comes from this social embeddedness.

At the start of the second day, Keith Decker gave an invited presentation on coordinating intelligent agents. This talk focused on how to get organizations—multiple software agents and humans—

To coordinate their activities when they are working on shared, loosely coupled problems, such as engineering design or information gathering. Decker described some useful representations (including TAEMS [Task Analysis and Environment Modeling System]) for annotating an agent's representation of its activities, and some approaches (including GPGP [Generalized Partial Global Planning]) to designing coordination mechanisms that are adapted to some particular problem-solving environment. Examples were drawn from various projects in distributed information gathering, distributed hospital patient scheduling, and a Boeing Rotorcraft collaborative design project.

Decker's research program is involved in developing intelligent software agents and *organizations* of these agents (including sometimes humans) that can operate in environments where there is a lot of uncertainty about what is happening and where there may be time pressures or deadlines. The agents will in general have many goals, some partially overlapping or conflicting. They are not (and cannot)

Realistically look for optimal solutions, but instead must satisfice — try to find a solution that is 'good enough' in the time and resources that are available. No agent can work completely alone.

This research program can be divided into three areas. First, how to formally represent and reason about these sorts of problems, both externally as a human software engineer and internally as a software agent. To this end the TAEMS task structure description language was developed (representing what are thought to be the important concepts) and the GPGP approach to coordination (a way to reason about TAEMS descriptions within each software agent so that a team of them acts coherently together). Secondly, software and tools are constructed for building actual software agents. This includes the RETSINA project that started with Katia Sycara at CMU, and the current DECAF project which is a Java version at the University of Delaware that combines features of RETSINA and Decker's work on coordination at UMass. Finally, we need to understand, model and even imitate human organizational structures in the context of software agents (both organizations of *all* software agents, and mixed human/software agent hybrid organizations). This is very important both because complex problems often need more than trivial organizational solutions, and because most real systems are embedding in existing human organizations (so they must respect the boundaries of those organizations and the roles of the people with whom they interact).

The TAEMS (Task Analysis and Environment Modeling System) language is used to formally define what a task structure is, what parts are known by what different agents, and what happens when agents execute these parts. TAEMS is often used as an annotation language on top of HTN (Hierarchical Task Network) plans, based on careful, functional descriptions and an underlying state- based model of computation. The basic idea is that each agent is trying to maximize performance, as described by some set of utility characteristics (summarized as *quality* for good characteristics, and *cost* for bad characteristics). Since the time that something gets done often affects these things a lot, we also track the *duration* of various activities. TAEMS task structure annotations describe how the actions of any agent affect the performance of that agent or others (by changing quality, cost, or duration). The basic relationship here is the *subtask*; but more important are various hard and soft relationships between tasks (i.e. *enables* where A must come before B, or *facilitates*, where doing A will cause B to be done better, cheaper, or quicker). All relationships have a formal, quantitative mathematical definition. TAEMS agents can reason about these task structures, and even use them as a language for communicating about coordination problems.

In *designing coordination mechanisms* using these representations, GPGP is a domain independent scheduling approach that makes several architectural assumptions. Most important of these is that the agent represents its current set of intended tasks using the TAEMS task structure representation language. An agent using GPGP provides a planner or plan retriever to create task structures that attempt to achieve agent goals, and a scheduler that attempts to maximize utility via the choice and temporal location of basic actions in the task structure. Each GPGP mechanism examines the change in task structure for certain situations, such as the appearance of a particular class of task relationship, and responds by making local and non-local *commitments* to tasks, possibly creating new communication actions to transmit commitments or partial task structure information to other agents. The set of coordination mechanisms is extendible, and any subset or all of which can be used in response to a particular task environment situation. Initially, GPGP defined five coordination mechanisms based on Durfee's PGP. By defining them in TAEMS terms, they can (and have been) applied to domains quite different from vehicle monitoring, such as hospital scheduling, and software process management.

Finally, Decker discussed using these ideas to build real software agents. In DECAF, which is their current agent toolkit, the agent's communicating/ planning/ scheduling/ execution are concurrent. The general data flow is that new KQML messages (i.e. ASK) create new *objectives*. The planner creates TAEMS task structures to achieve the objectives. There are usually many simultaneous plans and possible actions vying for agent resources—the scheduler creates an appropriate agenda of tasks. Finally, the execution monitor actually carries out the agenda. In DECAF, these are done concurrently and constantly. The agent is thus *constantly* (but efficiently!) re-planning and re-scheduling as the world changes dynamically about it, and in response to uncertain action outcomes that force it to interleave planning and execution1.

The final paper session was started by Schroeder of the University of Hannover who began by describing his work with Mora and Alferes, which is concerned with the further development of previous work on argumentation semantics for single agents. Argumentation semantics in extended logic programming for a single agent determines its beliefs by an internal argumentation process. Schroeder's work extends the initial argumentation framework to a multi-agent setting including both argumentation and cooperation. In this work, inference for multi-agent systems and an algorithm for inference are both defined, and an argumentation protocol sketched and demonstrated with an example implemented using vivid agents.

Finally, van Eijk of Utrecht University described work with de Boer, van der Hoek and Meyer on a programming framework for systems of interacting agents. This work extends previous work in the development of a programming language for interacting agents that is based on the semantically well-founded concurrent programming paradigms of CSP and CCP, in which agents can revise their beliefs, by formalizing some basic patterns of interaction between communicating agents. Agents interact with each other in a shared environment that is modeled as a mathematical structure. Each agent is assigned a part of the environment it can inspect and manipulate, known as its expertise. Observations are performed in the context of a *theory*, especially concerning information on those sections of the environment that an agent cannot directly interact with, which is constructed during the execution of the system. The theory is maintained by employing the ability to communicate with others agents via the exchange of information. On the basis of their theory and the current window on the environment, agents draw conclusions by means of some consequence operation that characterizes their reasoning pattern.

As the field of agent-based systems continues to expand, and the diversity of research grows (Howe and Parsons, 1998), the value of well-focused and directed, yet informal, workshops like UKMAS'98 becomes more pronounced. Indeed the way in which it has engaged communities from both academia and industry is demonstrated by the location and organization of the next workshop in the series, which will be held in Bristol in December 1999, chaired by Chris Preist of Hewlett Packard Labs, who have generously supported the previous workshops.

REFERENCES

[1] Aylett, R, Brazier, F, Jennings, N, Luck, M, Preist, C and Nwana, H, 1998. “Agent Systems and Applications” The Knowledge Engineering Review, 13(3): 303–308.
[2] Binmore, K, Castelfranchi, C, Doran, J and Wooldridge, M, 1998. “Rationality in Multi-Agent Systems” The Knowledge Engineering Review, 13(3): 309–314.
[3] d'Inverno, M, Fisher, M, Lomuscio, A, Luck, M, de Rijke, M, Ryan, M and Wooldridge, M, 1997. “Formalisms for Multi-Agent Systems” The Knowledge Engineering Review, 12(3): 315–321.
[4] 1More information about this work can be found at http://www.cis.udel.edu/ decker/.
[5] Doran, JE, Franklin, S, Jennings NR, and Norman, TJ, 1997. “On Cooperation in Multi-Agent Systems” The Knowledge Engineering Review, 12(3): 309–314.
[6] Fisher, M, M¨uller, J, Schroeder, M, Staniford, G and Wagner, G, 1997. “Methodological Foundations for Agent-Based Systems” The Knowledge Engineering Review, 12(3): 323–329.
[7] Howe, A and Parsons, S, 1998. “Special issue: perspectives on recent intelligent agent's research as viewed through two conferences” Knowledge Engineering Review, 13(2): 129–130.
[8] Luck, M, 1997. “Foundations of Multi-Agent Systems: Issues and Directions” The Knowledge Engineering Review, 12(3): 307–308.
[9] Luck, M, d'Inverno, M, Fisher, M and FoMAS'97 Contributors, 1998. “Foundations of Multi-Agent Systems: Techniques, Tools and Theory” The Knowledge Engineering Review, 13(3): 297–302