Investigating Utility Assignment Based Approaches to Multiagent Coordination

Steven Lynden, Omer F. Rana
Department of Computer Science
University of Wales Cardiff, POBox 916,
Cardiff, CF24 3XF. UK.
{S.J.Lynden, O.F.Rana} @ cs.cf.ac.uk

Abstract— **A potential problem exists when a system contains multiple autonomous self-interested entities, as is the case in a multiagent environment: agents can work at cross-purposes and exhibit behaviour that can be detrimental to the overall functionality of the system. Our approach to solving this problem is based on constraining the actions of agents so that individual agent behaviour is always beneficial to the global system; we also investigate a mechanism that allows agents to modulate their interactions and collaborate by sharing information/resources. This technique is based on assigning weighted relationships to other agents representing evaluations of previous interactions with the agents.**

I. Introduction

We investigate approaches to multiagent coordination, based on the techniques developed in the Collective Intelligence [2] [3] (COIN) work by Wolpert. We are concerned with multiagent systems (MAS) consisting of agents that have the ability to learn to adapt to their environment and maximise their local utility values, using various reinforcement learning (RL) [4] techniques or other learning mechanisms. We consider a MAS to be a system consisting of a society of agents, deployed by one or more owners to achieve a number of desired objectives. We make the assumption that we are allowed to assign local utility values to agents belonging to the MAS, that will affect their subsequent behaviour due to the fact they aim to maximise their local utility. We also assume the existence of a global utility value, G, that quantifies the overall performance of a MAS - if a MAS consists of agents representing multiple owners, G must quantify the degree to which the combined needs of the owners of the MAS are satisfied. We allow agents to interact with agents outside their MAS, using negotiation mechanisms such as auctions or other techniques, the outcomes of which may be included in the function used to calculate G. The investigation of negotiation techniques is not, however, the focus of our work.

II. Coordination

Our approach to coordination is to assign local utility values that reflect as accurately as possible an agent's contribution to G, with the intention that agents will learn to perform actions that will maximise their local utility values, in turn maximising G. The challenging issue here is to assign rewards that reflect the contribution of the agent towards the global goal or desired behaviour, also known as the credit-assignment problem [7]. We utilise techniques developed in the COIN work, in which the "reverse" credit-assignment problem is addressed and the performance of utility assignment techniques are analysed. In [3], Wolpert describes a descriptive framework for building COINs using various local utility assignment methods, with success in particularly achieved using Wonderful Life Utility (WLU), a local utility value calculated by clamping the states of all other agents over the history of the system to an arbitrary value and obtaining a local utility value for an agent that represents the increase in global utility due to the existence of that agent alone. WLU provides a very accurate reflection of an agent's contribution to its environment's global utility value, and is of particular interest in terms of its performance in systems with large numbers of agents. Our work differs from that of the COIN work in that we concentrate on a practical implementation within a FIPA compliant [6] multiagent environment. We develop a framework for multiagent coordination, maintained to be as domain independent as possible, so that it can be applied to a variety of areas in which agent oriented development is applicable, such as e-commerce, resource allocation, network management etc.

The global utility of a MAS and how it is calculated can depend on various characteristics of the aggregate behaviour of the MAS, and whether they are considered desirable or undesirable. The information needed to calculate G can therefore range from simple output values obtained from various agents, to the entire history of the internal states of the agents in the MAS. If we consider the internal state of an agent to be something that it can output, then the function used to calculate G is parameterised by a set of output values given by the agents in the system, hence, if all output from all agents can be encoded in a string format, we can calculate the global utility of a system using output provided by the agents transported within the messages of an agent communication language (ACL), such as FIPA ACL [6]. This allows the concept of the MAS environment to be implemented as an entity that can receive the output necessary to calculate G from the agents within the system and then distribute local utility values to the agents based on policies such as WLU. We refer to such an entity in our systems as an environment simulation node (ESN). An ESN is able to evaluate the performance of a MAS, and hence calculate G, by requesting information from the agents in the system using an ACL and evaluating the message sent in reply. Local utility values can then be distributed to the agents based on various policies, which reflect the contribution of each agent to the system's overall performance. Communication between the ESN and the system agents is achieved using FIPA ACL with a specialised content language called "environment", which enables agents to recognise messages that originate from an ESN. We assume the understanding of a common ontology between the ESN and the system agents, and that system agents will aim to maximise the local utility values distributed by the ESN.

III. Agent Interaction

Agents interact in order to share resources, exchange information and provide or obtain services from other agents. An agent can initially gain an indication of which other agents to interact with in many agent systems, including FIPA compliant systems, by querying a directory facilitator (DF) agent. A DF agent contains a description of the services offered by agents that have registered with it. The service description maintained by the DF agent may be suitable for gaining an indication of the services an agent "claims" or "intends" to offer, but may not be informative enough for an agent to accurately predict the outcome of an interaction. Evaluating interactions in terms of their value to an agent allows agents to build models of other agents and predict the expected outcome of interactions with certain agents. These predictions can then be used to make decisions on which agents to interact with and how to interact with them.

In order to investigate multiagent coordination in a variety of domains, a generic technique for optimising the interactions of agents is required. As a MAS expands, the potential number of interactive relationships in the system increases to such a degree that it becomes unrealistic for agents to maintain detailed models of all agents in large MASs. Our approach to building models of other agents is achieved using weights that represent role specific relationships with other agents. Each agent maintains a set of relationships with other agents, where each relationship contains a number of weights representing an evaluation of role specific aspects of previous interactions with the agent. If an interaction with a certain agent is useful in some way, the weight held for that interaction should be updated accordingly, so that subsequent choices concerning which agents to interact with are affected by this weight update. Weights are implemented as floating-point values and are updated using various RL based techniques, where the weight for an agent is initialised to an arbitrary value and then reinforced based on evaluations of the interaction it represents. The global set of weights maintained by the agents in the system is in effect a gating network [1], which modulates the interactions of the agents.

This use of this mechanism also allows agents to query each other concerning their evaluation of interactions with other agents, in effect recommending role specific relationships with certain agents. This differs from the information provided by the DF service description in that it is based on the experience of interacting with an agent and not the agent's published service description, which may be much less informative. We hope that the use of the gating network structure will enable agents to restrict interactions with other agents to be as advantageous as possible with respect to maximising their local utility values, leading to an emergent global collaboration, with agents sharing useful data and techniques as well as services.

IV. Implementation

We have developed software, aimed at MAS developers, that allows the construction of MASs with characteristics that allow us to investigate various applications of the aforementioned techniques. The system is implemented using the FIPA-OS [5] agent development toolkit. The architecture of our implementation is built around a "Node", which is a generic agent that can be specialised to build application specific agents. The Node contains the necessary functionality in order for agents to utilise the utility assignment and gating network approaches described previously. A library of tasks [5] is provided, from which agent functionality can be built in a modular fashion. The task library is maintained to be as domain independent as possible, with Java objects (written by the agent developer) used to parameterise these tasks and define application specific functionality. A basic knowledge of Java programming is required to develop agents using our framework, as agent behaviour must be specified by algorithms programmed in Java.

V. Conclusions

We have outlined the salient features of our approaches to MAS coordination. As ongoing work, we aim to investigate the use of multiple ESNs within the same MAS, where each ESN concentrates on a specific aspect of the performance of the system. Agents may be able to sum the local utility values distributed to them by multiple ESNs, meaning new ESNs can be created at runtime.

We aim to test the approaches outlined above empirically, with particular interest paid to the scalability of the techniques.

References

[1] R. Sun & T. Peterson: *Multiagent Reinforcement Learning: Weighting and Partitioning*, Neural Networks (Elsevier Science), 12:727-753, 1999.

[2] D. Wolpert & K. Tumer: *An Introduction to Collective Intelligence*, In Handbook of Agent Technology. AAAI Press/MIT.

[3] D. Wolpert, K. Wheeler & K. Tumer: *General Principles of Learning-Based Multi-Agent Systems*, Proceedings of the Third Anuual Conference on Autonomous Agents, May 1999.

[4] R. S. Sutton & A. G. Barto: *Reinforcement Learning: An introduction*, MIT Press, Cambridge, MA, 1998. Press, 1999.

[5] S.J. Poslad, P. Buckle & R. Hadingham: *The FIPA-OS Agent Platform: Open Source for Open Standards*,Proceedings of PAAM 2000, Manchester, UK, (April 2000).

[6] FIPA (Foundation of Intelligent Physical Agent) homepage: http://www.fipa.org

[7] S. Sen & G. Weiss: *Learning in Multiagent Systems*, In Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence The MIT Press (1999).