Chapter XVIII
A Framework for Dynamic Agent Organizations

Shaheen Fatima
Loughborough University, UK

Michael Wooldridge
University of Liverpool, UK

ABSTRACT

This chapter presents an adaptive organizational policy for multi-agent systems called TRACE. TRACE allows a collection of multi-agent organizations to dynamically allocate tasks and resources between themselves in order to efficiently process and incoming stream of tasks. The tasks have deadlines and their arrival pattern changes over time. Hence, at any instant, some organizations could have surplus resources while others could become overloaded. In order to minimize the number of lost requests caused by an overload, the allocation of resources to organizations is changed dynamically by using ideas from microeconomics. We formally show that TRACE has the ability to adapt to load variations, reduce the number of lost requests, and allocate resources to computations on the basis of their criticality. Furthermore, although the solution generated by TRACE is not always Pareto-optimal, TRACE has the properties of feasibility and monotonicity that make it well suited to time-constrained applications. Finally, we present experimental results to demonstrate the performance of TRACE.

INTRODUCTION

In a multiagent system two or more agents join together to achieve goals that individuals on their own cannot, or to achieve them more efficiently (Rosenschein & Zlotkin 94, Faratin, Sierra, & Jennings, 98, Kraus, 01, Falcone & Castelfranchi, 01). It is well recognized that the ability of a multiagent system to dynamically reorganize its structure and operation at run-time is highly valuable for many application
domains. For example, consider a multiagent information retrieval system with a number of information repositories. Access to these repositories is provided by information agents. Users request these agents for information, the agents access the repositories and use the available computational resources to process requests. In most practical cases, these requests have two main characteristics: deadlines, and an arrival pattern that is unknown a priori and varies with time. Here the resources used to carry out requests are limited, so they must be used optimally. Thus, such dynamic environments require the multiagent system to operate under a fluctuating load, schedule requests before their deadline, and allocate resources optimally.

Thus maximum possible requests must be processed with available resources. Since the load is time varying, some parts of the system may receive more requests than others and have a higher demand for resources. So the key problem is to develop a method that allows the system to reorganize itself (i.e., allocate resources dynamically in accordance with their demand) in order to adapt to a changing environment. A lot of research effort has been devoted to developing such methods (Waldspurger Hogg, Huberman, Kephart, & Stornetta, 92, Wellman, Birmingham, & Durfee, 96, Clearwater, 96, Wlash & Wellman, 99, Sandholm & Lesser, 01). But there is lack of a single method that addresses all the three issues of deadlines, fluctuating loads, and efficiency. In order to overcome this shortcoming, we present a comprehensive framework that copes with deadlines and a fluctuating load and also addresses the issue of efficient allocation of resources. Note that the objective of this chapter is not to focus on operating systems related issues such as ‘serializeability’ or ‘deadlock management’. These have been dealt with at length in distributed and real time operating systems (Douglass, 02, Tanenbaum & Steen,04). The main objective of this chapter is to focus on developing a market based protocol for resource allocation that works at a higher level than operating systems.

The proposed method is called TRACE (Task and Resource Allocation in a Computational Economy). In TRACE, a request is a task. TRACE is made up of two components: a task allocation protocol (TAP) and a resource allocation protocol (RAP). As agents have different capabilities, it is necessary for them to cooperate effectively. The task allocation protocol (TAP) allocates tasks to agents through a process of negotiation. The changes in load are handled by the resource allocation protocol (RAP), which uses ideas from microeconomics. The main focus of this chapter is the RAP although we give a brief overview of the TAP.

This chapter makes two extensions to the RAP proposed in (Fatima, 01). First, our new version of the RAP has the properties of feasibility and monotonicity So the reallocation of resources (in Fatima:01) can take place only after the RAP terminates normally. In contrast, the feasibility property of the new RAP allows reallocation to take place even before its normal termination. This has the advantage of speeding up the allocation process. Furthermore, since the RAP also has the property of monotonicity, the solution it generates after every iteration improves with the number of iterations. Second, this chapter also provides a comprehensive analysis of the properties of the RAP.

THE SETTING

We formally define the terms used in TRACE and the problem it aims to solve.

**Organizations:** We have a multiagent system, M, structured as a set of k organizations \( M = \{O_1, \ldots, O_k\} \). Each organization is a set of agents: organizations have no internal structure. Different organizations contain mutually disjoint sets of agents. We denote the set of all agents by \( AG = \{a_1, \ldots\} \). The members