Coordination Mechanisms for Dependency Relationships among Multiple Agents *

Wei Chen
Computer and Information Sciences
University of Delaware
Newark, DE 19716
wchen@cis.udel.edu

Keith S. Decker
Computer and Information Sciences
University of Delaware
Newark, DE 19716
decker@cis.udel.edu

ABSTRACT

A recent observation about multiagent coordination is that one can describe possible mechanisms in a domain-independent way, as simple or complex responses to certain dependency relationships between the activities of different agents. Thus agent programmers can separate encoding agent domain actions from the solution to particular coordination problems. This paper explores the specification of a large range of coordination mechanisms for the common hard *enablement* relationship between tasks at different agents. This paper also presents the implementation of this idea in the DECAF agent architecture and an initial exploration of the separation of domain action from meta-level coordination actions for eight simple coordination mechanisms.

Categories and Subject Descriptors

I.2.11 [**ARTIFICIAL INTELLIGENCE**]: Distributed Artificial Intelligence—*Coherence and coordination*

General Terms

Coordinating multiple agents and multiple activities, Coordination infrastructures

1. INTRODUCTION

Our research towards the study of coordination concentrates on the coordination mechanism development, but has a different prospective from previous approaches. For example, compared with MIT CCS's research [1], which is to describe, categorize, and analyze processes for separation into sub-processes for further study, our vision is to explore dependencies at a finer granularity. Since all dependencies are composed of enablement or facilitation relationships among agent tasks, we study these relationships themselves and associate a common set of mechanisms with every agent. With these embedded coordination mechanisms, each agent is able to learn over time which coordination mechanism is the best for a specific environmental context.

Decker [2] and Castelfranchi [3] found that an important way to express coordination mechanisms is to study the relationships of agents' goals, the planned tasks to achieve those goals, and required resources. Our work is a general solution to this problem, which is based mainly on two previous achievements, TAEMS [4] and GPGP [2]. The key point of GPGP is to assume each agent is capable of reasoning locally about its schedule of activities and possible alternatives. Thus coordination mechanisms of many different styles can be thought of as ways to provide information to a local scheduler that allow it to construct better schedules.

This paper concerns an implementation of these ideas using DECAF (Distributed Environment Centered Agent Framework) [6] as an architecture and set of agent construction tools for building real multi-agent systems. DECAF is being used to build applications in electronic commerce, bioinformatics, and plant alarm management.

2. TASK STRUCTURE ALTERATION

To achieve its desires, an agent has to select appropriate actions at suitable times with the right sequence. These actions can be represented by an agent's task structures via HTN planning. A *task* represents a way of achieving some objective and each task is related to a set of subtasks via a Quality Accumulation Function (QAF) that indicates how the execution of the subtasks affects the completion of the super task(such as an AND/OR relationship). Actions are described by a vector of characteristics (e.g. quality, cost, and duration) that allow predictive scheduling based on some dynamic agent utility function. Because of the inevitability of non-local dependencies, and the associated uncertainty of the action characteristics, the ability of and agent scheduler is severely limited if it can not acquire information about when and how the non-local dependencies are to be handled. It is this information that is provided by coordination mechanisms.

If the execution of a task affects, or is affected by, another task, there exists relationship between these tasks. These relationships among agent tasks can be classified as dependent relationships. In any case, if a task structure extends over tasks/actions located at different agents, then these *non-local* relationships that cross agent boundaries become potential coordination points.

We describe coordination mechanism generally as a set

*This work is supported by NSF Grant No. 9733004.

AAMAS'02, July 15-19, 2002, Bologna, Italy.

of protocols, possibly unique to the mechanism, and as an associated automatic re-writing of the specification of the domain-dependent task (expressed as an augmented Hierarchical Task Network [5]). For example, if the coordination mechanisms are a reservation and a deadline commitment protocol, then the re-writing changes "Act2 enables Act1" into "reserve-act enables deadline-cmt enables Act2 enables Act1". DECAF provides a facility for examining patterns of relationships in the current task structures between local tasks and *non-local* tasks, and re-writing them as required by the mechanism.

3. COORDINATION MECHANISMS

We have catalogued at least seventeen coordination mechanisms for enable relationships in the abstract. We only briefly describe a selected set of mechanisms here. They are *avoidance* (with or without some sacrifice), *reservation schemes*, *simple predecessor-side commitments* (to do a task sometime, to do it by a deadline, to an earliest-start-time, to notify or send result directly when complete), *simple successor-side commitments* (to do a task with or without a specific EST), *polling approaches* (busy querying, timetabling, or constant headway), *shifting task dependencies* by learning or mobile code (promotion or demotion), various third-party mechanisms, or more complex multi-stage negotiation strategies.

The input to these mechanisms are agents' task structures with structural information and annotations on the tasks. The result of the mechanisms might be either a structural alteration (i.e. removing or adding tasks) or an alteration of the annotations on the task structure, or both. Considering the scheduling problem that local agent may have no knowledge about the characteristics of non-local tasks, our approach removes this uncertainty and allow the local agent to make better scheduling decisions.

4. EXPERIMENT AND CONCLUSION

We created an experiment framework to test the performance of these mechanisms under various environment factors. In this framework *Error Rate*, *Task Repeat Rate*, and *Agent Load* are some of the changing input factors that describe an environment. Multiagent system performance is characterized by *Quality Change*, *Communication Load*, *Task Execution Time*, *Idle Time*, *Deadlines Missed* and *GPGP Coordination Time*. This experiment framework can be both simulation testbed for coordination mechanisms, and be applied to real applications, like query planning and execution in an information gathering system. For information gathering tasks, a query planning agent takes user input and decomposes it into sub-tasks distributed to multiple wrapper agents. Each wrapper agent queries remote databases, extracts the expected information, and sends the sub-results back to an agent for result composition. One wrapper agent may need the results of other wrapper agents' queries for either further sub-queries or further information extractions. The scheduling for a wrapper agent to access multiple information resources, the explicit enablement relationships between the reasoner and the wrapper agent, and the interdependencies among wrapper agents, provide coordination mechanisms with a perfect stage to perform on.

Avoidable mechanisms cost very little in terms of meta-level coordination time, so programmers are encouraged to apply them first unless not applicable in certain situations. The more complex coordination mechanisms result in higher quality, but require much longer time. If deadlines are the main factor for agents, *Coordination by Reservation* outperforms others; If the repeat rate of requested coordination task is high, *Demotion Shift* mechanism saves the transmission time by sending object code for further reuse; *Promotion Shift* has this advantage as well and saves one round of coordination communication, but with the constraint of extra domain knowledge; If the enablee agent is heavily loaded, *Coordination by Sending Result* distributes the computing task to the enabler agent; If the underlying network protocol is connectionless, *Polling* ensures the reliability of coordination protocol with continuous queries for the earliest presence of a result, or takes recovering actions early in case of failure; By publishing a well known timetable or fixed execution period, the mechanisms of *Timetabling* and *Constant Headway* unburden the enabler and enablee agents from tightly coupled coordination.

5. CURRENT WORK AND FUTURE WORK

We built an experiment framework to study a large number of coordination mechanisms for dependency relationships among agents, and the application of these mechanisms have been attempted on domains, such as query planning problem and Internet information gathering problem.

The relationships we explored are hard relationships, *enables*. The soft relationships are potential extension to our current research. We believe all kinds of non-local relationships can be explored within the general DECAF HTN task structures. Another future task is to apply the combinations of mechanisms, because under certain environments it is highly possible that combinations of mechanisms may result in better performance. Error tolerance is also an important issue associated with the mechanisms. Finally, potential learning approaches will enable agents to find the best actions in certain common situations.

6. REFERENCES

[1] Kevin Crowston and Charles Osborn. Modeling coordination from field experiments. Conference on Organizational Computing, Coordination and Collaboration: Theories and Technologies for Computer-Supported Work, Austin, TX, 1989.

[2] K. S. Decker and V. R. Lesser. Designing a family of coordination algorithms. ICMAS-95, San Francisco.

[3] C.Castelfranchi and R.Conte. Distributed artificial intelligence and social science: Critical issues. *Foundations of DAI, Chapter 20*, 1996.

[4] K. S. Decker and J. Li. Coordinating mutually exclusive resources using gpgp. *AAMAS*, 3, 2000.

[5] K. Erol, D. Nau, J. Hendler, and R. Tsuneto. A critical look at critics in htn planning. *IJCAI95*, Montreal, Canada, Aug. 1995.

[6] J.Graham and K.Decker. Towards a distributed, environment-centered agent framework. *ATAL*, Springer-Verlag, 2000.