Coordination of Distributed Knowledge Networks Using Contract Net Protocol

Jihoon Yang, Raghu Havaldar, Vasant Honavar, Les Miller and Johnny Wong
Artificial Intelligence Research Laboratory Department of Computer Science
226 Atanasoff Hall Iowa State University Ames, IA 50011. USA.
`http://www.cs.iastate.edu/~honavar/aigroup.html`

Abstract— **Tools for selective proactive as well as reactive information retrieval, information extraction, information organization and assimilation, knowledge discovery using heterogeneous, distributed knowledge and data sources constitute some of the key enabling technologies for managing the data overload and translating recent advances in automated data acquisition, digital storage, computers and communications into advances in decision support, scientific discovery and related applications. Such *distributed knowledge networks* (DKN) have to be able to effectively utilize multiple autonomous, often independently owned and operated information systems. Given the complexity of the such systems and the need for autonomy of the components, multi-agent systems, because of their modularity, offer an attractive framework for the design of DKN. In such multi-agent systems, satisfactory completion of the tasks at hand depend critically on effective communication and coordination among the agents. This paper describes an approach to coordination and control of DKN through inter-agent negotiation using the *contract net protocol* (CNP). The feasibility of the proposed design is demonstrated on tasks involving information retrieval and knowledge discovery using distributed data and knowledge sources.**

I. Introduction

Recent advances in high throughput data acquisition technologies, digital storage technologies, computers and communications have made it possible to gather and store scientific data (e.g., genome data), industrial data (e.g., design and manufacturing data) and military data (e.g., intelligence data), and business data in electronic form in databases and computerized information systems. In order to translate the advances in our ability to acquire and store data in increasing volumes and at increasing rates into gains in our understanding of the respective domains and new capabilities for effective decision-making, sophisticated tools are needed for information retrieval, knowledge discovery, and decision support [1].

Several application domains (e.g. military command and control, law enforcement, scientific discovery), require the use of multiple, geographically distributed, heterogeneous data and knowledge sources (e.g., sensors, satellites, intelligence reports, etc.) and analysis tools. Often, the data and knowledge sources and analysis tools are independently owned and operated and reside on different hardware and software platforms. Specific applications call for coordinated use of multiple resources and the exact combination of resources needed is not known a-priori. Multi-agent systems, because of their modularity, offer an attractive approach to the design of *distributed knowledge networks* that utilize multiple, heterogeneous, and often autonomous resources in decision support, scientific, distributed manufacturing, business, military command and control, and other applications. In such multi-agent systems, satisfactory completion of the tasks at hand depend critically on effective communication and coordination among the agents. In order to harvest the potential power of such systems in practical applications, it is essential that suitable mechanisms be devised to exercise adequate control over the behavior of such systems. In multi-agent systems, the notion of control suggests such functions as coordination among agents, synchronization among multiple agents, activation and deactivation of individual agents or groups of agents, selection among agents, creation of new agents when needed, elimination of agents that are no longer needed, adaptation of individual agents and agent populations to changes in the environments or task demands, learning (both at the individual as well as group levels) from experience, and (at a much slower timescale) evolution of agent populations toward more desirable behaviors. Both natural systems (e.g., cells, brains, immune systems, evolution, groups, social organizations, economies, societies) and artificial systems (computers, multicomputers, computer networks, programs, factories) offer rich sources of examples of a wide vari-

This research was partially supported by grants from the National Science Foundation (IRI-9409580) and the John Deere Foundation to Vasant Honavar.

ety of coordination and control mechanisms that can be beneficially incorporated into the design of complex information processing systems in general [2], [3], and distributed knowledge networks in particular: coordination that emerges from interaction among large number of agents that exhibit relatively simple behaviors inspired by organizations such as the ant colonies [4]; hierarchical control where the flow of control follows the structure of the hierarchy (e.g., in the military); coordination that emerges from interaction (including communication and negotiation) among self-interested agents as exemplified in the contract net protocol [5], [6], [7] and related negotiation mechanisms [8] and distributed routing in large self-managing communication networks [9]; control that emerges from competition for resources under the influence of environmental rewards as exemplified by evolutionary processes modeled by genetic algorithms [10].

This paper describes an approach to coordination and control of distributed knowledge networks through inter-agent negotiation using the contract net protocol (CNP) [5], [6], [7]. Preliminary results suggest the feasibility of using contract net protocol or its variants in distributed knowledge network applications for information retrieval and knowledge discovery using heterogeneous, autonomous, and distributed data and knowledge sources and analysis tools.

The rest of the paper is organized as follows: Section II describes the CNP and its implementation. Section III presents the results of experiments designed to evaluate the performance of CNP in a multi-agent system for information retrieval. Section IV concludes with a summary and discussion of some directions for future research.

II. Contract Net Protocol

The CNP was originally proposed as a tool for communication and control in a distributed problem solver [5]. Its use was demonstrated in a distributed sensing system in [5] and for a distributed delivery system in [6]. CNP provides a mechanism for agents to communicate and negotiate to solve a distributed problem via *contracts*. A contract is a set of tasks to be accomplished. Agents *announce* tasks that they need performed, make *bids* to perform tasks announced by other agents, *evaluate* the bids and *award* contracts. Each agent has a set of *capabilities* and *needs*. An agent's capabilities indicate the set of tasks that the agent can perform for other agents (perhaps at a certain cost). An agent's needs indicate the set of tasks that the agent needs to accomplish. The tasks that are part of an agent's needs candidates to be delegated to other agents: the tasks are announced to other agents, the bids for the tasks are collected, the best bid (which yields the highest utility e.g., profit, quality of the service, etc.) is determined and then the tasks are awarded to the agent submitting the best bid. Under certain assumptions, it can be shown that successive contracts each contract leads to a task allocation that is more beneficial to the entire society of agents that is involved in the negotiation process [11].

Our CNP implementation is based on a modular, object-oriented design which contains several domain-independent modules as well as a few domain-dependent (application-specific) modules. This design lends itself to being adapted for use in a broad range of applications. The domain-independent modules include: *Announcer, Bidder, Awarder, Awardee* as well as certain auxiliary modules. They work closely with the *Local Optimizer* module which calls application-dependent functions. The domain-independent modules operate as follows:

- The *Directory* is a special agent which provides assistance in locating agents. When an agent enters into the network, it registers with the directory.
- The *Messenger* handles sending and receiving of messages between agents. It queries the directory to locate agents. It also manages the storage and organization of messages. Other modules use these stored messages.
- *Phase Controller* controls the orderly execution of different stages of the negotiation process by calling the announcer, bidder, awarder and awardee.
- *Announcer* sends announcements to an appropriate set of agents. Each message defines the task to be fulfilled and sets any limitations (e.g., the maximum price it is willing to pay, the time limit it will be waiting to receive bids, and so on). The local optimizer selects a task from among the current needs for making an announcement. Only one announcement message is sent per cycle.
- *Bidder* generates bids in response to announcements received. The process is completed in two stages: It first discards announcements that correspond to tasks that can not be performed given the current capabilities of the agent or announcement that have expired. It then determines the appropriate bids by consulting the local optimizer.

- *Awarder* works with the local optimizer to process various bids (received within some time limit), generating awards based on some criteria (e.g., lowest cost, highest quality, etc.), and sends awards to appropriate agents. The agents whose bids are not selected are sent *loser* messages.
- *Awardee* accepts an award that is received, communicates with the local optimizer to update the state of the system (e.g., if it is supposed to supply some products, it decreases the quantity of products and increases the income/assets).

The *local optimizer* is a domain or application dependent module which incorporates domain-specific features to make decisions. Each phase of negotiation relies on certain functions performed by the local optimizer which ensures that each agent behaves rationally so as to maximize its utility.

The four phases of *Announcer, Bidder, Awarder* and *Awardee* are repeated until the negotiation reaches an equilibrium (i.e., no agent is in a position to improve its utility by delegating or accepting tasks to be performed) or an acceptable solution with anticipated utility is obtained. The following pseudo-code summarizes the CNP:

do forever (in *Phase Controller*)

Bidder:
- Read all the announcements received and check their expiration times;
- Generate bids within the limit of resources available (e.g., cash) for announcements that has not expired;
- Send the bids to the corresponding agents;
- Delete announcements processed;

Awarder:
- Read all the bids for announcements received, and drop any bids that are late;
- Pick the best bids for announcements whose time limit is expired;
- Send an awarding message to the winning agent, and send a loser message to all the other agents who had bid;
- Delete all expired announcements;

Awardee:
- Accept all awarding messages; Update the state of the system;
- Get all the loser messages;
- Remove all the announcements received corresponding to the awarding/loser messages;

Announcer:
- Pick one item from *Needs* set (that gives highest utility) and send out announcements to agents;
- Re-announce announcements that did not receive any bids within the time limit;

end

The overall view of the multi-agent system and the structure of an agent is shown in Figure 1 and Figure 2, respectively.

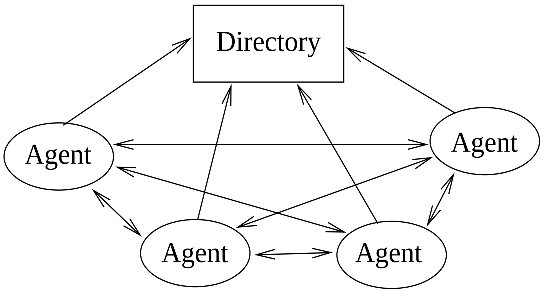

Fig. 1. Overall View of the System

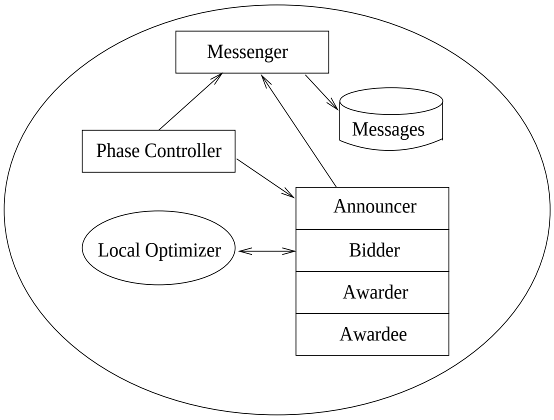

Fig. 2. Agent Design

III. Coordination of Information Retrieval and Knowledge Discovery Agents

In order to explore the feasibility of using CNP for coordination among multiple agents in a distributed knowledge network, we designed and implemented a multi-agent system for information retrieval and knowledge discovery from distributed data sources. Each agent maintains a needs table and a capabilities table. The needs table contains description of the data (e.g. a list of queries) that the agent needs, and the capabilities table contains description of the agent's capabilities (e.g., the information that is able to supply, or analysis functions that is able to perform on data) that can be utilized by other agents. Items in both tables have costs associated with them. The cost in needs table is the maximum cost the agent is willing to pay to get the information. The cost in capabilities table is the amount of money the agent wants to make by supplying the information. These costs are used by the local optimizer during the negotiation process. Each agent seeks to maximize its utility. The framework is flexible enough to allow an agent to provide value-added information to other agents. For example, an agent can obtain data from other agents, perform analysis of the data to extract some useful knowledge, and then provide the knowledge back to the agents that need it. Six agents were used in the CNP in our experiments.

The simulation results revealed the effectiveness of the CNP. Each agent was able to acquire the information that it needed from other agents at minimum cost attainable through the negotiation process. Sometimes, it was observed that some agents' bids were not sent in time (because of processing and communication delays) and thus were eliminated in the awarding phase. However, each agent always got the best deal among all the bids available at any given time.

IV. Summary and Discussion

CNP provides an attractive framework for negotiation and coordination among self-interested rational agents. Within this framework, each agent can announce tasks, make bids, evaluate bids made by other agents to complete the tasks, and offer contracts. Hence, it offers a possible mechanism for coordination among multiple autonomous agents in multi-agent systems.

Our implementation of the CNP, because of its modular design, lends itself to being adapted for use in a broad range of applications of distributed knowledge networks. The feasibility of this approach to coordination among multiple automous agents was demonstrated on an application involving information retrieval from distributed data sources. A number of issues including the scalability of the approach in large multi-agent systems remain to be explored. An interesting direction for further research involves hybrid coordination mechanisms that use CNP in combination with a hierarchical organization of multi-agent systems. Also of interest are adequately expressive languages for expressing the needs and capabilities of agents across different classes of applications of distributed knowledge networks.

References

[1] V. Honavar, L. Miller, and J. Wong, "Distributed knowledge networks," in *IEEE Information Technology Conference*, Syracuse, NY, 1998.

[2] V. Honavar and L. Uhr, "Coordination and control structures and processes: Possibilities for connectionist networks," *Journal of Experimental and Theoretical Artificial Intelligence*, vol. 2, pp. 277–302, 1990.

[3] L. Uhr, *Algorithm-Structured Computer Arrays and Networks: Architectures and Processes for Images, Percepts, Models, Information*, Academic Press, New York, 1984.

[4] D. Hofstadter, *Godel, Escher, Bach : an Eternal Golden Braid*, Basic Books, New York, 1979.

[5] R. Smith, "The contract net protocol: High-level communication and control in a distributed problem solver," *IEEE Transactions on Computers*, vol. C-29, no. 12, pp. 1104–1113, December 1980.

[6] T. Sandholm, "An implementation of the contract net protocol based on marginal cost calculations," in *11th National Conference on Artificial Intelligence (AAAI-93)*, Washington, DC, 1993.

[7] T. Sandholm, "Contract types for satisficing task allocation: I theoretical results," in *AAAI 1998 Spring Symposium: Satisficing Models*, 1998.

[8] J. Rosenschein and G. Zlotkin, *Rules of Encounter: Designing Conventions for Automated Negotiation Among Computers*, MIT Press, Cambridge, MA, 1994.

[9] A. Mikler, V. Honavar, and J. Wong, "Analysis of utility-theoretic heuristics for intelligent adaptive network routing," in *Proceedings of the Thirteenth National Conference on Artificial Intelligence (AAAI-96)*. 1996, vol. 1, pp. 96–101, AAAI Press.

[10] M. Mitchell, *An Introduction to Genetic algorithms*, MIT Press, Cambridge, MA, 1996.

[11] T. Sandholm, "Necessary and sufficient contract types for optimal task allocation," in *Fourteenth International Joint Conference on Artificial Intelligence (IJCAI-97)*, Nagoya, Japan, 1997.