AN ELECTRONIC COMMERCE FRAMEWORK FOR RESOURCE ALLOCATION AMONG MULTI-AGENT ENTERPRISES

Alcino Cunha **Orlando Belo**
Departamento de Informática, Escola de Engenharia, Universidade do Minho
Largo do Paço, 4709 Braga Codex, PORTUGAL
{alcino, obelo}@di.uminho.pt

Abstract

This paper presents an electronic commerce framework for resource allocation in multi-enterprise environments. Within each enterprise the resources are allocated by a multi-agent system in order to increase the allocation process efficiency, reactivity, fault tolerance, and extensibility. We propose a multi-agent enterprise architecture based on a dynamic organizational structure, which allows the enterprise to appear in external environment as a single and coherent negotiation agent. An automated negotiation mechanism based on marginal cost calculations is presented, and the problem of determining the partners for a negotiation process is specially addressed and formalized.

1 INTRODUCTION

Resource allocation is one of the major interest areas to which the *Distributed Artificial Intelligence* (DAI) community, specially the *Multi-Agent Systems* (MAS) area, dedicates many research efforts. This interest is partially due to the possibility to apply a resource allocation framework to almost every computational problem [12]. Additionally, its suitability to be implemented in a distributed fashion way reforces such interest. Characteristics like the spatial distribution of the resources, the increasing of the system efficiency, reactivity, fault tolerance, and extensibility, are strong motives to select such implementation strategy.

Resource allocation is usually done inside a single enterprise. Many advantages can be achieved by extending it to a multi-enterprise environment. Sometimes, the resources owned by an enterprise may not be sufficient to accomplish some tasks or even to execute them efficiently. Using short term alliances in order to allow tasks' transference to other enterprises may be a solution for both situations. However, the self-interest of the involved entities forces the existence of a negotiation mechanism to calculate the monetary compensations that must be payed as a tradeoff for the execution of foreign tasks.

Software agents can be successfully used in such negotiation context, since they have characteristics that one may find in a real multi-enterprise environment, namely [4]: autonomy, social ability, responsiveness and pro-activeness. However, the design of their internal behavior and their negotiation protocols are very difficult tasks because of the self-interested character and the bounded rationality that agents may have. The later constraint exists due to [5]: the limited communication bandwidth; the limited processing power; the self-interested behavior of the agents (that restricts the information sharing); and the dynamic character of the environment.

A framework for electronic commerce among bounded rational self-interested agents have already been proposed in [7, 8]. Such framework does not allow the existence of several agents controlling an enterprise's resources, since it was designed only for self-interested agents. The use of a multi-agent approach inside each enterprise rises some problems due to the duality of behavior that each agent must assume simultaneously: self-interested when dealing with agents from other enterprises, and cooperative within the same enterprise [9].

This paper presents an electronic commerce framework for resource allocation in multi-enterprise environments, where each enterprise's resources are controlled by a multi-agent system, that overcomes the mentioned behavior duality problem. A similar version was already used in a practical application for load distribution in similar environments [1].

0

2 MULTI-AGENT ENTERPRISES

Each enterprise has a set of tasks and a set of resources to execute them. The resources are organized as a graph of processing nodes. The only demand imposed to this kind of organization are that every node must be capable of executing the system's agents and that communication lines exist between them to allow message passing.

2.1 Architecture

Within each enterprise there are two types of agents: *allocation agents*, that control the enterprise resources and allocate its tasks, either through negotiation processes with other enterprises or through cooperation processes with the remaining enterprise's allocation agents; and *routing agents*, that are responsible for routing the messages between allocation agents and from outside agents to the allocation agent in charge of negotiation with the enterprise to which they belong. There exists one routing agent for each processing node of the enterprise.

The quantity of allocation agents, their placement in the environment of the enterprise, and their responsibilities, are specified by a dynamic organizational structure, that determines their roles and communication patterns, reduces the complexity of their control problem, and constrains the range of their decisions, providing the basis for a coherent cooperation [2].

Each allocation agent is characterized by the tuple $< Location, Resources, Enterprises >$, where *Location* defines the node where the agent is located, *Resources* defines the set of nodes and communication lines that the agent controls, and *Enterprises* defines the set of enterprises with which the agent interacts. In order to increase efficiency, it is not possible that two agents have the same location and that one resource or organization is controlled by more than one agent. An external agent can contact an enterprise by any of its nodes. The routing agent of the contacted node routes the message to the allocation agent, which includes the sender in its *Enterprises* set. The replies are also routed through the entry node. This mechanism hides the internal architecture to the outside environment, and allows the enterprise to look as a single and coherent self-interested agent.

The decision to change the organizational structure is distributed among the allocation agents, as a way to avoid bottlenecks and contention. A specific protocol was designed to rule this change and to allow the conflict detection and resolution [1]. Conflicts can emerge from communication latencies. The decision about changing is made every time the enterprise satisfaction level decreases more than a certain value, or after a certain period since the last decision. This process is very costly, both in communication and computation, and should be tuned so that reorganization is undertaken infrequently [3].

2.2 Negotiation Protocols

Due to the behavior duality, two communication protocols are used. Inside the enterprise, the allocation agents cooperate trough a slight variation of the original *Contract Net protocol* (CNP) [11]. For negotiation processes between different enterprises it is used the CNP extension proposed in [8]. The original CNP cannot be used in this context because it does not deal, explicitly, with the self-interest of the agents. The major innovations of this extension are the use of multiple levels of commitment and a penalty policy to prevent decommitment situations. Furthermore, we extended this protocol to allow the negotiation of long-term contracts. These contracts are resource oriented and the monetary compensation is negotiated per unit of time. Such contracts allow: i) task transference between two enterprises without the necessity of a previous negotiation process; ii) the possibility to contract communication lines, and iii) a more effective way to announce the enterprise availability when a resource surplus situation occurs.

Every time a long-term contract is made, a virtual node is created in the remote node of the contracted enterprise with a capacity equal to the contracted resources capacity. To allow control transparency these nodes also have routing agents, which allows their inclusion in the *Resources* set of the contractors as normal nodes. To simplify the access to a virtual node, the enterprise can also contract some communication lines (also in a long-term fashion) in order to connect itself to the remote node. Besides control transparency, this concept also provides the basis to the extension of the market where the enterprise operates. This is achieved in two ways: the organizational structure can change, so that an allocation agent is moved to the virtual node (providing a cheaper access to a larger group of enterprises); or new enterprises can get in touch with the owner of the virtual node concerning the cheaper access to one of its nodes.

3 THE AUTOMATED NEGOTIATION MECHANISM

To enable the automated negotiation the allocation agents must quantify the monetary compensation that the enterprise is willing to tradeoff for the remote execution of a specific task. This compensation depends on the task characteristics and on the enterprise current level of satisfaction. The utility function $U(M, P_1, \ldots, P_{Np})$, where M stands for the available money and $P_1, \ldots, P_{Np}$ for the remaining Np parameters which influence the enterprise satisfaction, is used to quantify such tradeoff. Since it assesses the enterprise satisfaction, the utility function also provides a mean to implicitly define its goals.

3.1 Global Utility Assessment

Each allocation agent has only a partial view of the utility function parameters. To compute their values, in the whole enterprise, it is necessary that each agent broadcasts its local values of the parameters. To define the actualization rate of these values the allocation agent are parameterized with the factor Max_Age. This parameter defines the maximum age of its partial view of the utility function parameters. The age of a parameter is defined as $Age(P) = t_{act} - t_{obt}(P)$, where t_{act} stands for the actual time and $t_{obt}(P)$ for the last time when P was obtained. Considering N as the number of allocation agents, M^k and P_j^k as the values of, respectively, the available money and the value of the parameter j in the allocation agent k, and that each agent is capable of assess immediately its local value of the parameters, then in each allocation agent i it should be true that

$$\frac{\sum_{k=1}^{N,k\neq i}(Age(M^k) + \sum_{j=1}^{Np} Age(P_j^k))}{(Np+1) \times (N-1)} \leq Max\_Age. \tag{1}$$

The value of Max_Age can be different for each agent, allowing a more precise tuning according to the local environment. There is always a tradeoff between the quality of the decision process and the interconnection network saturation that must be quantified when defining the value for Max_Age. When condition 1 becomes false, the agent should request the update of some parameters. The overhead of this procedure can be made minimal, because it can be incorporated in the cooperation protocol used inside the enterprise. The need for explicit updating messages is a rare situation.

3.2 The Task Market Extent

Every time a task allocation process starts, it is necessary to decide if it will be executed inside the enterprise or if it will be negotiated abroad. It is also necessary to define precisely the partners with whom it will be negotiated. Both decisions are transparently made due to the concept of dynamic definition of the task market extent. This concept was retrieved from microeconomics concept of *product market extent* [6]. The market of a set of tasks T is defined as $M(T)$, the set of allocation agents to whom a contract announcement will be sent. This set results from the maximization of the expected return of an announcement that is calculated as

$$\max_{M(T)\subseteq Others} \mathcal{E}(\Delta U^{M(T)}(T)) > \Delta U_{add}(T), \tag{2}$$

where $Others$ is the set of all allocation agents except for the one that makes the announce, $\mathcal{E}(\Delta U^{M(T)}(T))$ is the expected change in utility (if the tasks T are announced to the set of agents $M(T)$), and $\Delta U_{add}(T)$ is the change in the utility that occurs if the tasks T are executed by the agent that announces them. Equation 2 guarantees that $M(T) = \emptyset$ when the tasks can be executed locally in an optimal way. Let $U'_{after} = U'(T_{no\_dec} \cup T'_{dec} \cup T'_{off} \cup T)$ and $U'_{before} = U'(T_{no\_dec} \cup T'_{dec} \cup T'_{off})$. $\Delta U_{add}(T)$ is bounded below by

$$\Delta U^{-}_{add}(T) = \min_{T'_{dec}\subseteq T_{dec}, T'_{off}\subseteq T_{off}} [U'_{after} - U'_{before}]$$

and above by

$$\Delta U^{+}_{add}(T) = \max_{T'_{dec}\subseteq T_{dec}, T'_{off}\subseteq T_{off}} [U'_{after} - U'_{before}].$$

This limits are based on those determined in [7]. $U'(T)$ is the value of the utility function with the set of tasks T in execution and is calculated heuristicly by a specific scheduler. T_{no_dec} is the set of tasks from which an agent can not decommit (tasks from the same enterprise), T_{dec} the tasks from which it can decommit by paying the negotiated penalty, and T_{off} the tasks that can be assigned to the agent in consequence of the offers that it previously made. The value of $\Delta U_{add}(T)$ is determined as

$$\Delta U_{add}(T) = \Delta U^{-}_{add}(T) + \alpha \times (\Delta U^{+}_{add}(T) - \Delta U^{-}_{add}(T)),$$

where α $(0 < \alpha < 1)$ stands for the agent risk attitude. A low value for α leads to a risk-seeking agent, that will try to send announces for more agents (by making $\Delta U_{add}(T)$ a more negative value) in expectation of a good offer. High α values will cause the agent to act more carefully. We can also see α as defining the agent exploratory behavior. In fact, a risk-seeking agent sends announces to larger groups of agents and societies and thus searches a larger task allocation solution space.

$\mathcal{E}(\Delta U^{M(T)}(T))$ is bounded below by

$$\mathcal{E}^{-}(\Delta U^{M(T)}(T)) = \min_{O\subseteq M(T)} \mathcal{E}_{aux}(\Delta U^{M(T)}(T)),$$

and above by

$$\mathcal{E}^{+}(\Delta U^{M(T)}(T)) = \max_{O \subseteq M(T)} \mathcal{E}_{aux}(\Delta U^{M(T)}(T)),$$

where

$$\mathcal{E}_{aux}(\Delta U^{M(T)}(T)) = \mathcal{E}(\Delta U^{O}_{off}(T)) + \Delta U^{M(T)}_{com}(T).$$

$\mathcal{E}(\Delta U^{O}_{off}(T))$ stands for the expected change in utility if the set of agents O made a valid offer, and $\Delta U^{M(T)}_{com}(T)$ for the change in utility caused by the communication costs of sending announces to all the agents in $M(T)$. Because a riskier agent expects a higher return from the announce, we have

$$\mathcal{E}(\Delta U^{M(T)}(T)) = \mathcal{E}^{+}(\Delta U^{M(T)}(T)) - \alpha \times \mathcal{E}_{dif}(\Delta U),$$

where

$$\mathcal{E}_{dif}(\Delta U) = \mathcal{E}^{+}(\Delta U^{M(T)}(T)) - \mathcal{E}^{-}(\Delta U^{M(T)}(T)).$$

The expected change in utility if the set of agents O made a valid offer is determined as

$$\mathcal{E}(\Delta U^{O}_{off}(T) = \max_{W \in O}(\mathcal{E}(\Delta U^{W}_{choose}(T))),$$

where $\mathcal{E}(\Delta U^{W}_{choose}(T))$ stands for the expected change in utility if the chosen agent to execute T is W, and is calculated as

$$\mathcal{E}(\Delta U^{W}_{ex}(T)) + \Delta U^{W}_{send}(T) + \Delta U^{O-\{W\}}_{dec}(T),$$

where $\mathcal{E}(\Delta U^{W}_{ex}(T))$, $\Delta U^{W}_{send}(T)$ and $\Delta U^{O-\{W\}}_{dec}(T)$ are, respectively, the expected change in utility if the task is executed by W, the change in utility caused by the communication costs of sending the task to W and the change in utility caused by the penalties that must be payed to the remaining agents whose offers must be rejected.

The duality of behavior is handled transparently, because, when an agent is from the same enterprise, parameters like $\Delta U_{send}(T)$ or $\Delta U_{dec}(T)$ have values very close to zero or even zero. Thus, the equation 2 implicitly gives priority to the agents of the same enterprise when calculating $M(T)$. However, it is important to remember that all the decisions are taken in an economic perspective and try to maximize the expected utility of the consequent action. So, if the execution in an remote enterprise is expected to be more profitable, even with all the costs associated to the remote negotiation, that enterprise will be included in $M(T)$.

3.3 Learning How to Estimate

$\mathcal{E}(\Delta U^{W}_{ex}(T))$ is estimated based on its past values stored in the allocation agent's memory. The size of the memory is parameterized by ΔM. Lower values for ΔM augment the agent reactivity, because a smaller memory only allows to keep record of the most recent events. Higher values provide more secure and informed decisions. If N samples of $\mathcal{E}(\Delta U^{W}_{ex}(T))$, $X_1, \ldots, X_N$, exist in the agent's memory, then the average and the standard deviation are defined as

$$\overline{X} = \frac{\sum_{i=1}^{N} X_i}{N} \quad and \quad s = \sqrt{\frac{1}{(N-1)} \times \sum_{i=1}^{N}(X_i - \overline{X})^2}.$$

Let $t_{1-\alpha,N-1}$ and $z_{1-\alpha}$ be the critical points of, respectively, the t - Student distribution and the standard normal distribution that contain at their right a probability of $1-\alpha$. μ, an estimation to $\mathcal{E}(\Delta U^{W}_{ex}(T))$, can be determined as [13]

$$\mu = \begin{cases} \overline{X} + t_{1-\alpha,N-1} \times \frac{s}{\sqrt{N}} & if N \leq 30 \\ \overline{X} + z_{1-\alpha} \times \frac{s}{\sqrt{N}} & otherwise. \end{cases}$$

This value is used in order to bias the estimative according to the agent's risk attitude. Let

$$Age_{sample} = \begin{cases} \Delta M & if N \leq 1 \\ \min_{1 \leq i \leq N}(Age(X_i)) & otherwise, \end{cases}$$

and

$$\beta = \frac{\Delta M - Age_{sample}}{\Delta M}.$$

β is a biasing factor that changes according to the age of the sample, in order to give more credit to those that result from recent negotiations. Finally, the estimation of $\mathcal{E}(\Delta U^{W}_{ex}(T))$ is calculated as

$$\mathcal{E}(\Delta U^{W}_{ex}(T)) = \beta \times \mu + (1 - \beta) \times \gamma,$$

where γ stands for a parameterized default estimation when $N \leq 1$ (μ cannot be determined in such situation).

The calculations in the remaining phases of the negotiation process are done in a similar fashion. The parameters α and ΔM are used to bias almost all of them according to the desired behavior for the agents. These parameters are strongly related and should not be tuned separately. The relation between the exploratory behavior and the reactivity of an agent was studied in [10]. One of the major conclusions of that study was that, in order to obtain high performance, an exploratory agent should also be reactive. In our case, this implies that low values for α should be paired with low values for ΔM.

4 CONCLUSIONS

This paper presents an electronic commerce framework for resource allocation between multi-agent enterprises. The use of a negotiation mechanism based on marginal utility calculations and the dynamic definition of the task market extent provide the allocation agents with the ability to manage, transparently, the dual behavior that they must have in such environments. The use of a dynamic organizational structure allows each enterprise to appear as a single and coherent negotiation agent. The concepts of resource oriented contract and virtual node provide an effective way to explore long term synergies between enterprises. The possibility to define the internal behavior of the agents gives to the framework the necessary flexibility to adapt to the specificities and dynamic character of almost every enterprise and environment. The use of a multi-agent system to control the resources of each enterprise make it suitable to be applied in very large and spatially distributed enterprises and provides the necessary negotiation throughput in frequent outsourcing contexts.

ACKNOWLEDGMENT

The present work receives financial support from JNICT under the scholarship BD/9269/96.

References

[1] Alcino Cunha and Orlando Belo. A multi-agent based approach for load distribution in multi-enterprise environments. In *IASTED International Conference on Applied Informatics (AI-97)*, pages 306–309, Innsbruck, Austria, February 1997.

[2] Edmund H. Durfee, Victor R. Lesser, and Daniel D. Corkill. Coherent cooperation among communicating problem solvers. *IEEE Transactions on Computers*, C-36(11):1275–1291, November 1987.

[3] Edmund H. Durfee, Victor R. Lesser, and Daniel D. Corkill. Trends in cooperative distributed problem solving. *IEEE Transactions on Knowledge and Data Engineering*, 1(1):63–83, March 1989.

[4] Nick Jennings and Michael Wooldridge. Software agents. *IEE Review*, pages 17–20, January 1996.

[5] Victor R. Lesser. Multiagent systems: An emerging subdiscipline of ai. *ACM Computing Surveys*, 27(3):340–342, September 1995.

[6] Robert S. Pindyck and Daniel L. Rubinfeld. *Microeconomics*. Maxwell Macmillan International Editions, second edition, 1992.

[7] Tuomas Sandholm. An implementation of the contract net protocol based on marginal cost calculations. In *Eleventh National Conference on Artificial Intelligence (AAAI-93)*, pages 256–262, Washington D.C., 1993.

[8] Tuomas Sandholm and Victor R. Lesser. Issues in automated negotiation and electronic commerce: Extending the contract net framework. In *First International Conference on Multiagent Systems (ICMAS - 95)*, pages 328–335, San Francisco, 1995.

[9] Tuomas Sandholm and Victor R. Lesser. On automated contracting in multi-enterprise manufacturing. In *Improving Manufacturing Performance in a Distributed Enterprise: Advanced Systems and Tools*, pages 33–42, Edimburgh, Scotland, 1995.

[10] Andrea Shaerf, Yoav Shoham, and Moshe Tennenholtz. Adaptive load balancing: A study in multi-agent learning. *Journal of Artificial Intelligence Research*, 2:475–500, 1995.

[11] Reid G. Smith. The contract net protocol: High-level communication and control in a distributed problem solver. *IEEE Transactions on Computers*, C-29(12):1104–1113, December 1980.

[12] Michael P. Wellman. The economic approach to artificial intelligence. *ACM Computing Surveys*, 27(3):340–342, September 1995.

[13] Fadil H. Zuwaylif. *General Applied Statistics*. Addison-Wesley Publishing Company, 1972.