An Optimized Hierarchical Model for Agent-Mediated E-Commerce

CRISTEA BOBOILA
University of Craiova
Faculty of Mathematics and Computer Science
Al.I. Cuza Street, No. 13, Craiova RO-200585
ROMANIA

MARCELA S. BOBOILA
Stony Brook University
Computer Science Department
Stony Brook, NY 11794-4400
USA

Abstract: This paper presents an agent-based hierarchical model for electronic commerce. We advance our research in two possible directions that we propose for e-commerce environments, the Single Perspective Hierarchy and the Multi-Perspective Hierarchy models for resource discovery. We study their comparative performance with respect to traffic overhead and time efficiency. Our approach uses logarithmic search techniques, thus providing improved behavior in comparison with traditional linear methods used for resource discovery in distributed settings.

Key–Words: Electronic commerce, Business groups, Hierarchical model, Multi-agent systems, Distributed systems

1 Introduction

The future Internet that we envision is a collaborative environment motored by automated services. These characteristics are also reflected in one of its major components, electronic commerce, where a global infrastructure provides the benefits of a rich and rewarding environment, from which all business parties can nevertheless profit.

In this paper, we propose an agent-based solution that maps well on a collaborative scheme. In our model, we want to increase performance during the resource discovery in a multi-agent scenario. We render a structured communication scheme very efficient, and think that the degree of optimization employed by such a scheme overruns the difficulty of setting a structured communication environment.

We focus our research on a hierarchical communication model, where each agent is a node in a B-tree structure. A B-tree structure benefits from a logarithmic search time, which is a significant optimization for communication. Two logical models are further deployed on a hierarchical structure. The first one is the *Single Perspective Hierarchy (SPH)*, in which the configuration is established at the beginning, and all nodes have the same, unitary view of the network structure. In a SPH, new nodes can join the hierarchy if they appear in the collaborative group (Section 4), but only one root node can exist at a time. The second model is the *Multi-Perspective Hierarchy (MPH)*. In a MPH, each node is the root of a virtual hierarchic tree, and it has its own view of the network structure. We mathematically study the performance of each of these approaches, in terms of *round trip time* (Section 4) and communication overhead.

The rest of the paper is structured as follows: Section 2 presents related work in this area. Section 3 describes general features of the model proposed. Section 4 gives a comprehensive description of the SPH model, emphasizing on performance issues. Section 5 presents the second model, MPH, and outlines what it brings new in comparison with SPH. Section 6 introduces implementation details of the communication scheme. Section 7 describes simulation results that characterize the performance of the methods. Finally, Section 8 concludes the paper.

2 Related Work

Research in this area has been particularly directed towards agent negotiations [1][3][6], where possible scenarios are explored during the negotiation process until a mutual agreement is found. The negotiation is basically a constraint satisfaction problem, because the agents are limited by the offers advertised by the others. These offers are sometimes imprecise or incomplete, but agents still need to arrive to a consensus.

In a dynamic environment such as the electronic commerce, agents must carry out various tasks that can be viewed as a workflow of activities. Previous research on multi-agent systems has also been concerned with the relationships between multi-agent cooperation and workflow [7][8][10]. The automation of the business process may involve the execution of collaborative tasks. Each task represents a logical piece of work that contributes to a process [10] and can be associated to a role. Predefined agent roles such as query agents, resource agents, etc. can facilitate the functioning of a distributed, multi-agent system. Yet another substantial part of previous research has focused on mobile agents and methods to provide support for agents' migration and users' mobility [4][9].

3 General Model

A business group (BG) may contain several parties (entities) each with particular economic interests. Figure 1 depicts a BG in which a company (Company 1) wants to purchase a specific resource R. We can see from the figure that Company 2 can provide this type of resource. The process of resource discovery in a BG implies the identification of members with appropriate resources. We may require mechanisms for discovering and purchasing appropriate resources, for providing access to remote storage resources, and so forth.

In a dummy approach, Company 1 has to query each particular party involved whether it possesses that item or not. This method implies sequential communication with each member of the BG and leads to low performance. A multi-agent infrastructure formed of intelligent agents should be implemented in order to provide a robust and efficient BG structure.

We propose a hierarchical model of resource discovery, where the topology of the parties involved in the electronic commerce scenario is associated with a binary tree structure. Each branch represents a communication link and each node in the tree represents an entity that may possess and/or request resources during the business process. Each node can be an agent that wants to discover the necessary resources or items, a broker that provides various items, or it can have both functionalities [2].

4 Single Perspective Hierarchy

The single perspective hierarchical model (SPH) is presented in Figure 2. Each party has the same view of the communication structure in the BG. The hierarchy is therefore *absolute*. Any node in the binary tree - a leaf, the root or any intermediary node - may request an item, and any of them can provide that item. The algorithm has a series of steps that are described in pseudo-code below:

1. Node A, which denotes agent A, needs item I and creates a request message m.

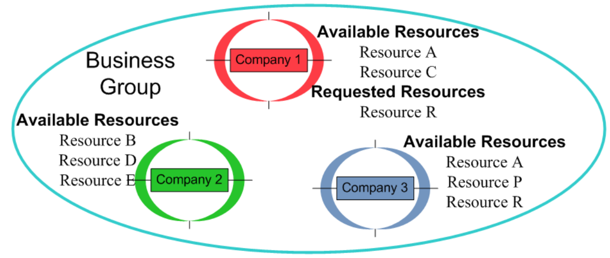

Figure 1: A business group (BG)

2. Node A sends m to its parent in the B-tree and its children (if it has any).

3. For all nodes n that receive m do

   If n has the requested item, reply.

   Else If m was received from a child then

   (a) Forward message m to the other child of node n.

   (b) If node n is not the root then forward message m to n's parent.

   Else forward m to your children.

We can observe two distinct parts of this algorithm. The first consists of a *bottom-up* approach (the *Else If* clause), where at each step, a node forwards the message it receives to its parent (if it is not the root) and the other son (the one from which it did not received the message). We call this *bottom-up* because the request travels up to the parent, towards the root. In the second part, after arriving at the root of the hierarchy, we have a *top-down* approach (the *Else* clause). The message is now always forwarded to the two children, traveling from the root to the leaves and covering the second branch of the tree relative to the root.

In a requester-responder communication, the *round trip time (RTT)* is the time necessary for the request message to travel from the requester to the responder and for the reply to get back to the requester. In the worst case, the requester and the responder are the furthest apart, on the leaf level (Figure 2). For example, the requester can be the leftmost leaf in the tree and the responder can be the rightmost leaf. In this case, the message has to climb all the way to the root and then go down on the right branch to the leaf. The number of levels in a B-tree with n nodes is logarithmic, given by $log_2 n$.

In a simplifying assumption, let us consider that all the links have the same length and traveling speed, and are covered in a unit of time. Therefore, the worst time that we obtain for a request-response message is:

$$T_{SPH} = 2 \cdot (L - 1) = 2 \cdot log_2 n - 2 \; , \qquad (1)$$

where n is the number of nodes (agents) and L is the number of levels in the tree.

After the provider has been found, a direct link with the requester can be established by calling the associated proxy (Section 6). Consequently, the RTT is given by the formula:

$$RTT_{SPH} = T_{SPH} + 1 = 2 \cdot log_2 n - 1 \; . \qquad (2)$$

The drawback of the method is related to the message traffic generated in the BG. At each step, the node that is forwarding the request must transmit the message on two routes: either to the parent and to the other child (the sibling that did not transmit the message), or to its two children. There are two exceptions though: if the current node is the root of the hierarchy or the initial node (the requester), the message must be forwarded on a single route. Therefore, each node usually sends the received message on one route and a copy of this message on the other. The number of messages thus doubles with each forwarding. In the worst case, the whole tree will be covered, meaning that a message will travel along each of the links between any two adjacent nodes. This leads to a worst case number of messages of $N - 1$, where N is the number of nodes in the binary tree.

We can see that a substantial reduction of the overhead induced by messages sent can only be achieved during the bottom-up part of the algorithm. At this stage, if a node can provide the requested item, it will stop and not continue to forward the message to the parent. Thus, a large part of the tree that was accessible through that parent does not have to be covered. We know that, if the responder node is the root, as much as half of the nodes in the binary tree will not be visited, otherwise even more than half of them will not be accessed, depending on the position of the responder. On the other hand, during the top-down stage of the algorithm, only a small part of the tree, namely the sub-tree of the responder will not be covered. There is no control over the rest of the routes.

5 Multi–Perspective Hierarchy

In the multi-perspective hierarchical model (MPH) that we propose, each agent has a personal, distinct view of the communication structure in the BG. The notion of hierarchy is therefore *relative* to each entity. Basically, each agent is the root of its own hierarchy represented as a binary tree, and therefore the solely requester. Any other node may provide the item and be the responder.

Each agent creates its personal hierarchical view according to some criteria. In this context, we use

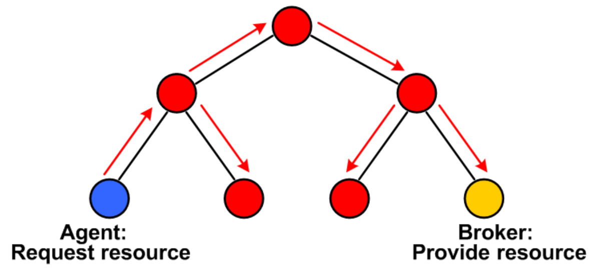

Figure 2: The SPH model

the notion of *priorities* to differentiate between parties. For a certain business entity A, an agent B may have a higher priority than another entity C, because it usually provides resources that are needed by A. As a result, the concept of *levels of priorities* emerges. The levels of priority are associated with the levels in the binary tree. The importance that the root gives to a node decreases towards the leaves. Since business entities with higher priority are closer to the root, the speed of resource discovery is increased.

The *Requester Ids* showed in the first column are unique id-s associated with a node when entering the group. The forwarding table (Table 1) for *Node 1* states how this node is viewed in the relative hierarchy of each other node and to what nodes it should forward a request message coming from them. For example, the first row states that in the hierarchical view with *Node 2* as root, the parent of *Node 1* is *Node 4* and it has no children (it is a leaf). This tabular representation of the forwarding information reduces the amount of memory necessary in a MPH approach.

The longest time required by a message to travel from the requester (who is always the root relative to its hierarchy in the MPH model) until it arrives at a node that can provide the item is:

$$T_{MPH} = L - 1 = log_2 n - 1 \; , \qquad (3)$$

with n being the number of nodes and L the number of levels in the binary tree. When the provider is found, a direct connection link can be established with the requester. That leads to a RTT value of:

$$RTT_{MPH} = T_{MPH} + 1 = log_2 n \; . \qquad (4)$$

With the multi-perspective hierarchical model, the RTT for the worst case is 2 times less than in the case of single perspective hierarchical model. Depending on the relative position of the requester and the responder for MPH and SPH, there can be cases where the RTT is less for the SPH model, although the worst case for MPH performs better. With MPH, since each node can only forward the message to its children,

Requester Id	Parent	Children
2	4	-
3	4	2,6
4	2	3,7
5	3	-
6	2	4,5
7	5	2,3

Table 1: Forwarding table for *Node 1*

Setting	Description
$N = 50$	Number of nodes (entities in the business group)
$t^{i,j} \in [0 \ldots 20]$	Travel time along link between nodes N_i and N_j, in time units
$N_{exp} = 50$	Number of runs with different (agent, broker) distributions

Table 2: Simulation settings

we always go down the tree, in a top-down approach. Therefore, we do not benefit from the message traffic reduction that may appear in the bottom-up part of the SPH algorithm (Section 4). We have a trade-off between the search speed (the worst case is double times faster) on one hand and the additional storage and message traffic on the other hand.

Imagine a MPH scheme where an agent wants to discover the possible provider for resource X. The requester is situated at the top of its hierarchy, and the message sent from the root will have to travel on all branches, except the subtree with the responder as root (because the responder will not forward the message to its children any more). We can see therefore that the message traffic in the network has to count all the links that the message travels along and is roughly given by the following formula:

$$T_m = \frac{n}{2} + \frac{n}{2^2} + \ldots + \frac{n}{2^i} + 1 \,, \tag{5}$$

where n is the number of nodes and i is the level in the tree where the responder is situated. We start with the root on level 0. The equivalent form for expression 5 is:

$$T_m = 1 + n \cdot \sum_{j=1}^{i} \frac{1}{2^j} \,, \tag{6}$$

The message traffic is pretty large, approaching n when the responder is a leaf node. A large part of the traffic cost is due to the fact that the search continues on additional branches in the tree, even if the resource provider was found. In order to leverage this cost, we propose an optimization method that reduces the traffic in the subtrees that do not contain the resource provider.

We introduce the concept of *checkpoint nodes*, which denote a few randomly chosen nodes from the integral set, with the role of *contact points*. We formalize this concept as below:

Definition 1 *For any binary tree distribution T over the set of nodes $N = \{N_i | 0 <= i <= n\}$, where n is the cardinal of set N, $|N| = n$, we define N_CK to be the subset $N\_CK \subset N$ with the property: $\forall N_i \in N, \forall N_j \in N\_CK, (N_i, N_j)$ are directly connected.*

The property states that checkpoint nodes can be reached from any node in the network through a direct link. Once the provider has been found, it contacts the checkpoint nodes thus stopping the forwarding of the message to the nodes below the checkpoints. Our simulation results (Section 7) show that for a large set of nodes, a substantial reduction of traffic cost can be achieved. Also, we emphasize the random choosing of the N_CK subset. With a good random generator, we can benefit from the uniform coverage of the whole network, without the hassle of picking particular nodes.

6 Implementation aspects

The hierarchical models proposed are based on a multi-agent communication framework, developed in a distributed environment. We are using the Java platform that facilitates portability on several operating systems. We are using the lookup facility that the JINI technology [11] provides, in order for agents to participate in communication. We add new service providers and agents to the network by registering via a Discovery protocol to a lookup server.

From the implementation point of view, agents inter-communicate through a proxy, using semantic objects. The proxy is obtained by querying the JINI lookup service for a service implementing the agent or broker interface. The XML format for semantic objects favors a very flexible communication, where elements of interest can be user-defined. Below is an example of request for a specific resource that an agent can send.

```
<TRANSACTION_ID> 724 </TRANSACTION_ID>
<ACTION_TYPE> request </ACTION_TYPE>
<SENDER_ID> 7 </SENDER_ID>
<RESOURCE>
   <NAME> resource X </NAME>
   <QUANTITY> 1 </QUANTITY>
```

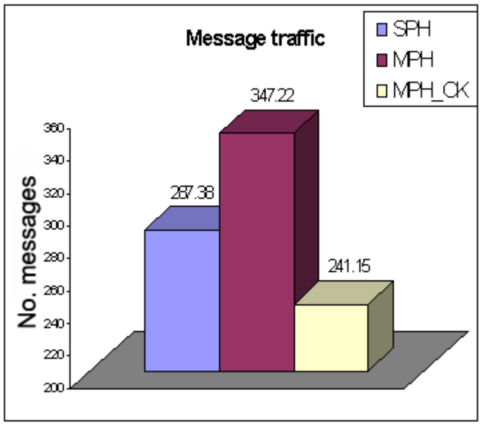

Figure 3: Traffic overhead in the business group

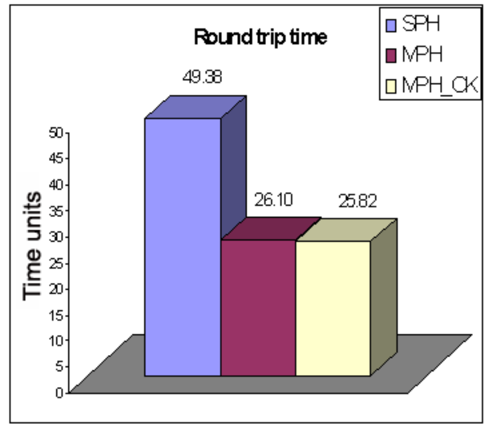

Figure 4: Average round trip time for messages

```
      <QUALITY> Extra </QUALITY>
      <DESCRIPTION>
         buyer’s description,
         specific requirements
      </DESCRIPTION>
   </RESOURCE>
   <TIMESTAMP>
      2/5/07 4:05:07pm
   </TIMESTAMP>
   <LIFETIME> 24 hours </LIFETIME>
```

We point out the importance of the timestamp and lifetime fields included in the description of a semantic object. Their presence was introduces in order to define a life period for any message in the network, and implicitly for any request from an agent (potential buyer). A message should not remain active for an indefinite period of time. An expired message would be useless and would only introduce iverhead in the communication network. Therefore, the lifetime period specifies a time in which the message is of interest for the emitting entity. After this period has passed, the message can be discarded.

7 Simulation results

We have studied the performances of the hierarchical models proposed in a simulated scenario with 50 nodes. The (agent, broker) distribution pairs, namely the agent that requests a specific resource and the broker that provides the resource were designated randomly from the entire set of nodes, in order to determine the average efficiency. We have run the experiments over 50 different (agent - requester, broker - provider) distributions in the hierarchy.

In real business environments, it is highly unlikely that the travel time of the message between two nodes is the same for any pair of nodes. The time variation can be due to several factors, ranging from different physical distances to different communication bandwidth over the internet. Therefore, we have considered connection links of different length, with distinct travel time along the links. In the simulated experiments we assign random travel time along links, given by a number of time units in the interval $[0 \ldots 20]$. A synthesis of the simulation settings is presented in Table 2.

The simulation results in Figure 3 show the average message traffic in the business group. The message traffic during a run of the algorithm is determined by the number of messages (representing the same semantic object) sent in the network. In over words, the traffic during a run is given by the number of links along which the messages describing a particular semantic object travel. Our results are given as an average measure of all the simulations carried out.

The average message overhead in the network is mathematically described as:

$$\overline{T_m} = \frac{\sum_{i=1}^{N_{exp}} \left(\sum_{j=1}^{N_{l_i}} t_j^{x,y} \right)}{N_{exp}} , \tag{7}$$

where N_{exp} is the number of runs carried out and N_{l_i} is the number of links covered during the i-th run. $t_j^{x,y}$ is the travel time along the link j, that connects nodes N_x and N_y.

We can see that the lowest message overhead is obtained in the third case, where we employ the MPH method, with checkpoint nodes. We can infer that the presence of checkpoints has a high influence over the message traffic, leading to substantial reductions. With a uniform distribution of checkpoint nodes we can obtain significantly good results, because large areas of the hierarchy, below the checkpoints, are no longer searched once the provider is found. Of course, in some cases, the agent may want to find the best fit for his request, and not stop when the first provider responded. He would rather trade efficiency (speed) and traffic reduction for quality of solution (resource). Therefore, the semantic object representing the request would also incorporate information with respect to the search model preferred. The results also show that the SPH model produces less message overhead than the MPH, without checkpoint nodes. That is due to the reduction in the number of messages which can be obtained during the bottom-up stage, as discussed in Section 4.

We also studied the average round trip time for each of the methods (Figure 4). We can see that the last two methods produce similar timing results with respect to finding the provider and delivering the result to the agent. They also give significantly better results than the first method, which is approximately double times slower, in average. The speed-up of the last two methods is determined by the way in which the scanning of the tree is performed, while searching for the provider. The scanning is carried out in a top-down approach, starting from the root of the hierarchy. In contrast, the first method, Single Perspective Hierarchy, may also perform bottom-up searches which slow the process down.

8 Conclusions

In the context of dynamic nature of electronic commerce, various business entities want to have a fast access to resources that exist in a business group. This paper proposes two methods of optimizing the discovery process of available resources: the Single Perspective Hierarchy method and the Multi-Perspective Hierarchy method. We improve upon usual linear methods and study a logarithmic search algorithm that uses binary trees to speed up the search.

In the Single Perspective Hierarchy method, we have a single view of the whole tree, shared by all business parties. Each node must transmit the message either bottom-up or top-down, or stop forwarding if it can provide the requested resource. If the resource is found during the bottom-up stage, the message overhead is reduced. With the Multi-Perspective Hierarchy method, each node has its own hierarchy, based on priority levels. The resource discovery is usually faster in this case, but we have to pay in terms of additional storage required by the forwarding tables. Furthermore, we do not benefit from overhead reduction obtained if the resource is found during the bottom-up stage, since the search with MPH always starts from the root and not from an arbitrary node in the tree. A significant improvement of the MPH method is obtained with the introduction of checkpoint nodes, that reduce the traffic overhead. As future directions of research, we intend to build upon the current communication framework to support large business groups in distributed environments. We will direct our study towards efficient algorithms and implementation extensions for collaborative, dynamic economic groups that perform transactions in e-business scenarios.

References:

[1] R. Kowalczyk and V. Bui, On Constraint-based Reasoning in e-Negotiation Agents, *LNAI (2000)*, Springer-Verlag, pp. 31 - 46, 2000.

[2] L. Palopoli, D. Rosaci, D. Ursino, Agents' roles in B2C e-commerce, *AI Communications*, IOS Press, Vol. 19, Issue 2, pp. 95 - 126, 2006.

[3] N. Jennings, P. Faratin, A. Lomuscio, S. Parson, C. Sierra and M. Wooldridge, Automated Negotiation: Prospects, Methods and Challenges, *Journal of Group Decision and Negotiation*, 2001.

[4] S.R. Tate, K. Xu, Mobile Agent Security Through Multi-Agent Cryptographic Protocols, *IC 2003*, pages 462-468, 2003.

[5] F. Buccafurri, D. Rosaci, G.M.L. Sarn and D. Ursino, An Agent-Based Hierarchical Clustering Approach for E-commerce Environments, *LNCS*, *DEXA*, Vol. 2455, pp. 109 - 118, 2002.

[6] P. Faratin, C. Sierra, N. Jennings and P. Buckle, Designing Flexible Automated Negotiators: Concessions, Trade-Offs and Issue Changes, Institut d'Investigacio en Intelligencia Artificial Technical Report, *RR-99-03*, 1999.

[7] M. Griss, G. Bolcer, L. Osterweil, Q. Chen and R. Kessler, Agents and Workflow - An Intimate Connection, or Just Friends, *TOOLS99*, 1999.

[8] S. Bassil, M. Benyoucef, R.K. Keller and P. Kropf, Addressing dynamism in e-negotiations by workflow management systems, *DEXA*, 2002.

[9] Y. Wang, K.-L. Tan, X. Pang, A Parallel Dispatch Model with Secure and Robust Routes for Mobile Agents, *LNCS*, *DEXA*, Vol. 2455, pp. 129 - 138, 2002.

[10] Q. Chen, M. Hsu, U. Dayal and M. Griss, Multi-Agent Cooperation, Dynamic Workflow and XML for E-Commerce Automation, HP Laboratories, *HPL-1999-136*, 1999.

[11] A. Hampshire, G.S. Blair, JGrid: exploiting jini for the development of grid applications, *FIDJT 2002*, Springer Verlag, pp. 132-42, 2002.