A Brokering Protocol for Agent-Based E-Commerce

Kwang Mong Sim and Raymond Chan

Abstract—This paper explores issues of engineering agents that partially automate some of the activities of information brokering in e-commerce. In particular, it focuses on addressing the problem of connecting buyers and sellers. An algorithm that uses multiple criteria to match buyers and sellers based on prespecified user profiles was devised and implemented. The process of matching and connecting buyers and sellers are divided in four stages: selection, evaluation, filtering and assignment. Ideas of the approach of connection are realized in a testbed on securities trading consisting of a society of information agents (a broker agent, a recommendation agent and a record agent) and trading agents (buyer and seller agents) that communicates via a blackboard database. In addition, an information brokering protocol was devised and implemented to structure the interactions and information exchange among agents in the testbed. A series of experiments that were carried out showed favorable results in executing the protocol.

Index Terms—Agent-mediated e-commerce, information agent, information brokering, multi-agent system.

I. INTRODUCTION

Given the unprecedented surge in information sources globally the demand for acquiring valuable and strategic business information for (electronic) trading has been ever increasing. In particular, the difficulty of finding information about trading partners in the global business arena accentuates the need for electronic intermediaries to assist, navigate, and mediate the invocation of these services. This paper explores the issues of engineering a society of agents that partially automate the task of connecting buyers and sellers trading electronically.

Objectives: The goals of this research are to

1) design and implement an algorithm that connects buyer and seller agents;
2) devise a brokering protocol for specifying and structuring the interactions among electronic intermediaries and trading (buyer and seller) agents;
3) design and engineer a testbed that models and simulates some of the activities of information brokering in the domain of securities trading.

In Section II, the main highlights of this research: an approach for connecting buyer and seller agents [1] is exposed. The overall architecture of the agent-based brokering testbed (consisting of a broker agent, a record agent, a recommendation agent, buyer agents and seller agents.) is explicated in Section III. Section IV describes the broker protocol and explicates stages of interaction and information exchange among agents in the testbed. Some experimental results and evaluation are illustrated in Section V. In Section VI, the agent-based brokering testbed is compared with some extant agent-based brokering systems. Section VII concludes this paper.

II. CONNECTING BUYERS AND SELLERS

While issues of the connection problem in the domain of resource and task allocation have been addressed by Smith [2], this research explores the issues of matching profiles of buyers and sellers in electronic trading. Much like allocating resources (agents) to tasks in the contract net protocol [2], connections of buyers and sellers are based on a set of predefined selection criteria that is domain specific. In this research the process of connecting buyers and sellers are carried out in four stages: selection, evaluation, filtering and assignment as shown in Fig. 1.

Selecting requests and advertisements. This stage compares requests from buyers and advertisements from sellers. In selecting buyers (sellers) for an advertisement (request) three selection criteria are used. The algorithm prefers to select requests and advertisements

1) that are not expired or withdrawn;
2) with profiles and preferences that are closely matched;
3) that have not been previously assigned.

Criterion 1) ensures that only valid advertisements (requests) are selected. This is because advertisements (requests) can be withdrawn at anytime and housekeeping jobs are normally done periodically in batch. By preferring advertisements (requests) that are not expired or withdrawn in the early stage saves system resources in connecting invalid buyer or seller agents and avoids recommending invalid connections to users. Table I illustrates connections with and without criterion 1).

While criterion 2) provides focus decisions [2] to connect trading partners with profiles and preferences that coincide, criterion 3) bears some resemblance in realizing the property of balanced loading [2] in the contract net protocol. For example, in the domain of securities trading, the stock type (for example,
Example 1: Connection with and Without Criterion 1

Suppose that there are 3 sellers A, B, C and 3 buyers 1, 2, 3 such that seller A’s advertisement has expired and request from buyer 2 is withdrawn.

Without criterion (1), nine connections are formed such that only four (thick lines in the above diagram) are valid.

With criterion (1), only four valid connection are formed. Less resource is required in handling these four (when compared to nine) connections is expected.

Table 1: Example 1: Connection with and Without Criterion 1

<table>
<thead>
<tr>
<th>Seller A</th>
<th>Buyer 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Seller B</td>
<td>Buyer 2</td>
</tr>
<tr>
<td>Seller C</td>
<td>Buyer 3</td>
</tr>
</tbody>
</table>

Asian Bank’s Stock) specified in a buyer’s request that matches the stock name/type specified in a seller’s advertisement (for example, Hang Seng Stock Index) with comparable buying/selling price and buying/selling volume satisfied criterion 2). If traders are recommended some services/products that are far from their expectation, they are more likely to repost their requests (advertisements). This may generate higher load for the system (which corresponds to the regeneration of subtasks in the contract net protocol [2] that leads to combinatorial explosion). Experiment 3 in Section V in this paper demonstrates that with criterion 2), the testbed can recommend connections that are quite close to users’ preferences. Criterion 3) ensures that every trading agent has quite uniform (albeit, not necessarily equal) opportunity of being connected to other trading partners and this is demonstrated in experiment 4 in Section V. The algorithm for selecting buyers and sellers is given in algorithm 1.

Evaluating Connections. In this stage the utility of each of the connection of trading partners is determined. The utility $U$ is a function that consists of a list of domain specific attributes $\langle a_1, a_2, \ldots, a_n \rangle$. One of the methods for determining the utility of a connection $C_i$ is given as follows:

$$ U(C_i) = \sum_{j=1}^{n} w_j \times a_j $$

where $w_j$ is a coefficient of arbitrary user-specified value that indicates the relative weighting of each attribute in calculating $U$. For instance, if an organization is more concerned about the delivery date but willing to absorb higher price, then a higher weighting is placed on delivery date than price.

The set of domain specific attributes collectively determines the selection of a service or product. For example, in the domain of securities trading, price ($P$), volume ($V$) and desirability ($D$) are three possible attributes with values from 1 (very poor) to 10 (very good). $D$ is determined by analyzing the transaction histories of a trading agent. In evaluating trading agents, the factors that contribute to higher rating (desirability) may include: the numbers of previously successful transactions and the payment records of the traders (that is, the traders deliver payment in time).

Algorithm 1: Selecting Buyers and Sellers.

**Procedure: Select Connection**

Input: $\{$request$\}$, $\{$advertisement$\}$ /* Request queue and advertisement queue */

Output: $\{$connection$\}$ /* Connection queue */

Processing:

Set $\{$connection$\} = \emptyset$ /* Criterion 1 (not expired, not withdrawn) and criterion 3 (not assigned) */

For each $r_{k,i} \in \{$request$\}$ which is not expired, withdrawn or assigned

For each $a_{j,k} \in \{$advertisement$\}$ which is not expired, withdrawn or assigned

/* Criterion 2 */

If profile and preference of $r_{k,i}$ match those of $a_{j,k}$ and connection for $r_{k,i}$ and $a_{j,k}$ does not exist then

Generate a new connection $\text{conn}_{k,i}$ from $r_{k,i}$ and $a_{j,k}$

Set $\text{conn}_{k,i} \cdot \text{connected} \cdot \text{count} = \text{conn}_{k,i} \cdot \text{connected} \cdot \text{count} + 1$

Set $a_{j,k} \cdot \text{connected} \cdot \text{count} = a_{j,k} \cdot \text{connected} \cdot \text{count} + 1$

End-if

End-for

Set $r_{k,i} \cdot \text{select_cycle} = r_{k,i} \cdot \text{select_cycle} + 1$

End-for

Algorithm 2: Evaluating Connections.

**Procedure: Evaluate Connection**

Input: $\{$connection$\}$, desirability $D$

Output: utility $U$

Processing:

For $j = 1, \ldots, n$ set each coefficient $w_j$ to an arbitrary value /* Calculate the value of utility for all connections in the connection queue */

For each $\text{conn}_{k,i} \in \{$connection$\}$ which is not assigned

Find desirability $D$ of $\text{conn}_{k,i} \cdot \text{buyer}$ and $\text{conn}_{k,i} \cdot \text{seller}$ from local database

Calculate $U = U(C_i) = \sum_{j=1}^{n} w_j \times a_j$

Update $U$ in $\text{conn}_{k,i}$

End-for

The utility function for securities trading is defined as follows:

$$ U(C_i) = w_1 \times P + w_2 \times V + w_3 \times D $$

Values of $P$, $V$ and $D$ range from 0 to 1 with the following interpretations:
TABLE II
EXAMPLE 2: OUTCOME OF FILTERING STAGE WITH DIFFERENT VALUES OF β

<table>
<thead>
<tr>
<th>Selection cycle</th>
<th>Connection found</th>
<th>Utility for the connection</th>
<th>After filtering (with β = 2)</th>
<th>After filtering (with β = 6)</th>
<th>After filtering (with β = 8)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>conn_1</td>
<td>5</td>
<td>accepted</td>
<td>deleted</td>
<td>deleted</td>
</tr>
<tr>
<td></td>
<td>conn_2</td>
<td>7</td>
<td>accepted</td>
<td>accepted</td>
<td>deleted</td>
</tr>
<tr>
<td></td>
<td>conn_3</td>
<td>9</td>
<td>accepted</td>
<td>accepted</td>
<td>accepted</td>
</tr>
<tr>
<td>2</td>
<td>conn_4</td>
<td>6</td>
<td>accepted</td>
<td>accepted</td>
<td>deleted</td>
</tr>
<tr>
<td></td>
<td>conn_5</td>
<td>8</td>
<td>accepted</td>
<td>accepted</td>
<td>accepted</td>
</tr>
<tr>
<td>3</td>
<td>conn_6</td>
<td>9</td>
<td>accepted</td>
<td>accepted</td>
<td>accepted</td>
</tr>
<tr>
<td>4, 5</td>
<td>no connection found</td>
<td></td>
<td></td>
<td>(selection completed after 5 cycles)</td>
<td></td>
</tr>
</tbody>
</table>

If β = 2, selection will be completed after 2 cycles because the system found the number of expected connections (with utility 5,7,9,6,8). If β = 6, 3 cycles were needed and the number of connection found is 5 (with utility 7,9,6,8,9). If β = 8, 5 cycles were executed and the number of connection found is 3 (with utility 9,8,9). With β = 6, more connections with better utilities will be found.

1) smaller difference between buying and selling price results in a higher value of P;
2) smaller difference between buying and selling volume results in a higher value of V;
3) trading agent with larger number of previous successful transactions and better payment records receive a higher value of D. For trading agents with no transaction history, a middle value of 0.5 will be assigned since the broker is neutral in assessing their desirability.

The algorithm for evaluating buyer and seller agents is given in algorithm 2.

Filtering: This stage filters out connections in two steps. The first step drops connections with U lower than a predefined cutoff point β that is an adjustable parameter with value from one to ten. The use of β ensures that connections with low utility would not be recommended to buyers and sellers while maintaining the total number of recommendations. The optimizing value of β can be determined through experiments (see Appendix A). While a smaller value of β may result in shorter connection time (more connections found but with lower utilities), a larger value of β may decrease the number of connections (but results in higher utilities). An optimizing value of β that generally and frequently gives the most promising results is (from Appendix A) 5 or 6. Table II illustrates how the connection results vary with different value of β. This example shows that better connection results (more connections with higher utilities) are obtained when β is set to 6.

If the number of connections (after the first filtering step) is larger than the expected number of responses specified in the transaction profiles, the second step of filtering selects and filters out connections with the lowest ratings. This step ensures that the number of recommendation will not exceed users’ expectation. The whole filtering process ensures that users are given more desirable connections. The algorithm for filtering is given in algorithm 3.

Assigning buyers and sellers. This stage marks the completion of the selection cycle for a request (advertisement) and releases the connection results to buyers and sellers for review. The connection algorithm terminates if either one of the following two stopping conditions is met:
1) number of connection equals to the expected number of response specified by the buyers (sellers);
2) request (advertisement) has already gone through a pre-specified number of cycles of selection and filtering.
Algorithm 3: Filtering Connections.

Procedure: Filter Connection
Input: \{connection\}, \{request\}, \{advertisement\}
Output: \{connection\}
Processing:
Set an arbitrary value for $\beta$ (cutoff point) between 1 to 10
/* Step 1: Remove connections with utilities lower than the cutoff point $\beta$ */
For each $\text{conn}_k \in \{\text{connection}\}$ with assign flag $= N$
If $\text{conn}_k \cdot \text{utility} \ < \beta$ then
    Set $\text{conn}_k \cdot \text{expected_response} = \text{conn}_k \cdot \text{expected_response}$
End-if
End-for
/* Step 2: Remove connections with lowest utility if the total number of connections is larger than user’s expected value */
For each $\text{req}_k \in \{\text{request}\}$
If $\text{req}_k \cdot \text{connected_count} \ > \text{req}_k \cdot \text{expected_response}$ then
    Find $\text{conn}_k$ of $\text{req}_k$ with lowest utility
    Set $\text{conn}_k \cdot \text{connected_count} = \text{req}_k \cdot \text{connected_count}$
    Set $\text{adv}_j \cdot \text{connected_count} = \text{adv}_j \cdot \text{connected_count} - 1$
    Where $\text{adv}_j \cdot \text{advertisement_id} = \text{conn}_k \cdot \text{advertisement_id}$
    \{\text{connection}\} = \{\text{connection}\} \ - \ \text{conn}_k$
End-if
End-for
The algorithm to reduce the number of connections (when the total number of connections is larger than user’s expectation) for each $\text{adv}_j$ is similar to that for each $\text{req}_k$

Algorithm 4: Assigning Buyers and Sellers.

Procedure: Assigning Connection
Input: \{connection\}, \{request\}, \{advertisement\}
Output: \{connection\}
Processing:
Set an arbitrary value for max_select_cycle
/* Update the assign flag */
For each $\text{req}_k \in \{\text{request}\}$ with assign_flag $= N$
If $\text{req}_k \cdot \text{connected_count} \geq \text{req}_k \cdot \text{expected_response}$ /* Condition 1 */
    or $\text{req}_k \cdot \text{select_cycle} \geq \text{max_select_cycle} /* Condition 2 */$
    then
        Set $\text{req}_k \cdot \text{assign_flag} = Y$
        Set assign_flag of all connections to $\text{req}_k = Y$
End-if
End-for
The algorithm to update the assign_flag for each $\text{adv}_j$ is similar to that for each $\text{req}_k$

Condition 1 ensures that the expected number of responses is found after filtering and therefore the connection cycle can be terminated. Condition 2 checks if the requests (advertisements) are already in the system for a period of time and have gone through a predefined number of selection cycles but the expected number of responses cannot be found. While condition 1 minimizes the connection time, condition 2 prevents infinite waiting for connections of requests (advertisements) in the system. For example, if a user wants three recommendations and the system already finds three connections, it seems wise not to keep the request (advertisement) in the system any longer. Requests and advertisements not connected will be picked up in the coming selection cycles. The algorithm for assigning buyers and sellers is given in algorithm 4.

The filtering stage and the assignment stage together provide an optimization path that allows buyers and sellers to go through multiple cycles of connection selection. This will increase the chance of finding better trading partners.

To realize the ideas of the connection algorithm and to provide an electronic infrastructure for bolstering information brokering, an agent-based information testbed (Section III) and a brokering protocol (Section IV) were designed and implemented.

III. AGENT-BASED INFORMATION BROKERING TESTBED

The agent-based information brokering testbed [3] is shown in Fig. 2. It consists of a society of intermediary or information agents (the broker agent, the record agent and the recommendation agent) and trading agents (buyer agents and seller agents) using a blackboard database as a central repository. The features and functions of the testbed include:

- enabling buyers to post requests and sellers to post advertisements;
- connecting buyers and sellers (see descriptions in Section II);
- maintaining and documenting user profiles;
- documenting successful connections;
- making recommendations to potential buyers or sellers.

In addition to making connection between buyers and sellers, the testbed also has a mechanism for making recommendation to potential buyers or sellers.

The above features are bolstered by a society of agents described below:

Buyer Agent: Buyer agents provide user interface for the users of the testbed. They post requests or withdraw requests from the blackboard, view results of each request, and make purchase decisions.

Seller Agent: Seller agents are similar to buyer agents but act on behalf of human sellers.

Broker Agent: The broker agent connects buyer and seller agents together using the connection algorithm described in Section II. In addition, it manages and controls the transactions and accesses to the blackboard.

Record Agent: The record agent documents and records the transaction histories of users in the record database. Using the information in the database, user profiles can be traced and determined.

Recommendation Agent: By accessing documented transaction histories, the recommendation agent proactively seeks
potential buyers (sellers) when there is no matching sellers (buyers) or when the buyer to seller (seller to buyer) ratio is too large. Employing this ‘push’ technology, recommendations are sent to potential trading partners via e-mail.

Blackboard Database: The blackboard accepts 1) requests from the buyer agents in the request queue and 2) advertisements from the seller agents in the advertisement queue. It is used to record buyer to seller connection information from the broker agent in the connection queue, and the transaction results and user satisfaction from the buyer and seller agents in the result queue.

IV. AN INFORMATION BROKERING PROTOCOL

A. Stages of the Protocol

While the testbed provides an electronic infrastructure for the information and trading agents, the brokering protocol specifies the interactions and information exchange among the information and trading agents. The stages of the brokering protocol (shown in Fig. 3) are listed as follows:

1) Buyers send requests and sellers send advertisements to the broker agent to specify their profiles and preferences. Buyers and sellers may also withdraw requests or advertisements previously sent. For securities trading, buyers’ specifications may include the type of products (for example, stocks of banks) and the broker agent attempts to match them with actual product names based on information stored locally. Sellers’ specifications may include the product name (for example, stocks of Citibank).

2) The broker agent attempts to connect buyers and sellers (see Section II) by matching the specifications of requests and advertisements. It determines the rating of each connection made and filters out those with rating lower than a predefined cutoff point. Multiple connections for a request or an advertisement are also possible.
3) The protocol does not assume that connections for requests and advertisements will always be made. If advertisements (requests) with matching specifications cannot be found, the recommendation agent proactively 'push' e-mail recommendations to trading agents by accessing the record database to determine their profiles and transaction histories. Table III illustrates how the recommendation agent select advertisements and requests. For selecting advertisements/requests, the recommendation agent prefers to contact trading agents:

a) with the largest number of successful transactions,
b) that has most recently completed a transaction (that is, sold/bought a similar product/service).

The e-mails may contain additional information of the stock or the company found using Internet search engines.

4) Buyers and sellers review connections made by the broker agent.

5) Buyers and sellers complete the transaction. In this stage, trading agents can either take or reject recommendations made by the broker agent. They can also provide feedback to the brokering testbed by indicating their satisfaction level. Users’ satisfaction will be used as one of the evaluation criteria of the agent-based information brokering testbed.

B. Information Exchange Among Agents

The various types of messages that are transmitted and received by agents include: requests from buyer agents, advertisements from seller agents, connections of request to advertisements, withdrawals of requests or advertisements, and results of decision. For securities trading, the messages may contain the following information.

Request: Requests sent by buyer agents to the broker agent contain the following information: type of stock required, valid
Table III: Selecting Trading Agents for Recommendation

Suppose that the current content of the record database is in chronological order given as follows:

<table>
<thead>
<tr>
<th>Buyer Agent</th>
<th>Seller Agent</th>
<th>Stock</th>
<th>Transaction Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>B</td>
<td>1001</td>
<td>150</td>
</tr>
<tr>
<td>C</td>
<td>D</td>
<td>1031</td>
<td>360</td>
</tr>
<tr>
<td>E</td>
<td>B</td>
<td>1001</td>
<td>145</td>
</tr>
<tr>
<td>A</td>
<td>D</td>
<td>1001</td>
<td>155</td>
</tr>
<tr>
<td>F</td>
<td>B</td>
<td>1001</td>
<td>140</td>
</tr>
<tr>
<td>E</td>
<td>C</td>
<td>1001</td>
<td>160</td>
</tr>
</tbody>
</table>

If no seller agent with matching advertisement can be found for the following request:

<table>
<thead>
<tr>
<th>Buyer Agent</th>
<th>Stock</th>
<th>Buying Price</th>
<th>Buying Volume</th>
</tr>
</thead>
<tbody>
<tr>
<td>G</td>
<td>1001</td>
<td>144</td>
<td>2000</td>
</tr>
</tbody>
</table>

The recommendation agent prefers to select trading agents from the record database:

1. with the largest number of successful transactions, or
2. that has most recently completed a transaction.

Based on criteria 1, trading agent B is selected because it has the largest number of transactions of the same stock. Based on criteria 2, trading agent C and F are selected. C is selected because it has most recently sold a similar stock and may be interested to purchase more. F is selected since it has most recently bought a similar stock with a lower price. Hence, it may be interested to sell the stock (on this account, F may also be a potential seller). The recommendation agent notifies B, C and F via e-mail that a request for 2000 shares of stock 1001 at a price of 144 is available.

Example 3: Selecting Trading Agents for Recommendation

period of the request, highest buying price, volume required, and the number of expected responses.

request (request_id: id.type;
    buy.agent.id: id.type;
    buy.stock.name: stock.name;
    buy.stock.type: stock.type;
    buy.valid.period: period;
    buy.volume: volume;
    buy.price: double;
    expected.response: integer;)

where

id.type is integer;
stock.name is characters;
stock.type is {stock.name1,...,stock.name_n};
period is integer; /* number of days */
volume is integer; /* number of shares */

Advertisement: Advertisements sent by seller agents to the broker agent contain the following information: stock name available, valid period of the advertisement, lowest selling price, volume available, and the number of expected requests.

advertisement (advertisement.id: id.type;
    sell.agent.id: id.type;
    sell.stock.name: stock.name;
    sell.valid.period: period;
    sell.volume: volume;
    sell.price: double;
    expected.request: integer;)

Connection: The broker agent sends connection information to buyers and sellers that contain the following information: request id or advertisement id, stock name, expiry date of the connection, price, and volume.

connection (connection.id: id.type;
    request.id: (id.type1,...,id.type_n);
    advertisement.id: (id.type1,...,id.type_n);
    buy.agent.id: (id.type1,...,id.type_n);
    sell.agent.id: (id.type1,...,id.type_n);
    conn.stock.name: stock.name;
    conn.expire.date: date;
    conn.volume: (volume1,...,volume_n);
    conn.price: (price1,...,price_n);

Withdrawal: Requests and advertisements posted can be withdrawn by the originators.

withdrawal (request.id: id.type;
    advertisement.id: id.type;
    agent.id: id.type;)

Result: After reviewing the connection results, buyers and sellers will decide if the transaction can be completed or not. Buyers and sellers can notify the broker agent their decisions
Fig. 4. Result of experiment 1 (request connection time).

and give feedback (on a prescribed point scale) on how well the recommendations were.

```plaintext
result (advertisement.id: id.type;
  request.id: id.type;
  successful.connection.id: id.type;
  agent.id: id.type;
  result: result.type;
  satisfaction: feedback.type;)
```

where

```plaintext
result.type is {accept, reject};
feedback.type is {1...10};
```

V. EXPERIMENTATION AND EVALUATION

The performance of the testbed was determined using several parameters, including the rate at which requests and advertisements are generated and the number of buyer and seller agents. Four representative experiments in the domain of securities trading are presented below to demonstrate the approach taken in evaluating the testbed.

A. Experiment One: Average Connection Time

In the first experiment, the performance of the testbed was examined by measuring the average time used in making connections for buyers and sellers. The average connection time and the standard deviation of 50 connections were recorded. Fifty requests and 50 advertisements were generated by five buyer agents and five seller agents respectively. Each buyer (seller) agent generates one request (advertisement) every minute. The connection time for each request and advertisement were measured and shown in Figs. 4 and 5, respectively. The mean and standard deviation of connection time were recorded as follows:

requests connection time:
- mean $\chi = 21.8 \text{ s}$
- standard deviation $\sigma = 5.4 \text{ s}$
- % of requests within $(\chi \pm \sigma) = 41/50 = 82\%$

advertisements connection time:
- mean $\chi = 22.5 \text{ s}$
- standard deviation $\sigma = 5.0 \text{ s}$

Although an average connection time of about 22 s appeared to be relatively large for a sample of 50 connections, it is reminded that this also included the time to wait for requests (advertisements) to arrive in the system under the above experimental environment. Nevertheless, the results showed relatively stable average connection time with relatively small standard deviation.

B. Experiment Two: Users’ Satisfaction

In the second experiment, the relationship between user satisfaction and time was studied. User satisfaction ($U_s$) is a function of $\Delta P$ (the percentage difference of buying and selling price), $\Delta V$ (the percentage difference of buying and selling volume) and $\Delta C$ (the percentage difference of expected number of responses and the number of connections recommended). $U_s$ is mapped to a scale from 0 to 99.

In part 1 of experiment 2, the average buyer satisfaction for 20 requests was measured at 20 s interval from 40 s to 200 s. Advertisements were generated by 9 seller agents. Each seller agent generated one advertisement in 20 s. The connection parameters including price and volume in each request and advertisement were generated randomly. The results were shown in Fig. 6. In part 2 of experiment 2, the average seller satisfaction of 20 advertisements was measured. Requests were generated by nine buyer agents with same rate as in part 1. The result was shown in Fig. 7. Both the results of part 1 and part 2 showed an increase of $U_s$ when the time for making connections increased. The experiments showed that given reasonable amount of time (about 200 s), the system was able to connect buyers to sellers.
with an almost complete user satisfaction. It only took about 80 s before a high user satisfaction (80) can be achieved.

C. Experiment Three: Focus Decisions

In the third experiment, the difference of selling/buying price and volume was studied to determine if the testbed is able to connect buyer and seller agents with closely matched profiles and preferences (criterion 2) of the selection stage in Section II). In this experiment, the percentage difference of selling/buying price $\Delta P$ plus the percentage difference of volume $\Delta V$ is mapped to a scale from 0 to 100 where higher point means more the preferences matched. The matching of profiles is calculated as follows:

$$M_i = \frac{\sum_{1 \leq n \leq N} ((1 - \Delta P) \times 50 + (1 - \Delta V) \times 50)}{N}$$

where

- $M_i$: match of profile for execution $i$;
- $\Delta P = \frac{P_{\text{seller}} - P_{\text{buyer}}}{\max(P_{\text{seller}}, P_{\text{buyer}})}$;
- $\Delta V = \frac{V_{\text{seller}} - V_{\text{buyer}}}{\max(V_{\text{seller}}, V_{\text{buyer}})}$;
- $N$: total number of connections formed.

In each execution, $\Delta P$ and $\Delta V$ of connections formed from 20 requests and 20 advertisements were measured and $M$ was calculated. The connection parameters including price and volume in each request and advertisement were generated randomly. In total, ten executions with the same setup were performed. The results in Fig. 8 showed that the testbed was able to connect buyers and sellers with a degree of more than 80% similar in user preferences and therefore showed that the testbed was able to select buyer and seller agents with closely matched profiles and preferences in forming connections. It seemed that the testbed could generally connect buyers and sellers that have similar interests.

D. Experiment Four: Balanced Loading

The fourth experiment studied the property of providing every trading agent with uniform opportunity of being connected to other trading agents (criterion 3) of selection stage in Section II). In this experiment, the number of connections for each seller/buyer were measured with different number of advertisements/requests.

In the first part of this experiment, the number of connections of nine buyers was measured with 18, 36, 54 and 72 advertisements while in the second part, the number of connections of nine sellers with 18, 36, 54 and 72 requests was measured. The results shown in Figs. 9 and 10 indicated that the testbed connected different number of requests/advertisements among nine sellers/buyers quite evenly. It seems that every trading agent in the testbed has quite uniform opportunity of being connected to other trading partners. This property seems to be similar to the balanced loading property in the contract net protocol.

VI. RELATED WORK

The information brokering protocol in this paper bears some resemblance to the contract net protocol [2] in the way that both protocols are used to solve the connection problem. While the contract net protocol provides a mechanism for structuring high-level interactions between nodes for co-operative task executions, the information brokering protocol defines high-level interactions between buyers, sellers, and the broker in the brokering process. Desirable properties in the contract net protocol such as balanced loading and focus decision are also present in the information brokering protocol.
Some of the extant brokering systems that are similar to this research include Andersen’s Bargain Finder [4], FAST Broker from University of Southern California [5], WARREN multi-agent portfolio management system from Carnegie Mellon University [6], Firefly [7], and Amalthaea [8]. Like the broker agent in this research, the Bargain Finder provides connection service to on-line stores for the lowest price of compact disc available. Unlike the broker agent, the Bargain Finder make comparison solely on price while the broker agent consider multiple factors when making connection. FAST is a purchasing agent with access to a large number of distributors and manufacturers. Like the brokering system in this paper, FAST does not support negotiation between buyers and sellers. The WARREN from CMU is a multi-agent system for the management of financial portfolios. It experiments with several design constraints and related models of middle agents including brokering and matchmaking. Like the broker agent, the WARREN keeps track of the current state of the situation, environment, and user information needs. Unlike the recommendation agent and record agent, the WARREN does not ‘push’ information to users with reference to the transaction history and currently available products and requests in the system. Firefly is a recommender system that supports only the product brokering stage while the brokering system in this research supports both product and merchant brokering, identified in [9] as two of the essential stages of the consumer’s buying behavior model. Similar to the brokering system in this research, Amalthaea is a multi-agent system and makes recommendations to users. To evaluate users’ satisfaction, Amalthaea accepts users’ feedback directly through a user interface while the broker agent determines users’ satisfaction by computing the differences of the connection parameters (such as price and volume) of requests and advertisements.

The evaluation approach used by the broker agent seems to provide a more objective comparison than the credit feedback mechanism in Amalthaea.

VII. CONCLUSION AND FUTURE WORK

The agent-based brokering system exposed in this paper partially solves the connection problem in the product brokering and the merchant brokering stages of e-commerce. In solving the connection problem using the contract net protocol, two of the desirable properties are focus decision and balanced loading [2]. As demonstrated in experiment 3 and 4, these two properties are also present in the broker protocol in this research. In this paper, the design and engineering of the brokering system that consists of user interfaces, the broker agent, the record agent and the recommendation agent was briefly introduced. The structure of the broker agent and the approach of making connection based on multiple factors was described. The stages of the broker protocol that define the brokering process were outlined. Finally, favorable results from several experiments show that the system has relatively good performance. As part of a future expansion, designs on message passing to facilitate communication among agents in the testbed are currently being considered. In addition, instead of accepting user-supplied coefficients for computing the utility of a connection, this research hopes to enhance the broker agent with the capability of learning these coefficients. Furthermore, as a middle agent, the support of negotiation stage of e-commerce is being devised.

APPENDIX

EXPERIMENT ON THE VALUE OF CUTOFF POINT $\beta$

In this experiment, the buyer (seller) satisfaction against filtering cutoff point $\beta$ were measured in the hope that an optimizing value of $\beta$ can be found for the testbed. As illustrated in Section II, the value of $\beta$ is expected to be not too small or not too large. While a small value of $\beta$ may result in fast connection time and more connections but with lower utilities, a large value of $\beta$ may decrease the number of connections but with higher utilities.

In part $1$ of this experiment, the average buyer satisfaction of $20$ requests was measured while $\beta$ was varied from one to nine. Requests were generated by five buyer agents at an interval of $20$ s. Advertisements were generated by five seller agents at an interval of $10$ s. The connection parameters including price and volume in each request and advertisement were generated randomly. The result was shown in Fig. 11. In part $2$ of this experiment, the average seller satisfaction of $20$ advertisements was measured while $\beta$ was varied from one to nine. Advertisements were generated by five seller agents at an interval of $20$ s. Requests were generated by five buyer agents at an interval of $10$ s. The result was shown in Fig. 12. The results of part $1$ (part $2$) showed a peak of buyer (seller) satisfaction when $\beta$ had a value of five (six). The results indicated that a medium value of $\beta$ would be favorable to the testbed as what was expected.

REFERENCES


Kwang Mong Sim received the B.Sc. (Hon.) degree (summa cum laude) in computer science from the University of Ottawa, Ottawa, ON, Canada, and the M.Sc. and Ph.D. degrees in computer science from the University of Calgary, AB, Canada.

He is a Faculty Member with the Department of Information Engineering, Chinese University of Hong Kong. His research interests include agent-based Internet commerce, Internet agent technology, and intelligent agent technology. He led a group of ten M.Sc. students in engineering agent-based e-commerce systems such as learning and negotiation agents, brokering agent, information filtering softbot, ontology agent, and security agent. He was a program committee member for two artificial intelligence in e-commerce workshops in Sydney, Australia, (1999) and Melbourne, Australia, (2000), and a session chair for the Intelligent Agent Technology Conference (1999). He has work experience in five countries spanning the continents of North America, Asia and Europe.

Dr. Sim is listed in the Marquis’ Who’s Who in the World Millenium Edition and will be listed in the 18th (2001) edition.

Raymond Chan received the M.Sc. degree in information technology with distinction from the Hong Kong Polytechnic University, Hong Kong, and the B.Sc. (Hon.) degree in physics from the Chinese University of Hong Kong.

He entered the field of software development in 1988, and has worked as a system analyst and application programmer in both private and public sections. His research interests are agent-based e-commerce and information brokering.