Electronic Commerce Research and Applications (ECRA), Elsevier Science, Vol. 3, No. 3, 2004, pp 294-310

PumaMart: A Parallel and Autonomous Agents based Internet Marketplaces

Yan Wang
Department of Computing
Macquarie University
Sydney, NSW 2109
Australia
yanwang@ics.mq.edu.au

Kian-Lee Tan
Department of Computer Science
National University of Singapore
3 Science Drive 2, Singapore 117543
tankl@comp.nus.edu.sg

Jian Ren
Department of Computer Science
National University of Singapore
3 Science Drive 2, Singapore 117543
renjian@comp.nus.edu.sg

Abstract: Software agents are flexible, autonomous, and dynamic computational entities. For B2C e-commerce applications, the wide variety of choices to the consumers has also introduced the problem of information overloading. Meanwhile, there are so many e-shops and products for the consumers that it has become too time-consuming to find the best deal.

In this paper, we present *PumaMart*, a **P**arallel and a**u**tono**m**ous **a**gents based Internet **Mar**ke**t**place, which deploys several novel models to facilitate autonomous and automatic online buying and selling by software agents (stationary and mobile) while providing fast response to consumers. These techniques include a 2-phase evaluation model, a parallel dispatch model and an auction-like negotiation model. Both evaluation model and negotiation model are based on the fuzzy evaluation criterion with clustering based grading function. What a consumer needs to do is to submit requests including the information for the desired products, selection preferences, through a web page in a Java-enabled browser. These information will be sent to the master agent at the server of the Agent Service Provider (ASP), which will employ its worker agents for subsequent shop searching/filtering, offer gathering/evaluating, negotiating, even booking and payment.

Keywords: B2C e-commerce; Autonomous agent; 2-phase evaluation model; Fuzzy evaluation criterion; Negotiation model

1 Introduction

The advances of web technologies such as the Internet, HTML, Java and XML have greatly pushed the development of Electronic Commerce (EC). Today, many online shops (e-shops) publish their product catalogue on the Internet, offering a wide variety of products. More and more consumers are turning to the Internet for such information as well as to purchase products online.

However, it is quite difficult for consumers to find the optimal offers in today's electronic commerce for two reasons. First, consumers' decisions are based on multiple objectives – while he may want to find the cheapest item, he would also want to ensure that the e-shop is reliable or the item can be shipped within a certain period of time. Very often, these objectives may conflict. Second, there are so many online shops selling similar products, and the offers (price, delivery, warranty, etc) can be significantly different. To get the most attractive deal, the consumer has to browse and compare the offers of as many sites as possible.

The recent development in the technology of software agent, stationary or mobile, offers attractive solutions in the field of EC. Software agents are computational entities that are flexible, autonomous, and dynamic [1]. Mobile agent approach aims to extend the traditional client/server model to a three-tire model, namely client/agent/server [1]. When being dispatched to a remote server with an encapsulated task, a mobile agent can execute autonomously and benefit from executing locally. The results can be sent back through a message. During this period, it does not require constant connection as traditional RPC (Remote Procedure Call) so it is applicable to devices with limited bandwidth and computing resources, and long-tern transactions without constant interaction. The server in the three-tier model can be partially or purely a data server with the complexity of various computation functionalities partitioned to mobile agents from different parties. Otherwise, the servers should enable functionalities to respond the all kinds of requests, standard or specific, from different parties. The mobile agent approach is also suitable for deploying parallel processes over distributed sites on the Internet [2]. The tasks can be decomposed and encapsulated to multiple mobile agents. Every mobile agent can run independently to accomplish its task. And all the mobile agents can run in parallel on distributed hosts so that the whole tasks can be completed in a short time.

Based on these features, as pointed out by [3], future e-commerce models will enhance current models by using software agents. In our real life, people can turn to a few agents or agencies for buying something such as an air ticket, renting or buying a house. They can choose a satisfactory one from multiple provided plans. Similarly, the introduction of autonomous agents acting on behalf of end-consumers could reduce the effort required from users to conduct EC transactions by automating a variety of activities and add values to three primary electronic commerce dimensions: fresh information gathering and retrieval, information filtering, as well as dynamic and flexible execution of transactions [4, 5, 6, 7]. Additionally, the use of mobile agents can automate evaluation and filtering with different criteria from consumers by different types of mobile agents from different parties with only required data returned (after filtering). In this scenario, the complexity with management servers and e-shops are reduced with respect to responding to clients' requests.

In this paper, we present *PumaMart*, a **P**arallel and a**u**tono**m**ous **a**gents based Internet **Mar**ke**t**place that facilitates parallel and autonomous processing. Based on our 2-phase fuzzy evaluation model, parallel dispatch model and one-to-multiple auction-like negotiation model, this system aims to enable autonomous and automatic buying and selling by software agents in agent-mediated B2C Internet marketplaces while providing fast response to consumers. The 2-phase evaluation model not only evaluates shops and offers, but also controls the scale of dispatched mobile agents. The negotiation model is based on the same fuzzy evaluation criteria as the 2-phase evaluation model. All that a registered consumer needs to do is to input information for desired products, selection preferences, through a web page in a Java-enabled browser. These information will be sent to the master agent at the server of Agent Service Provider (ASP), which will employ its worker agents for subsequent shop searching/filtering, offer gathering/evaluating, negotiating and payment. All these processes are transparent to end-users. The source code of PumaMart can be downloaded at [8].

The rest of this paper is organized as follows: Section 2 presents some related work. The infrastructure of our Internet marketplace and its process flow are briefly presented in Section 3. Section 4 describes the 2-phase evaluation model. Section 5 presents the parallel dispatch model and Section 6 presents the negotiation model. In Section 7, the implementation and user interfaces of PumaMart are presented. Section 8 concludes our work.

2 Related work

There has been an increasing amount of research activities to exploit software agents, especially mobile agents, to support electronic markets or enterprises.

A fundamental description for setting up an electronic market with mobile agents is from [9], which proposed an architecture for online market. The system consists of conductors and members. The conductor manages the market and members participate in electronic commerce activities. Members are providers, shops and consumers. The conductor provides the framework of the market and manages the setup of members, product ontology and member information. [9] introduces some internal activities that these agents should do. But this work addresses only an individual market without any focus on an electronic market community on distributed sites and corresponding implementation.

Lange and Oshima also briefly introduced a mobile agent based marketplace architecture in [10] and showed that the Aglet Software Developing Kit (ASDK) system [11, 12] can be used to support an electronic marketplace and the meeting pattern and communication mechanism of Aglets, which are mobile Java objects, can be adopted to meet the requirements for representing the behaviors of mobile agents. In his framework the consumer agent visits marketplaces one by one to perform the shopping request and negotiation activities. Another related work on MAgNet online trading system [13] aims at making mobile agents serve both buyers and suppliers.

The above-mentioned works, addressing agent-based marketplace on a platform with central management servers, benefit much from the deployment of mobile agents, such as good mobility, high autonomy as well as the role simulation that present the realistic simulation to the real commercial activities. But they simply put mobile agents in a serial working pattern and their global control architecture and in-depth processes are not presented. As discussed in [14], the performance of the serial agent model could not be satisfactory for a large-scale marketplace involving many e-shops.

Some work has been done on evaluation models and negotiation models incorporated in agent-mediated e-commerce systems, such as [5, 15-22]. Kwang's work [15] explores the problem of matching buyer agents and seller agents using multiple criteria based on pre-specified user profiles. The process of matching and connecting buyers and sellers is based on a utility evaluation function and a filtering algorithm.

Kowalczyk and Faratin's work [16, 17] explores the issues in applying fuzzy concept in the agent-aided e-commerce system. [16] presents a customable **F**uzzy **e**-**N**egotiation **A**gent**s** (FeNAs) system for limited common knowledge and imprecise preference. The FeNAs system uses the principles of utility theory and fuzzy constraint-based reasoning to find a consensus that maximize the agent's utility. The work in [17] presents an algorithm using the notion of fuzzy similarity to enable agents to make trade-off in negotiation.

Zeng and Sycara presented *Bazaar* in [18], an experimental system, for updating negotiation offers between two intelligent agents during bilateral negotiations and studied the benefit of learning in negotiation. *Bazaar* supports single-attribute negotiation on price. So when applied to multi-attribute scenarios, an evaluation model is essential.

MIT *Kasbah* marketplace [19, 20, 21] is an e-commerce negotiation system. This marketplace sets up an environment where a user can create an autonomous agent to buy or sell a product, and even negotiate the product price on his/her behalf. In this system, mobile agents are used.

MIT *Tête-à-Tête* [5] provides a negotiation approach to retail sales. Unlike most other online negotiation systems that competitively negotiate over price, Tête-à-Tête agents co-operatively negotiate across multiple terms of a transaction. The shopping agents follow an argumentative style of negotiation with sales agents and use the evaluation constraints captured during the brokering stages as dimensions of a multi-attribute utility. This utility is used to rank merchant offers based on consumer's preference.

These models work well in their respective environments. In our context, some issues should be addressed.

1. Only the attributes correlated to the products are considered in both evaluation and negotiation phases in these works. Some models only consider the price of a product. But in real-life, the consumers consider not only the offer, but also the commercial reputation of shops. In an electronic environment, the security ranking of e-shops is a very important consideration w.r.t. the reputation.

2. In the evaluation models mentioned above, the grading functions are fully based on the pre-defined grading standard (e.g., specifying the scope for a desirable price as [$1000, $1200] or giving more detailed price categories. But it has to be changed from time to time otherwise it may be rigid and inflexible to reflect the dynamic changes of markets. If the standard is from consumers, it requires them to have good knowledge on markets and product prices.

3. The negotiation process in most works is bilateral that cannot benefit from the competition among the e-shops. Moreover, all these approaches require the consumer to specify his public initial offer, which is the beginning offer of the negotiation process, and private border offer, which is a maximum limit that must be respected in reaching a deal. But in real world, a consumer has very limited information for him/her to be able to define the border offer precisely.

3 The infrastructure for Internet marketplaces

3.1 System components

In [14] we presented the agent-based infrastructure for Internet Marketplaces, on which PumaMart is based. Here we briefly review it.

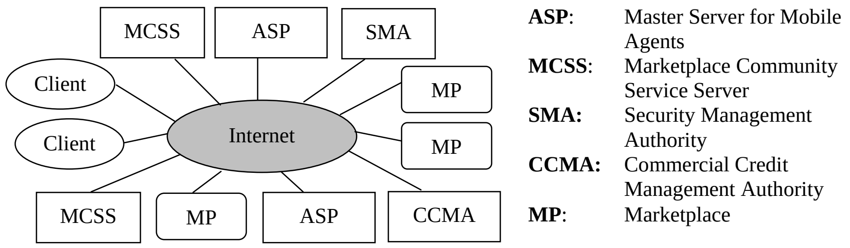


Figure 1 Overview of the Marketplace

In the agent-based framework, there exists a set of marketplaces (see Figure 1). They are connected to the Internet. The Agent Service Provider (ASP) is an execution environment for software agents, including mobile agents. A consumer-agent can be created at ASP at the client's request. Such a consumer-agent can reside in the server of ASP, act as a master agent and dispatch its worker agents to related marketplaces (MP) to gather and evaluate offers and negotiate with e-shops. Meanwhile, a set of ASPs are set up and distributed globally.

In the proposed architecture, there is a set of Marketplace Community Service Servers (MCSS). A MCSS is responsible for maintaining the directory information of MPs and e-shops in the MPs. The Security Management Authority (SMA) assesses the security levels of authorized e-shops based on attack reports and periodically reports the updated security rankings of MPs and e-shops to the MCSS. CCMA is the authority that assesses and manages the commercial credit of all e-shops. Successful transactions will help to upgrade the commercial credit while merchant cheating will downgrade the commercial credit of an e-shop. In this paper we will not address the algorithm for calculating commercial credit. Detailed and complex reputation mechanisms can be found in [23, 24, 25]. A client here is a registered user of the ASP. A MP is the Internet marketplace consisting of a set of e-shops that run simultaneously on different servers.

3.2 Procedures of setting up a MP and an e-shop

When setting up a MP, the MPM (MP Manager) should register the MP to MCSS by sending the following:

(1) MP’s name, domain name and IP address

(2) MP’s certificate including its public key obtained from SMA

(3) identification of the MPM

(4) IP addresses, certificates, directories and goods catalogue of all e-shops in the MP

(5) identifications of corresponding shop-agents

(6) current time

When an e-shop is set up in a MP, the shop-agent should register to the MPM by sending the following:

(1) e-shop's name and IP address

(2) e-shop's certificate including its public key

(3) e-shop's goods catalogue

(4) identification of the shop-agent

(5) current time

Information (1) and (2) are put in the MPDS (MP Directory Server) by MPM. If the catalogue of an e-shop is changed, the shop-agent will notify the MPM and MPM will report these changes to MCSS. The MPM will also report to MCSS when any e-shop withdraws or the whole MP withdraws.

3.3 Process flow

Based on our framework, the process enabling buying and selling can be described as follows:

1. **Input Request**: For a client, he/she chooses an ASP to input the information of a product such as the name, model and type for the product. He/she is also required to input consumer preference. The client can either choose some predefined preference standards or customize the weight of each item.

2. **First-phase Evaluation (Shop-Evaluation)**: With the request of a client, a consumer-agent is created at the server of the ASP, who will act as a master agent and dispatch a mobile agent to MCSS to search, evaluate and filter e-shops. With the information, relevant e-shops selling the product are evaluated over their reputation (i.e., commercial credit and security ranking).

3. **Parallel Dispatch**: After the shop-evaluation phase, the consumer-agent will dispatch a pool of worker agents to qualified e-shops in parallel to gather offers. In the best case when mobile agents are dispatched in binary, the time complexity for dispatching n mobile agents is $O(log_2 n)$. More details about parallel dispatch models can be found in Section 5.

4. **Second-phase Evaluation (Offer-Evaluation)**: With the returned offers, the second-phase evaluation is conducted on both offers' attributes and e-shops' reputation. More details on the two-phase evaluation are described in Section 4.

5. **Negotiation**: According to the client's selection preference and the results of the second-phase evaluation, a few e-shops will be selected for negotiation over offers' attributes by the negotiation-agent. The negotiation process can be performed partially in different search regions (see Section 5). More details about the negotiation model are presented in Section 6.

6. **Book and Payment**: With the successful result of negotiation, one e-shop will be selected to book the product and make an online secure payment [26]. We will not address the secure payment issue in this paper. If the consumer is not satisfied with the result after negotiation, he could specify new preference and initiate a new process from evaluation to negotiation. This can be an interactive process before the consumer makes a booking and payment.

4 Two-phase evaluation

As there are hundreds or thousands of shops on the web selling the same kind of product, it is unnecessary and impossible for a consumer or even an agent to browse/visit all of them. Moreover, the number of offers may be very much more than the number of shops since a shop may provide multiple offers. So a consumer must find a way to evaluate as many shops/offers as possible to identify the promising ones for further transaction.

In most real-world situations, preferences and constraints may be imprecisely defined, such as low price, high quality, short delivery/shipment time, and so on. As pointed in [27], a rational approach toward decision-making should take into account subjectivity and knowledge uncertainty. Hence the autonomous agents should have the capability to consider imprecise information. Fuzzy-set theory is a good means for modeling uncertainty arising from mental phenomena [27, 28]. In [29], the author used fuzzy intelligent agents to help Web advertisers make decisions regarding Web site visitors. Each agent has a small knowledge set to judge if a Web visitor is “good” or not according to his age and salary. In [30], only two attributes “age” and “salary” are considered.

In our evaluation model, we applied the principle of utility theory and the fuzzy-set technologies on multi-attribute that have the capability to naturally represent human conceptualization and process imprecise information. These attributes come from both shops and offers.

4.1 Fuzzy evaluation criterion

Following the principle of fuzzy-set theory [27], we define the fuzzy evaluation criterion as follows:

Assume a value x consists of a number of attributes:

$$x=\{x_1, x_2, \ldots, x_n\}$$

1. For each attribute x_i, calculate its firing level F_i as:

   $$F_i=K_i(x_i)$$

   where K_i is the grading function for attribute x_i.

2. Calculate the scores T_i of each attribute as:

   $$T_i=S_i(F_i)$$

   where S_i is the score function that maps the attribute score in up to 5 intervals, i.e. [1, 5].

3. Calculate the overall utility $U(x)$ of the value x as:

   $$U(x)=\sum w_i T_i$$

   where the relative importance assigned to each attribute is modeled as a weight w_i. We assume the weights are normalized, i.e. $\sum w_i=1$. All weights are given by a consumer via predefined preference standards.

4. Calculate the overall firing level $O(x)$ of value x as:

   $$O(x)=K(U(x))$$

   where K is the grading function.

To summarize, the overall firing level $O(x)$ for a value $x=\{x_1, x_2, \ldots, x_n\}$ in the multi-dimensional space is defined as:

$$O(x)=K(\sum w_i * S_i(K_i(x_i))) \quad (1)$$

The grading function K converts the attribute values into the corresponding firing levels. These levels can be represented as linguistic values, such as “very good”, “good”, “moderate”, “poor” or “very poor” for 5 categories.

In our work, we use the k-means clustering algorithm [30] as the grading function, which is simple and cost-effective. In addition, even if the resultant clusters are not very accurate [31], the first-phase evaluation will leave “very good”, “good” and even “moderate” e-shops to the second phase. This ensures that most good e-shops are included and will be visited. And offers may be further improved through negotiation. In real markets, the attributes, particularly the prices, vary from time to time. A previously “very good” price may be not attractive now (e.g., currently it becomes a “good” one or even a “poor” one). Clustering algorithm instead of the one to grade attribute values into pre-defined levels has the capability to reflect the dynamic changes of markets.

4.2 A two-phase evaluation model

The evaluation is conducted in two phases, namely, shop-evaluation and offer-evaluation, according to the selection preference chosen by the consumer.

4.2.1 First phase evaluation: shop-evaluation

The shop-evaluation helps locate good and convenient shops to the consumers in online trading. This phase takes into account attributes including commercial credit, security rank, and delivery/shipment service.

The commercial credit of an e-shop is set based on the number of its previous successful transactions and e-shops with fast delivery/shipment will get high value for delivery/shipment service. Those attributes for the first-phase evaluation are stored in the MCSS with the product types for all e-shops. One worker agent is dispatched to MCSS to accomplish the evaluation task and send the results back to the master consumer-agent. Only after shop-evaluation will worker agents be dispatched to those qualified e-shops to search for offers in parallel.

For shop-evaluation, the overall firing level for $x=\{x_1, x_2, \ldots, x_n\}$ is calculated as follows:

$$O(x)=K(\sum w_i * S_i(K_i(x_i))) \qquad x=\{x_i\}=\{C, S, D\} \tag{2}$$

where x_i can be one of the following:

- *C* denotes the commercial credit of the e-shop.
- *S* denotes the security rank of the e-shop. Higher security rank means higher security level for the shop.
- *D* denotes the time for delivering the product to consumers.

After evaluation, by default the shops that are rated as “moderate” or above are returned to the consumer-agent.

4.2.2 Second phase evaluation: offer-evaluation

Based on the result of shop-evaluation, the master consumer-agent will dispatch a pool of worker agents in parallel to qualified e-shops. When all worker consumer-agents send back offer details of the product, such as the price, stock status, warranty service, the offer-evaluation will be conducted.

In this phase, the overall firing level for $x=\{x_1, x_2, \ldots, x_n\}$ is calculated as follows:

$$O(x)=K(\sum w_i * S_i(K_i(x_i))) \qquad x=\{x_i\}=\{P, D, W, C, S\} \tag{3}$$

where x_i can be one of the following:

- *P* denotes the price of the product
- *D* denotes the time for delivering the product to the consumer
- *W* denotes the warranty time
- *C* denotes the commercial credit of the shop
- *S* denotes the security rank of the e-shop

After offer-evaluation, “good” and “very good” offers and the corresponding e-shops where these offers are from are selected for negotiation for further benefit.

4.2.3 Fuzzy preference standards

In our application, we design fuzzy preference standards and sub-standards with crisp weights displayed in the UI (User Interface) of PumaMart. A consumer can choose the standards he desires. While providing convenience for consumers, these predefined standards are not able to cover all consumers’ preferences. In our system, the consumer can make minor adjustment if necessary after choosing predefined standards or totally customize the weight of each attribute. So it combines the benefits of the two methods mentioned above and the preference selection becomes simple and intuitive.

Considering the reputation of shops, the preference standards are classified into two groups, i.e., *cautious consumer group* and *incautious consumer group*. In *cautious consumer group*, the consumer pays much attention to the reputation of the e-shops. The commercial credit and security level get higher weights in the evaluation process. In contrast, in the *incautious consumer group*, the consumer does not care too much about the reputation of the e-shops. The commercial credit and security rank get low weights in the evaluation process. There are four sub-groups in each group reflecting the properties of offers.

1. *Cautious consumer group*:

- *Price Priority Sub-Group*. In this sub-group, the price gets higher weight than any other attributes. The offer with lower price may get higher score. There are three choices in this sub-group:

  1) Extreme price priority 2) *Moderate price priority* 3) *Relative price priority*

  In *extreme price priority*, price's weight is evidently higher than all other attributes. In *relative price priority*, the weight of price is relatively a bit higher than other attributes. While in *moderate price priority*, the weight of price lies between the above two choices.

- *Delivery Priority Sub-Group*. The offer with shorter delivery/shipment time gets higher score. There are three choices in this Sub-group:

  1) Extreme delivery priority 2) *Moderate delivery priority 3) Relative delivery priority*

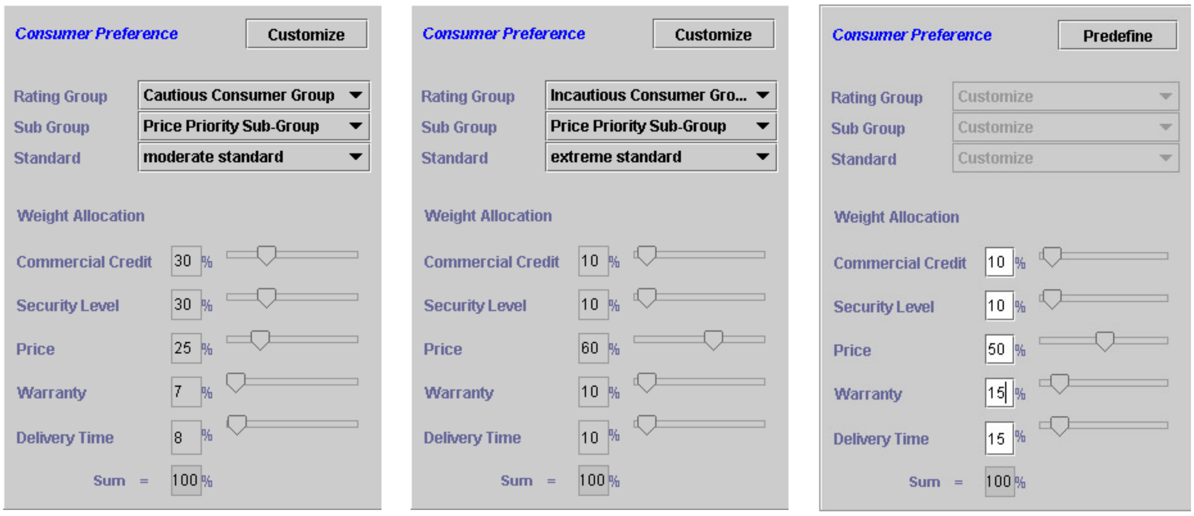
- *Warranty Priority Sub-Group*. This sub-group is suitable for the customers who prefer longer warranty time. There are also three choices in this sub-group:

  1) Extreme warranty priority 2) Moderate warranty priority 3) Relative warranty priority

- *Balance Priority Sub-Group*. In this sub-group, the weights of different attributes are the same.

2. ***Incautious consumer group***: There are the same four sub-groups in this group.

In Figure 2, we illustrated several cases for preference standards.



(a) Cautious Consumer Group/Moderate Price Priority Sub-group

(b) Incautious Consumer Group/Extreme Price Priority Sub-group

(c) Customize Weights

Figure 2 Preference Standards

5 Parallel dispatch of mobile agents

In many mobile agent-based e-commerce applications, one mobile agent is employed to visit a set of e-shops. In terms of performance, it is not a good choice especially when the number of e-shops is large. In this section, we introduce the proposed parallel dispatch model. As shown in Figure 3, master agent A_0 is responsible for dispatching PWAs (Primary Worker Agent) and distributing tasks to them. A PWA is the special WA (Worker Agent) that should dispatch other mobile agents. A WA is only responsible for locally asking offers, accessing data and returning the result. A PWA can also have a task of performing data access depending on the application. Suppose A_0 has to dispatch 16 agents to different hosts. Now, they can be divided into 2 groups led by two PWAs, say A_1 and A_9. When agents A_1 and A_9 are dispatched, each of them has 8 members including itself. For A_1, it will dispatch A_5 and distribute 4 members to it. Then A_1 will transit to the same layer (i.e., L_2) as A_5, which is called a virtual dispatch. But now A_1 has 4 members only. Following the same process, A_1 will dispatch A_3 and A_2. At last, after all dispatch tasks have been completed, A_1 will become a WA and can start its data-accessing task if it has. As a whole, since all PWAs are dispatched to different hosts, the dispatch process can be preformed in parallel. When there are $n=2^h$ mobile agents and Δt is the average time for dispatching a mobile agent, $(h+1)\Delta t$ will be the time for dispatching n mobile agents in the binary way. So, the dispatch time complexity will be $O(log_2 n)$. Thus, the proposed model is efficient. More discussions on parallel dispatch models can be found in our previous work in [14].

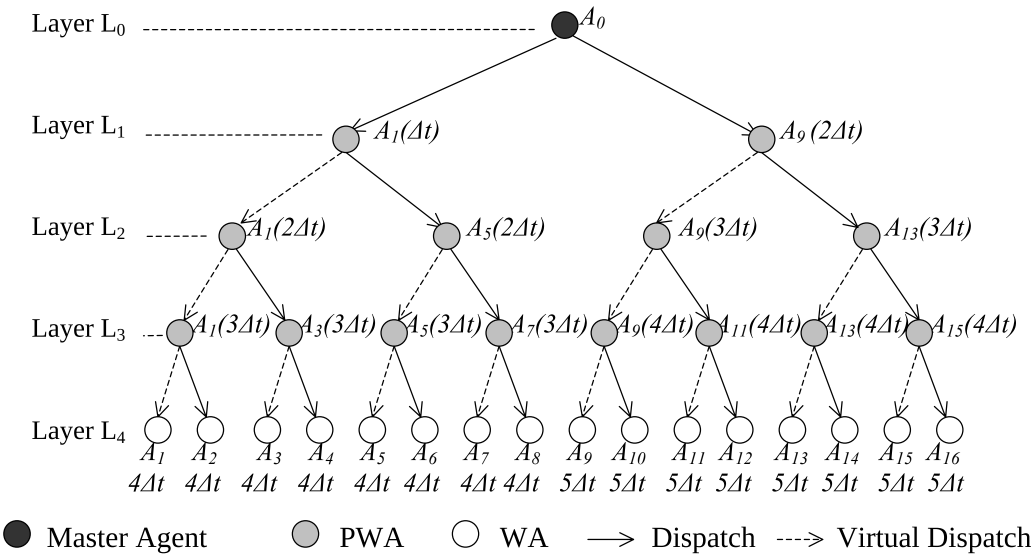


Figure 3 Dispatch Tree with 16 Mobile Agents

Additionally, different from the models presented in [14], if several e-shops that should be visited are within the same marketplace (intranet), one agent can be dispatched there to make a partially serial migration. Hence, the total number of agents will be fewer than that of visited e-shops. However, parallel process is necessary for dispatching agents to different marketplaces.

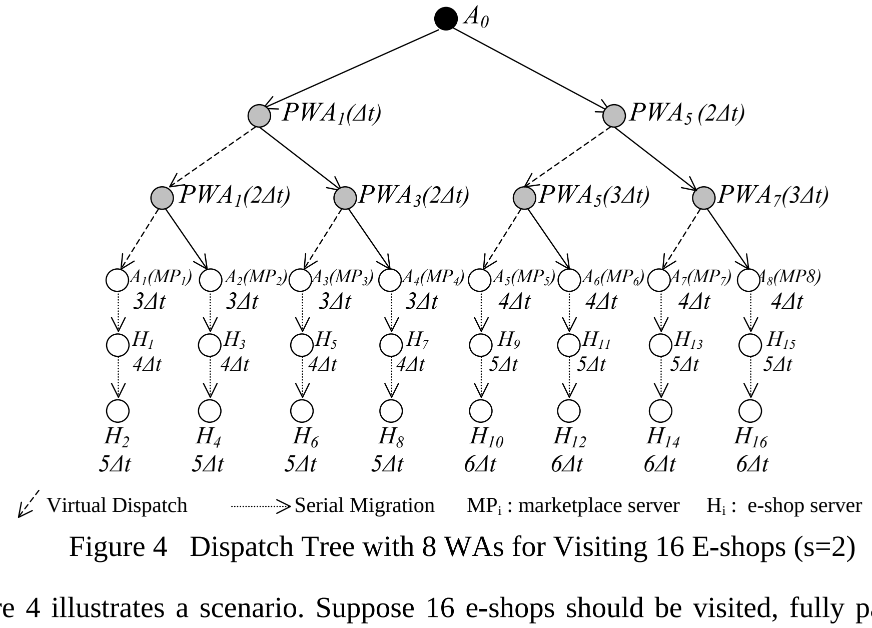


Figure 4 Dispatch Tree with 8 WAs for Visiting 16 E-shops (s=2)

Figure 4 illustrates a scenario. Suppose 16 e-shops should be visited, fully parallel dispatch model (see Figure 3) needs 16 mobile agents. If every 2 e-shops are within the same marketplace, 8 mobile agents can be employed and each one visits 2 e-shops serially (i.e., threshold $s=2$). If every 4 e-shops are within the same marketplace, only 4 mobile agents are needed (i.e., threshold $s=4$). That means 75% of the amount of mobile agents is reduced.

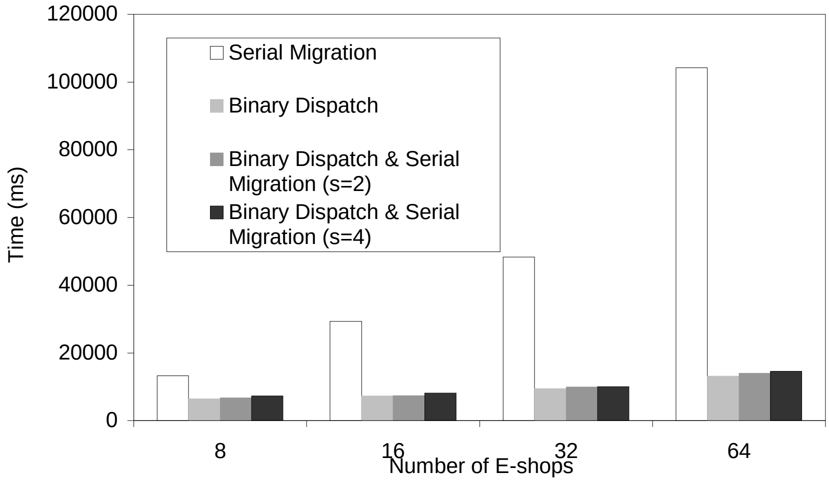


Figure 5 Comparison of Dispatch/Migration Time

Figure 5 shows the performance comparison of different models. The experiment is based on a cluster of PCs with the same configuration of Pentium 200MHz CPU and 64 Mbytes RAM ignoring the time spent on local data access. So the performance difference comes solely from the differences of the models. The result illustrates that parallel dispatch models can significantly outperform serial migration model even when the number of e-shops is 8. The time of serial model increases significantly with the number of e-shops. However the time of parallel dispatch model increases slowly while the performances of 3 parallel dispatch models are close to each other. When having 64 e-shops, parallel models obtain 86% saving percentage in average in comparison with the serial model that is widely used in agent-based e-commerce environments.

In Internet, since the network delay is longer and the time for local data accessing will be counted, the combined model will become inferior to purely parallel dispatch model. But if the number of serially visited e-shops is in a rational scale, the combined model will be more realistic since the number of dispatched mobile agents can be evidently reduced.

But a significant benefit from the combined model is that a dispatched agent can be assigned the function to complete a partial negotiation with the e-shops within one marketplace or several marketplaces (in corporation with other agents). Thus each set of results returned form such an agent includes optimized offers only. The optimized offer sets from different areas can be combined so that the final-stage negotiation will occur among the corresponding e-shops with optimized offers. For example, to search offers, a set of agents is dispatched globally. One of them is dispatched to e-shops in Singapore. Another is dispatched to Malaysia. An initial negotiation can be conducted among these e-shops. The optimized offers can be combined with those from other regions (e.g., Europe and North America) so that the final-stage negotiation will be performed among the selected e-shops.

6 One-to-multiple negotiation model based on fuzzy evaluation algorithm

Generally speaking, auction is a very efficient method of allocating goods/services in dynamic situations to the entities that value them highly [32]. Among various auctions, four common types of single-sided auction are English auction (first-price descending), FPSB (First-price sealed-bid), Vickery auction (second-price sealed-bid) and Dutch auction (first-price descending). The most common type of double-sided auction is the continuous double auction (CDA), which allows buyers and sellers to continuously update their bids/asks at any time in the trading period. It is widely used in stock markets where multi-buyers and multi-sellers trade.

The auction process in PumaMart is the type of bilateral auction, where the consumer agent is the auctioneer and shop agents are bidder. But it is a haggling process where both auctioneer and bidders can propose counter offers and proposed offers change "forwardly" or "backwardly" instead of change in a continuous ascending or descending order as in standard Dutch auction or English auction. But as a whole, the offers change towards an agreement point to make a deal. In our model, a negotiation-agent is responsible for negotiating with multiple e-shops simultaneously proposing counter offers based on e-shops' responses and proposals as well as the consumer's preference. E-shops will compete with each other in the negotiation process to bid for the best offer in order to make the deal.

The steps of the negotiation protocol are as the follows:

Step1: The negotiation-agent proposes the initial offer based on the results of the two-phase evaluation. The offer generation algorithm is presented in Section 6.2.

Step2: Repeat the negotiation round until at least one of the termination conditions is satisfied. The termination conditions are described in Section 6.1.

A. The negotiation-agent sends the newly proposed offer $O=\{x_1, x_2, \ldots, x_n\}$ to selected e-shops, where x_i is the attribute of the offer.

B. Each e-shop evaluates the offer O based on its own strategy and makes the decision to respond to the offer. The strategy of sellers is not addressed in this paper. Readers can refer to [33, 34]. But the response from a seller can be simply one of the follows:

- Accept the offer
- Reject the offer
- Propose a new counter offer

C. The negotiation-agent collects all the responses from the e-shops and checks if the termination conditions are satisfied.

If "Yes", negotiation stops. Go to step 3.

If "No", the negotiation-agent will propose a new counter offer O and repeat step 2.

Step3: Collect the negotiation results and process offer-evaluation. The offer-evaluation algorithm is presented in Section 4.2.2.

6.1 Termination conditions

In the negotiation process, if one of termination conditions is satisfied, the negotiation will stop. The termination conditions are described as follows:

1. The negotiation round limit reaches or negotiation time is up. When the limit reaches as specified in the input argument, the negotiation process has to terminate.
2. All the e-shops reach their border offers and terminate their negotiation.

After the negotiation terminates, the last offer that the e-shop accepted or the e-shop proposed is saved as the best offer of this e-shop for further processing. If the e-shop doesn't accept any offer or propose any counter offer, the original offer that the e-shop proposed before the negotiation is saved as its best offer.

6.2 Offer generation algorithm

Assume an offer O consists of a number of attributes:

$$O=\{x_1, x_2, \ldots, x_n\}$$

The consumer-agent can select the best value of each attribute in the shops' offers to compose the *initial offer* O^0 as follows:

$$O^0=\{x_{1best}, x_{2best}, \ldots, x_{nbest}\} \tag{4}$$

where x_{ibest} means the best value of attribute x_i in all shop offers.

Based on the initial offer, the consumer-agent can also use adjustment-factors to generate *counter offer* O^1 as:

$$O^1=\{(1+\beta_1)*x_{1best}, (1+\beta_2)*x_{2best}, \ldots(1+\beta_n)*x_{nbest}\} \tag{5}$$

where $\beta=\{\beta_1, \beta_2, \ldots \beta_n\}$ is the set of *adjustment-factors*.

The *adjustment-factor* determines the "step" of each attribute changed in the negotiation process. If the step is too small, the consumer cannot get better negotiation results within the round limit. On the other hand, if the step is too big, the consumer may miss the chance to make its offer accepted. The master consumer-agent adjusts the adjustment-factors automatically based on the following principles:

- In a certain negotiation round, if over half shops reject the new offer proposed by the consumer agent for the first time, it means it is too aggressive for the e-shops. The consumer will keep on negotiating using “smaller” adjustment-factors (i.e. smaller $|\beta_i|$) in the next round to adjust the rejected offer.
- In a certain negotiation round, if all e-shops accept the offer, it means the offer is less aggressive. The negotiation-agent will use a “bigger” adjustment-factor in next negotiation round.

For example, suppose the lowest price of a product after two-phase evaluation is 1000 dollars and the price of the consumer-agent’s initial offer is 800 (here β= -20%). If all e-shops rejected this price, the adjustment-factor is adjusted to -15% ($|\beta|$ is smaller this time) and a new price 850 is proposed for the next round. If $850 is widely accepted, the agent will adjust the factor to -18% (now $|\beta|$ is larger than |-15%|) and a more aggressive price $820 is proposed.

6.3 Features

The auction process in PumaMart is the type of bilateral auction. We allow both the buyer and sellers propose counter offers. But different from CDA, in PumaMart only one buyer agent exists in each auction process. This is the common feature of single-sided auctions (where one seller may face multiple buyers too). In the process of auction, a bidder has no information about others. So long as the buyer-agent continues proposing new counter offers, it shows the competition among sellers may last till the end of the auction.

Table 1 Features of PumaMart Auction-like Negotiation Processing

Auction Models	duration time	ratio of Buyer-Seller	bidder information revealed to others	settlement price among bidders	termination
English	multi-round	1-to-m or m-to-1	yes	highest price	when there are no more bids for a time period
FPSB	single-round	1-to-m or m-to-1	no	highest price	when time is reached
Vickery	single-round	1-to-m or m-to-1	no	second highest price	when time is reached
Dutch	multi-round	1-to-m or m-to-1	yes	highest price	when a reserve price is reached
CDA	multi-round	m-to-m	yes	trades take place any time at different prices	when there are no more bids for a time period
PumaMart	multi-round	1-to-m	no	The best offer is determined by the attributes of the offers, the attributes of the e-shops as well as the preference of the consumer. The winner may be not the last bidder.	when time is reached or when there are no more bids for a time period

At the end of the negotiation, the negotiation-agent can evaluate all the offers using the fuzzy evaluation criterion to find the best one. Being consistent with the two-phase evaluation model, the commercial credit and the security level of the e-shops are also taken into account in the evaluation process. The best offer is determined by the attributes of the offers, the attributes of the e-shops as well as the preference of the consumer. This means the last offer reached in the negotiation may be not the best one accepted by the consumer. If the last shop bidder has “very good” commercial credit and security ranking, it should be the winner. However, if its reputation is just “good” or “moderate”, the shop must make its offer evidently better than others so as to become the winner. Otherwise, the winner will be other bidder. This is a bit different from some Dutch auction or English auction based negotiation models, such as [35], where the last bid offer will do win.

In Table 1 we outline the features of our model. But this does not mean we have proposed a new type auction model since there are many different types of auction [36] and [37] defines taxonomy of auction parameters allowing for approximately 25 million types of auction. Our model just incorporates some features from different conventional models and it is consistent with the evaluation criteria in PumaMart.

7 Implementation

PumaMart has been implemented based on Java, IBM Aglet system and SAX APIs (the Simple API for XML) from Sun Microsystems [11, 12, 38, 39] running on a cluster of PCs connected in a LAN by 100Mbps network cards. In PumaMart, data are stored in XML documents.

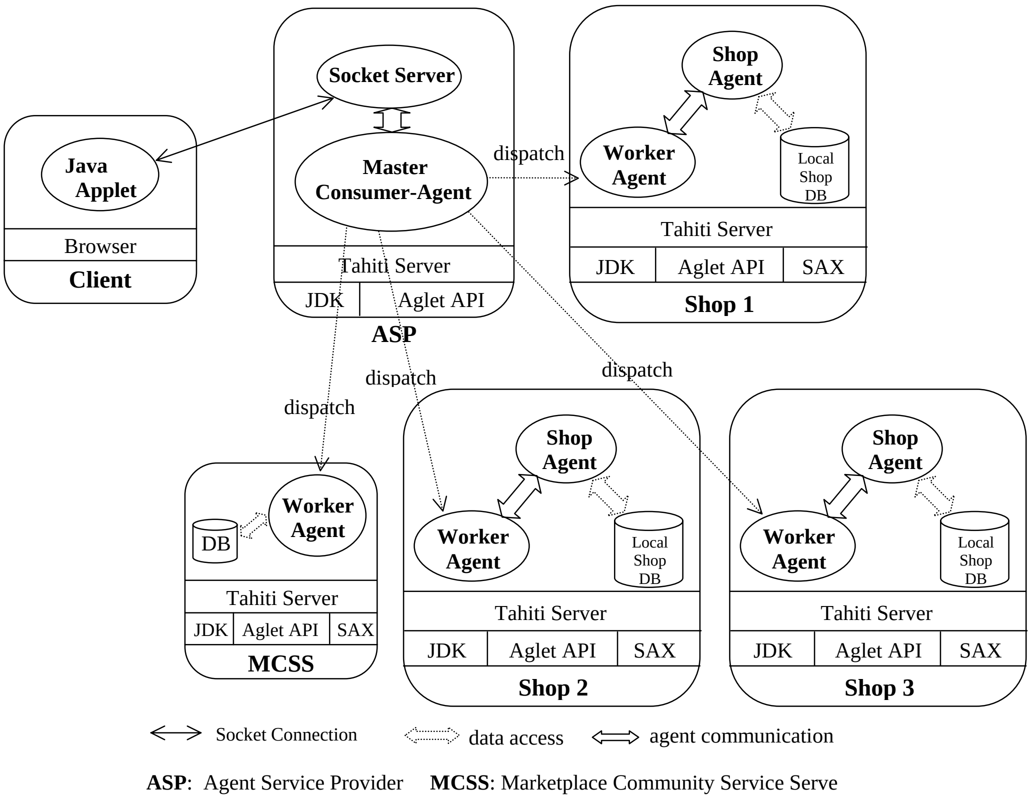


Figure 6 Implementation of PumaMart

Figure 6 briefly illustrates the framework of PumaMart. A socket server runs at ASP as a bridge between the client and the master agent. Through it, requests from the consumer are sent to the master consumer-agent and results are returned to the end user. At each shop server, a shop-agent runs on top of a Tahiti server from IBM ASDK, which is also an execution environment for coming agents. The shop-agent communicates with coming agents dispatched by the master consumer-agent.

Next, we briefly present the user interfaces of PumaMart with simulation data.

The main UI is shown in Figure 7. In this example, the product and selection preference that the consumer specified are as follows:

Product:	*Laptop*
Manufacturer:	*Toshiba*
Rating Group:	*Cautious Consumer Group*
Sub-Group:	*Price Priority Sub-Group*
Rating Standard:	*Moderate price priority*
Filter Ratio:	*Top 30%*

The 2-phase evaluation model is presented in Figures 8 and 9. According to the consumer's selection preference, some e-shops are chosen after being evaluated and filtered. Then a set of mobile agents is dispatched to these e-shops asking their offers. An e-shop may provide several offer combos. All returned offers are evaluated and sorted. In the two phases, the reputations of e-shops are taken in accounted.

Figure 10 presents the process of negotiation.

To illustrate all processes, essential information on statuses is transferred to the applet running in a browser at the consumer's side. But as a matter of fact, after submitting the request, these processes are automatically completed by the master agent at the ASP server, which employs a set of WAs at different stages, till the best offer is returned. All processes are transparent to the end user.

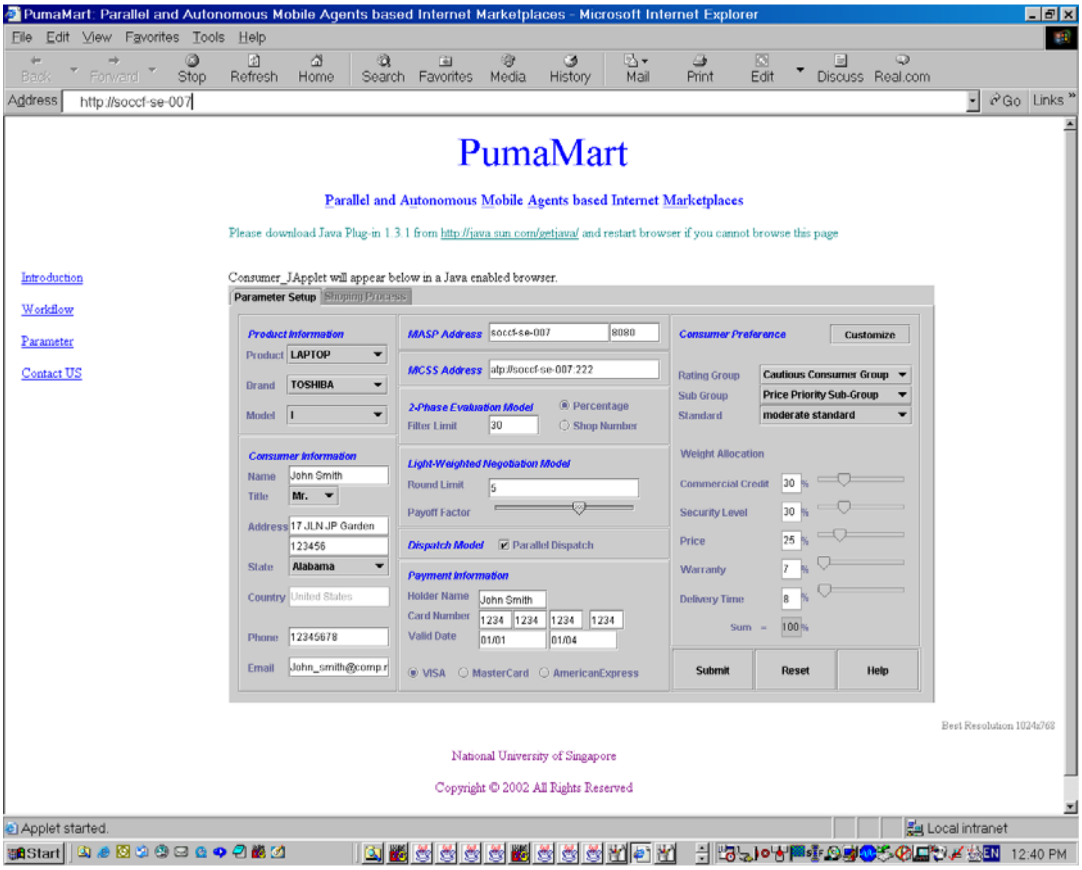


Figure 7 UI and Request Input

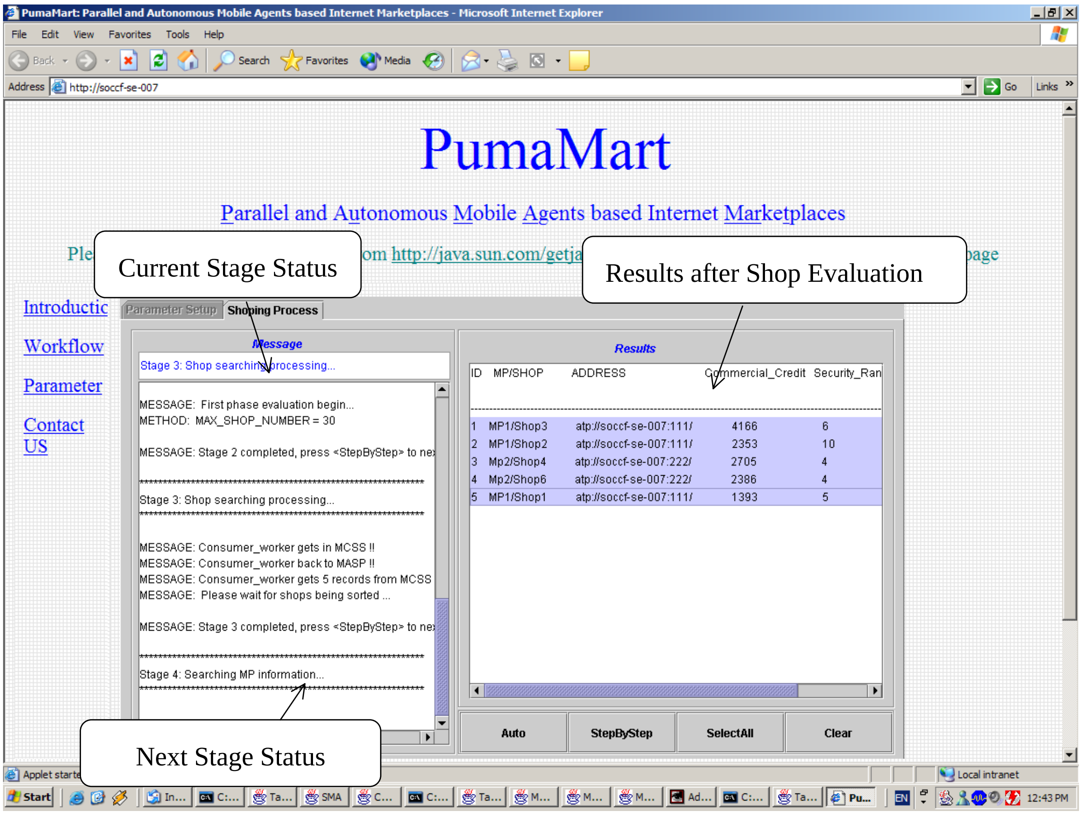


Figure 8 UI for Shop Evaluation of the 2-phase Evaluation

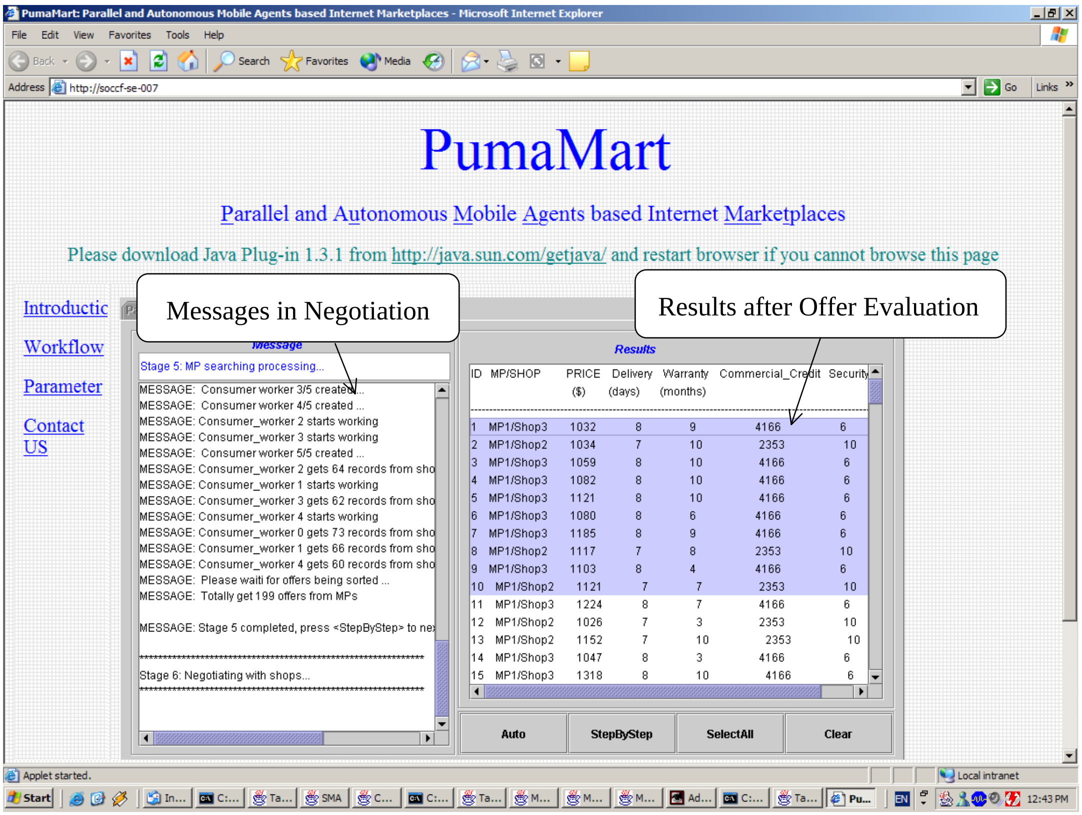


Figure 9 UI for Offer Evaluation of the 2-phase Evaluation

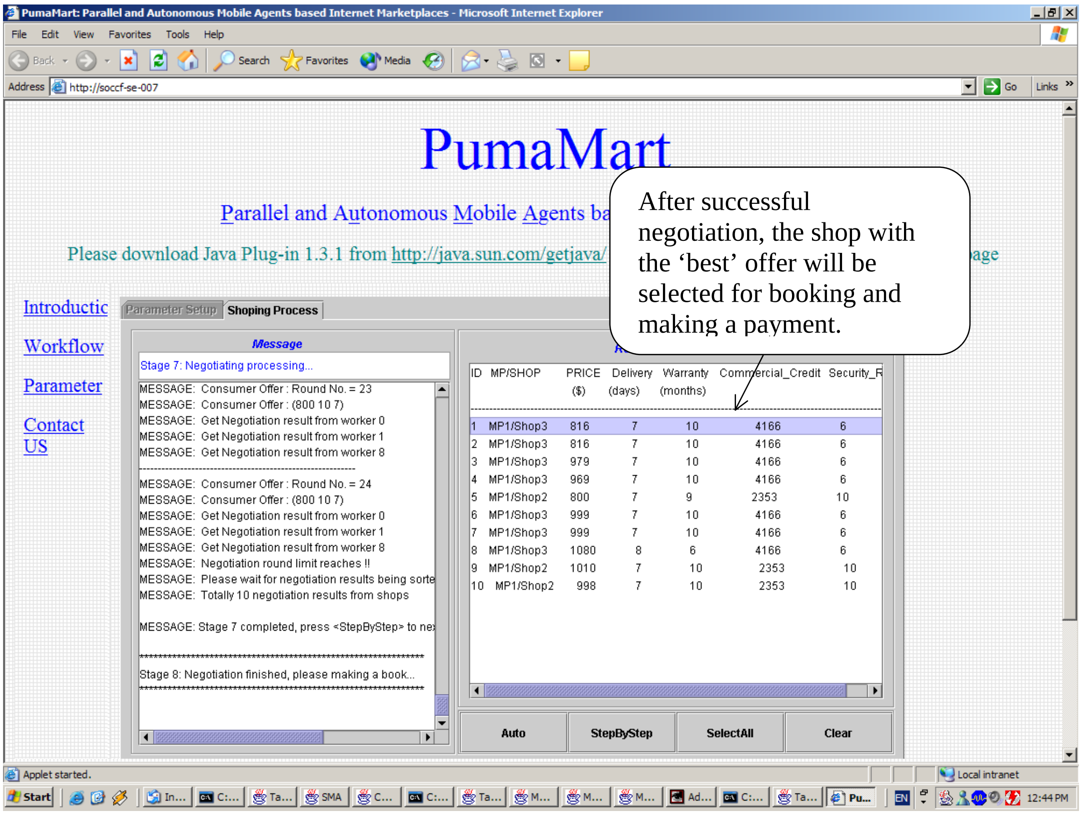


Figure 10 UI for Negotiation

8 Conclusions

In this paper, we have presented a parallel and autonomous agents based B2C Internet marketplace. To summarize, some features of our system are as follows:

1. A consumer only needs to input information for the product and his selection preferences through a web page in a Java-enabled browser. Then these requests are sent to the master agent at ASP, which employs worker agents for subsequent searching, evaluating, and negotiating. The total processes are transparent to end-users.

   In PumaMart, Parallel dispatch strategy is adopted when a large number of e-shops should be visited. In comparison to serial migration strategy, parallel dispatch can ensure that the consumer can get a response very fast.

2. Both attributes of e-shops and the attributes of offers are taken into account to evaluate offers. The attributes of e-shops include commercial credit and security ranking. The attributes of offers include price, warrantee service, and delivery time. Basic notions from fuzzy set theory are employed in evaluation process [27]. With the clustering based grading function [30], the evaluation process can reflect the real-time situation of markets. Additionally, some predefined fuzzy preference standards with crisp weights are provided for the convenience of users.

3. The evaluation process is performed in two phases. The first one evaluates and filters e-shops. It can also help to reduce network load by filtering unqualified e-shops. The 2nd phase evaluates offers where attributes of both e-shops and offers are taken into consideration.

4. Negotiation is a one-to-multiple bilateral and auction-like process. The process is based on the same fuzzy evaluation criteria used in the 2-phase evaluation model. The negotiation process starts after offer-evaluation and it is operated on those e-shops with "good" and "very good" offers. So the consumer need not specify his initial offer and border offer. The best offer is determined by the attributes of the offers, the attributes of the e-shops as well as the preference of the consumer. The winner may be not the last bidder.

This system incorporates several models for different stages. However, some more complex models should be developed to meet more complex requirements. For example, the requirements for group buying and negotiation or multi-target-dependant negotiation (e.g. a package including air-ticket, hotel booking and car renting). We also plan to incorporate some security mechanisms presented in our previous work [26, 40] into our system. Additionally, issues on commercial credit management and security rank management are still open.

Acknowledgement

This work is supported by the NSTB/MOE funded project on Strategic Program on Computer Security (R-252-000-015-112/303).

References:

[1] D. Lange and M. Oshima, Seven good reasons for mobile agents, CACM 42 (3) (1999) 88-89.

[2] S. Papastavrou, G. Samaras and E. Pitoura, Mobile agents for World Wide Web distributed database access, IEEE Trans on Knowledge and Data Engineering 12 (5) 2000 802-820.

[3] T. D. Rodrigo, and A. Stanski, The Evolving Future of Agent-based Electronic Commerce, in: S. M. Rahman and M. S. Raisinghani (Eds.), Electronic Commerce: Opportunity and Challenges, Idea Group Publishing, 2000, pp. 337-351.

[4] A. Corradi, R. Montanari and C. Stefanell, Mobile agent integrity in e-commerce application, in: Proceedings of 19th IEEE International Conference on Distributed Computing Systems, 1999, pp. 519-533.

[5] R. Guttman and P. Maes, Agent-mediated integrative negotiation for retail electronic commerce, in: Proceedings of the Workshop on Agent Mediated Electronic Trading (AMET'98), 1998, pp 70-90.

[6] H. Nwana, J. Rosenschein, T. Sandholm, C. Sierra, P. Maes, and R. GuHman, Agent-mediated electronic commerce: issues, challenges, and some viewpoints, in: Proceedings of AMET'98 Workshop, 1998, pp. 189-196.

[7] P. Maes, R. Guttman and A. Moukas, Agents that buy and sell, CACM 42 (3) (1999) 81-91.

[8] PumaMart free code, http://www.ics.mq.edu.au/~yanwang/PumaMart_v2.zip, also available at http://xena1.ddns.comp.nus.edu.sg/demo_download_page.html

[9] S. Sohn. and K. J. Yoo, An architecture of electronic market applying mobile agent technology, in: Proceedings of 3rd IEEE Symposium on Computers and Communications (ISCC '98), 1998, pp. 359-364.

[10] D. Lange and M. Oshima, Mobile agents with Java: the Aglet API, in: D. Milojicic, F. Douglis and R. Wheeler (Eds.), Mobility: Process Computers and Agents, Addison-Wesley Press, 1999, pp. 495-512.

[11] D. Lange and M. Oshima, Programming and Deploying Java Mobile Agents with Aglets, Addison-Wesley Press, 1998

[12] ASDK, http://www.trl.ibm.co.jp/aglets/

[13] P. Dasgupta, N. Narasimhan, L. E. Moser and P. M. Smith, MAgNET: Mobile agents for networked electronic trading, IEEE Transactions on Knowledge and Data Engineering, 11 (4) (1999) 509-525.

[14] Y. Wang and J. Ren, Building Internet marketplaces on the basis of mobile agents for parallel processing, in: Proceedings of 3rd International Conference on Mobile Data Management (MDM2002), IEEE Computer Society Press, pp. 61-68.

[15] M. S. Kwang and C. Raymond, A brokering protocol for agent-based e-commerce, IEEE Transaction on Systems, Man and Cybernetics, 30 (4) (2000) 474-484.

[16] R. Kowalczyk and V. Bui, On fuzzy e-negotiation agents: autonomous negotiation with incomplete and imprecise information, in: Proceedings of 11th International Workshop on Database and Expert Systems Applications, 2000, pp. 1034-1038.

[17] P. Faratin, C. Sierra and N. R. Jennings, Using similarity criteria to make negotiation trade-offs, Artificial Intelligence 142 (2) (2002), pp. 205-237.

[18] D. Zeng and K. Sycara, Benefit of learning in negotiation, in: Proceedings of 15th National Conference of Artificial Intelligence, 1997, pp. 36-41.

[19] A. Chavez, D. Dreilinger, R. Guttman and P. Maes, A real-life experiment in creating an agent marketplace, in: Proceedings PAAM97, 1997, pp. 160-179.

[20] D. Paula, G. E. Ramos and F. S. Ramalho, Electronic commerce negotiation model formalization, Technical Report UFPE-DI 99-20, 1999

[21] A. Chavez and P. Maes, Kasbah: an agent marketplace for buying and selling goods, in: Proceedings of First International Conference on the Practical Application of Intelligent Agents and Multi-Agent Technology, 1996, pp. 75-90.

[22] S. Kraus, Automated negotiation and decision making in multiagent environments, LNAI 2086, Springer-Verlag, 2001, pp. 150-172.

[23] Y. H. Tan and W. Thoen, An outline of a trust model for electronic commerce, Applied Artificial Intelligence, 14 (8) (2000), pp. 849-862.
[24] G. Zacharia and P. Maes, Trust management through reputation mechanisms, Applied Artificial Intelligence Journal, 14 (9) (2000), pp.881-908.
[25] J. Carbo and J. M. Molina, Trust management through fuzzy reputation, International Journal of Cooperative Information Systems, 12 (1) (2003), pp. 135-155.
[26] X. Pang, K.-L. Tan, Y. Wang and J. Ren, A secure agent-mediated payment protocol, in: Proceedings of ICICS2002, Springer-Verlag, LNCS Vol. 2512, pp. 422-433.
[27] S.-J. Cheng and C. L. Hwang, Fuzzy multiple attribute decision making - methods and applications, Lecture Notes in Economics and Mathematical Systems, Springer, 1992.
[28] R. E. Bellman and L. A. Zadeh, Decision-making in a fuzzy environment, Management Science, 17 (4) (1970), pp. 141-164.
[29] R. R. Yager, Targeted e-commerce marketing using fuzzy intelligent agents, IEEE Intelligent Systems, 15 (6) (2000), pp. 42-45.
[30] J. A. Hartigan, Clustering Algorithms, Wiley Press, 1975
[31] D. Plaehn and D. Lundahl, A comparison of a self-organizing neural network vs. traditional cluster analysis models in classification of consumer response data, available at http://www.camo.com/downloads/sensometrics_p3%5B1%5D.ppt
[32] M. Amin and D. Ballard, Defining new markets for intelligent agents, IT Professional, 2 (4) (2000), pp. 29-35.
[33] P. R. Wurman, Dynamic pricing in the virtual marketplace, IEEE Internet Computing, March/April 2001, pp. 36-42.
[34] F. Lopes, N. Mamede, A. Q. Novais, and H. Coelho, Towards a generic negotiation model for intentional agents, in Proceedings of 11^{th} International Workshop on Database and Expert Systems Applications, 2000, pp. 433–439.
[35] E. David, R. Azoulay-Schwartz and S. Kraus, An english auction protocol for multi-attribute items, in Proceedings of AAMAS2002 Workshop on Agent-Mediated Electronic Commerce, LNAI 2531, Springer-Verlag, pp. 52-68.
[36] M. He, N. R. Jennings and H. Leung, On Agent-mediated electronic commerce, IEEE Trans on Knowledge and Data Engineering, 15 (4) (2003) 985-1003
[37] P. R. Wurman, M. Wellman and W. Walsh, The michigan internet auctionbot: a configurable auction server for human and software agents, in: Proceedings of 2nd International Conference on Autonomous Agents, 1998, pp. 301-08.
[38] JDK, http://java.sun.com/products/
[39] SAX API, http://developer.java.sun.com/developer/products/xml/docs/api/overview-summary.html
[40] Y. Wang and X. Pang, Security and robustness enhanced route structures for mobile agents, ACM/Kluwer: Mobile Networks and Applications (MONET), ACM Press/Kluwer Academics Publisher, 8 (4) (2003) 413-423.