# Intelligent Shopping Using Migrating Agents

**Lianne Lee and Nandan Parameswaran**
University of New South Wales Sydney Australia
paramesh@cse.unsw.edu.au

**Abstract**

An automated shopping framework which employs migrating agents as a means to implement today's practices of online shopping is introduced. The system encompasses a new business market architecture which provides support for retailer agents and migrating shopper agents, with agent communication adhering to IEEE-FIPA specifications. The proposed system focuses on shopping input and protocols for the migrating shopper agent. Our experience shows that highly communicative migrating agents save more time in shopping on a large scale. However, migrating agents are susceptible to background noises that add load to the system.

## Introduction

E-commerce as a means of shopping to globally connect suppliers, retailers, and shoppers result in overload of information offered on the Internet, and repetitive direct manipulation (Maes 1994). Using migrating agents, the shopping process can be automated to a large extent for consumers, while business can still maintain some level of privacy. This paper aims to propose an adaptive and interactive FIPA-compliant migrating agent desktop shopping system.

After a brief survey in the next section, we propose a shopping architecture. We then discuss our simulation of a system and in the following section we present the evaluation results. Final section is the conclusion.

## Related Work

Maes's six-stage consumer behaviour model (Maes and Guttman 1999) for online shopping comprises of: a) Need Identification; b) Product Brokering; c) Merchant Brokering; d) Negotiation; e) Purchase; and f) Delivery. From Mae's work it is seen that information overloading occurs in the first three stages of her consumer behavior model and the last three stages address user-initiated-interaction via commands and direct manipulation. Autonomous agents can solve both these problems where the user is involved in a cooperative manner communicating her intentions and monitoring the tasks with the agents (Maes 1994).

Attempts have been made to automate shopping utilizing static intelligent shopping agents (shop bots) such as BargainFinder, ShopBot, PersonaLogic, Ringo, and Tete@Tete. Widely available comparison shop bots include Google Product Search, Yahoo Shopping, MySimon, DealTime, and StaticICE.

In Sohn and Kwan's architecture, the agents are organized into conductors which manage the market, and members who participate in electronic commerce activities, and can consist of providers, shops and consumers (Sohn 1998). Keegan and O'Hare introduced EasiShop, an agent-based, location-aware, automated ubiquitous commerce system to partially automate real-world shopping (Keegan 2002). The BestDeal system (Nipur 2009) and (Ren 2001) address security and fault tolerance using a Hierarchical Fault Tolerance Protocol (HFTP). PumaMart (Wang 2004) uses a two-phase evaluation model employing fuzzy logic to evaluate which online stores to visit, a parallel agent dispatching model, and an auction-like negotiation model.

## A Proposed Migrating Architecture

The shopping process is initiated at the client side where the user provides input including payment and the basic configurations for the behavior of migrating agents. These configurations are sent to a master server for resource allocation by Managerial Shopper Agents with the help of a Directory Facilitator Agent, where migrating Shopper Agents are created and assigned the specified resources (e.g. shopping list, budget, and shopping routes). Shopper Agents are then dispatched to retail stores for shopping where queries, purchases, and negotiations can be made with a Retail Agent. The Shopper Agents travel from one virtual outlet to another, and repeat the process as long as the client wants or until all items are bought. Meanwhile, the user has the choice to remain connected to the Managerial Shopper Agent to monitor the shopping process or leave the agents to do the job. The Directory Facilitator Agent (DFA) provides a model of the market and manages the tracking of retail stores, product scheme and Managerial Shopper Agents.

The Shopper Agent (SA) is responsible for searching, negotiating (if permitted at e-shop), and purchasing from retail stores, given a shopping list and several other configuration parameters. It does this by communicating with Retailer Agents. Constant updates will be posted if the user wishes for close monitoring of this agent. Apart from shopping, a shopper agent listens for commands coming from the MSA and responds accordingly. Depending on the configuration, a shopper agent will also be able to communicate with the other shopper agents in charge of a particular shopping.

A team is made up of several individual shopper agents organized as a tree. The chain of command starts from the MSA which receives commands from the client. Commands are passed on to the specific shopper agent or flooded to all shopper agents one level below and continue until the agents at the leaf are reached.

The Retailer Agent (RA) reads a product catalogue from the store's database to search for items, and modifies the database after a sale.

Communication Protocols: There are three types of message protocols (FIPA compliant).

• Shopper Agent vs. Retailer Agent: See Figure 1.
• Managerial Shopper Agent vs. Shopper Agent.
• Shopper Agent vs. Shopper Agent.

The following is the list of performatives used:

ACCEPT_PROPOSAL: SA to RA - SA informs the RA to add an item to the SA's cart after a proposal from the RA has been received.

CALL_FOR_PROPOSAL: SA to RA - SA queries for a proposal of an item.

INFORM: (1) RA to SA – RA informs SA that an action has been successfully completed. (2) SA to MSA - SA informs MSA of the success of either a shopping list update or a budget update.

PROPAGATE: (1) SA to SA - SA notifies other recipients to forward the message received. (2) to MSA - SA notifies the MSA of a purchase so that the message is forwarded to its children.

PROPOSE: (1) RA to SA - RA announces availability of an item and to offer a product to an SA. (2) MSA to SA - MSA proposes shopping list and budget updates to SA.

REQUEST: (1) SA to RA - SA sends a copy of SA's cart to RA for removal of an item from cart, negotiation, or checkout. (2) MSA to SA - MSA sends a request to SA for a progress update from the SA.

FAILURE: Any agent - Denotes any agent performing a task instructed whenever a failure occurs.

REFUSE: Any agent - Denotes any agent performing a task whenever an action requested is not allowed to be performed.

Commands that are supported include:

User commands: Choice of products, shops, purchase conditions and additional tasks.

Shopping-specific commands: Start a new purchase, Specify a shopping list, Specify budget; Specify retail

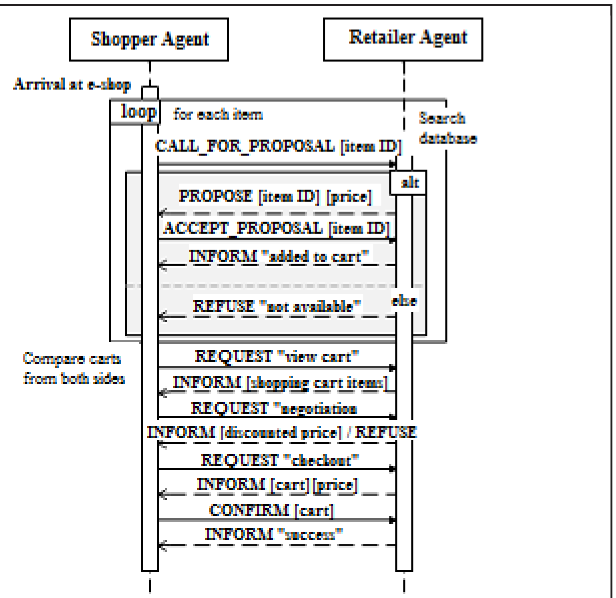


*Figure 1 Communication Protocol between SA and RA*

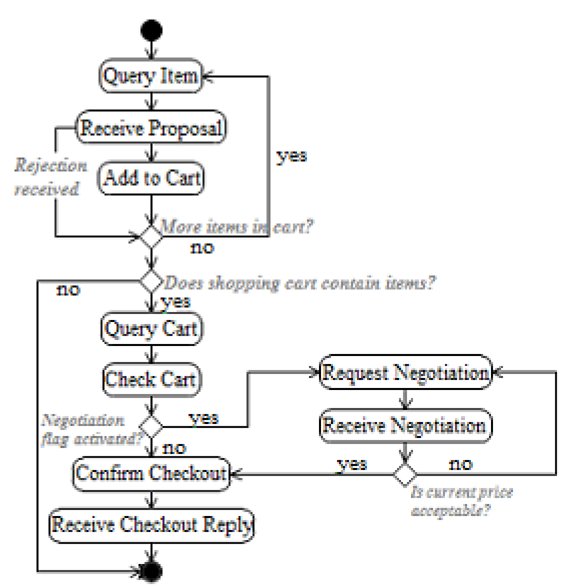


*Figure 2 A Shopper Agent's shopping cycle*

stores (optional); Specify time limit (optional); and Specify recurrence (optional).

Shopper Agent-specific Commands: Add agent, Specify type of agent, Specify sub-shopping list (optional), Shopping Agent, Launch agents, Modify shopping list, Modify budget, Return/ Return all, and Kill/ Kill-all Monitor agents.

Post Shopping Operations: Following Yan's PumaMart (Wang 2004) system, users in our system will be able to provide feedback in the form of ratings as a tool for e-shop reputation recording and also for future improvements using the commands: Rate retail store(s), Describe retail store(s), Report retail store(s), Save shopper agent(s).

## The SHOPSMART System

Our system ShopSmart was implemented in JADE. The distributed platform was achieved when containers are executed on different hosts. Intra- or inter-platform agent communication can happen, enabling in-shopping agent communication, when the user intends to monitor the progress of the agents. User can instruct the agents affecting their shopping behavior dynamically.

Shopper Agents: Agents within a group (for example, teamed agents) that are constantly sending and receiving messages amongst themselves about specific topics can register to uniquely created topics.

Performance Evaluation: Static vs. Migrating Agent Shoppers - Inter-Platform Resource Utility. The static shopper appears to shop at a faster rate with small shopping lists, while the mobile shopper takes off at a slow start, but remains almost consistent with the time taken to shop for 5000 unique items and for 1000.

Figure 3 shows that the performance of the static shopper deteriorates with the increasing number of shopping list size, as opposed to the mobile shopper. An increase in time taken to shop for both shoppers is observed as the number of messages exchanged between the shoppers and the retailer agents increases. In addition, the static shopper is in a worse position because there is a propagation delay for each message sent over the network.

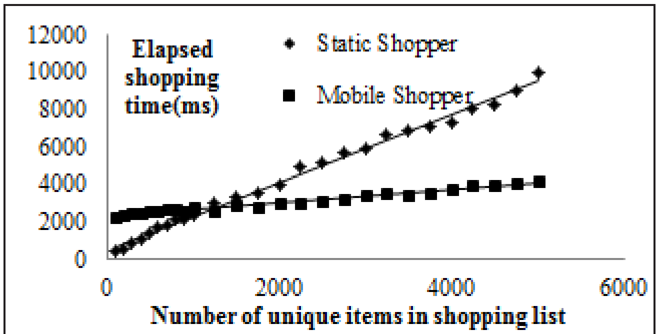


*Figure 3 Elapsed shopping time for static vs. mobile shopper with increasing shopping list size in an inter-platform node*

Static agents can be used when a high communication level is not needed. Migrating agents, on the other hand, have to overcome the hurdle of migration before being able to perform efficiently.

### Effects of Chatter on Mobile Shoppers

Figure 4 shows the time taken for a 10-node shopping cycle against the number of talkative agents per node, with chat frequencies of 1 message per second, 1.25 messages per second, and 1.5 messages per second. Talkative agents are static agents that reside in their own containers, sending a message and receiving a reply to and from an agent residing in the same container. The talkative agents are assigned a delay of 1 second, 0.8 seconds, and 0.667 seconds for each experiment to achieve a chat frequency of 1 message per second, 1.25 messages per second, and 1.5 messages per second for each agent. The number of talkative agents in each node was incremented by 25 each time, until a maximum of 325 talkative agents for each of the ten nodes is reached. The migrating agent was then assigned a shopping list with 100 items, with 10 items to shop for in each node.

With a chat frequency of 1 message per second, the mobile shopper was able to complete shopping in all ten nodes with 325 talkative agents in each node. On a chat frequency of 1.25 messages per second, the mobile shopper was able to shop at six out of the ten nodes with 325 talkative agents in each node before freezing and finally failing completely. As for the chat frequency of 1.5 messages per second, the mobile shopper was not able to reach the 325 talkative agents mark, and only managed to shop at four out of the ten nodes with 300 talkative agents each. On further analysis, the mobile shopper seems to be able to complete the 10-node shopping task at a constant rate until it encounters nodes with 250 talkative agents and more. The elapsed shopping time increases infinitely beyond this point. This proves that mobile shoppers are highly vulnerable to background noise, even if it does not affect the mobile shopper. Therefore, one can assume that a mobile agent can be attacked by increasing the noisy agents on the system.

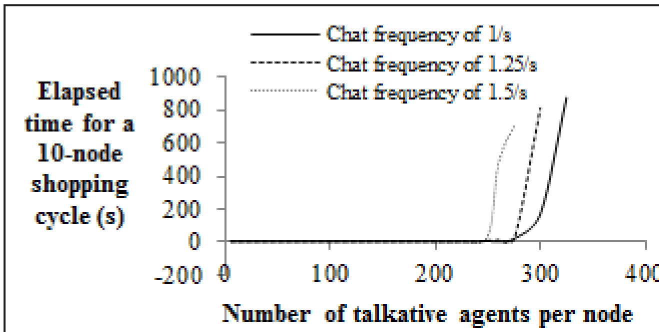


*Figure 4 Elapsed time for a 10-node shopping cycle with varying numbers of talkative agents and chat frequencies*

## Significance of Teamed Agents

Figure 5 shows a comparison of the average time taken to shop for varying shopping list sizes between a team of one, two and three agents. An individual agent is used to produce the control data set. Teams with two or more agents have highly interactive shopper agents, where each agent reports to its team mates regarding the products purchased after visiting each store. Shopping times are measured for shopping lists with 100 to 1000 products, incremented by 100 unique products each time. Shopper agents are allocated the exact same shopping list, but with different stores to visit in their shopping route. A team had to visit a total of 10 stores for each round of shopping. Hence, the individual shopper was assigned 10 stores to visit, while each agent in the paired shopping team was assigned 5 stores, and agents in the team of three had to visit 3, 3, and 4 stores each.

The results show that shopper agents working in a team proved to be faster in general, compared to the individual shopper agent. This is mostly because of the workload carried by each agent, which is lessened in a team as the number of shops to be visited has decreased. Furthermore, the shopping list size decreases as soon as a purchase is made by a team mate.

Despite a significant difference in the elapsed shopping time for an individual agent and teamed agents, the difference in shopping times between the team of 2 shopper agents and 3 shopper agents appears to fluctuate between positive and negative results. This could be explained by the ratio of the communication frequency to the size of the shopping list. On an average, shopper agents in the team of three would receive more team updates from its team members, since it has one more team member than the team of two. In addition, the messages received would have to be parsed in order to deduct the products already bought by its team members. During this process, the shopping process is entirely paused, to prevent duplicate purchases. Therefore, the level of communication between agents for a team of three shopper agents may be somewhat redundant given the size of the shopping list, and the user might be better off using a team of two, given that the graph depicts only a minor difference between the two teams. However, the difference in time for the experiment does not represent a real life scenario, as the current difference in time between results is merely ±2 seconds, which can be greatly amplified on a large-scale shopping system.

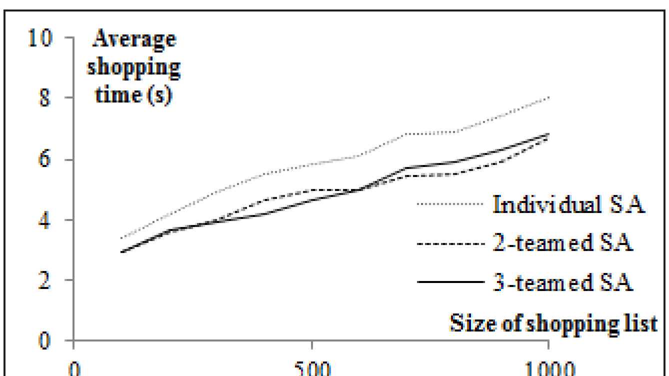

*Figure 5 Elapsed shopping time for an individual vs. teamed SA*

## Conclusion

Migrating agents proved to save resources (time and bandwidth) on a large scale compared to static agents. The results have also proven that migrating agents are susceptible to attacks, even if the attacks are not targeted towards any specific agent. Teamed agents work better than individual shopper agents in general, although much more experimentation has to be done in order to deduce a suitable team size which will correspond to the length of the shopping list addressing some of the issues raised in (Trillo 2007). The problem solving model of the migrating agents will be influenced by the shopping behavior of humans assisted by agents on their mobile phones.

## References


Keegan, S. and O'Hare, G. 2002. *Easishop: Context Sensitive Shopping for the Mobile User through Mobile Agent Technology*, In The 13th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, vol.4, 1962-1966.

Maes, P. 1994. *Agents that Reduce Work and Information Overloa*d, In Communications of the ACM, vol. 37, No. 7.

Maes, P., Guttman, R., and Moukas, A. 1998. Agents that Buy and Sell: Transforming Commerce as We Know It, In Communications of the ACM.

Nipur, H. and Garg, K. 2009. *A Fault Tolerant Comparison Internet Shopping System: BestDeal by using Mobile Agent*, In International Conference on Information Management and Engineering, 541-545.

Ren, J., Wang, Y., Pang, X., and Tan, K. 2001. *A 2-Phase Evaluation Model for Agent-mediated Internet Marketplaces*, In Web Information Systems Engineering, Proceedings of the Second International Conference, vol. 1,63-71.

Sohn, S. & Yoo, K. 1998. *An Architecture of Electronic Market Applying Mobile Agent Technology*, In Proceedings of Third IEEE Symposium on Computers and Communications 1998, 359-364.

Trillo, R., Ilarri, S., Mena, E. 2007. *Comparison and Performance Evaluation of Mobile Agent Platforms*, In Third International Conference on Autonomic and Autonomous Systems, 41-47.

Wang, Y., Tan, K. and Ren, J. 2004. *PumaMart: a Parallel and autonomous Agents Based Internet Marketplace*, In Electronic Commerce Research and Applications, 294-310.