



An agent-based approach to flexible commerce in intermediary–centric electronic markets

Duen-Ren Liu*, Tzyy-Feng Hwang

*Institute of Information Management, National Chiao Tung University, 1001 Ta Hsueh Road,
Hsinchu 300, Taiwan*

Received 11 April 2002; received in revised form 4 June 2003; accepted 4 June 2003

Abstract

The growth of electronic-markets (e-markets) necessitates the processing of many commerce protocols (processes). Each protocol handles different messages and flows of transactions among customers, merchants and intermediaries. Developing applications for each commerce protocol is costly and impractical. Accordingly, developing a flexible model of Internet commerce to support various commerce protocols is important. Moreover, the emergence of intermediaries is inevitable as e-markets grow. This work analyzes various trading processes and the roles of intermediaries, and proposes a flexible agent-based model of intermediary–centric commerce. The proposed model applies basic transaction functions to compose various commerce protocols. Agent technology and an event-driven approach are employed to process flexibly the complex trading interactions among the participants. The agent-based approach enhances scalability and makes complex trading interactions more tractable. Furthermore, an agent-based prototype system is deployed. The novel system is extensible and facilitates flexible trading interactions to support various commerce protocols in e-markets.

© 2004 Elsevier Ltd. All rights reserved.

Keywords: Agent; Electronic market; Intermediary; E-commerce; Flexible commerce

1. Introduction

As the popularity of the Internet has increased, trading on the Internet has risen dramatically. The problems of realizing Internet shopping in electronic markets (e-markets) have become important. In e-markets, many commerce protocols are used

* Corresponding author. Tel.: +886-3-5712-121x57405; fax: +886-3-572-3792.

E-mail address: dliu@iim.nctu.edu.tw (D.R. Liu).

to process transactions among customers, merchants and intermediaries. A *commerce protocol* involves a sequence of basic transaction functions, including placing orders, making payments, delivering goods, giving refunds, negotiating and repairing goods. Each commerce protocol manages transaction flows in a particular sequence. Each protocol may also use different messages to communicate among the parties involved in a transaction. Customers cannot easily become familiar with all of these commerce protocols, and so they cannot conveniently shop on the Internet. Costs are increased and designing all applications to accept all commerce protocols is impractical. Consequently, a key issue is the development of a unified Internet commerce model that can adapt to various commerce protocols and so enable customers to use various commerce protocols easily.

Ketchpel et al. (1997) introduced the idea of flexible shopping model. Their model incorporates some basic functions of merchants, customers and service providers. Customers can use various shopping mechanisms to trade with a merchant, since the shopping model is sufficiently flexible to support various commerce protocols. However, the model does not integrate the intermediary, who plays an important role in e-markets. Although the customer can deal with the merchant directly over the Internet, some services such as consultation services provided by an intermediary, cannot be replaced.

Intermediaries can evaluate various merchants on the Internet, and deal only with those who are reliable, thereby making Internet shopping more secure. An intermediary can also provide search and consultation services. Customers can shop and search for goods through the intermediary to reduce the risks and inconvenience of shopping, and save time in finding goods. Integrating intermediaries to support various shopping services is important in promoting Internet shopping in e-markets.

Software agents are programs that act on behalf of their human users or other software agents to perform laborious information-gathering tasks (Guha and Lenat, 1994; Lesser, 1999; Nodine et al., 2000). Agents communicate with each other using agent communication language (ACL, Finin et al., 1994). An ACL message is a Knowledge and Query Manipulation Language (KQML) expression in which the ‘arguments’ are terms or sentences in Knowledge Interchange Format (KIF) formed from words in the ACL vocabulary (ontology). Software agent technology is well suited to search and integrate data from heterogeneous environments (Bayardo, 1997; Nodine et al., 2000). Agent-based technology has been used to support e-business (Papazoglou, 2001) and to develop e-catalog systems (Keller and Genesereth, 1996; Liu et al., 2001) and e-commerce systems (Glushko et al., 1999; Maes et al., 1999).

This work investigates the feasibility of deploying a flexible commerce system that can support various commerce protocols in various phases of trading, including pre-trade (such as consultation), trade and post-trade (such as after-sale service). Notably, a commerce protocol is composed of a sequence of basic transaction functions. Internet commerce in e-markets is separated into two parts—the service providers that provide basic transaction functions, and the commerce model that coordinates the trading interactions among the participants. The basic service-providers include e-broker systems (Bichler and Segev, 1999; Keller and Genesereth, 1996; Liu et al., 2001; Lee et al., 2001), negotiation systems (Rebstock, 2001; Van de Walle et al., 2001) and payment systems (Abrazhevich, 2001; SET, 1997). This work focuses on the commerce model. The intermediary is incorporated in the commerce model to support various commerce

protocols covering pre-trade, trade and post-trade phases. An agent-based approach is also proposed to construct an intermediary–centric commerce model on top of the service systems, which provide various basic transaction functions. The commerce model is assumed to be able to integrate basic transaction functions through various service-provider agents.

The proposed commerce model adopts an agent-based framework that flexibly coordinates trading interactions among the trading participants. The agent-based approach increases scalability and makes complex trading interactions tractable. The model consists of five agents—the commerce-coordination agent, the customer agent, the merchant agent, the intermediary agent and the service-broker agent. Each agent uses an event-driven approach to communicate and cooperate with other agents in processing the transaction flow associated with a commerce protocol. An agent-based prototype system is deployed to realize the proposed model. The proposed system is extensible and facilitates flexible trading interactions to support various commerce protocols in e-markets.

The remainder of this paper is organized as follows. Section 2 analyzes various commerce protocols available in different trading phases. The proposed agent-based commerce model is presented in Section 3. Section 4 illustrates some commerce protocols to show how various commerce protocols can be flexibly composed based on some basic transaction functions. The implementation and demonstration of the commerce system are illustrated in Section 5. Section 6 compares our design with related research work. Finally, Section 7 presents the conclusions and suggests future work.

2. Analysis of the transaction process

This section analyzes the trading processes in Internet shopping to determine the available commerce protocols and the basic transaction functions.

The trading process of Internet shopping is divided into three phases—pre-trade, Trade, and post-trade. In the *pre-trade phase*, the primary goal is to find the goods that customers require. Customers may search for information about goods on the Internet, and then compare all goods to make a decision. The main tasks of the *trade-phase* include order, payment and delivery. The *post-trade phase* includes after-sale services. Customers may request repair, refund and exchange services from the merchant or the intermediary. These services are becoming more important to attract customers to shopping on the Internet. Table 1 lists some commerce protocols available in each phase and the basic functions of

Table 1
Commerce protocols vs. basic functions

PHASE	Commerce protocols	Basic functions
Pre-trade	Consultation-service, search-service	Search, suggest, compare
Trade	Subscription, session shareware, pre-paid	Order, payment delivery, negotiation
Post-trade	Repair service, exchange service, refund service	Service-request, return, repair, refund, negotiation, payment, delivery

each phase. Notably, various commerce protocols are composed of different sequences of basic functions.

Pre-trade phase. The intermediary collects information about merchants and products, and is thus able to provide *Consultation* and *Search services* to customers.

Trade phase. The second phase includes trading processes such as order, payment and delivery. The commerce protocols available in this phase are subscription, session, shareware and pre-pay (Ketchpel et al., 1997). For example, the Subscription service allows customer to order goods and then receive them after the bill is paid.

Post-trade phase. The post-trade phase includes after-sale services that involve basic functions such as repair, refund and exchange. In this phase, the interplay between the participants is quite complex. The commerce protocols defined in this work are as follows. (1) *Exchange service*: a customer requests this service if the product he has received has flaws or is unacceptable. He may return an item and receive another one for free (Exchange-Free) or by making some additional payment (Exchange-Bill). (2) *Repair service*: customers may issue a repair service request to the intermediary or merchant. He may return goods to the intermediary or merchant and receive them after they have been repaired. The repair may require a charge (Repair-Bill) or be free (Repair-Free). The customer may prefer to pay some additional money to exchange the damaged goods for new goods (Repair-to-Exchange). (3) *Refund service*: customers may purchase goods that do not meet their needs. Customers who ask for a refund will receive one.

3. Agent-based approach to flexible commerce

This section presents the agent-based commerce model, and then illustrates the functionality of the proposed agents.

3.1. Agent-based commerce model

The Internet shopping system can be separated into two parts—the basic service provider that provides basic transaction functions, and the commerce model that coordinates trading interactions among the participants. Fig. 1 depicts the proposed agent-based commerce model. The model comprises five agents—merchant, customer, service-broker, commerce coordinator and intermediary. The commerce model supports various services in all three trading phases. Adding an intermediary makes the trading interactions among the participants more complex. Trading interactions may occur between a pair of participants, such as the customer and the intermediary or the customer and the merchant, and also among all of the participants. For example, customers may contact the intermediary to place an order and make a payment. If the intermediary does not have the ordered goods in stock, it may inform the merchant to deliver the goods directly to the customer.

Agents communicate to cooperate in trading interactions by sending messages to each other. When an agent receives a message, it determines which event handler should be responsible for this message (transaction event). The event handler of each agent acts according to the transaction event. For instance, the payment event-handler associated

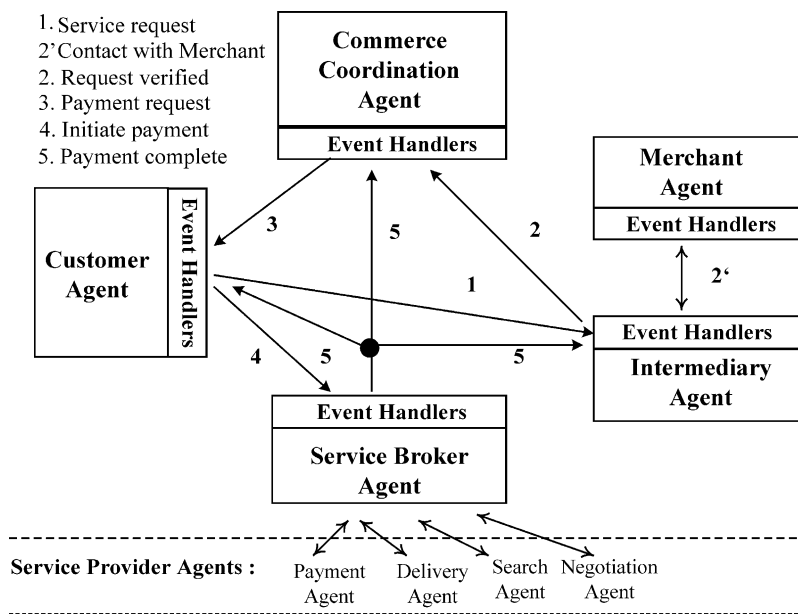


Fig. 1. Agent-based commerce model.

with each agent will handle its corresponding payment actions when a 'Payment' message is received. The payment event-handler of the customer agent deals with customers' payment actions while the payment event-handler of the intermediary agent deals with intermediary-related payment actions. Section 3.2 details the functions and the event handlers of the agents.

Fig. 1 depicts the operation of this model, and shows how participating agents communicate by sending trading messages. The trading interactions among the participating agents are as follows. (1) The customer agent starts by sending a service request (placing an order) to the intermediary agent. (2') If the intermediary does not have the ordered goods in stock, it may contact the merchant to determine whether the ordered goods are available. (2) The intermediary agent confirms the service request, and then sends a message to inform the commerce coordination agent that the request has been verified. (3) The commerce coordination agent requests that the customer makes a payment. (4) The customer agent makes the payment through the service broker agent. (5) The service broker agent signals to both the commerce coordination agent and the intermediary agent that the payment has been made. The intermediary will then deliver the goods to the customer.

Fig. 2 shows the integration of the proposed commerce model and basic service functions. The proposed model is built on top of various service providers. The service provider can provide some basic functions such as search, payment, negotiation and delivery, to be used by the commerce system, through various service-provider agents. The agent-based commerce model communicates with service-provider agents to request

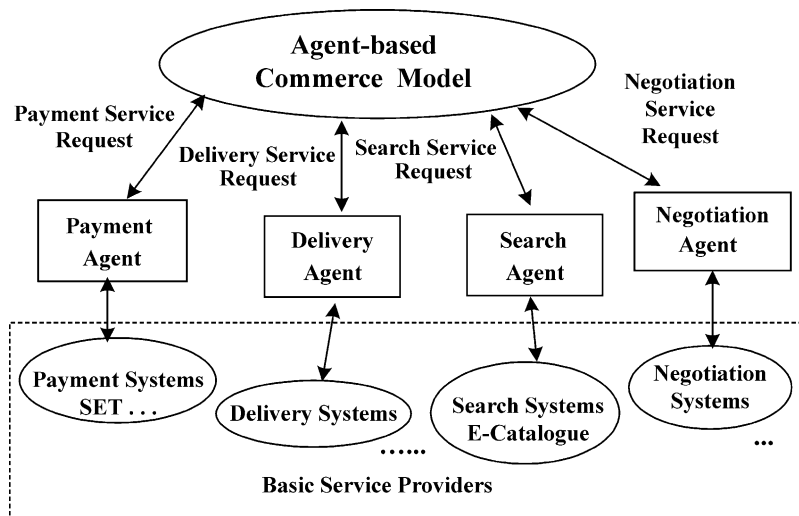


Fig. 2. Integration of commerce model and basic functions.

basic functions. For example, the model may issue a request to the search agent to apply a search function provided by the e-catalogue system.

3.2. Agents' functionality

This section details the functions of each agent. Five participants are involved in the trading process; each participant must handle transaction events at its own site. Accordingly, five types of agents have been developed to manage the transaction activities associated with the five participants in the trading process. The customer agent is responsible for handling the transaction events at the customer site. The merchant and intermediary agents handle the transaction events at their own sites. The customer may place an order, make a payment, and request a refund or repair service. The merchant or intermediary may process customer orders and payment, and then deliver the goods. The intermediary can act as an agency that either owns actual inventory or sells products directly. By providing information to customers and perhaps making suggestions, the intermediary helps customers to make purchasing decisions. The intermediary may also ask the merchant to provide further services to customers. The service-broker agent communicates with various service-provider agents to conduct the necessary basic functions.

In this proposed model, each agent takes an event-driven approach to communicate and cooperate with other agents. A sequence of trading interactions among the five participating agents represents a commerce protocol that is composed of a sequence of basic transaction functions. The trading interactions among the five participants are triggered in an event-driven way that determines each subsequent step of the transaction function. Thus, various sequences of trading interactions are flexibly facilitated, without

the need to implement each commerce protocol individually. Accordingly, the proposed commerce model flexibly supports various commerce protocols.

The event handler of each agent handles transaction events in an event-driven way. Additionally, transaction states are used to achieve flexible trading interactions. When an agent receives a specific message (event), it determines which event handler should handle the corresponding message and take actions according to the transaction state information. The commerce coordination agent determines the next step of the transaction according to the current transaction state and the transaction flow of the commerce protocol. The commerce coordination agent then sends messages to the participating agents to inform them of their next action in performing basic functions.

Event handlers. Tables 2–4 list some of the event handlers of each agent and the corresponding actions in response to a message (event). The event handlers of each agent deal only with issues related to the participants with which they are associated. For example, all of the refund event handlers of each agent handle refund activities, but the actions of each refund handler are different. The customer agent's refund handler, CustomerRefund, deals with customer-related issues. It receives the refund from the merchant or the intermediary. The merchant and the intermediary agents also have refund handlers, but their task is to pay the refund to the customer. The refund handler of the commerce coordination agent, CommerceRefund, determines the next step after the refund is completed.

Transaction state. In the proposed model, the transaction-state plays an important role in achieving the flexibility of the commerce model. Each commerce protocol is composed of a sequence of transaction states. After receiving a message from the commerce model, the customer agent determines which event handler will handle the message, and executes the corresponding actions. When the commerce coordination agent receives a message (event) that indicates that a transaction action is completed, it determines the next state and initiates the next action of the transaction. Notably, several messages may need to be communicated among the agents to cooperate the action associated with a transaction state. All transaction actions in each commerce protocol can be constructed on the basis of transaction states.

Table 5 shows sequences of transaction states for some commerce protocols. For the example of the Repair-Bill service, the transaction flow of the Repair-Bill service contains six states. The first transaction state is Requesting-Service. If the request is accepted, the next

Table 2
Event handlers of the customer agent

Event handlers	Actions
CustomerOrder	Place an order
CustomerPayment	Make payment
CustomerDelivery	Receive the goods
CustomerService	Request an after-sale service to the merchant or the intermediary
CustomerReturn	Return the goods to the merchant or the intermediary
CustomerRefund	Receive the refund
CustomerNegotiation	Negotiate the contract with the merchant or the intermediary

Table 3
Event handlers of the commerce coordination agent

Event handlers	Actions
CommerceOrder	Confirm the order, determine and initiate next state
CommercePayment	Confirm the payment, determine and initiate next state
CommerceDelivery	Confirm the delivery of goods, determine and initiate next state
CommerceService	Confirm the service request, determine and initiate next state
CommerceReturn	Confirm the return of goods, determine and initiate next state
CommerceRepair	Confirm the repair of goods, determine and initiate next state
CommerceRefund	Confirm the refund, determine and initiate next state
CommerceNegotiation	Confirm the result of negotiations, update the message, determine and initiate next state

state is Returning-Goods with the action to send the goods to be repaired back to the merchant. If the return of goods is successful, the commerce coordination agent determines the next state is the Negotiating state, and initiates negotiations between the merchant agent and the customer agent. With the negotiation, the customer and the merchant achieve an agreement on the repair cost and time. The next transaction state is Repairing-Goods, and so on. Moreover, event handlers process the actions according to the events (messages) and the transaction states. For instance, the payment event-handler in each agent will handle its corresponding payment actions of the participant at the ‘Making-Payment’ state. This model provides the flexibility to support various commerce protocols. To create a new commerce protocol, the system should define a sequence of transaction states, and the contents of the message for the interactions among the participants.

Notably, the commerce coordination agent needs to determine the next state and message receiver of the protocol. The commerce model stores the state sequence of each protocol, and the message receiver that the commerce coordination agent should send the message to in each state of the commerce protocol. The agent determines the next state and the receiver according to the current state, state sequence and corresponding receiver information. Moreover, for transaction with long process duration, such as the repair function, the intermediary or merchant agents should store the message information in order to resume the transaction after the repair function is completed.

Table 4
Event handlers of the merchant (intermediary) agent

Event handlers	Actions
MerchantOrder (IntermediaryOrder)	Check the order from customers
MerchantPayment (IntermediaryPayment)	Receive the payment from customers
MerchantDelivery (IntermediaryDelivery)	Deliver the goods to customers
MerchantService (IntermediaryService)	Check the service request from customers
MerchantReturn (IntermediaryReturn)	Receive the goods returned by customers
MerchantRepair (IntermediaryRepair)	Process the repair of goods
MerchantRefund (IntermediaryRefund)	Pay the refund to customers
MerchantNegotiation (IntermediaryNegotiation)	Negotiate with customers

Table 5
State sequences of commerce protocols

Protocol name	Sequence of transaction states
Consultation	Searching → comparing → suggesting, searching → suggesting → comparing
Subscription	Placing order → making payment → delivering goods
Session	Placing order → delivering goods → placing order → delivering goods → ... → (end of session period) → making payment
Pre-paid	Making payment → placing order → delivering goods → placing order → delivering goods → ... → making payment
Exchange-free service	Requesting service → returning goods → exchanging new goods
Exchange-bill service	Requesting service → returning goods → negotiating → making payment → exchanging new goods
Refund-service	Requesting service → returning goods → refunding
Repair-free service	Requesting service → returning goods → repairing goods → delivering goods
Repair-bill service	Requesting service → returning goods → negotiating → repairing goods → making payment → delivering goods
Repair-to-exchange	Requesting service → returning goods → negotiating → making payment → exchanging new goods

4. Illustrated examples of commerce protocols

This section illustrates some examples of commerce protocols to show how the proposed commerce model can be applied to support various commerce protocols. The examples are mainly presented to demonstrate that the system is flexible to support various commerce protocols using the basic functions discussed in Section 3.

4.1. Exchange-free service

The sequence of messages communicated among the participating agents is shown in Fig. 3. The meaning of each step is detailed in the following. (1) *Requesting a service*. The customer agent starts with sending a service request to the intermediary. The ‘service request’ message contains the information of the exchange service, identity of the customer, purchasing record and product information (merchant and receipt). The service-request handler of the intermediary agent determines what protocol of the exchange service should be used. (2) *Verifying the service request*. The service-request handler of the intermediary agent verifies the integrity of the request, checking if the customer has provided enough information and ensures that it is appropriate for the exchange service, and then sends a message to inform the commerce coordination agent that the request is verified. (3) *Return request*. Assuming that before the new goods are being delivered to the customer, the customer may return the old product for exchange first. The commerce coordination agent notifies the customer agent (message 3) that the customer should return the goods. (4) *Return*. The return handler of the customer agent is responsible for sending the goods back to the intermediary. The customer agent issues a request for delivery function (message 4) to the service-broker agent. The service-broker agent then seeks a service provider, capable of providing the delivery function, to conduct the delivery

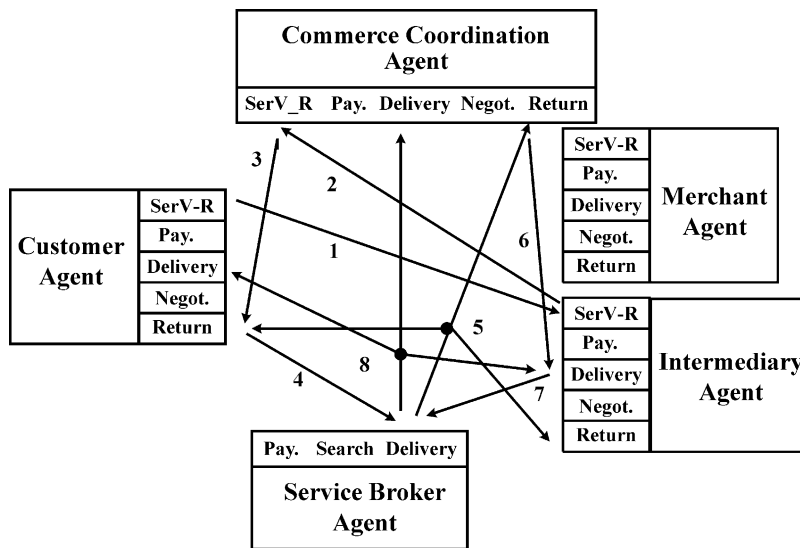


Fig. 3. Exchange-free service.

process. (5) *Acknowledging return*. As the goods are returned, the delivery handler of the service-broker agent sends an acknowledgement (message 5) to the commerce coordination agent, intermediary agent and customer agent to inform them that the goods have been delivered successfully. (6) *Request for sending goods*. As the commerce coordination agent receives the acknowledgement, it determines the next step, and then informs the intermediary agent to send new merchandise to the customer. The commerce coordination agent notifies the intermediary agent to initiate the delivery of the new goods (message 6). (7) *Initiating delivery*. This step is similar to step 4, except that a new goods is delivered from the intermediary to the customer (message 7). (8) *Acknowledging delivery*. As the new merchandise is successfully delivered, the commerce coordination agent, customer agent and intermediary agent are notified (message 8). Upon receiving the acknowledgment, the commerce coordination agent recognizes that this is the last step of the transaction.

4.2. Exchange-bill service

The example discussed above is at the cost of free exchange. The customer may need to pay additional money for exchanging new goods, as shown in Fig. 4. The transaction steps are similar to the steps in the above case. The difference between these two cases is that in this case, the customer agent uses the Negotiation function to negotiate with the intermediary agent. Notably, the steps 1–5, which are not shown in Fig. 4, are the same as the steps 1–5 of the Exchange-free service. An agreement on exchanging the goods with an additional payment can be achieved (message 6 and 7), and the customer agent issues a request for payment function (message 8–10) to the service-broker agent, as shown in

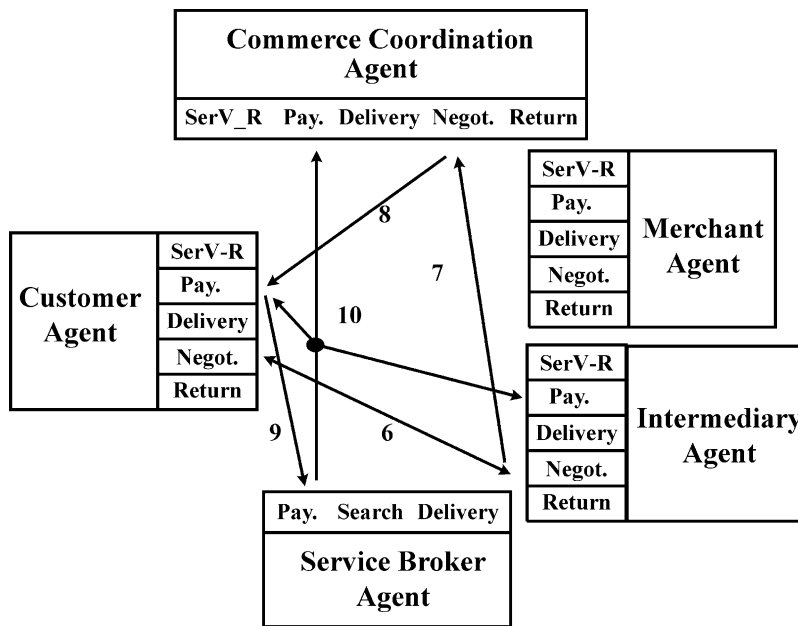


Fig. 4. Exchange-bill service.

Fig. 4. The following steps are the same as the steps 6–8 of the Exchange-free service, and are omitted in Fig. 4.

4.3. Repair-bill service

In this case, the customer agent sends a repair request to the intermediary agent. The intermediary agent may request the merchant agent to provide the repair service to the customer. The transaction steps are similar to the steps in the case of exchange-service. The main difference is that the negotiation function is conducted to estimate the cost and time of the repair. Moreover, for transaction with long process duration, such as the repair function, the intermediary or merchant agents should store the message information in order to resume the transaction after the repair function is completed. As soon as the repair is completed, the repair handler of the intermediary or merchant agent should inform the commerce coordination agent and resume the transaction.

4.4. Repair-to-exchange service

The customer may want to have another choice for the repair service. The customer may prefer to pay some additional money for an exchange of new goods, when the intermediary or merchant agent informs the customer agent about the cost of a repair. The customer agent can negotiate with the intermediary or merchant agent to change the *repair service* to the *exchange-bill service*.

The examples presented above are only a few of the available commerce protocols. Those examples demonstrated the concept of the commerce model, and showed how the event handlers of each participating agent could work together to support various commerce protocols. Although only some basic transaction functions are used to compose commerce protocols, when agents work together, a broad range of complex trading interactions (trading processes) can be generated.

5. System implementation and demonstration

This section illustrates the implementation of the proposed system. Moreover, the demonstration of our prototype system is presented. The agent-based system was implemented using JAVA and Voyager, a JAVA-based agent class developed by ObjectSpace Inc. MySQL is used as the database to store transaction related information such as the transaction record, transaction state and commerce protocols. The participating agents are running on different sites and they communicate messages through the Internet.

We demonstrate the process of the repair service as an example. In this case, we suppose the intermediary can offer the repair service, and the customer needs to pay additional money to repair the goods. The first state is requesting service. The customer sends a repair request to the intermediary (Fig. 5). The intermediary informs the commerce

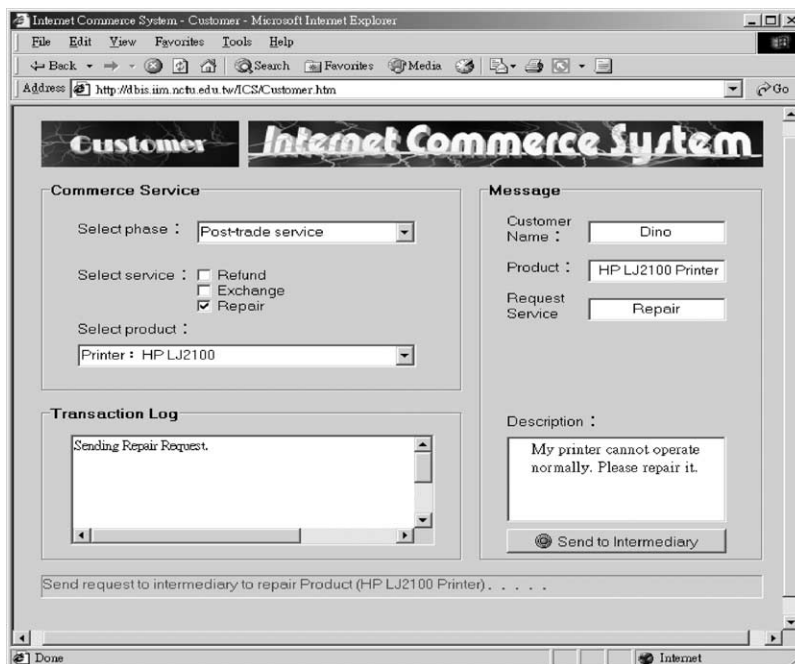


Fig. 5. Customer requests for repair service.

model after accepting this request, and then the commerce model determines the next state and asks the customer to return the goods.

After the customer returns the goods via the delivery mechanism, the commerce model informs the intermediary to estimate the cost and time of the repair (Fig. 6), and the intermediary starts to negotiate with the customer. The intermediary’s negotiation handler sends the repair contract including the time and cost to the customer. Once the customer receives the repair contract, he decides to accept the terms of the repair.

After the negotiation is completed, the commerce model informs the intermediary to initiate the repair. When the repair is completed, the intermediary’s repair handler resumes the transaction, and informs the commerce model. Then the customer is notified by the commerce model to initiate a payment for the repair cost. After the payment has been made through the payment service provider, the intermediary sends the repaired goods to the customer by the delivery mechanism, and then the commerce mode determines if the transaction is completed. Notably, the negotiation is one basic service function provided via basic service providers. In this work, we focus on the coordination and composition of basic service functions to generate various commerce protocols. Thus, we assume that the negotiation function with various negotiation strategies provided by the negotiation agents can be integrated with our commerce model through the service broker agent. In current implementation of our system, we employ the simple take-it-or-leave-it strategy.

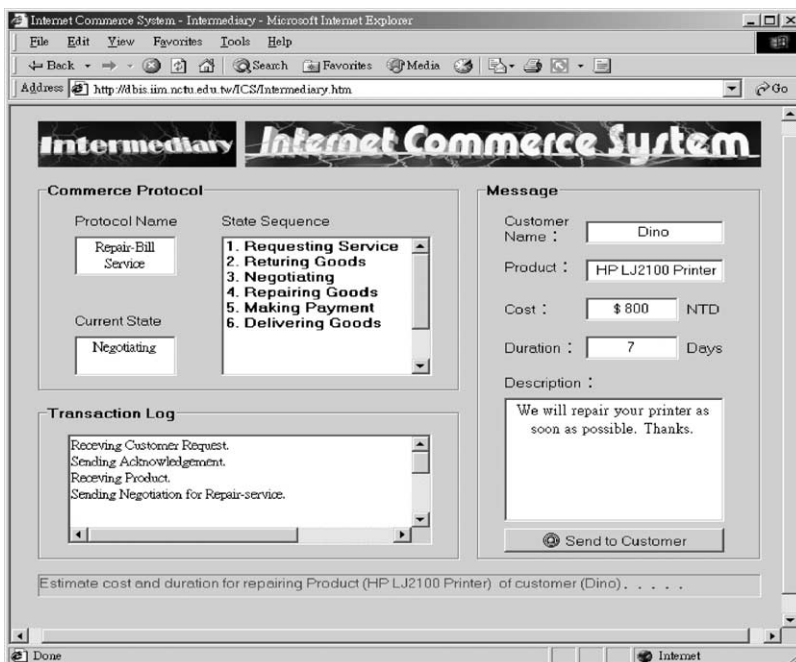


Fig. 6. Intermediary’s estimation of repair cost.

6. Discussion and comparison

Research on electronic brokers (Bichler and Segev, 1999) mainly focused on designing brokering architectures to provide various important brokerage services, such as merchandise filtering, suggestions, search, negotiation (Rebstock, 2001; Van de Walle et al., 2001) and electronic catalogs (Baron et al., 2000; Keller and Genesereth, 1996; Liu et al., 2001; Lee et al., 2001). The primary function of an electronic catalog (e-catalog) system is to support Internet-based product searches. In contrast, our proposed commerce model controls the process of trading interactions among the participants based on the composition of basic transaction functions supported by various service providers. Our proposed agent-based commerce model is built on top of the service providers. The e-broker is also a service provider that can provide the search and negotiation function to be used by our commerce model. The agent-based commerce system can be extended to integrate various service providers through the service-broker agent.

Ketchpel et al. (1997) proposed a shopping model which has a flexible architecture for information commerce. They demonstrated that how a small number of basic services can be used to construct a wide range of shopping protocols. Examples of the shopping interactions are illustrated to demonstrate the flexibility. However, their work mainly focused on the trade-phase, i.e. order, payment, and delivery. Services in the pre-trade and post-trade phase are not considered. Moreover, their model does not include the intermediary. We have extended their idea to develop an agent-based commerce system for flexible commerce in intermediary-centric e-markets.

The proposed agent-based system has incorporated with the intermediary to support various services in three trading phases. The advantage of using agent-based technology is the extensibility to integrate various service providers through the service-broker agent. Consequently, the novel system is more flexible to support various commerce protocols. As the popularity of Internet shopping has increased, more commerce protocols will be needed to suit diverse transaction behaviors. The agent-based system enhances scalability and makes complex trading interactions tractable, and thus is able to flexibly support new commerce protocols in e-markets.

7. Conclusions and future work

A flexible agent-based commerce model is proposed to support various commerce protocols in various trading phases. As electronic markets continue to grow, more protocols need to be deployed. Using the agent-based approach, the proposed model can flexibly compose new commerce protocols. Moreover, the proposed commerce model incorporates the intermediary to support services in three trading phases. A commerce system that includes an intermediary will inevitably attract customers to Internet shopping in e-markets. This work will be the basis of further research on intermediary-centric Internet commerce in e-markets.

The proposed agent-based commerce model is built on top of basic service providers that can provide basic transaction functions. This work focused on the commerce model, but integrating various service providers is very important in taking full advantage of

the commerce model. Moreover, the commerce system may need to select a service provider that can satisfy the needs of a particular customer. A standard description language is required to define various transaction functions, to enable the integration of, and search for, suitable service providers. Recently, standards and related technology of *Web services* have been discussed (Tsalgaidou and Pilioura, 2002). Further work should investigate the integration of various Web services, provided by various service providers, into the commerce model. Currently, the system is being enhanced to integrate the personalized e-catalogue system (Liu et al., 2001). Moreover, security is critical in electronic commerce. Additional work must be done to ensure that the commerce model fully meets security requirements.

Acknowledgements

The authors would like to thank the National Science Council of the Republic of China for financially supporting this research under Contract No. NSC 88-2213-E-009-092.

References

- Abrazhevich D. Classification and characterization of electronic payment systems. Proceedings of Second International Conference on Electronic Commerce and Web Technologies, Munich, Germany; 2001. p. 81–90 (Lecture notes in computer science, LNCS 2115, Springer).
- Baron JP, Shaw MJ, Bailey Jr. AD. Web-based E-catalog systems in B2B procurement. *Commun ACM* 2000; 43(5):93–100.
- Bayardo Jr.RJ. InfoSleuth: agent-based semantic integration of information in open and dynamic environments. Proceedings of the 1997 ACM SIGMOD International Conference on Management of Data, Arizona, USA; 1997. p. 195–206.
- Bichler M, Segev A. A brokerage framework for internet commerce. *Distributed Parallel Databases* 1999;7(2): 133–48.
- Finin TW, Fritzson R, McKay D, McEntire R. KQML as an Agent communication language. Proceedings of the third ACM International Conference on Information and Knowledge Management (CIKM'94), Gaithersburg, Maryland; 1994. p. 456–63.
- Glushko RJ, Tenenbaum JM, Meltzer B. An XML framework for agent-based E-commerce. *Commun ACM* 1999;42(3):106–14.
- Guha RV, Lenat DB. Enabling agents to work together. *Commun ACM* 1994;37(7):126–42.
- Keller AM, Genesereth RM. Multi-vendor catalogs: smart catalogs and virtual catalogs, EDI FORUM. *J Electron Commerce* 1996;9(3):87–93.
- Ketchpel S, Garcia-Molina H, Paepcke A. Shopping model: a flexible architecture for information commerce. Technical Report; 1997. at URL: <http://dbpubs.stanford.edu/pub/1997-52>.
- Liu D-R, Lin Y-J, Chen C-M, Huang Y-W. Deployment of personalized E-catalogues: an agent-based framework integrated with XML metadata and user models. *J Network Comput Appl* 2001;24(3):201–28.
- Lee J, Wang P, Lee HS. A visual one-page catalog interface for analytical product selection. Proceedings of Second International Conference on Electronic Commerce and Web Technologies, Munich, Germany; 2001. p. 240–9 (Lecture notes in computer science, LNCS 2115, Springer).
- Lesser VR. Cooperative multiagent systems: a personal view of the state of the art. *IEEE Transact Knowledge Data Engng* 1999;11(1):133–42.
- Maes P, Guttman RH, Moukas AG. Agents that buy and sell. *Commun ACM* 1999;42(3):81–91.
- Nodine M, et al. Active information gathering in Infosleuth. *Int J Cooperative Inf Syst* 2000;9(1/2):3–28.

- ObjectSpace Inc. Voyager 2.0B2. At URL: <http://www.objectspace.com/products/voyager>
- Papazoglou MP. Agent-oriented technology in support of e-business. *Commun ACM* 2001;44(4):71–7.
- Rebstock M. An application architecture for supporting interactive bilateral electronic negotiations. Proceedings of Second International Conference On Electronic Commerce and Web Technologies, Munich, Germany; 2001. p. 196–205. (Lecture notes in computer science, LNCS 2115, Springer).
- SET secure electronic transaction specification. Version 1.0, Book1: Business description. MasterCard, VISA, May 31; 1997, at URL: http://www.setco.org/download/set_bk1.pdf
- Tsalgatidou A, Pilioura T. An overview of standards and related technology in web services. *Distributed Parallel Databases* 2002;12(2/3):135–62.
- Van de Walle B, Heitsch S, Faratin P. Coping with one-to-many multi-criteria negotiations in electronic markets. Proceedings of Twelfth International Workshop on Database and Expert Systems Applications, Munich, Germany; 2001. p. 747–51.