

A framework for applying intelligent agents to support electronic trading

Ting-Peng Liang^{a,*}, Jin-Shiang Huang^b

^a Department of Information Management, National Sun Yat-sen University, Kaohsiung 80424, Taiwan

^b Department of Information Management, Ming Chuan University, Taipei, Taiwan

Abstract

The purpose of this paper is to study how Intelligent Agents (IAs) can be used to facilitate electronic trading. An IA is a software program designed for performing a specific task based on its own knowledge and the message it received. Given the increased complexity of Internet services, many IAs are useful to make electronic markets more effective.

In the paper, activities and structures of electronic markets are reviewed and discussed with respect to the coordination mechanism and primitive activities. This is followed by an analysis of IAs useful for electronic commerce (EC). A three-layer architecture for organizing IAs for EC is developed. Finally, application of the framework to support EC and related issues are presented. The findings are useful for implementing a more effective environment for EC. © 2000 Elsevier Science B.V. All rights reserved.

Keywords: Electronic commerce; Intelligent systems; Intelligent agents; Internet stores

1. Introduction

Electronic commerce (EC) is the cutting edge for today's business. The widespread of Internet opens an enormous amount of business opportunities. More and more organizations are facing the challenge of this new technology [8,10,16,31]. There are several motivations for doing business electronically. First, Internet users have become a fast growing group that forms a promising market. A report estimated that there were more than 28 million Internet users up to

1996 in the US alone [11]. America Online (AOL) alone is claiming to have 15 million users in 1998. The market size is estimated to be tens of billions by the year of 2000 [18]. This is attractive to virtually any business. Second, Internet offers a new way of doing business, which can overcome time and geographic barriers. Traditional businesses have specific business hours and can only serve customers within a geographic range. Since Internet links customers all around the world, customers can purchase virtually any products from anywhere in the world as long as they are available on the Internet. This will have a tremendous impact on traditional stores [3,24]. Third, EC allows service providers to have more information about their customers and customers are also likely to receive a better service. For instance, sellers can collect customer information through the

* Corresponding author. Tel.: +886-7-525-2000; fax: +886-7-525-4799.

E-mail addresses: liang@mis.nsysu.edu.tw (T.-P. Liang), jshuang@mcu.edu.tw (J.-S. Huang).

network, whereas customers can easily obtain better products or services by accessing more stores or choosing from more available alternatives. New business ideas and channel revolution are also underway [3].

Although EC is attractive; its transaction process is often complicated. Involved parties may need to collect and analyze information, negotiate contracts, execute transactions safely, and provide follow-up services over the Internet. Therefore, it is critical to develop environments that can handle the growth of electronic markets and control the increase in complexity. In fact, controlling information overload in EC may be a key to its future.

One way to reduce information overload is to delegate some activities to software agents. An agent is a person or business authorized to act on another's behalf. In software development, an agent is a computer program that can operate autonomously and accomplish unique tasks without direct human supervision [32]. A software agent possesses the properties of autonomy, social ability, reactivity, and proactiveness [30]. It is often used to manage information, support decision-making, and automate repetitive office and personal activities on behalf of the user. In some cases, an agent needs internal knowledge to perform the task intelligently. This is called an Intelligent Agent (IA). The agent approach has been applied to e-mail filtering [22], simulation [28], learning [25], programming [27] and many other domains. Therefore, it is reasonable to anticipate that, if properly applied, IAs can effectively reduce the load on the user and hence increase the performance of electronic transactions.

The purpose of this paper is to study the role of IAs in EC and to develop a framework for applying IAs to support business activities in electronic environments. The remainder of the paper is organized as follows. First, literature in EC and IAs are reviewed to give an overview of the roles, transaction mechanisms, and existing applications of IAs in EC. Then, activities involved in electronic transactions are analyzed and classified. An agent-based framework for EC is presented. It groups agents into three levels: market, contract, and activity. Agents at a lower level support those at the higher level to achieve a high integrity. Finally, sample applications of the framework are illustrated.

2. Literature review

In order to know how IAs can be applied to support EC, we need to know different types of electronic transactions and their respective IA applications.

2.1. Electronic commerce

EC is a business practice associated with the buying and selling of information, products, and services via Internet. Given the rapid proliferation of World Wide Web, Internet is becoming a new channel for business. Based on differences in product order and delivery, there are at least four types of EC on the Internet.

1. Off-line order, off-line delivery: Information is available from the Internet, but both ordering and delivery are executed off-line. For instance, a car dealer builds a homepage that provides adequate information to the customer. If the customer decides to purchase, he needs to contact the dealer off-line to order a car.
2. On-line order, off-line delivery: Both product information and ordering are available through the Internet. Once ordered, the product will be delivered off-line.
3. On-line order, on-line delivery: The same as the previous type, except that the product or service is delivered to the customer on-line. This is usually useful for information services.
4. Off-line order, on-line delivery: Customers order in a traditional way, but the product or service is delivered through the Internet. Although this is possible, it is a little unusual.

As an information and business channel, the Internet has a few features. First, the set-up cost is low. The standardization of network communication technology has significantly reduced the cost for installing a virtual store on the Web. The unit cost for information transmission also becomes virtually negligible. Second, the market structure may be changed due to changes in transaction costs. The decrease in transaction costs may eliminate the traditional inter-

mediaries such as wholesalers and create more direct and personalized marketing [2,20]. Finally, the bargaining power between buyers and sellers may change. It is generally believed that the buyer has an increased bargaining power on the electronic market due to more complete information.

No matter how their relative power changes, players in traditional commerce still exist in EC, but may perform functions differently. For instance, suppliers may promote their products or services through e-mail. Consumers may search, compare, and order products without leaving their offices. In transaction cost economics, contractual relations are classified into market, trilateral, bilateral, and unified governance [29]. Although Malone et al. [20] argued that market governance would prevail in EC due to asset specificity, a good environment for EC must support different governance mechanisms of transactions.

Given a governance mechanism, there are different trade types. For instance, the market governance includes barter, bargaining, and bidding. The trilateral governance includes auction and clearing. The bilateral governance includes contracting. In EC, these trade types must be supported.

2.2. IAs in EC

EC is a complicated process that includes information search, alternative evaluation, negotiation for terms, order and delivery of products, and post-sales service [15,16]. Most of these stages need some sort of support. Although there is no literature presenting comprehensive applications of IAs in EC, some specialized applications are available in information search and decision support.

The amount of data available on the Web makes information search and screening an early application domain for IAs. A typical approach is to use keyword matching to locate a document or to measure the relevance of a document. Many artificial intelligent techniques, such as rules, best-first search, and genetic algorithms, have been used to capture the knowledge necessary for searching intelligently [5,6,7,13,23]. Other applications include filtering e-mail and news [18], learning user profile in teaching [21].

IAs have also been applied to support decision-making. For example, Ba et al. [1] developed a client–broker–server framework that supported deci-

sions over the Internet through the coordination of interface agents, gateway agents, and information retrieval agents. Goul et al. [10] proposed a framework that applied the contract net approach to view different decision models as different IAs. In the framework, different agents must bid for service. Bhargave and Krishnan [4] presented the Decision-Net system that provides certain modeling services through an agent on the Internet. Liang and Doong [17] and Maes [19] reported results from using bargaining agents.

Recently, IAs are also developed for matching providers and consumers on the Web [13] and supporting competitive contract [14]. They divide the necessary IAs into two categories, customer and vendor, and design communication protocols between them. These works have shown the feasibility of applying IAs to support EC. However, the classification of customer and vendor may be too simple to capture various trade types. It is useful to develop a more complete framework that covers a spectrum of trade types.

3. Analysis of trading activities

In most commercial process, there are three major players: buyer, vendor, and broker. Buyers are customers who purchase certain products or services. Vendors are product or service providers. Brokers are intermediaries who help the buyer and the vendor to complete a transaction. The buyer and vendor must exist in any trading, while the broker exists only in certain conditions. These players form different trade types. Each type has a set of activities to be performed. A good environment for EC must support various trade types. In this section, we define and analyze six common trade types to identify their basic activities. These basic types are barter, bargaining, bidding, auction, clearing, and contract.

3.1. Six trade types

Common trade types can be *bilateral*, i.e., buyers and vendors trade directly, or *trilateral*, i.e., a broker exists to facilitate buyers and vendors. The common bilateral types are barter, bargaining, bidding, and contract, whereas common trilateral types are auction and clearing.

3.1.1. Barter

Barter is a trade type in which both sides offer their products for an exchange. A deal is reached if both sides have a higher preference on what the other is offering than those of their own goods. In EC, bartering exists in exchanging laser disks, compact disks, books, and others.

A typical bartering process includes: (1) the initiator announces the bartering information such as the goods, its condition, and probably what the initiator is looking for, (2) those receiving the information assess the value of the product and respond with what they would like to offer, (3) the initiator assesses the value of the offered goods to decide whether to deal, and (4) if both sides like the other's offer, a deal is made, (5) they document the deal and deliver the product to each other.

3.1.2. Bargaining

Bargaining is a trade type in which the buyer negotiates terms with the seller until an acceptable deal is reached. Usually, the buyer finds a seller, examines product price or other terms, negotiate to obtain a better deal. If the deal failed, the buyer finds another seller to bargain again. A typical bargaining process includes: (1) the buyer finds a seller, (2) they negotiate on the price or other terms, (3) if an agreement is reached, they have a deal. Otherwise, the process continues, and (4) the buyer pays for and the seller delivers the product or service.

3.1.3. Bidding

Bidding is a trade type that involves a buyer and many potential sellers. The buyer compares the received bids and chooses the best. A typical bidding process includes: (1) the buyer calls for bidding after determining the specification, amount, and base price, (2) bidders submit their bids, (3) the buyer chooses the best bid, whose price is lower than the base price (usually the lowest one among all bids), and (4) the buyer pays for and the winner delivers the product or service.

3.1.4. Auction

Auction is a trilateral trade type that involves a seller, many potential buyers, and a broker handling the auction. The buyers bid sequentially to compete for the object to be sold. A typical process includes:

(1) the seller decides the bottom price of the object to be auctioned, (2) the broker announces the object and calls for an auction, (3) potential buyers assess the value and bid for the object sequentially, (5) the broker chooses the buyer who offered the highest price (and higher than the bottom price), (6) the winner pays and the seller delivers. Commission fees are paid to the broker.

3.1.5. Clearing

Clearing is a trade type involving multiple buyers, multiple sellers, and a broker. A typical example is the stock exchange. Both buyers and sellers submit their requests. The broker tries to match the requests. A typical clearing process includes: (1) both buyers and sellers submit their requests and terms of transaction to the broker, (2) the broker compares the submissions to match them, (3) the broker informs both sides if a match is found, (4) the buyer pays and the seller delivers. Commission fees are paid to the broker.

3.1.6. Contract

Contract is a trade type in which both the buyer and the seller are governed by a set of mutually agreed rules. If there is no contract, then both sides need to negotiate for an agreement. If a contract already exists, then ensuring accurate implementation of individual orders under the regulation of the contract becomes the key. The transaction processes differ in the above two situations. For EC, negotiating contracts is more tedious and probably too difficult to complete on the Web. Therefore, supporting contracted transactions over the Web is more feasible at least for the time being. A typical trade process under contract often includes: (1) the buyer informs the provider to deliver certain products, (2) the provider confirms the request, and (3) the buyer pays and the provider delivers the product according to the contract terms.

3.2. Activities in the trading process

From the trading process of different types, we can find many similarities. Since each trading process is also a decision process, the activities in the trading process can, in fact, be classified into several

Table 1
Activities at different decision stages

Intelligence	Design	Choice	Implementation
<i>Data collection</i>	<i>Alternative generation</i>	<i>Evaluation</i>	<i>Transaction</i>
<ul style="list-style-type: none"> • Goods valuation • Specification • Vendor information • Buyer information 	<ul style="list-style-type: none"> • Solicit offers • Negotiate terms • Set-up trading conditions 	<ul style="list-style-type: none"> • Multi-attribute models • Comparative models • What-if analysis 	<ul style="list-style-type: none"> • Documentation • Payment • Delivery • Deal closure • Exception handling
		<i>Decision</i>	<i>Service</i>
		<ul style="list-style-type: none"> • Accept goods • Accept bid/offer • Accept contract • Choice models • Model solving • Choose partner • Deal notification 	<ul style="list-style-type: none"> • User profile • Process track • Learning

groups based on Simon's decision process model [26], as shown in Table 1.

At the information stage, the traders need to collect information about other traders and the object of concern. Data to be collected may include description, specification, value, and price of the object, and the ability and reputation of the trading partner. Information searching is very important for trading over the Web, especially for merchandise that cannot be inspected physically.

At the design stage, the main purpose is to identify plausible alternatives. In price-oriented trading processes, such as bidding, auction, and clearing, the focus of alternative generation is to determine proper offering prices. For other trade types, negotiation for trading terms is very important. In some cases, setting up trading conditions is also useful. For instance, one may want to barter a hard disk for a CD-ROM and set up a condition that two books will go with the hard disk if the deal is sealed in 2 days.

At the choice stage, alternative evaluation and decision making are the focal point. The trading partner evaluates the available alternatives to find the best one. Different techniques or models may be used for alternative evaluation at different trade types. For example, a simple comparative model that ranks the bids by their prices and chooses the best one is adequate for bidding, whereas a multi-attribute model that integrates concerns on different attributes may be necessary for a complicated bargaining case. What-if analysis may also be useful in some cases.

Decisions are made based on the result from alternative evaluation. Different trade types may need different decision models. For example, bidding chooses the lowest price bid; auction chooses the highest one, whereas clearing and barter pick up the best match. Once a decision is made, the partner must be notified, followed by the actual implementation of the transaction.

During the execution of transactions, all trading terms and conditions must be documented. The buyer then pays for and the seller delivers the object. If any unexpected situations occur, exception handling becomes important. Finally, some additional services such as building up user profiles, tracking the trading process, and learning the trading experience may be useful.

4. Framework for EC agents

Given different trade types, applying IAs can play a significant role in EC. For instance, a bidding agent can help call for bidding, solicit bids, notify winners, and perform transactions over the Web. In fact, applying IAs can also change traditional trade types. For example, agents serving as brokers on the Web may add a third party to bilateral transactions.

The discussion in Section 3 indicates that IAs can be used to support EC at three different levels: market, contract, and activity, as shown in Fig. 1. At the market level, the agent helps determine the proper

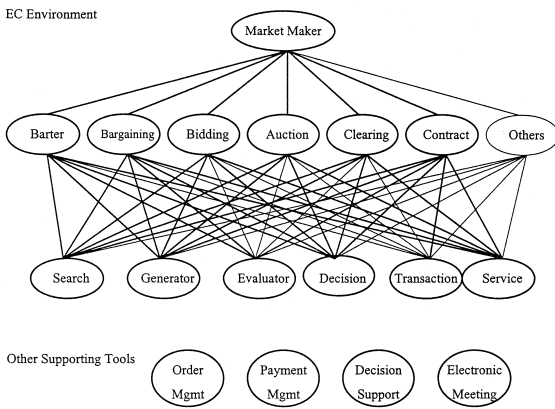


Fig. 1. The framework IAs-based EC.

trade types. At the contract level, a particular trade type has been chosen. The corresponding agent helps performing and monitoring the transaction process. At the activity level, each agent is capable of performing certain tasks such as information search or alternative evaluation.

In Fig. 1, each node stands for an agent, while the edge indicates the relationship of its connecting agents. At the top of the framework is an agent called *market maker*. It serves as a window between the user and other agents. If a transaction does not have a predetermined trade type, then the market maker will select the proper one based on its internal knowledge and activates proper agents at the lower level.

Multiple agents exist at the transaction level. Each of them is responsible for performing a certain type of transactions. For example, the barter agent will be activated upon the request of the user or the market maker. Six agents corresponding to six trade types are shown in Fig. 1. Of course, other trade types may be defined and added into the framework later on if necessary.

Activity agents at the third level perform certain activities for different trade agents. They are defined based on agent activities discussed in Section 3.2. The search agent supports data collection, the generator supports alternative creation, the evaluator supports alternative comparison, the decision agent supports the choice stage, and so on.

In addition to the basic trade mechanisms, the transaction process often needs other supports. For

example, an electronic meeting system may be useful in supporting the buyer and seller to negotiate for trading terms. In this case, a negotiation support system that is capable of analyzing the negotiator's preference may be useful.

4.1. Functions of agents

For IAs to perform well, a good operational definition is necessary. The major process for an IA to perform includes: (1) scanning the environment to receive messages, (2) process the messages based on its own knowledge, and (3) take actions. Therefore, we can represent the functions of an IA with respect to its environment, perception, knowledge, and action.

4.1.1. Market maker

The major task of the market maker is to choose a proper trade type based on user requirements. Fig. 2 shows the knowledge structure of the market maker. Upon receiving a message, the agent uses the knowledge to determine which type of trading should be chosen. The major criteria adopted by the agent includes: (1) whether the trading partner is fixed, (2) whether a product is traded for money or physical objects, (3) whether price is checked sequentially or simultaneously, and (4) whether the final deal is determined by one side or both sides. After selecting a trade type, the market maker activates the agent at the transactional level to start the trading process.

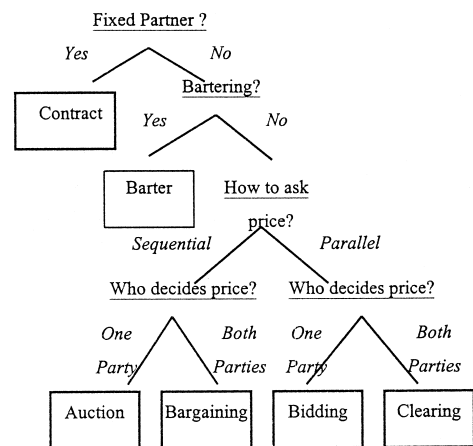


Fig. 2. Knowledge of the market maker.

To summarize, the functions of the market maker can be illustrated in the following four aspects.

Environment: EC users, all other agents.

Perception: Message from the user or other agents.

Knowledge: As shown in Fig. 2.

Action: Interacts with the user for necessary inputs, messages that activate other agents.

4.1.2. Agents at the transactional level

The major task of the agents at the transactional level is to manage the process, once a particular

trade type is chosen. It receives messages from the market maker, applies its internal knowledge to activate proper agents at the activity level to execute a transaction. Their functional illustration is as follows.

Environment: EC users, market maker, and other agents.

Perception: Messages from the user or the market maker.

Knowledge: Activities and their control sequence. The process and knowledge of the agents at this level are shown in Tables 2 and 3.

Table 2
Agent knowledge and actions at the transactional level

Agent	Process	Action
Barter	(1) Search for bartering information	(1) Request a search agent for bartering information
	(2) Solicit offers	(2) Request a generator to solicit offers
	(3) Compare alternatives	(3) Request an evaluator to compare offers
	(4) Choose an alternative	(4) Request a decision agent to make a choice
	(5) Tailor transaction details	(5) Request a transaction agent for execution
	(6) Transaction execution	(6) Request a service agent to help, if necessary
Bargaining	(1) Search for negotiation information	(1) Request a search agent for negotiation information
	(2) Negotiate for proper terms	(2) Request a generator to generate terms
	(3) Evaluate different offers	(3) Request an evaluator to assess offers
	(4) Make a choice	(4) Request a decision agent to make a choice
	(5) Tailor transaction details	(5) Request a transaction agent for execution
	(6) Transaction execution	(6) Request a service agent to help, if necessary
Bidding	(1) Search for bidding information	(1) Request a search agent for bidding information
	(2) Solicit bids	(2) Request a generator to solicit bids
	(3) Evaluate bids	(3) Request an evaluator to assess bids
	(4) Choose the highest bid	(4) Request a decision agent to make a choice
	(5) Tailor transaction details	(5) Request a transaction agent for execution
	(6) Transaction execution	(6) Request a service agent to help, if necessary
Auction	(1) Search for auction information	(1) Request a search agent for auction information
	(2) Solicit offers	(2) Request a generator to solicit offers
	(3) Evaluate offers	(3) Request an evaluator to assess offers
	(4) Choose the best offer	(4) Request a decision agent to make a choice
	(5) Tailor transaction details	(5) Request a transaction agent for execution
	(6) Transaction execution	(6) Request a service agent to help, if necessary
Clearing	(1) Search for clearing information	(1) Request a search agent for clearing information
	(2) Solicit requests for clearing	(2) Request a generator to solicit requests
	(3) Evaluate requests	(3) Request an evaluator to assess terms
	(4) Find matching options	(4) Request a decision agent to match requests
	(5) Tailor transaction details	(5) Request a transaction agent for execution
	(6) Transaction execution	(6) Request a service agent to help, if necessary
Contract	(1) Search for contract information	(1) Request a search agent for contract information
	(2) Negotiate for proper terms	(2) Request a generator to process negotiation terms
	(3) Evaluate agreement	(3) Request an evaluator to assess agreements
	(4) Finalize a contract	(4) Request a decision agent to finalize a contract
	(5) Tailor transaction details	(5) Request a transaction agent for execution
	(6) Transaction execution	(6) Request a service agent to help, if necessary

Table 3

Functions of agents at the transactional level

○ indicates required functions; * indicates optional functions.

Types of knowledge	Barter agent	Bargaining agent	Bidding agent	Auction agent	Clearing agent	Contract agent
<i>Data collection</i>						
Good valuation	○	○	○	○	*	○
Specification	○	○	○	○	○	○
Vendor information	○	○	○	*	*	○
Buyer information	○	○	*	○	*	○
<i>Alternative generation</i>						
Solicit offers	○	○	○	○	○	○
Negotiate terms	○	○	*	*	*	○
Set-up conditions	○	○	○	○	○	○
<i>Alternative evaluation</i>						
Multi-attribute models	○	○	*	*	*	○
Comparative models	○	○	○	○	○	○
What-if analysis	○	○	○	○	○	○
<i>Decision</i>						
Accept goods	○					
Accept bid/offer		○	○	○	○	
Accept contract						○
Choice models	○	○	*	*	*	○
Model solving	○	○	*	*	*	○
Determine partner	○	*	○	○	○	*
Deal notification	○	○	○	○	○	○
<i>Transaction</i>						
Documentation	○	○	○	○	○	○
Payment		○	○	○	○	○
Delivery	*	*	*	*	*	*
Deal closure	○	○	○	○	○	○
Exception handling	○	○	○	○	○	○
<i>Service</i>						
User profile	*	*	*	*	*	*
Process track	*	*	*	*	*	*
Learning	*	*	*	*	*	*

Action: Actions corresponding to the transaction process are shown in Table 2.

and transaction agents in a proper sequence. Their functional illustration is as follows.

4.1.3. Agents at the activity level

The agents at the activity level perform their functions upon receiving requests from the trading agents. For instance, a bidding agent would activate the search agent, generator, evaluator, decision agent,

Environment: EC users, all agents.

Perception: Messages from IAs at the transaction level.

Knowledge: Each agent at the activity level has its own knowledge for performing certain functions. For example, a search agent possesses a set of knowl-

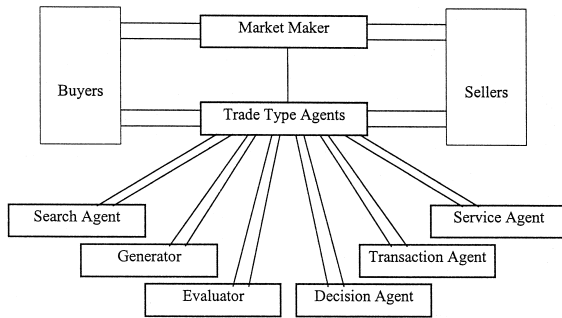


Fig. 3. The operation of IA architecture in EC.

edge for collecting product value, specification, and other related data. An evaluation agent possesses knowledge of how to evaluate an offer when different trade types are adopted.

Action: Execute the requested task properly.

4.1.4. Other supporting tools

In addition to the agents at three levels, other tools are useful in supporting EC. For example, agents for order and payment management may be used to support the transaction agent, while the electronic meeting agent may be used to support bargaining.

Fig. 3 shows the coordination mechanism for applying the IAs in EC described in this section. Both the buyer and seller can initiate a trade by sending requests to the market maker. The market maker interacts with the players over the Web to know the user requirements and choose a trade type accordingly. The trade agent, once activated by the market maker, sends messages to proper activity agents and controls the transaction process.

5. Illustrative examples

To demonstrate the above agent-based EC environment in detail, an illustrative example is presented in this section. For agent communication, we adopt the Knowledge Query and Manipulation Language (KQML) [9] as the outer language, and an inner language similar to UNIK-OBJECT [12] to describe contracts. Three most commonly used com-

mands (called *performative* in KQML) are ask-if, evaluate, and reply.

ask-if

```

:reply-with < expression >
:sender < word >
:receiver < word >
:content < expression >
  
```

Ask-if is a performative that allows the sender to ask *the receiver to check the conditions stated in the content section*. The receiver responds to the sender with the given *reply-with* label.

evaluate

```

:reply-with < expression >
:sender < word >
:receiver < word >
:content < expression >
  
```

Evaluate is a performative that allows the sender to ask *the receiver to provide certain services defined in the content section*. After fulfilling the requirement, the receiver replies to the sender with a proper message given in the *reply-with* section.

reply

```

:in-reply-to < expression >
:sender < word >
:receiver < word >
:content < expression >
  
```

Reply is a performative that allows the receiver to *reply the sender's request (in-reply-to) with a message included in the content section*.

Given the above performatives, we can use KQML to show agent communications in the following example.

(1) Assume that the company EW would like to purchase 10 notebook PC over the Web, the command that initiates the electronic transaction is to send a request to the market maker. We can either send a request for a certain type of trade or a statement of conditions and let the market maker to

choose a proper trade type. The proper agent messages in the latter case would be the following.

```
(evaluate
:reply-with 981201-01
:sender EW
:receiver MARKETMAKER)
:content
  ((TITLE REQUEST_FOR_TRADING_
TYPE)
  (CONTRACT_TYPE DETERMINED_
BY_MARKETMAKER)
  (REQUIREMENTS
  (PRODUCTS
  (ITEM_NAME NOTEBOOK PC)
  (QUANTITY 10)
  (SPECIFICATIONS
  (PROCESSOR (> = "486")))))
```

(2) Upon receiving the request, the market maker activates the bidding agent if explicitly stated as in the first case, or chooses a trade type based on its own knowledge in the second. In this example, the message does not indicate that EW would like to order from any particular seller. The market maker suggests *bidding* to pursue the lowest price. Therefore, it sends a message to the bidding agent.

```
(evaluate
:reply-with 981201-02
:sender MARKETMAKER
:receiver BIDDING_AGENT)
:content
  ((TITLE REQUEST_FOR_PROPOSAL)
  (CONTRACT_TYPE BIDDING)
  (REQUIREMENTS
  (PRODUCTS
  (ITEM_NAME NOTEBOOK PC)
  (QUANTITY 10)
  (SPECIFICATIONS
  (PROCESSOR (> = "486")))))
```

(3) Upon receiving the message, the bidding agent has to send messages to the agents at the activity level to solicit and evaluate bids, and choose the best one for EW. It first sends requests to a generator to

solicit bids and a search agent to collect information about the bidder.

```
(evaluate
:reply-with 981201-03
:sender BIDDING_AGENT
:receiver GENERATOR SEARCH_AGENT)
:content
  ((TITLE SEARCH_FOR_PROPOSAL)
  (CONTRACT_TYPE BIDDING)
  (REQUIREMENTS
  (PRODUCTS
  (ITEM_NAME NOTEBOOK PC)
  (QUANTITY 10)
  (SPECIFICATIONS
  (PROCESSOR (> = "486")))))
```

(4) Potential bidders submit their bids using *reply*. For example, if the company XY would like to bid by offering Pentium500 machines at the price of US\$1000 each, then it replies the following message to the generator.

```
(reply
:in-reply-to 981201-03
:sender XY
:receiver SEARCH_AGENT)
:content
  ((TITLE PROPOSAL)
  (CONTRACT_TYPE BIDDING)
  (PROPOSAL
  (PRODUCTS
  (ITEM_NAME NOTEBOOK PC)
  (QUANTITY 10)
  (SPECIFICATIONS
  (PROCESSOR
  (= "Pentium500")))))
```

(5) After receiving the bid from XY, the Generator may pass a message to the search agent to check for other information of the company.

```
(evaluate
:reply-with 981201-04
:sender GENERATOR
:receiver SEARCH_AGENT)
:content
```

```
((TITLE SEARCH_FOR_PROPOSAL_DE-
TAIL)
(CONTRACT_TYPE BIDDING)
(COMPANY XY))
```

(6) The search agent launches the search to find relevant information and then reply to the generator.

```
(reply
:in-reply-to 981201-04
:sender SEARCH_AGENT
:receiver GENERATOR)
:content
((TITLE ANSWER_FOR_PROPOSAL_DE-
TAIL)
(CONTRACT_TYPE BIDDING)
(COMPANY XY)
(DESCRIPTION (ESTABLISHED IN
1930 SALES 15M))
```

(7) The generator puts together all data and passes it back to the bidding agent.

```
(reply
:in-reply-to 981201-03
:sender GENERATOR
:receiver BIDDING_AGENT)
:content
((TITLE ANSWER_FOR_COMPLETE_
PROPOSAL)
(CONTRACT_TYPE BIDDING)
(COMPANY XY)
(DESCRIPTION (ESTABLISHED IN
1930 SALES 15M)
(PROPOSAL
(PRODUCTS
(ITEM_NAME NOTEBOOK PC)
(QUANTITY 10)
(SPECIFICATIONS
(PROCESSOR
(= ‘Pentium500’))))))
```

(8) After receiving adequate bids, the generator passes all bids to the evaluator. The evaluator chooses the best one and passes the message back to the bidding agent, which then passes the result to the transaction agent to document and execute the transaction.

The performative *ask-if* may be used for query. For example, if the bidding agent would like to know

whether there are firms capable of providing notebook PC whose CPU is 486 or higher, then it may ask the search agent to announce it on the Web:

```
(evaluate
:reply-with 981201-05
:sender BIDDING_AGENT
:receiver SEARCH_AGENT)
:content
((TITLE SEARCH_FOR_PROVIDER)
(REQUIREMENTS
(PRODUCTS
(ITEM_NAME NOTEBOOK PC)
(QUANTITY 10)
(SPECIFICATIONS
(PROCESSOR (> = ‘486’))))))
```

The search agent checks to see whether firms willing to bid exist on the Web by passing the following message to the search agent.

```
(ask-if
:reply-with 981201-06
:sender SEARCH_AGENT
:receiver COMPANY)
:content
((TITLE ANNOUNCE_FOR_PROPOSAL)
(CONTRACT_TYPE BIDDING)
(REQUIREMENTS
(PRODUCTS
(ITEM_NAME NOTEBOOK PC)
(QUANTITY 10)
(SPECIFICATIONS
(PROCESSOR (> = ‘486’))))))
```

If there are firms available, the search agent will reply to the bidding agent. The above example shows that different trade types can be implemented in an IA-based environment. The proposed framework for EC should work well.

6. Concluding remarks

EC is becoming an important channel for future business. In this paper, we have presented a three-level framework for using IAs to support EC. It groups agents into three layers: market, transaction,

and activity. Agents at the market level accept requests from the user and choose a proper trade type. Agents at the transaction level ensure that the selected type is executed properly. Agents at the activity level perform a specific task in the user's decision making process. Because the framework proposed in the paper takes into account major trade types as well as the stages in decision process, including intelligence, design, and choice, it is more useful than previous work focused on developing a single agent.

The contribution of the research is that the framework provides guidelines for developing a friendly electronic trading environment. Given the complexity of EC, it is critical to have IAs to help out. Although more new agents may be defined, this work is a good beginning toward studying the application of IAs to EC. The research is, of course, not without limitations. First, the research is more conceptual at this stage, more comprehensive prototyping may be useful to discover unfound problems. Second, applying IAs to EC is a new area that lacks significant cases. Some activities such as bargaining are extremely complicated and may be difficult to coordinate or control. It takes more time to develop and evaluate those agents proposed in the framework. Finally, agent coordination is critical in an agent-based environment. Our approach uses higher level agents to coordinate activities at the lower level. This works but may not be the best. More effort is necessary to make such a friendly environment become the reality.

Acknowledgements

The authors thank the reviewers for their comments on earlier versions of the manuscript. The research was partially funded by a grant from National Science Council to the first author.

References

- [1] S. Ba, R. Kalakota, A.B. Whinston, Using client-broker-server architecture for intranet decision support, *Decision Support Systems* 19 (1997) 171–192.
- [2] R. Benjamin, R. Wigand, Electronic markets and virtual value chains on the information superhighway, *Sloan Management Review* 37 (1995) 62–72, Winter.
- [3] P. Berthon, L.F. Pitt, R.T. Watson, The World Wide Web as an advertising medium: toward an understanding of conversion efficiency, *Journal of Advertising Research* 1 (1996) 43–54, January/February.
- [4] H.K. Bhargave, R. Krishnan, Decision support on demand: emerging electronic markets for decision technologies, *Decision Support Systems* 19 (1997) 193–214.
- [5] C.M. Bowman, P.B. Danzig, U. Manber, F. Schwartz, Scalable internet resource discovery: research problems and approaches, *Communication of the ACM* 37 (8) (1994) 98–107.
- [6] H. Chen, Y.M. Chung, M. Ramsey, Intelligent spider for Internet searching, *Proceedings of the 30th Hawaii International Conference on System Science*, IEEE Press, Los Alamitos, CA, 1997, pp. 178–188.
- [7] P. DeBra, R. Post, Information retrieval in the World Wide Web: making client-based searching feasible, in: *Proceedings of the First International World Wide Web Conference* (Geneva, Switzerland), 1994.
- [8] S. Dekleva, J. Zupancic, Key issues in information systems management: a Delphi study in Slovenia, *Information and Management* 31 (1) (1996) 1–11.
- [9] T. Finin, J. Weber, G. Wiederhold, M. Genesereth, R. Fritzson, J. McGuire, S. Shapiro, C. Beck, Specification of the KQML agent-communication language plus example agent policies and architecture, Draft, The DARPA Knowledge Sharing Initiative, External Interface Working Group, 1993.
- [10] M. Goul, A. Philippakis, M.Y. Kiang, D. Fernandes, R. Otondo, Requirements for the design of a protocol suite to automate DSS deployment on the World Wide Web: a client/server approach, *Decision Support Systems* 19 (1997) 151–170.
- [11] D.L. Hoffman, W.D. Kalsbeek, T.P. Novak, Internet and Web use in the US, *Communication of the ACM* 39 (12) (1996) 36–46.
- [12] D.L. Hoffman, T.P. Novak, Marketing in hyper-media computer-mediated environments: conceptual foundations, *Journal of Marketing* 60 (2) (1996) 50–68.
- [13] J.K. Lee, D.W. Cheung, B. Kao, J. Law, T. Lee, Intelligent agents for matching information providers and consumers on the World Wide Web, *Proceedings of the 30th Hawaii International Conference on System Science*, IEEE Press, Los Alamitos, CA, 1997, pp. 189–199.
- [14] J.K. Lee, W. Lee, Intelligent agent based contract process in electronic commerce: UNIK-AGENT approach, *Proceedings of the 30th Hawaii International Conference on System Science*, IEEE Press, Los Alamitos, CA, 1997, pp. 230–241.
- [15] T.P. Liang, J.S. Huang, An empirical study on consumer acceptance of products on electronic market: a transaction cost model, *Decision Support Systems* 24 (1998) 29–43.
- [16] T.P. Liang, H.J. Lai, Development of electronic shopping in Taiwan, *Proceedings of the APEC Conference on Information Literacy*, Tokyo, Japan, 1997.
- [17] T.P. Liang, H.S. Doong, Effect of bargaining in electronic Commerce, *International Journal of Electronic Commerce*, 5 (3), (2000).

- [18] R.C. MacClaren, The Web means business, *PROMO: The Magazine of Promotion Marketing* 9 (1995) 31, December.
- [19] P. Maes, Agents that reduce work and information overload, *Communications of the ACM* 37 (7) (1994) 31–40.
- [20] T.W. Malone, J. Yates, R.I. Benjamin, Electronic markets and electronic hierarchies, *Communications of the ACM* 30 (6) (1987) 484–497.
- [21] T. Mitchell, R. Caruana, D. Fritag, J. McDermott, D. Zabowski, Experience with a learning personal assistant, *Communications of the ACM* 37 (7) (1994) 80–91.
- [22] L.F. Motiwalla, An intelligent agent for prioritizing e-mail messages, *Information Resources Management Journal* (1995) 16–24, Spring.
- [23] B. Pinkerton, Finding what people want: experiences with the Webcrawler, *Proceedings of the Second International World Wide Web Conference*, Chicago, IL, 1994.
- [24] E.M. Roche, Business value of electronic commerce over interoperable networks, *Information Infrastructure and Policy* 4 (1995) 307–325.
- [25] T. Selker, Coach: a teaching agent that learn, *Communications of the ACM* 37 (7) (1994) 92–99.
- [26] H. Simon, *The New Science of Management Decision*, Prentice-Hall, Englewood Cliffs, NJ, 1977.
- [27] D.C. Smith, A. Cypher, J. Spohrer, KidSim: programming agents without a programming language, *Communications of the ACM* 37 (7) (1994) 54–67.
- [28] M. Tambe, W.L. Johnson, R.M. Jones, F. Koss, J.E. Laird, P.S. Rosenbloom, K. Schwamb, Intelligent agents for interactive simulation environments, *AI Magazine* 16 (1995) 15–39, Spring.
- [29] O.E. Williamson, Transaction-cost economics: the governance of contractual relations, *The Journal of Law and Economics* 22 (2) (1979) 233–261.
- [30] M. Wooldridge, J.R. Jennings, Agent theories, architectures, and languages: a survey, In: M. Wooldridge, J.N. Jennings (Eds.), *Proceedings of ECAI-94 Workshop on Agent Theories, Architectures, and Languages*, Amsterdam, The Netherlands, 1994, August.
- [31] H.L. Yang, Key information management issues in Taiwan and the US, *Information and Management* 30 (5) (1996) 251–267.
- [32] J. Yen, Human agents and intelligent agents: an experiment on the internet, *Proceedings of the 30th Hawaii International Conference on System Science*, IEEE Press, Los Alamitos, CA, 1997, pp. 242–251.



Ting-Peng Liang is Professor of Information Systems and Director of the Software Incubator of the National Sun Yat-sen University in Kaohsiung, Taiwan. Prior to the current position, he was the Director of the Graduate Institute of Information Management and Dean of the College of Management. His research interests include electronic commerce, decision support systems, model management, and intelligent systems. His papers have appeared in journals

such as *Management Science*, *Operations Research*, *Decision Sciences*, *Decision Support Systems*, *MIS Quarterly*, *Journal of MIS*, among others. He is also on the editorial boards of many academic journals.



Jin-Shiang Huang is an Assistant Professor of Information Systems at Ming Chuan University in Taipei, Taiwan. He received his MBA and PhD degrees in Information Management from National Sun Yat-sen University. His research interests include economics of electronic commerce, decision support systems, and negotiation support systems. His papers have appeared in *Decision Support Systems* and conference proceedings.