An Intelligent Agent Based Competitive Contract Process: UNIK-AGENT

Jae Kyu Lee and Woongkyu Lee

Graduate School of Management,

Korea Advanced Institute of Science and Technology

E-mail: jklee@msd.kaist.ac.kr

An earlier version of this paper “Intelligent Agent Based Contract Process in Electronic Commerce : UNIK-AGENT Approach”, was presented at the Thirtieth Hawaii International Conference on System Sciences (HICSS-30) in January 1997 and has won the best paper award in the Internet and Digital Economy Track.
Abstract

As the second generation of electronic commerce, the framework of the agent based commerce is proposed. Since the communication messages and solution methods of agents are significantly influenced by the contract type, we have defined the grammar of contracts first. By using this generalized contract grammar, we can express new creative contract types as well as traditional ones like bids and auctions. The messages in agent based commerce have adopted three layers: Agent Communication Language, Electronic Commerce, and Product Specification. For electronic commerce, the agent is regarded as an extension of expert systems with the additional capability of communication control and meta-problem-solving. A prototype of the agent based commerce development environment, UNIK-AGENT, is developed to implement the idea. The procedure of the message generation and solution method selection is illustrated with the case of a computer purchase.
1. Introduction

In the Internet based electronic market, a customer can afford to contact many vendors worldwide to buy even small pieces of products owing to the lower-than-ever search and transaction costs. However, this convenience will push up the number of transactions astronomically, resulting in the manual handling of transactions uneconomical or physically impossible. So we need to automate the product search, selection, and contract process as much as possible using intelligent customer and vendor agents to reduce the degree of human’s effort. To automate communication between the agents, our first task is to define the contract type, which determines the protocol of purchase and sales.

In the electronic market, we can define new contract types which are more diverse and flexible than traditional ones. A contract type can be identified with a set of variables and their corresponding values, and the new contract types applicable in electronic commerce are identified in section 3. Next, we have to design the intelligent agent which can aid the customer and vendor respectively according to their respective selected contract type.

In this study, we propose an architecture of intelligent agent named UNIK-AGENT (Lee, 1995; Lee and Lee, 1995; Lee and Lee, 1997; Lee, et. al., 1994) which is an extension of an expert system with additional capabilities of communication control and meta-problem-solving. To fulfill the communication among the customer and vendor agents, we propose three message layers: Agent Communication Language Layer, Electronic Commerce Layer, and Product Specification Layer. In this paper, the process for a contract type is illustrated showing the contingent messages and problem solving methods selected.
The remainder of this paper is organized as follows; Section 2 reviews the related literature. In Section 3, we define the new contract types in electronic commerce. In Section 4, an architecture of intelligent agent named UNIK-AGENT is suggested. In Section 5, the format and types of messages during the contract process is proposed in three layers. In Section 6, the intelligent agent based contract process is illustrated with an example. Finally, Section 7 concludes with the summary of contributions.

2. Literature Review

2.1 Intelligent Agent

Researchers in agent technology have offered a variety of definitions. These definitions range from the simple to the lengthy and demanding. The reason for such a variety of definitions is that each of them grew out of the exemplar agents that the definer had in mind (Franklin and Graesser, 1996). Therefore, there is no single definition of an agent or intelligent agent. However, the widely accepted attributes of agents are autonomy, communication, reasoning, and learning capability as described in Table 1. Other typical attributes include mobility and reactivity (Nissen, 1996; Franklin and Graesser, 1996; Foner, 1996; Belgrav, 1996).

Insert Table 1 Here

In this paper, we regard intelligent agent as an extension of the expert system with additional communication and meta-problem-solving capabilities (Lee, 1995; Lee and Lee, 1995; Lee and Lee, 1997; Lee, et. al., 1994).

2.2 Intelligent Agents in Electronic Commerce
Electronic commerce is mercantile transactions (for example, buying and selling information, products and services) via computer networks. Broadly speaking it is a new way of conducting, managing and executing business transactions using computer and telecommunication networks.

One of the most important applications of intelligent agents in electronic commerce is comparison shopping. In comparison shopping, agents assist in searching for product items on the Internet on behalf of a customer. A customer interacts with his/her agent by submitting a set of requests. Upon receiving the request, the agent searches the relevant online shops throughout the Internet seeking for the items that match the search criteria. The agent returns with a detailed description and the price about the sought items and addresses of the virtual stores that deal with the items. The agent may format the information for effective comparison, for instance, by sorting according to the price order (Aoun, 1996). One of the earliest such agent is BargainFinder developed by Andersen Consulting (Krulwich, 1996), which searches multiple CD retailers. Other prototypical comparison shopping agents are Bargainbot and ShopBot as listed in Table 2.

Insert Table 2 Here

However, the current agents cannot have a full fledged interpretation capability because the vendors’ products information is described in unstructured HTML files in a natural language. This implies the product specification and price information need to be more structured until the natural language processor with a common sense reasoning capability becomes a reliable technology.

The intelligent agents idea suggested - but not yet implemented - by the Agents Cooperation has some reasoning capabilities specifically for electronic commerce (Agent Cooperation, 1996). The
Decision Agent and Demand Agent simulates the role of the customer and vendor respectively. These two types of agents are equipped with learning capability. For example, the Demand Agent predicts the future demand by analyzing the history of sold products, and the Decision Agent recommends appropriate products based on the learned preferences.

2.3 Agent Communication Language (ACL)

In order to communicate with other agents, an agent should generate messages and interpret the received ones. For this purpose, the formal and common representation of messages, a so-called agent communication language (ACL), is necessary. ACL consists of two parts: outer language and inner language (Genesereth and Ketchpel, 1994). In this paper, we adopt the Knowledge Query and Manipulation Language (KQML) as the outer language and UNIK-OBJECT (an object-oriented tool) as the inner language (Lee, at. el., 1996).

KQML was developed as one of the principal outcomes of the DARPA Knowledge Share Effort, and is performed by the External Interface Working Group. KQML displays communicating attitude about information such as querying, stating, believing, requiring, achieving, subscribing, and offering (Finin, at. el., 1996). Since KQML is indifferent to the content of the information per se, it often employs another inner language (or content language) like KIF or UNIK-OBJECT.

ACL is devised for communication among agents independent of domain. Therefore, to apply ACL specifically to the electronic commerce domain, we should additionally define an inner language that can suitably express electronic commerce. To implement this idea, we adopt three message layers: the ACL Layer (a domain independent outer language), Electronic Commerce Layer (terms necessary for electronic commerce), and Product Specification Layer as depicted in Figure 3 (Lee and Lee, 1997).
3. Contract Types in Electronic Commerce

Since the contract procedure is critical in determining the communication between agents (customer agents and vendor agents) for electronic commerce, let us review the traditional competitive contract procedure and design new types of contracts that are possible in electronic commerce.

3.1 Competitive Contract Process in Traditional Commerce

The process of the traditional bid for purchase is composed of four steps: announcement of the invitation for bids, estimation of price, bidding, and selection of a successful bidder.

1) Announcement of the Invitation for Bids: The invitation for a bid should be announced to the potential bidders. The invitation for a bid includes items like required specification, quantity, bidding date, etc.

2) Estimation of price: The estimate price that will be used as a criterion for the selection of a successful bidder should be determined. The written estimated price should be kept secret, and never revealed in advance.

3) Bidding: Bidders are gathered in the same place at the same time and submit their own bid prices in a sealed envelope.

4) Selection of a Successful Bidder: Open the sealed bid prices, and select the one with the minimal price lower than the estimated price. If there is no bid price lower than the estimated price, the bidding process is repeated again until a bid price lower than the estimated price can be found within a certain iteration (Kim, 94; Park, 90).
In this bidding process, there can be some variations in selecting the potential bidders. In general, registration to bid is open to any vendor. However, bidders may be limited to pre-nominated ones. This pre-nomination is allowed if the amount of purchase is relatively small; if there are few vendors available; and if the product to be ordered is so special that the bidders may be limited to only those who can satisfy such criteria. The other modification may happen if the bid is for on-going products with relatively cheap prices. In this case, the bidding price may not be a total amount, but a unit price.

An auction is another type of competitive contract with a single seller and multiple potential bidding customers. In an English auction, the current bid price is open to all other bidders, and the bid price is upgraded until no higher bid is proposed. The last and highest price is the successful bid price (Kim, 1994). In a Dutch auction, however, instead of the bid prices proposed by the bidders, the seller begins with an announced initial maximal price and gradually lowers the price until a bidder takes the offer (Lee, et. al., 1995). The procedure of each contract type should be strictly kept once the type is announced to the public. The language for electronic commerce should be able to express the entire contract procedure.

3.2 Creative Contract Types in Electronic Commerce

New contract types can be created in the electronic commerce because the execution cost of sophisticated contracts can be economically justified. So we attempt to represent the contracts in grammatical form which can represent not only the traditional contracts like bidding and auction, but also new types that did not formally exist in the traditional commerce yet. To do so, we suggest variables and their values that describe contract types. These variables and values are represented by the Backus-Naur Form (BNF) in Figure 1 (Lee and Lee, 1997).
We can observe that the traditional bidding and auction are two special cases, and several variations can be adopted for the creative contracts. For instance, a customer may want to announce the successful bid price of the first round and switch to the second round of auctioning in search of a even lower price. In this case, the bidding price in the first round by the vendor agent may be different from the price for the one round cut-off bidding. Since such variations will influence the messages and problem solving methods as such, the agents should be sensitive to the contract types and situation.

4. Architecture of UNIK-AGENT

Now let us design a tool for an Agent Based Commerce Development Environment (ABCDE). As mentioned earlier, UNIK-AGENT implement ABCDE regarding agents as an extension of expert systems.

4.1 Overview of UNIK-AGENT

Overall, UNIK-AGENT is composed of a problem solver and communication controller. In contrast to the traditional expert systems that have an inference (solution) engine, UNIK-AGENT has a problem solving manager that selects the appropriate solution method over multiple solution engines as depicted in Figure 2 (Lee and Lee, 1997; Lee and Lee, 1995; Lee, et. al., 1994).
The Problem Solver has two components: Problem Solving Manager (which selects an appropriate solution method for a specific problem) and Solution Engines (various inference engines or algorithms) because an agent may have more than one solution method.

Problem Solving Manager has two tasks. First, it controls the whole contract process. Second, it selects an adequate solution method for the messages received, and decides what to tell to fulfill the agent's goal. For instance, the Problem Solving Manager of the Customer agent decides whether to generate an adequate message or invoke the bidding decision algorithm depending upon the context.

Solution engines are collection of problem solving methods like forward chaining, case-based reasoning, filtering out dominated alternatives, a neural network based estimation, scheduling, and message generation.

4.3 Communication Controller

There are five layers in the Communication Controller.

1) Message Manager Layer:

ACL Layer in the messages is interpreted and validated in this layer. That is, performative and its parameters are validated.

2) Directory Consulting Layer

For outgoing messages, this layer consults a directory agent in the network about the appropriate agents address. The cash memory may be consulted instead of external directory agent if the internal memory exists and does not have to be refreshed. For the maintenance of consistency between internal cash memory and external agent’s memory, we need to adopt the agent consistency protocol (Lee and Kim, 1996).

3) Individual Message Layer:
For outgoing messages, this layer breaks down the broadcasting messages into individual messages.

4) Message Queue Management Layer:

This layer handles the incoming and outgoing message queues, rejects overdued proposals, and initiates problem solving procedure.

5) Message Gate Layer:

This layer transmits the messages following the protocol like TCP/IP.

5. Message Layers in Electronic commerce

As mentioned earlier, messages for electronic commerce consist of three layers as depicted in Figure 3. These layers are above the TCP/IP layer. The global standard of messages is critical to make the agent based commerce compatible world wide.

The top layer consists of a KQML performative and its parameters which are the domain independent messages as reviewed in Section 2.3. Let's call this layer ACL layer.

The middle layer named Electronic Commerce layer adds details to the KQML content with specific parameters necessary for electronic commerce. For instance, TITLE corresponds to the message name and CONTRACT_ID an identification code of a contract. The messages are constructed by combination of TITLE, CONTRACT_ID, and the others. As depicted in Figure 3, REQUIREMENT, PROPOSAL, and SELECTED_PROPOSAL commonly have their own subparameters which are related with the content of a contract such as PRODUCT,
DELIVERY_METHOD, DELIVERY_DATE, and PAYMENT_METHOD. REQUIREMENT is necessary for the customer to specify his/her requirement; PROPOSAL for vendor to submit a proposal; and SELECTED_PROPOSAL for customer to inform the selected proposal to the successful bidder.

The bottom layer named the Product Specification layer defines the specification of items more specifically by product types. This layer consists of the specification of items to be contracted. In this paper, we illustrate the Product Specification Layer with the personal computer.

6. Message Generation and Solution Procedure

In this section, let us design messages and solution methods necessary for an illustrative contract process. The contract type adopted for an illustration is a buyer centered bidding. The relevant message and solution method can be identified by the situation-action type forward chaining rule based system. In UNIK-AGENT, the Problem Solving Manager has adopted the forward chaining tool UNIK-FWD (Lee, at. el., 1996). The overall process of message generation among agents is depicted in Figure 4.

Insert Figure 4 here

6.1 Initiation of the Contract Process by the Customer Agent

A contract can be initiated as a customer requests his/her requirements to the Customer agent. This step consists of three sub-steps conducted by the Problem Solving Manager.
(1) The identification code of a contract is generated. This identification code is specific to the customer.

(2) A human customer determines a contract type for a particular purchase. The knowledge for this contract type is used for the activation rules in Problem Solving Manager both in the Customer agent and the Vendor agents. Since the identification of contract type requires the selection of values for many variables, it may not be an easy task to novice users. So several popular contract types may be identified in advance and saved so that customers can easily select from them. For instance, four buyer centered contract types may be prespecified: Normal Bid, Prenominated Bid, Unit Price Bid, and Negotiable Bid. These four contract types may be saved by the customer’s choice. The Normal Bid is the same as the traditional bidding type, and Prenominated Bid is the same as the Normal Bid with the exception of prenominated bidders. In this example, suppose the human customer has selected the Normal Bid modified to allow human involvement. Therefore, the contract type in this example is identified as the following object.

```{Contract_Type_960924
    is-a: Contract_960924
    number_of_bid_round: 1
    competitor_price_referable: not_referable
    announced_estimated_price_limit: enforced
    number_of_proposals_for_each_bidder: 1
    bid_price_change: not_allowed
    bid_price_open_time: at_predetermined_time
    buyer_bidder_prenomination: prenominated_by_customer
    human_involvement: enforced
    bidding_price_type: total_amount
    buyer_rule_for_selection_of_successful_bidder: min_price
    buyer_nego_between_price_&_spec: not_allowed}```
The contract type does not mention the products at all. So this information along with other requirements like bidding time, delivery method, payment method, delivery date, item name, quantity, and price belongs to Electronic Commerce layer.

(3) As the next step, the requirement for the products to be purchased should be identified. The products may be specified differently depending upon whether it is specified by a vendor or customer. Vendors may share a complete standardized product specification (as illustrated in section 6.4.2). However, the customer may not request with a full specification. For instance in the following example, only three factors of main memory, hard disk capacity and processor are identified.

\[
\text{ProductSpecification}_960924
\begin{align*}
\text{is-a:} & \quad \text{Contract}_960924 \\
\text{Main\_Memory:} & \quad (\geq 840\text{MB}) \\
\text{Hard\_Disk:} & \quad (\geq 16\text{MB}) \\
\text{Processor:} & \quad (\geq \text{Pentium90})
\end{align*}
\]

Specifying the customer’s requirement based on product specification is a basic way of representation. However, customers may want to express at a higher level like “a personal computer for a high school student having word processor and Internet access”. This is exactly the process that the expert systems should support to transform to the basic level of product specification. So the customer agent needs either to be a knowledge intensive system or to consult an intelligent products catalog.

6.2 Construction of the RFP by the Customer Agent
The Message Manager constructs the message Request_for_Proposal (RFP) in three layered format based on the customer’s request. In the ACL layer, the performative “evaluate” is selected to request the proposals with the parameters: \textit{sender}, \textit{receiver}, and \textit{reply-with}. We may use the performatives in KQML with the interpretation on electronic commerce, or define new performatives that precisely express the meaning for electronic commerce. At this stage, we adopt the former approach. An identification code of the Customer agent is the value of \textit{sender}. The values of \textit{receiver} are dependent on the contract type. If the customer wants to prenominate the bidders, s/he may input the bidders referring to the directory about the vendors. Otherwise, all the vendors in knowledge base will be automatically selected. The completed RFP form is as depicted in Figure 5.

\textit{Insert Figure 5 Here}

6.3 Initialization of the Vendor Agents

Upon receiving the RFP from the Customer agent, the Message Manager in the Vendor agent passes the RFP to Problem Solving Manager on time (upon arrival or at a predetermined time). The Problem Solving Manager interprets the \textit{TITLE} in \textit{CONTENT}, and identifies that it is an RFP. So the contract process is triggered by the following rule.

\textit{Contract Process Initiation Rule}

\textit{IF } ReceivedMessage.TITLE IS RFP
\textit{THEN } (INITIALIZE ContractProcess)

6.4 Selection of Adequate Products by the Vendor Agents
6.4.1. Selection of a Solution Engine. In this example, since buyer’s negotiation between price and specification is not allowed, product selection method adopted should be the following \textit{Exact Matching} rule.

\textbf{Exact Matching Rule}

\begin{align*}
\text{IF } & \text{ buyer\_nego\_between\_price\_&\_spec IS Not\_Allowed} \\
\text{THEN } & \text{ Product\_Selection.Solving\_Method IS Exact\_Matching}
\end{align*}

Exact matching is nothing more than retrieving the products that precisely satisfy the customer’s requirement (main memory, hard disk, processor, and price in this example) from the product database in the Vendor agent.

However, if the negotiation between price and specification is allowed, the matching does not have to be exact. Therefore, the case-based retrieval algorithm with an incomplete customer’s requirement may be used as the following \textit{Case Based Reasoning Rule}.

\textbf{Case Based Reasoning Rule}

\begin{align*}
\text{IF } & \text{ buyer\_nego\_between\_price\_&\_spec IS Allowed} \\
\text{THEN } & \text{ Product\_Selection.Solving\_Method IS Case\_Based\_Reasoning}
\end{align*}

A point we need to note is the fact that object oriented product database is appropriate for agent’s flexible and efficient retrieval of relevant products. Hypertext is not efficient until the natural language becomes perfect without significant overhead. Nevertheless, the hypertext may be stored in objects for human’s comprehension.

6.4.2. Selection of Product. The retrieved products may be more than one. Then the Vendor agent has to decide the rule of selecting the proposed products. The rule may select ‘all retrieved
products’ or ‘the products with the minimum price’ depending upon the number of proposals allowed to each vendor. The rule may also consider the bidding method in a particular contract type. For instance, if the bidding rule is the selection of the minimum price as far as the functional specification is satisfied, the minimum priced proposal will be the best strategy. The demerit with the blind minimum price strategy is that the price factor can be overemphasized without considering the quality, optional functionality, expandability, after service, delivery date, and credit of vendor. This negative effect may be reduced partially, however, by employing rich specification capability of products. The other way is to get the customer involved in negotiating the price with specification supporting the impact analysis for trade-offs. This is particularly necessary for high quality and high price products. To help in balancing functionality and price, the Constraints and Rule Satisfaction Problem (CRSP) approach with interactive reasoning capability is an effective method (Lee and Kwon 1993). Suppose in this example, a product is selected as follows.

\[
\begin{align*}
\text{(Amount} & \quad (= 2000)) \\
\text{Specification} & \\
\text{IS-A:} & \quad Contract_960924 \\
\text{Main_Memory:} & \quad (= 64MB) \\
\text{Hard_Disk:} & \quad (= 1GB) \\
\text{Processor:} & \quad (= PENTIUM100) \\
\text{Monitor_Size:} & \quad (= 20INCH) \\
\text{CD_ROM:} & \quad (= 8X) \\
\text{Sound_Card:} & \quad (= 32K))
\end{align*}
\]

6.5 Construction of Proposal by the Vendor Agents

The constructed proposals as in Figure 6 are sent through Message Gate on TCP/IP layer. The proposal message has to be constructed with the proposing product information. This procedure is basically the same as the one for the construction of the RFP in the Customer agent.
6.6 Interpretation of Proposals by the Customer Agent

6.6.1. Interpretation of ACL Layer of Proposals. Since the performative of the proposal is 'reply' which is a response type, the Message Manager of the Customer agent confirms the value of 'in-reply-to' with the Message IDs in the message base. If there exits a matching Message ID, the message is valid. Otherwise, the message is invalid and will be ignored in this step.

6.6.2. Filtering Proposals. The valid message is passed to the Problem Solving Manager. In this step, messages are validated again with regard to the title, contract identification code, authorized bidder, and bid time restriction according to the following rule:

Valid Proposals Rule

\[
\text{IF} \quad \text{Received\_Message.Title IS Proposal} \\
\text{AND} \quad \text{Received\_Message.Contract\_id IS IN Contracts} \\
\text{AND} \quad \text{Received\_Message.Time } \geq \text{ Bid\_Begin\_Time} \\
\text{AND} \quad \text{Received\_Message.Time } \leq \text{ Bid\_End\_Time} \\
\text{AND} \quad \text{Received\_Message.Sender IS IN Contract\_Bidders} \\
\text{THEN CALL (ADD Received\_Message Candidates)}
\]

If a proposal passes this validation process, such a proposal becomes a candidate in which valid proposals are collected. In this example, there are five candidates as shown in Figure 7.

Insert Figure 7 Here

6.7 Selection of Successful Bid by the Customer Agent
At the bidding time, an alternative is selected out of the candidates. In this example, although R5's bidding price is the minimum in candidates, it does not satisfy the hard disk requirement, so it is dropped. Therefore, R3 is selected because its bid price is the minimum in feasible candidates set.

6.8 Stop or Next Round Decision

The last step of each round should be checked whether the entire procedure is finished or the next round should be invoked. This decision depends upon the number of successful bidders: a single alternative, multiple non-dominated alternatives, or no alternative. The following rules demonstrate the “stop” or “go to the next round” decision.

(In case of a Single Alternative)

**Select the alternative as a successful bid**

\[
\text{IF } \text{(NUMBER Selected Alternative)} = 1 \\
\text{AND Contract Type.Number of Bid Rounds} = \text{Current Round} \\
\text{AND Contract Type.Human Involvement IS Not Forced} \\
\text{THEN Selected Alternative IS Final Alternative}
\]

**Go to the next round with the single alternative**

\[
\text{IF } \text{(NUMBER Selected Alternative)} = 1 \\
\text{AND Contract Type.Number of Bid Rounds} > \text{Current Round} \\
\text{AND Contract Type.Human Involvement IS Not Forced} \\
\text{THEN BID IS Next Round}
\]

**Ask whether the customer would like to select the current single alternative**

\[
\text{IF } \text{(NUMBER Selected Alternative)} = 1 \\
\text{AND Contract Type.Human Involvement IS Forced} \\
\text{THEN (HUMAN SELECTION Selected Alternatives)}
\]

(In case of Non-dominated Alternatives)

**Let human select among the non-dominated alternatives**
IF (NUMBER Selected_Alternative)  >  1 
THEN (HUMAN_SELECTION Selected_Alternatives)

(In case of Case of No Alternative )

There is no satisfied alternative, and so terminate the bidding process
IF (NUMBER Selected_Alternative)  =  0
AND ContractType.Number_of_Bid_Rounds  =  <Current Round>
THEN BID IS Failed

There is no satisfied alternative, so go to the next round seeking other proposals
IF (NUMBER Selected_Alternative)  =  0
AND Contract_Type.Number_of_Bid_Rounds  >  <Current Round>
THEN BID IS Next_Round

In this example, since human involvement is enforced and the allowed number of round is only one, proposal R3 is selected, and is shown to a human customer for final approval, possibly in comparison with other candidates. If a human customer is satisfied with R3’s proposal, this bid is successful. The bid result message depicted in Figure 8 should be sent to each Vendor agent.

Insert Figure 8 Here

7. Conclusions

Standardization of messages and the agent's architecture are very important for the global agent based commerce as the second generation of electronic commerce. Toward this end, this research has contributed in four aspects.

First, we have proposed the structure of intelligent agent as an extension of expert systems by adding the communication control and meta-problem-solving capability. A prototype UNIK-
AGENT is designed with this idea, and is implemented using Visual C++ 4.0, UNIK-OBJECT, and UNIK-FWD that run on Window95.

Second, contract types usable in electronic commerce are identified in a formal grammar. This grammar can represent not only the traditional contracts like bidding and auction, but also new types that may be useful in electronic commerce owing to the inexpensive transaction cost.

Third, three layered messages (ACL Layer, Electronic Commerce Layer, and Product Specification Layer) for the communication among agents for electronic commerce are proposed with illustration.

Fourth, the message generation and solution selection procedure for the agent based contract is proposed and illustrated with an example.
References


Park, W., Modern Purchasing Management, Parkyoung Press, 1990 (in Korean)

<table>
<thead>
<tr>
<th>Attributes</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Autonomy</td>
<td>Take actions without triggering by the end-user</td>
</tr>
<tr>
<td>Communication Ability</td>
<td>Communicate with other agents</td>
</tr>
<tr>
<td>Reasoning Capability</td>
<td>Infer based on the current knowledge and experience</td>
</tr>
<tr>
<td>Learning Capability</td>
<td>Learn from past experiences</td>
</tr>
</tbody>
</table>

Table 1 Common Attributes of Agents
<table>
<thead>
<tr>
<th>Agent</th>
<th>Developer</th>
<th>Item</th>
<th>URL</th>
</tr>
</thead>
<tbody>
<tr>
<td>BargainFinder</td>
<td>Anderson Consulting</td>
<td>CDs</td>
<td><a href="http://bf.cstar.ac.com/bf">http://bf.cstar.ac.com/bf</a></td>
</tr>
<tr>
<td>Bargainbot</td>
<td>Curtin University</td>
<td>Books</td>
<td><a href="http://www.ece.curtin.edu.au/~saunb/bargainbot">http://www.ece.curtin.edu.au/~saunb/bargainbot</a></td>
</tr>
<tr>
<td>ShopBot</td>
<td>Univ. of Washington</td>
<td>CDs and Softwares</td>
<td><a href="http://www.cs.washington.edu/homes/bobd/shopbot.html">http://www.cs.washington.edu/homes/bobd/shopbot.html</a></td>
</tr>
</tbody>
</table>

**Table 2 Comparison Shopping Agents**
<contract_type> := <buyer_centered_contract> | <seller_centered_contract>


<seller_centered_contract> := <auction_type> <number_of_bidding_rounds> <competitor_price_referable> <announcer_estimated_price_limit> <number_of_proposals_for_each_bidder> <bid_price_changeable> <bid_price_open_time> <seller_bidder_prenomination> <bidding_price_type> <seller_rule_for_selection_of_successful_bidder> <seller_negotiation_between_price_&_spec>

<auction_type> := English_auction | Dutch_auction

<number_of_bidding_rounds> := <a certain number> | unlimited.

<competitor_price_referable> := referable | not_referable

<announcer_estimated_price_limit> := enforced | just_referred

<number_of_proposals_for_each_bidder> := <a certain number> | unlimited

<bid_price_change> := not_allowed | allowed_once | allowedCertain_times | allowed_unlimited

<bid_price_open_time> := at_a_predetermined_time | upon_received

<buyer_bidder_prenomination> := unlimited | prenominated_by_buyer | prenominated_by_criteria | prenominated_by_registration

<seller_bidder_prenomination> := unlimited | prenominated_by_seller | prenominated_by_criteria | prenominated_by_registration

<human_involvement> := enforced | not_enforced

<bidding_price_type> := total_amount | unit_price

<buyer_rule_for_selection_of_successful_bidder> := min_price | min_price_below_estimated_price | max_price_below_estimated_price | proprietary_rule

<buyer_negotiation_between_price_&_spec> := not_allowed | allowed_once | allowedCertain_number | allowed_unlimited

<seller_rule_for_selection_of_successful_bidder> := max_price | proprietary_rule

<seller_negotiation_between_price_&_spec> := not_allowed | allowed_once | allowedCertain_number | allowed_unlimited

---

Figure 1 Contract Type Grammar in Backus-Naur Form (BNF)
Figure 2 Architecture of UNIK-AGENT
Figure 3 Three Message Layers
Figure 4 Overall Contract Process in Electronic Commerce
Figure 5 An Illustrative Request for Proposal
Figure-6 An Illustrative Proposal

```xml
<REPLY>
  <SENDER> R3 </SENDER>
  <RECEIVER> CB </RECEIVER>
  <IN-REPLY-TO> 96A0924 </IN-REPLY-TO>
  <ONTODGUE> Agent Based Commerce </ONTODGUE>
  <LANGUAGE> UNIX-OBJECT </LANGUAGE>
  <CONTENT>
    <TITLE> PROPOSAL </TITLE>
    <PROPOSAL>
      <PRODUCTS>
        <ITEM_NAME> PC </ITEM_NAME>
        <COMMODITY> 1 </COMMODITY>
        <AMOUNT> 2000 </AMOUNT>
        <SPECIFICATIONS>
          <MAIN_MEMORY> 32MB </MAIN_MEMORY>
          <HARD_DISK> 1GB </HARD_DISK>
          <PROCESSOR> PENTIUM100 </PROCESSOR>
          <MONITOR_SIZE> 17INCH </MONITOR_SIZE>
          <CD_ROM> 8X </CD_ROM>
          <SOUND_CARD> 32K </SOUND_CARD>
        </SPECIFICATIONS>
      </PRODUCTS>
    </PROPOSAL>
  </CONTENT>
</REPLY>
```
<table>
<thead>
<tr>
<th>Proposal</th>
<th>Specifications</th>
<th>Amount</th>
</tr>
</thead>
<tbody>
<tr>
<td>R1</td>
<td>(MAIN_MEMORY = 64MB)</td>
<td>2250</td>
</tr>
<tr>
<td></td>
<td>(HARD_DISK = 1GB)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(PROCESSOR = PENTIUM100)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(MONITOR_SIZE = 20inch)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(CD_ROM = 8X)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(SOUND_CARD = 32K)</td>
<td></td>
</tr>
<tr>
<td>R2</td>
<td>(MAIN_MEMORY = 64MB)</td>
<td>2125</td>
</tr>
<tr>
<td></td>
<td>(HARD_DISK = 840MB)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(PROCESSOR = PENTIUM90)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(MONITOR_SIZE = 20inch)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(CD_ROM = 8X)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(SOUND_CARD = 32K)</td>
<td></td>
</tr>
<tr>
<td>R3</td>
<td>(MAIN_MEMORY = 32MB)</td>
<td>2000</td>
</tr>
<tr>
<td></td>
<td>(HARD_DISK = 1GB)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(PROCESSOR = PENTIUM100)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(MONITOR_SIZE = 17inch)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(CD_ROM = 8X)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(SOUND_CARD = 32K)</td>
<td></td>
</tr>
<tr>
<td>R4</td>
<td>(MAIN_MEMORY = 64MB)</td>
<td>2375</td>
</tr>
<tr>
<td></td>
<td>(HARD_DISK = 1GB)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(PROCESSOR = PENTIUM120)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(MONITOR_SIZE = 20inch)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(CD_ROM = 8X)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(SOUND_CARD = 32K)</td>
<td></td>
</tr>
<tr>
<td>R5</td>
<td>(MAIN_MEMORY = 32MB)</td>
<td>1875</td>
</tr>
<tr>
<td></td>
<td>(HARD_DISK = 420MB)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(PROCESSOR = PENTIUM100)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(MONITOR_SIZE = 17inch)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(CD_ROM = 4X)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(SOUND_CARD = 16K)</td>
<td></td>
</tr>
</tbody>
</table>

Figure 7 Illustrative Candidates
Figure 8 An Illustrative Bid Result