Agent-Based Negotiation in E-Marketing

V.K. Murthy
University of New South Wales, Australia

E.V. Krishnamurthy
Australian National University, Australia

INTRODUCTION

This article describes in brief the design of agent-based negotiation system in e-marketing. Such a negotiation scheme requires the construction of a suitable set of rules, called protocol, among the participating agents. The construction of the protocol is carried out in two stages: first expressing a program into an object-based rule system and then converting the rule applications into a set of agent-based transactions on a database of active objects represented using high-level data structures.

BACKGROUND

An agent is a code-containing object, that along with data and execution context can migrate autonomously and purposefully within a computer network. Thus an agent knows what to do with the information obtained from its environment. Agents behave like actors and have intentions and actions. In addition, agents are flexible, proactive and have multithreaded control. In this overview, we describe in detail how a set of agents can be used for negotiation in e-marketing. For this purpose we need to have a model of the multi agent-based paradigm for executing the negotiation process analogous to what we humans do. Negotiation is an interactive process among a number of agents that results in varying degrees of cooperation, competition and ultimately to commitment that leads to a total agreement, consensus or a disagreement. It has many applications, including economics, psychology, sociology and computer science.

MAIN THRUST OF THE ARTICLE

The following subsections bring out the main thrust of this chapter, namely: what is a multi-agent system, what is planning, reasoning and negotiation, and how agents can be useful in modeling e-market and e-auction. Also we will briefly describe how a coalition among agents can cause a speculation bubble or a crash in e-share market.

A MULTI-AGENT SYSTEM

A simple model of an agent that is suitable for our purpose is shown in Figure 1. This is a unified model based on several important contributions made by the following authors: (Chen & Dayal, 2000; Dignum & Sierra, 2001; Fisher, 1995; Genesereth & Nilsson, 1987; Ishida, 1994; Murthy, 2002; Woolridge, 2002).

As shown in Figure 1, an agent consists of the following subsystems:

(1) Worldly states or environment U: Those states which completely describe the universe containing all the agents.

(2) Percept: Depending upon the sensory capabilities (input interface to the universe or environment), an agent can partition U into a standard set of messages T, using a sensory function Perception (PERCEPT): PERCEPT :U $\rightarrow$ T.
PERCEPT can involve various types of perception: see, read, hear, smell. The messages are assumed to be of standard types based on an interaction language that is interpreted identically by all agents.

(3) Epistemic states or Mind M: Weassumethattheagent has a mind M (that is essentially a problem domain knowledge consisting of an internal database for the problem domain data and a set of problem domain rules) that can be clearly understood by the agent without involving any sensory function. The database D sentences are in first order predicate calculus (also known as extensional database) and agents' mental actions are viewed as inferences arising from the associated rules that result in an intentional database, whcih changes (revises or updates) D.
The agent's state of belief, or a representation of an agent's state of belief at a certain time, is represented by an ordered pair of elements (D, P). D is a set of beliefs about objects, their attributes and relationships stored as an internal database and P is a set of rules expressed as preconditions and consequences (conditions and actions). When T is input, if the conditions given in the left-hand side of P match T, the elements from D that correspond to the right-hand side are taken

Figure 1.

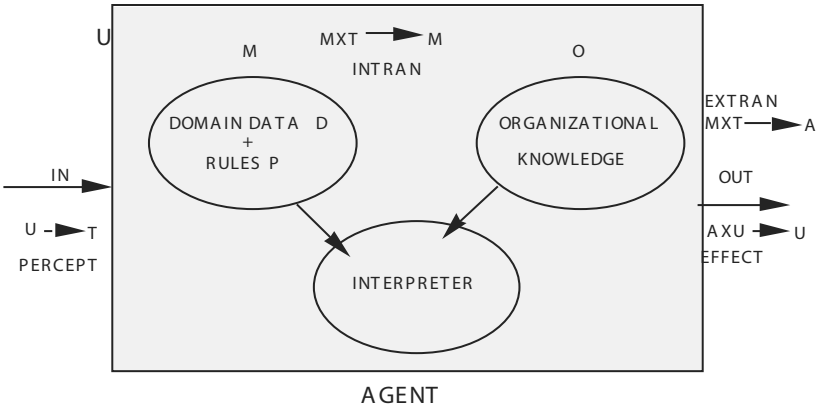

Figure 1

from D and suitable actions are carried out locally (in M) as well as on the environment.

(4) Organizational Knowledge (O): Since each agent needs to communicate with the external world or other agents, we assume that O contains all the information about the relationships among the different agents. For example, the connectivity relationship for communication, the data dependencies between agents, and interference among agents with respect to rules. Information about the location of different domain rules is in O.

(5) INTRAN: M is suitably revised or updated by the function called internal transaction (INTRAN).

Revision: Revision means acquisition of new information about the environment that requires a change in the rule system P. This may result in changes in the database D.

Example: The inclusion of a new tax rule in the tax system.

Update: Update means adding new entries to the database D; the rules P are not changed.

Example: Inclusion of a new tax-payer in the tax system.

Both revision and update can be denoted in set-theoretic notation by: INTRAN: M X T $\rightarrow$ M

(6) EXTRAN: External action is defined through a function called global or external transaction (EXTRAN) that maps an epistemic state and a partition from an external state into an action performed by the agent. That is: EXTRAN: M X T $\rightarrow$ A
This means that the current state of mind and a new input activates an external action from A.

(7) EFFECT: The agent also can affect U by performing an action from a set of actions A (ask, tell, hear, read, write, speak, send, smell, taste, receive, silent), or more complex actions. Such actions are carried out according to a particular agent's role and governed by an etiquette called protocols. The effect of these actions is defined by a function EFFECT, which modifies the world states through the actions of an agent:

EFFECT: A X U $\rightarrow$ U; EFFECT can involve additions, deletions and modifications to U.

Thus an agent is defined by a set of nine entities, called a 9-tuple:

(U,T,M(P,D),O,A,PERCEPT,INTRAN,EXTRAN,EFFECT).

The interpreter repeatedly executes selected rules in P, until no rule can be fired.

We can interpret all the abstract machine models (such as a finite state machine or a Turing machine) and parallel computational models (such as classifier systems) as subclasses of the agents, by suitably formulating the definitions.

The nature of internal production rules P, their mode of application, and the action set A determines whether an agent is deterministic, nondeterministic, probabilistic or fuzzy. Rule application policy in a production system P can be modified by:

(1) Assigning probabilities/fuzziness for applying the rule

(2) Assigning strength to each rule by using a measure of its past success

(3) Introducing a support for each rule by using a measure of its likely relevance to the current situation

The preceding three factors provide for competition and cooperation among the different rules. Such a model is useful for negotiation in learning, as well as in e-marketing that involves interactions between many agents.