Category: Applications of E-Technologies

Intelligent Product Brokering and Preference Tracking Services

Sheng-Uei Guan
National University of Singapore, Singapore

BACKGROUND

It has been projected that electronic commerce conducted via mobile devices such as cellular phones and PDAs will become a whopping $25 billion market worldwide by 2006 (Frost & Sullivan, 2002). Some of the driving factors behind mobile commerce (m-commerce) have been attributed to the compactness and high-penetration rate of these mobile devices.

Despite all the hype and promises about m-commerce, however, several main issues will have to be resolved (Morris & Dickinson, 2001; Nwana & Ndumu, 1996, 1997). Clumsy user interfaces, cumbersome application, low speeds, flaky connections, and expensive services have soured many who have tried m-commerce, and security and privacy concerns have also dampened enthusiasms for m-commerce.

Taking these concerns into account, the developers will have to offer something unique. One of the potential killer applications for m-commerce could be intelligent programs that are able to search and retrieve a personalized set of products from the Internet for their users. These programs are called software agents. Agent-based e-commerce has emerged and software agents have demonstrated tremendous potential in conducting transactional tasks via the Internet.

According to the model from the Maes's group (Guttman & Maes, 1999), the consumer buying behavior (CBB) can be divided into six stages, namely, need identification, product brokering, merchant brokering, negotiation, payment and delivery, and product service and evaluation. Among these stages, product brokering plays an important role. It involves gathering product or service information, filtering information, recommending products, and so forth.

A user searching for a particular product on the Internet will normally have to use popular search engines and enter keywords that describe the product. These search engines will process these keywords and generate a large number of links for the user to visit. Neither the search engine nor the Web site knows the preference of the user and therefore might provide information that is irrelevant to the user. Hence, user preference tracking becomes one of the fundamental tasks of product-brokering agents.

Agents act on behalf of their users by carrying out delegated tasks automatically. A product-brokering agent will search for the products in the background with minimal user intervention, thereby allowing the user to concentrate on other aspects of the transaction, such as product purchase, bidding, negotiation, and so forth. The agent could be programmed with the user's preferences in mind and filter out irrelevant products automatically. The agent could also detect shifts in the user's interests and adjust accordingly to suit the user. Personalized product-brokering agents require a profile of the user in order to function effectively. The agent would also have to be responsive to changes in the user's interests and be able to search and extract relevant information from outside sources.

At MIT Media Labs, Maes and Sheth (Maes, 1994; Sheth & Maes, 1993) came up with a system to filter and retrieve a personalized set of USENET articles for a particular user by creating and evolving a population of information filtering agents using genetic algorithms (Holland, 1973). Genetic algorithms have been widely used in various applications (Chen & Shahabi, 2002; Farhoodi & Fingar, 1997a). Their superior and flexible performance has motivated their use in software agents.

Some keywords will be provided by the user that represents the user's interests. Weights are also assigned to each keyword, and the agents will use them to search and retrieve articles from the relevant newsgroups. After reading the articles, the user can either give a positive or negative feedback to the agents via a simple GUI. Positive feedback increases the fitness of the appropriate agents and also the weights of the relevant keywords (vice versa for negative feedback). In the background, the system periodically creates new generations of agents from the fitter species while eliminating the weaker ones. Initial results obtained from their experiments showed that the agents are capable of tracking its user's interests and recommend mostly relevant articles.

Whereas the researchers at MIT required the user to input their preferences into the system before a profile could be created, Crabtree and Soltysiak (1998) believed

that the user's profile could be generated automatically by monitoring the user's Web and e-mail habits, thereby reducing the need for user-supplied keywords.

Crabtree and Soltysiak's approach was to extract high-information-bearing words that occur frequently in the documents opened by the user. This is achieved by using a text summarizer that can generate a set of keywords to describe the document and can also determine the information value of each keyword. A clustering algorithm is then employed to help identify user's interests, and some heuristics are used to ensure that the program could perform as much of the classification of interest clusters as possible.

However, they (Crabtree & Soltysiak, 1998) have not been completely successful in their own experiments. The researchers admitted that it would be very difficult for the system to classify all the user's interests without the user's help. Nevertheless, they believed that their program has taken a step in the right direction by learning user's interest with minimal human intervention.

Widyantoro, Ioerger, and Yen (2003) proposed a method to keep track of changes in user interests from a document stream. Their method integrates a pseudorelevance feedback mechanism, an assumption about the persistence of user interests, and incremental method for data clustering. They claimed that their method significantly improves the performances of existing user-interest-tracking systems without requiring additional, actual relevance judgments. Kiss and Quinqueton (2001) also presented a machine learning method designed to predict preference knowledge in a multiagent context.

A new product-brokering agent usually does not have sufficient information to recommend any products to the user, and therefore it has to get product information from somewhere else. A good source of information will be the Internet. In order to do that, a method suggested by Pant and Menczer (2002) involves inplementing a population of Web crawlers called InfoSpiders that searches the World Wide Web on behalf of the user. Information on the Internet will be gathered based on the user's query and will then be indexed accordingly. These agents initially rely on traditional search engines to obtain a starting set of URLs that are relevant to the user's query. The agents will then visit these Web sites and decode their contents before deciding where to go next. The decoding process includes parsing the Web page, looking at a small set of words around each hyperlink, and giving a score based on their relevance to the user. The link with the highest score is then selected, and the agent visits the Web site.

DESCRIPTION OF INTELLIGENT PRODUCT-BROKERING AGENTS

Intelligent product-brokering agents, a new design and implementation of product-brokering agents, is briefly presented here (Guan, Ngoo, & Zhu, 2002, 2003). The intelligent ontology-based product-brokering agents implemented are capable of providing a personalized service for their users. They learn user preferences over time and recommend products that might interest the user. This work has two main features: One feature is that the product brokering is closely integrated with user preference tracking, and the other feature is that a genetic algorithm is used to adjust parameters to track user preferences.

Some assumptions have been made about the system, such as that the user is a rational person and will select a product rationally and that the value a user places on a product can be calculated, such as the price of product. Before the product brokering agent is able to explore the Internet and retrieve product information for the user, the agent needs to have some prior knowledge, such as the URL of some relevant Web sites, keywords, or some quantifiable attributes that can be used to describe the product. It could be tedious if the user has to enter such information into the agents when he or she wants to search for a particular product. An alternative to this is to create a product ontology, which involves defining the meaning of each term that is used to describe the product, their valid range of values, and their relationship with one another.

In this design, each agent will have an evaluation function that will be used to calculate the value of each product. Products that have a higher value will have a higher chance of been recommended by the agent. This evaluation function has some tunable parameters that characterize the user's preferences for a particular category of products.

Before recommending a product to the user, the agent should first be able to evaluate which product would best fit the user's requirements. A proposed method is to use some quantifiable attributes, such as performance, cost, and so forth, to evaluate the products. As an example, two weights—perf_weight (the weight of the performance of the product) and cost_weight (the weight allocated to the product cost)—represent the weights that the user could give to each attribute. These two parameters are actually used to represent the user's preferences and are incorporated inside the agent. If perf_weight has a higher value, it means that the user place more emphasis on the perfor-