Intelligent Software Agents for Electronic Commerce

Kristin Michele Tolle  
Management Information Systems Department  
University of Arizona  
Tucson, Arizona 85721  
(520) 621-3927  
kolle@bpa.arizona.edu

Hsinchun Chen  
Management Information Systems Department  
University of Arizona  
Tucson, Arizona 85721  
(520) 621-4153  
hchen@bpa.arizona.edu

Introduction
Electronic commerce (EC) and software agents are two of the hottest fields of research in information science. As the Internet is rapidly becoming a popular marketplace for consumers and sellers of goods and services, combining these two research areas offers lucrative opportunities both for businesses wishing to conduct transactions over the World Wide Web (WWW) and for developers of tools to facilitate this trend.

The focus in this chapter will be on software agents specifically designed for electronic commerce activities. We will briefly describe the history of agent research in general, defining characteristics of agents, and will touch on the different types of agents. Following this introduction we will describe the learning and action mechanisms that make it possible for agents to perform tasks. Finally, we will describe the issues associated with the deployment of electronic commerce agents (ECAs).

Readers interested in more detailed coverage of the topics described in this chapter are invited to investigate the following web sites and the documents they reference: University of Maryland’s AgentWeb site http://www.cs.umbc.edu/agents/ and MIT Media Lab’s Software Agent Group http://agents.www.media.mit.edu/groups/agents/.

What Are Software Agents?
It can be argued that there are nearly as many definitions of what an agent is as there are researchers and developers in the field—perhaps more. Some definitions give broad characterizations, while others carefully detail what their authors perceive to be the “best” definition of an agent. The examples that follow define a specific type of agent with which this chapter is concerned—those typically referred to as “intelligent” or “autonomous” software agents.

An IBM white paper found at http://www.networking.ibm.com/iag/iagwp1.html defines a software agent as: “An Intelligent agent is software that assists people and acts on their behalf. Intelligent agents work by allowing people to delegate work that they could have done, to the agent software. Agents can, just as assistants can, automate repetitive tasks, remember things you forgot, intelligently summarize complex data, learn from you, and even make recommendations to you.”

Hayes-Roth’s description found in [13] includes the temporal aspect in defining intelligent agents: “Intelligent agents continuously perform three functions: perception of dynamic conditions in the environment; action to affect conditions in the environment; and reasoning to interpret perceptions, solve problems, draw inferences, and determine actions.”

From these descriptions it is apparent that agents are assistive computer applications designed to perform a set of tasks for their users. Agents operate in a variety of ways and are employed in a diverse set of domains. Some can learn how to accomplish tasks for the user by interfacing directly, either through direct programming manually, while others learn by employing artificial intelligence (AI) techniques to observe users’ actions and attempt to replicate them. Some agents are stand-alone single applications designed for a specific purpose, while others are highly interactive with users and other agents to gain insight into how they accomplish their goal-seeking activities. The key characteristics and learning methods are described in more detail later in this chapter.

Agent History
The beginnings of agent research can be traced back to 1977 during the early studies in distributed artificial intelligence (DAI). Carl Hewitt described his concurrent actor model [15] as a computation agent which has a mail address and a behavior. Actors communicate with each other via message passing, carrying out their actions concurrently. In 1986, Marvin Minsky gave a sim-
ilar description in his book *The Society of Mind* where he described societies of goal seeking agents. Many of the questions he asked about how agents should interact, learn, and carry out tasks \[22\] later became a model for researchers in the creation of intelligent agents.

The study of intelligent agents began around 1990 when people such as Alan Kay \[16\] and Nicholas Negroponte \[23\] proposed that human computer interfaces needed to be drastically changed to evolve into devices which could interact with users in a more human-like manner. Since that time a great deal of the research in this area has been fostered by the MIT Media Lab, Cambridge, Massachusetts. Researchers at MIT are making great contributions to a field of study known as "behavior-based" agents. Behavior-based is a term used to describe an agent which attains most or all of the key characteristics of an intelligent, autonomous agent.

**Why are Agents Important?**

Agents are typically employed to perform tasks that are too difficult or mundane for their human counterparts \[13\]. Much of intelligent agent research is centered around the fact that more and more "untrained consumers" are using computers and interfacing with the World Wide Web (WWW) and thus require more efficient, easier, user-friendly ways to navigate and do business in this dynamic digital world \[8\]. This is especially true due to the myriad problems being caused by the increasing amount of online information available on the WWW \[5\]. Intelligent agents have the potential to hide the technical details while providing a high level of access to "newbies" or novice users. This potential capability is often a driving force behind the business strategies for the development of many of the electronic commerce shopping agents.

Software agents are big business. Many corporations are touting the inclusion of agent technology in their software applications. Large firms such as IBM, Microsoft, HP, Sun, Oracle, Apple, Lotus, and AT&T have either gone on record as researching agent usage or announced the inclusion of agents in their products. Most notably, Sun and IBM offer workbench tools that allow for easy incorporation of agents into Java applications. Academicians are also capitalizing on the popularity of software agents with such noted researchers as MIT Media Labs' Pattie Maes and University of Washington Computer Science professors Daniel Weld and Oren Etzioni have started venture-backed companies Agents, Inc. and NETbot respectively.

How big is the agent market? It has been predicted that in the next 5 to 10 years most new information technology development will be affected and many consumer products will contain embedded agent-based systems \[11\]. The estimated market for agent software and products for the year 2000 is 3.9 billion, in contrast to the estimated 476 million for 1995. Specifically related to electronic commerce, in 1996 more than half of US Internet users purchased merchandise online, and sales totaled 500 million US dollars \[12\]. The author of this article called these figures "much understated" because they failed to include users who consulted the Internet before purchasing offline.

**Characteristics of an Intelligent Agent**

Agent researchers often attribute very humanistic characteristics to intelligent agents. Key characteristics listed below are similar to those found in \[27\] as well as many other papers.

- **Autonomy:** An intelligent agent must be able to act independently of the user. It should sense changes in its perceived environment and be able to take necessary actions to complete the goals which have been collaboratively contracted between the agent and the user. To classified as autonomous an agent must be:
  - Goal-oriented: accept user requests and be responsible for deciding how and where to satisfy those requests.
  - Collaborative: interact with users to request clarification; may modify or refuse to satisfy certain requests.
  - Flexible: have the ability to choose which actions to invoke and in what sequence, depending on the state of its external environment.
  - Self-starting: be active not only when it is invoked by the user, but also sense changes in the environment and determine when to take which action.

- **Personalizable/Adaptability:** An intelligent agent must be able to work within an agreed upon framework to accomplish tasks specific to a particular user. It must have the ability to learn a user's preferences and then act accordingly on behalf of the user. This learning can be accomplished either by the agent being manually programmed by the user or by "watching over the user's shoulder." The latter method employs machine learning techniques in order to discover how the user performs some task. After the agent is "trained" it can then gradually take over accomplishing the task for the user \[8\].

Even if the same agent software is employed at two different locations, over time each agent should build and refer back to information it has gathered from its specific user. By doing so, agents which were identical at the onset will often react differently to a given situation once they are "trained" to perform tasks which match a particular user's preferences \[21\].

**Communication Ability/Mobility/Discourse:** Intelligent agents should have the ability to communicate easily with users and other agents across different system architectures and platforms. This ability is critical
in order for the agent to develop a good understanding of its user and the environment in which it must operate. Being able to interact with other agents of a similar nature enhances the agent’s knowledge acquisition capability by providing it with multiple sources of information for the development of its knowledge base. This assists in overcoming the “slow learning curve” typically associated with agents that are trained through repetitive user interaction.

Risk and Trust: The agent must perform with a level of ability that results in the user trusting it to act on his/her behalf. In order for the user to become comfortable with delegating duties to an agent, even for simple, non-critical tasks, there must be confidence that the agent will act as expected in most situations. Without trust the user is unlikely to use an agent.

Temporal Continuity: The agent must remain active over extended periods without user interaction. For some agents, time not spent carrying out user tasks is spent analyzing memory and determining correlations between features and actions taken [20]. Some action triggering events often occur when the user is not directly interfacing with the agent. The agent can meet user needs more effectively, and with better perceived performance, by handling tasks as they are presented, rather than delaying action until it is invoked by the user.

Domain/Expectations: Interaction between users and agents are more successful if expectations are clearly defined. In dynamic and unpredictable environments, which is often the case in real-world applications, users should expect that the agent will, at least occasionally, pursue an incorrect course of action—even after an extensive training period. However, an agent should be expected to obtain a reasonably effective level of performance within a specific domain.

Cooperation: The user and the agent should be able to communicate to construct a contract specifying the tasks to be performed. This implies that on-going “conversations” will result in a collaborative agreement about tasks, goals, and actions the agent is to perform. Intelligent agents respond to user stimuli as peers, making suggestions and requesting additional information. This type of cooperation enables an agent to perform well in highly dynamic environments.

Types of Agents
It is impossible to list all the agents currently available, particularly because new agents are surfacing all the time. Figure 1 is an extension of a model found in [9] and attempts to show graphically the major categories of agent development and domains. As the diagram depicts, the domains of agent deployment are diverse.

Artificial Life has been placed as a sub-family under Entertainment Agents as that is the classification it is given by its creators along with knowbots and MUDs. Artificial Life agents are “behavior-based” agents that attempt to synthesize biological life forms. They are often employed as interactive games as they are in the demos available from the Artificial Life homepage http://gracco.irmkant.rm.cnr.it/luigi/lupa.algorithms.html. Knowbots are agents that occupy multi-user dimension (MUDs) game spaces. They offer assistance to game players in the MUD and often employ natural language processing techniques to interpret and respond to user dialogues.

Much recent agent research deals with a more lucrative category, Task Specific Agents. Task or Domain Specific Agents shown in Figure 1 have four sub-families: Personal Assistant or Shopping Agents, Information/Internet Retrieval, Education, and Optimization or System Administration Agents.

Personal Assistants and Shopping Agents generally operate in a small specific domain of a user’s computing environment. Their purpose is to make users’ computing time more efficient. Examples of this type of agents are e-mail assistants that sort, and in some cases delete, e-mail messages or data categorization agents, which try to categorize large amounts of information into manageable chunks. We will discuss in more detail task-specific agents designed to be personal shoppers who assist consumers in purchasing products electronically over the WWW and electronic meeting facilitation agents which assist in the categorization of textual information.

Information Retrieval (IR) and Internet Agents are search engines which typically assist users in locating information on local Intranets, information databanks and the WWW. Most of these agents have a single common entry point for all users and therefore cannot be customized. We will discuss an agent that is designed to locate information on Intranets and Internet more precisely than typical implementations of Internet agents.

Education Agents interact with users in a training capacity. Most of them are targeted to interact with children and teach specific topics, such as spelling, in a game-like atmosphere.

Optimization/Systems Administration Agents are designed to do the work of computer systems operators. For instance, they are able to determine where in a multi-host system applications will run most efficiently or what location of data files will result in the best data access times. Since computing environments are usually dynamic relative to some patterns of peak computing times, these agents are able to interact with the host
systems more efficiently than humans to make decisions that optimize system performance [3]. A common implementation of an agent’s decision-making strategy is based on a “bidding” or marketplace model.

The final sub-family under software agents, which will not be covered in this chapter, is viruses.

**How Agents Perform Goal Seeking Behavior**

The two major components of an individual agent are the way in which it learns and the way in which it carries out its tasks. Learning mechanisms define an agent’s ability to gain and retain information about a particular user. Many agents contain no learning component—they simply act on a predefined set of programmed rules, which are updated as necessary by the developers. Agents of this type cannot be individualized to a particular user. However, agent research has been evolving and many agents contain some capacity for learning that makes them individualizable and deployable in dynamic environments such as the Internet.

Action mechanisms define how agents accomplish tasks. Sometimes an agent is a complete system. More frequently, the agent is made of many separate components, often referred to as specialized agents, which may have conflicting goals. We next discuss different types of learning and action mechanisms.

**Learning Mechanisms**

There are three methods of developing agents: user programmed, knowledge-based approaches, and machine learning approaches [20]. Although the last of these is the generally accepted method for creating an autonomous agent, the others merit mentioning because they can be used in conjunction with machine learning approaches to improve the initial success of an intelligent agent.

**End-User Programmed**: This method requires the user to program the agent to perform required tasks. The user sets up the foundations by which the agents makes decisions and carries out tasks. This method can work well for static environments, but has not been as viable in more dynamic environments.

- **Advantage**: User will more readily trust the agent to act independently. The user can be certain of the action an agent will take, given a particular situation.
- **Disadvantage**: Creating and maintaining the agent is too difficult for most users. There is no method by which an agent learns or develops its perceptions about the environment, thus the user must continuously revise the agent’s decision-making parameters in order for it to operate in dynamic environments.

**Knowledge-Based**: Another name for a knowledge-based agent is an expert system. A knowledge-based agent typically exists in a highly domain-specific field. A software engineer provides the agent with a large set of background knowledge about the user and the environment within which the agent will operate. This type of implementation is common for intelligent user interfaces such as web browsers.

- **Advantage**: Provides a homogeneous interface for all users.
- **Disadvantages**: Takes a lot of domain specific knowledge. It difficult for software engineers to implement and maintain. There is no custom interface which is specific to the user and the user’s preferences are forgotten as soon as the application closes.

**Machine Learning**: Machine learning implies that the agent will adapt and learn through interaction with a user.

- **Advantages**: The user builds a relationship with the agent through training, thereby reducing some of the issues surrounding competence and trust. It is much
easier for software engineers and end users to implement a machine learning approach. The agent continues to improve its performance over time without the need for additions to its knowledge base.

- **Disadvantages:** Extensive repetitive behavior is required to train the agent.

**Combination Approaches:** Intelligent agents must have some minimum background knowledge to perform initially. Machine learning can be combined with either knowledge-based or user-developed approaches in order to achieve better performance at the onset. This essentially provides the advantages of the methods above, while reducing or removing many of the disadvantages.

**Machine Learning Approaches**

There are four methods by which an agent can learn and improve its behavior over time [21]:

1. **Observing and imitating the user:** Observing the user means that the agent monitors the user’s activities and then creates rules which are stored in a database and referenced by the agent. This database is also referred to as an agent’s knowledge base. The information contained within the database is highly dynamic.

2. **Receiving positive and negative feedback from the user:** Receiving positive and negative feedback is an indirect method by which an agent learns. In this case, an agent may make some recommendation about what it thinks the user will do. If the user positively reinforces the agent by agreeing with the agent’s chosen action, the agent gives the weight of the rule which resulted in reaching the proper conclusion a higher weighted value. If the user selects another alternative, the agent will then lower the weight of that rule. These weights are stored along with the rules in a database.

3. **Receiving explicit instructions from the user:** Explicit instructions are received when an agent and a user go through a training set which is set up for the purpose of teaching the agent about the user’s preferences.

4. **Asking other agents for advice:** This learning method involves agent collaboration [17]. The agent can interface with other agents that are similar in nature and ask their advice given a specific event. This usually occurs either when an inexperienced agent is unable immediately to reach to determine what action should be taken or when an agent initiates exploratory communication with fellow agents for certain classes of events.

**Action Mechanisms**

Given a task, an agent will choose and act on the basis of:

- **The perceived environment:** before acting the agent makes an assessment of the current situation in the user’s environment.
- **The internal needs of the agent:** The goal the agent is to act upon is reviewed.
- **The agent’s recent history:** The situation is compared with the database of information the agent has amassed, and the knowledge is applied to the situation [20].

As previously mentioned, an agent relies on a knowledge base of information which enable the agent to make judgments about possible actions. Each sentence is evaluated and given a point estimate or score. The sum of the differences for all weighted sentences is used to suggest a course of action. A confidence level is generated based on the proximity of neighbors, the closest neighbor’s recommended course of action, and previously memorized examples of similar situations. Depending on the resulting confidence level, a percentage score from 0 to 1, the agent will act independently, suggest a course of action, or ask the user for advice.

In many behavior-based applications, user-controlled threshold levels are allowed to be set. If the confidence level falls within the “do it” threshold the agent will act on the users behalf autonomously. If the confidence level is below the “do it” threshold, but above the “tell me” threshold, the agent will suggest a course of action, but not take the action without the users approval. If the confidence level is below the “tell me” threshold, the agent is unsure how to proceed and requires the user to “show” it the correct course of action. As the user’s confidence in the agent increases, the threshold can be changed to a level at which the agent rarely has to ask the user for advice on how to act, only in unusual or unique situations.

Communication with other agents is also important. Cooperation among agents fosters “societies” of agents [22], which collaborate to meet user goals. Collaborative implementations focus on the interaction between agents with potentially conflicting goals, communication between agents, and decomposition and distribution of tasks. It is this interaction rather than an individual effort which enables the agent system as a whole to meet user needs. Agents can exist in functional hierarchies to accomplish heterogeneous modular tasks or collaborate on tasks in a homogeneous way [6]. More collaborative implementations allow for the sharing of information between agents, which further facilitates the learning process.

**Electronic Commerce Agents**

Electronic Commerce Agents (ECAs) often incorporate a financial or business component into the agent architecture. Either they are directly responsible for the ex-
change of funds over the Internet (for example an agent designed to pay remittances electronically) or they are implemented to generate or locate information which is necessary to the user or business which "employs" them. Information accessibility is becoming of particular importance in today's dynamic marketplace—providing support and success to firms which make investments in technology to support the gathering and assimilation of both internal and external information available to the firm.

Electronic commerce of the future is described as a "massive economy of online services linked together so that businesses routinely outsource functions such as fulfillment and shipping" [12]. Distributors will "go virtual" by outsourcing physical warehousing and movement of goods. As we move from local to global economies, ECAs will become increasingly important to users and businesses.

**ECA Characteristics and Examples**

The most commonly available implementation of ECAs to the general public are shopping agents, also known as "metashoppers." We will discuss in more detail two shopping agents of varying degrees of complexity and capability which allow for purchasing products over the Internet: Jango and FireFly. An information gathering agent, the Itsy Bitsy Spider, is an example of an information generation agent—one which provides better precision in the locating of information than traditional search spiders.

Jango (http://jango.excite.com/index.dcg) from NETbot, Inc., is a shopping agent created by agent researchers Daniel Weld and Oren Etzioni. Jango is deployed under the Internet search engine Excite (http://www.excite.com/) as a shopping service. Similar to Internet and IR agent applications, Jango has a common user interface, rather than a customizable one. It has no built in capacity to store information about a user's searching preferences. Each time the user visits the site, it is as if it were the first time. Jango searches Internet web sites to find different sources for items the consumer may be interested in purchasing. Like a regular search engine, Jango searches based on the information entered by a user and returns all possible sources for requested items displaying all possible alternatives. Consumers must visit the Internet suppliers' sites in order to actually purchase products.

One of Jango's strengths is that it can search for a wide variety of products from more expensive items such as automobiles, furniture, and golf clubs to incidental items such as videos, spirits, and groceries. It has fast response time, but it may not perform an exhaustive search. Users are not given the opportunity to specifically select supplier sites. Another weakness is the common user interface. Jango retains no information about a consumer's preferences. However, this type of implementation matches the user search strategies of most Internet search engines such as Excite. Also, since it only reports the source and pricing for the goods and services the consumer is interested in, it relies on the supplier's source site to handle the purchasing aspect. This is a much easier strategy to implement as it avoids the necessity of dealing with electronic commerce issues such as security.

Firefly from Agents, Inc. is a more sophisticated type of shopping agent. Interestingly, unlike the newly available Jango, it was one of the first shopping agents available on the WWW (commercially available circa 1996). Unlike Jango, Firefly has its own site instead of being embedded in a larger environment and users are required to provide a login name and password, since Firefly retains user information. Unlike other web sites which subsist on targeted advertising, Firefly doesn't require the user to fill out demographic questionnaires. Instead Firefly "learns" the preferences of the user through interaction and stores them for future visits. The more the user interacts with its agent, the better the system gets at predicting the user's preferences.

Firefly's main purpose is to match consumers with music they might be interested in purchasing (although for a short period, it also offered video cassettes). Users train the agent by ranking music they have previously listened to on a scale of 1 to 7. The agent tracks the rankings along with genre preferences of a particular user. After a brief training period, the agent can predict and recommend other similar music to the user and can even predict his/her ranking of a music selection not previously encountered, by comparing the ratings and music tastes of the user to those of other users with similar musical tastes.

The greatest advantage of Firefly is that it remembers each individual user. Each time a user logs in and interacts with the agent, Firefly updates its information. Therefore, the agent can adjust when a user's purchasing preferences change. Update occurs automatically, unseen to the user. Agents, Inc. and its advertisers reach customers by using the psychographic profile of the user maintained by the agent. Specifically targeted advertisements can be sent to the user's screen—an advantage that common user interfaces do not possess.

A disadvantage is that the agent must be trained and thus requires periodic interaction in order to remain helpful. However, this disadvantage is outweighed by the marketing potential. A more serious disadvantage is the limited product range—music CDs and tapes. However, the links to other interesting shopping sites and the fact that many users log in simply to converse with
other users and/or enjoy the novelty of the agent’s musical suggestions overrides this disadvantage. Thousands of people visit Firefly’s site daily and while they visit the site they are shown advertisements which are likely to be of interest to them.

The Itsy Bitsy Spider (http://ai.bpa.arizona.edu/) from the AI Lab at the University of Arizona is an Internet/Intranet information retrieval agent. The agent’s main purpose is to combat the information overload problem occurring on the Internet. The need for such an agent becomes apparent when simple queries to Internet search engines result in thousands of hits, most of which are only marginally relevant.

Instead of focusing on standard keyword searching like most Internet agents and web crawlers, the Itsy Bitsy Spider (IBS) attempts to match entire pages. The user inputs the URL of a starting page or pages which represents the type of information the user is seeking. Through the user specified criteria, the agent searches for pages on the Internet or a local Intranet which are most similar to the starting page(s) and reports the results back to the user.

The page matching is a fairly simple concept which has dramatic results in the precision of retrieved pages. The agent converts the starting page(s) into a vector of search terms. Pages located on the Web which have a higher number of co-occurring terms with the starting page are selected as matches. Users can specify the number of pages to retrieve, the number of levels to search from the original page and/or the length of search time.

The greatest advantage of IBS is the ability to retrieve a concise number of highly relevant URLs. The accuracy is driven by support for an entire textual vector of search terms (all terms that appear in the starting page). A disadvantage is that the tool is slower than other search engines. However, by using the agent’s Java interface, users can see interim results instead of waiting for the entire search to complete before being able to look at retrieved items. Also, users have control over the waiting period by being able to specify search-time duration and the number of levels to search. As with Jango, IBS does not have the ability to conform to a particular user’s re-
requirements. All users rely on the same interface to enter query information.

These examples represent only a few of the ECAs available on the World Wide Web. Among the list of newly available metashoppers are Roboshopper (http://www.roboshopper.com) and Shopping Explorer (http://shoppingexplorer.com). Both provide features similar to Jango’s. They query online stores for specific products in parallel and then return the results in a list that allows the user to choose which stores to shop. Other search engines, such as Yahoo, are also rumored to be investigating the implementation of metashoppers [1].

Role of ECAs

Agents are designed to engage and help all types of end users [24]. They are not designed as replacements, but rather counterparts or helpmates to users in a complex and dynamic world.

An interesting paper by Chen et al. [4], describes one of the few comparisons of agents and their human counterparts. It was discovered that an agent was less precise in categorizing concepts generated during a GroupSystems(r) electronic meeting. However, the agent was five times faster than its human counterpart.

The rigorous study compared the performances of 12 professional meeting facilitators to those of a meeting categorizing agent for four different electronic meeting sessions. The conclusion was that it was most effective to combine use of the agent and a human counterpart to create meeting categories more rapidly. Employing the agent as a first cut categorizer followed by refinement by a human facilitator reduced the categorization time and resulted in excellent precision.

Previous research has shown that systems designed to replace humans in a specific domain (usually in reference to expert systems) have experienced low adoption rates [2], but users appear to be more receptive to agents because they are perceived to reduce workload by assisting them in accomplishment of tasks. Instead of purporting to be human replacements, as did the expert systems of the past, agents recently are gaining research support.
and usage as user enhancements.

Caveats

Ethics and Legal Issues

Who should have access to the data maintained and updated by an intelligent agent? Agents will often interface with similar agents to improve their overall performance. It is conceivable that an agent seeking to collect information about users could represent itself to the user’s agent as a similar agent just looking for assistance [20].

Is there a difference between companies that maintain and sell databases of their customers’ addresses and buying preferences and those that create and store user’s e-mail addresses and psychographic profiles so advertisers can better target their customers? (These types of applications and how the information they generate is used are described in more detail in [7, 10]) No legal precedent has been set on this issue. However, companies like Agents Inc. have made it their policy to protect their users’ privacy. In a recent article in IEEE expert, Vice President Yezdi Lashkari stated, “The user is the owner of private data. We’re simply the custodian” [14].

Security issues regarding the Internet are a pervasive and well-publicized problem. Hackers stealing credit card numbers and students breaking Internet security codes have filled the headlines ever since the introduction of the WWW. Convincing consumers that their credit information is secure is something that must occur before Internet commerce moves from the early-adopter stage to a main stream activity. With the standardization of communication languages between Internet agents comes the possibility of agents being created to gather information surreptitiously.

Larry Foner plainly states in [8] that “The sometimes conflicting goals of utility and privacy can be reconciled if the system is designed from the beginning to protect privacy while enabling distributed, shared computation.” He suggests we motivate the development of a toolkit for agents which incorporates privacy. Many of the new Java toolkits claim to have this ability without compromising inter-agent communication [17, 18]. Prominence of this issue has been driven by companies conducting electronic commerce and electronic data interchange over the Internet.
Economic Issues

People often associate the ability of an ECA's ability to "shop around" with the possibility of paying a lower price. It is assumed that since users can look at a variety of sources before they buy and can ultimately choose the vendor with the lowest price that EC agents can create lower overall prices for goods purchased over the Internet. However, there have been documented cases in which Internet buyers may actually have paid more for an item than a non-Internet purchaser.

Since agents (and unassisted Internet shoppers) often rely on pricing available on the web, they are not aware that an unadvertised lower price for the same item may exist off the Internet. Lee [19] reported that car buyers in Japan actually paid more for cars they purchased over the Internet than the average Japanese car buyer. There are also issues associated with being unable to look at an item, or in the case of the Japanese car market, test drive an item to make sure it meets the user's needs and that it is not defective in some way.

Many companies are investigating the incorporation of billing into their web sites [25]. As these types of sites move from the planning stages into reality consumers will be able to have their agents take care of the mundane task of paying bills. Paying bills over the Internet offers advantages to consumers and and billers alike. Consumers will no longer have to mail in payments, saving postal costs. Billers will be able to "close the payment loop" between the consumer and the banking industry. However, consumers must be convinced of the security of payments made in this manner before they move from the early-adopter stage to routine consumer behavior. Lack of infrastructure, along with virtually nonexistent legal and regulatory frameworks, have forced businesses and users to be cautious in the capitalization of electronic commerce on the Internet [12].

Social Impacts

With the amount of development underway in the field of software agents and the increased focus on agents by academic institutions and industry, it is clear that agents will have a strong impact on the way that we use and will use software in the future [26]. By definition, an agent's purpose is to make computer use easier. Interfaces are becoming more intuitive, customizable, and user-friendly. Reducing the complexity of using computers is opening the door to a new class of computer users. The statement "Grandma is online" is no longer a funny story, but an example of the accessibility offered by the new agent interfaces to the WWW.

Author Biographies

Kristin M. Tolle is a doctoral student in the Department of Management Information Systems at the University of Arizona, where she received her Master of Science in MIS in 1997. She is of a recipient a research fellowship from the National Library of Medicine and the Oak Ridge Institute for Science and Education. Her research interests are intelligent agents, information retrieval, natural language processing, and neural network decision support systems.

Hsichunchen Chen is an associate professor of Management Information Systems and director of the Artificial Intelligence Lab at the University of Arizona. He has received an NSF Research Initiation Award, the Hawaii International Conferences on System Sciences Best Paper Award, and an AT&T Foundation Award in Science and Engineering. Author of more than 30 articles about semantic information retrieval and digital libraries, Dr. Chen received his Ph.D. in Information Systems from New York University.

REFERENCES


