Software Agents: Characteristics and Classification

Qusay H. Mahmoud

School of Computing Science, Simon Fraser University
Burnaby, BC. Canada V5A 1S6
qmahmoud@cs.sfu.ca

Abstract. Since the arrival of the World-Wide Web, and the explosive growth of information on the Web, researchers in the field of Software Agents got interested in information agents (also known as Internet agents). However, for the past couple of years, agent technology has been a hot topic, and most likely, this is mainly due to the popularity of the Java programming language, which represents an ideal language for implementing software agents as it is the "*Write Once Run Anywhere*" language. This is an important feature for software agents as it allows them to run on all platforms of the Internet. The word "agent" however, has been misused. People involved in agent research have a variety of definitions for that word. This paper presents a brief overview of software agents, with emphasis on mobile agents. We classify agents in different types along several primary characteristics that agents should exhibit.

1 Introduction

Before we try to define what is a Software Agent, let us first try to understand the meaning and characteristics of an agent. We are all, in one sense or another, familiar with the concept of an agent. Probably most of us have dealt with travel agents and we know the role undertaken by them. The main representative role of a travel agent, for example, is that it *acts on behalf of others*. This characteristic can be considered as the first fundamental property of agency. A travel agent acts on behalf of a traveler in a variable degree of autonomy. In other words, when a travel agent calls an airline to reserve a seat, they do so autonomously; that is, they do not tell the airline to whom do they need a seat, they just say we need a seat. This is the second characteristic of agency – *autonomy*. A third characteristic of an agent is the degree of *proactivity* and *reactivity* in their behavior. For example, once an agent receives the details of its tasks, it tries proactively to attain the goals defined by the assigned tasks. And it reacts to the changes in the

available data by modifying its plan. Agents may also exhibit other attributes, including: *learning*, *co-operation*, and *mobility.*

2 What is an Agent?

Based on the above discussion, an agent can be defined, along with its characteristics as follow:

"*An agent is an entity that:*

- *acts on behalf of others in an autonomous fashion*
- *performs its actions in some level of proactivity and reactivity*
- *exhibits some levels of the key attributes of learning, co-operation, and mobility.*"

This definition, which is based on the definition given in [5], is equally true for software agents. But what exactly is a software agent and how does it differ from a software object? You may think of a software agent as one (or more) software object(s) that conforms to the above characteristics of agents and can be described as inhibiting computers and networks, assisting users with computer-based tasks. It is the responsibility of the programmer, however, to determine what an agent can do, as well as the information required from the user or software to have an agent performs its actions in a reactive manner. The behavior of the agent can be set by another software, which you can think of as a sort of a super agent, that forks (or clones) new agents when a task requires extra help.

3 Classification of Software Agents

Classification refers to the study of types and entities. There are several dimensions to classify existing software agents. They can be classified according to: the tasks they perform; their control architecture; the range and effectiveness of their actions; the range of sensitivity of their senses; or how much internal state they posses [6]. In this paper we classify agents, along several ideal and primary characteristics that agents should exhibit. We identify three characteristics: *autonomy*, *learning*, and *cooperation*, which we have discussed in Section 1. Our classification here is based on [4]. Let us start by elaborating a bit more on these characteristics. Autonomy refers to the characteristic that an agent can operate on its own without the need for human guidance. In other words, an agent has a set

of internal states and goals, it acts in such a manner to meet its goals on behalf of the user. In order to do so, an agent has to be proactive in the sense that it has the ability to take the initiative rather than acting simply in response to its environment [15]. Cooperation with other agents is necessary to accomplish a complicated task. In order to cooperate however, an agent must posses a social ability that allows it to interact with other agents. Finally, an agent is intelligent if it is able to learn and sense as it acts and reacts to its external environment. These three characteristics of agents are used to derive some types of agents to include in our classification as shown in Figure 1.

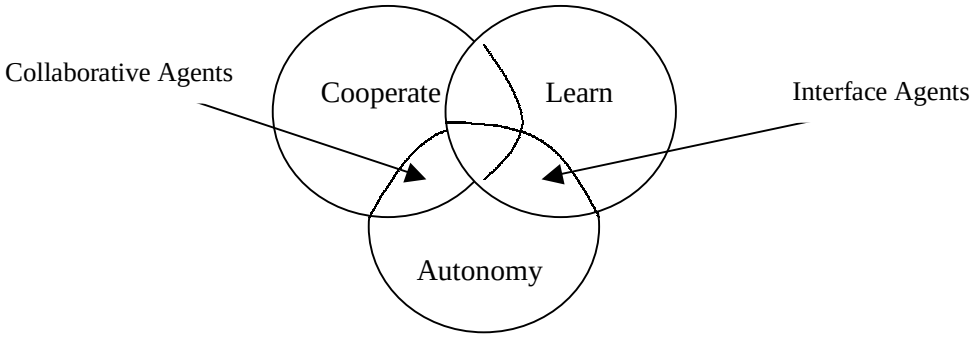

Figure 1: A partial view of agent classification

3.1 Interface Agents

Interface agents perform tasks for their owners by emphasizing autonomy and learning [16]. They support and provide assistance to a user learning to use a particular application such as a spreadsheet. The agent here observes the actions being carried out by the user and tries to learn new short cuts, then it will try to suggest better ways of doing the same task. As P. Maes points out, the key metaphor underlying interface agents is that of a personal assistant who is collaborating with the user in the same work environment. Interface agents learn to better assist its users in four ways [8]:

- By observing and imitating the user
- Through receiving positive and negative feedback from the user
- By receiving explicit instructions from the user

- By asking other agents for advice

While interface agents ask other agents for advice (learning from peers), their cooperation with other agents however, is limited.

The objective of P. Maes work on Interface Agents was initially to migrate from the direct manipulation metaphor [9] to one that delegates some of the tasks to software interface agents in order to accommodate novice users. With the explosive growth of the Web, and as we come close to the 20th century, we are having as many untrained computer users as we have microprocessors. Therefore, there is a gap that is coming more apparent as we go along with computers. Current computer interfaces, for example, do nothing unless you give commands from the keyboard, mouse, or touch screen [5]. Therefore, the computer is merely a passive entity waiting for your commands to execute them. It does not provide us with any help whenever we wish to perform some complex tasks. In the future, interface agents as well as information agents (as discussed later) may know users' interests and act autonomously on their behalf. That is what researchers in the area of agent technology have high hopes for.

3.2 Collaborative Agents

As the proliferation of computer communication networks was a big step toward the development of "virtual societies". Collaboration between individuals (in the virtual society) requires that communication links be established and used effectively. Distributed Artificial Intelligence, which is a subfield of Artificial Intelligence, is concerned with a virtual society of problem solvers (agents) interacting to solve a common problem [5].

The goal of collaborative agents is to interconnect separately developed collaborative agents, thus enabling the ensemble to function beyond the capabilities of any of its members. Implementing efficient ways of cooperation among agents is actually one of the central issues for Multi-Agent Systems development [19].

One of the motivations for having collaborative agents is to provide solutions to inherently distributed problems, such as distributed sensor network [20], or air-traffic control.

3.3 Information Agents

The explosive growth of information on the Word-Wide Web has given a rise to information agents (also known as Internet agents) in the hope that these agents will be able to help us manage, manipulate, or collate information from many distributed resources [7]. One may notice however, that information agents seem a bit similar to interface agents [16]. However, it is important to note that not all types of agents discussed here started at the same time. So, with the explosive growth of information, and the need for tools to manage such information, one would expect a degree of overlap between the goals of some agents. One distinction between interface and information agents, however, is that information agents are defined by *what they do*, in contrast to interface agents which are defined by *what they are*.

There are some interface agents, developed at MIT Media Lab, that have been deployed for Web-based roles [8], and hence they can be considered as information agents. One important characteristic of Information Agents is that they may be static or mobile.

Information agents are most useful on the Web where they can help us with mundane tasks. For example, we carry out actions that may consume long time (e.g. searching the Web for information). Why does not the computer (e.g. an information agent) carries out such tasks for us and later on present us with the results?

3.4 Reactive Agents

Reactive Agents act and respond in a stimulus-response [18] manner to the present state of the environment in which they are embedded. P. Maes highlights the following three key ideas which underpin reactive agents [13].

- Emergent functionality: the dynamics of the interaction leads to the emergent complexity.
- Task decomposition: a reactive agent is viewed as a collection of modules which operate autonomously and responsible for specific tasks (e.g. sensing, computation, etc.).
- They tend to operate on representations that are close to raw sensor data.

Up till now, there is a relatively few number of reactive software agent-based applications. A good application area for them seems to be the entertainment industry. Researchers at Philips are already working on digital video and 3-D graphics-based reactive agent animation [4].

3.5 Hybrid Agents

Hybrid Agents refer to those agents whose constitution is a combination of two or more agent philosophies within a singular agent [4]. These philosophies may be mobile, interface, information, collaborative, … etc. The goal of having hybrid agents is the notion that the benefits accrued from having the combination of philosophies within a single agent is greater than the gains obtained from the same agent based on a singular philosophy. An example of this is *collaborative interface agents* [17].

NOTE: The author believes that mobile agents (discussed next) can actually be considered hybrid agents, because in order for a mobile agent to be useful, it may have to reactive, proactive, and collaborative, for example.

3.6 Mobile Agents

A software agent is a *mobile* software agent if it is able to migrate from host to host to work in a heterogeneous network environment. This means we must also consider the software environment in which mobile agents exist. This is called the mobile agent environment, which is a software system distributed over a network of heterogeneous computers and its primary task is to provide an environment in which mobile agents can run. Note that not only an agent transports itself, but also its state. When it reaches the new host, the agent should be able to perform appropriately in the new environment.

3.6.1 Anew Paradigm for Distributed Computing

The central principle of today's distributed programming is remote procedure calling (RPC). The RPC approach, which was conceived in the 1970s, views computer-to-computer communication as enabling one computer to call a procedure in another. In RPC, all messages go through the network, each either requests or acknowledges a procedure's actions. This approach, however, has its own limitations. Most notably, all interactions between the client and server must go through the network as shown in Figure 2.

Figure 2: RCP-based Client/Server Computing Paradigm

Another approach that is forming a new paradigm for distributed computing is one that employs mobile agents. Initially this approach was known as Remote Programming [1]. The Remote Programming approach views computer-to-computer communication as one computer not only to call procedures in another, but also to supply the procedures to be performed. Each message that goes through the network comprises a procedure that the receiving computer is to perform and data that are its arguments. The procedure and its state are termed a mobile agent as they represent the sending computer even while they are in the receiving computer as shown in Figure 3.

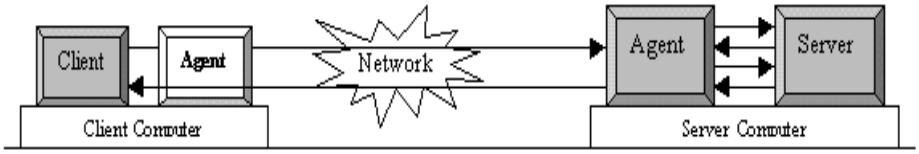

Figure 3: mobile agents-based computing paradigm

This approach is attractive since the reliability of the network is not crucial for the following reasons:

- Mobile agents do not consume much network bandwidth. They only consume bandwidth when they move.
- They continue to execute after they move, even if they loose network connectivity with their creators.

Therefore, if a client requires extensive communications with a particular server somewhere on the network, then implementing such a system using mobile agents is attractive. This is due to the fact that an agent can move closer to the remote

server, reducing the network traffic, performs all tasks and comes back. During that period the client machine doe not have to be switched on. It will have to be switched on only when it is time to welcome back the agent. Figure 3, shown above, illustrates the idea of this paradigm.

At this point, someone may say that this is exactly what *process migration* is all about and this has been done in the 60's. That someone would be absolutely right. However, mobile agents are different in the sense that they exhibit the characteristics of an agent as discussed in Section 1.

3.6.2 Applications of Mobile Agents

An attractive area to use mobile agents is in processing data over unreliable networks. In such networks, the low-reliability network can be used to transfer agents, rather than a chunk of data, from place to place. In this paradigm, the agent travels to the nodes on the network, process the information on those nodes (without the risk of network disconnection) and then return home. Another interesting area that is attracting lots of attention is electronic commerce. For example, instead of spending a huge amount of time going through on-line bookstores to find the best deal on a book, firing up an agent to do this task would save us a considerable amount of time. The agent would be programmed to visit a number of bookstores and find the best deals on books we need.

Another interesting area of applications for mobile agents is in network management. In today's heterogeneous network environments, network operators are required to have extensive knowledge of the diverse networks in order to manage them. Such management requires the collection of large amount of data from nodes on the network. Mobile agents represent an ideal tool for collecting the data and analyzing it.

3.6.3 Advantages of Mobile Agents

Mobile agents seem to be useful for many different applications. One may however, claim that virtually any task that can be performed with mobile agents can be performed with other technologies (e.g. remote method invocation). Despite the fact that there are not many distributed computing problems that cannot be solved without mobile agents, nevertheless mobile agents make certain applications easier to develop and may improve reliability and efficiency. Table 1, which is based on [5] contains some of the claimed advantages of mobile agents over conventional approaches.

Advantage	Justification
Efficiency	Mobile agents consume fewer network resources since they move the computation to the data instead of the data to the computation.
Less Bandwidth	Most communication protocols involve several interactions, which cause a lot of network traffic. Mobile agents consume bandwidth only when they move.
Robustness and Fault Tolerance	The ability of mobile agents to react dynamically to adverse situations makes it easier to build fault tolerance behavior in complex distributed systems.
Support for Hetero-geneous Environments	Mobile agent systems are computer and network independent. Therefore, a Java mobile agent can target any system that has a Java Virtual Machine.
Support for Electronic Commerce	Mobile agents are being used to build electronic markets since they embody the intentions, desires, and resources of the participants in the market.
Easier Development Paradigm	The construction of distributed systems can be made easier with mobile agents. Mobile agents are inherently distributed in nature. Therefore, they are a natural view of a distributed system.

Table 1: possible advantages of mobile software agents

3.6.4 Security in Mobile Agents

Despite the fact that mobile agents represent a new useful paradigm for distributed computing, they have, however, seldom been used in practice. The main reason for this is due to the fact that there are a number of related technical as well as social challenges to implementing mobile agent-based applications. Technical issues include, among others, difficulties with portability and security. The security issue may also be considered as a social challenge – convincing people to use mobile software agents to buy items over the Internet. As B. Laurel [10] pointed out “few of us would hire an agent to push the buttons on our calculator; most of us would hire an agent to scan 5,000 pieces of junk mail”.

The security challenge, however, is a complicated one. Mobile agents raise issues similar to Java Applets. There are several security issues to be considered in mobile agent-based computing. Some people in the computing community think of mobile agents as viruses sine they may exhibit similar behavior.

Mobile agent security can be split into two areas [5]:

- Protection of host nodes from destructive agents.
- Protection of mobile agents from malicious nodes.

One approach to protect the host nodes from destructive agents it through authentication and agent signatures where all unknown agents are rejected. However, this does not really seem like a good solution, because how would we know that a mobile agent is whom it claims to be?

The other area of security deals with the issue of protecting mobile agents from hosts which may want to scan the agent for information; alter the agent's state; or even kill the agent. The crucial issue here is that the agent will have to expose its data and information to the host in order to run on it [11]. Current research has shown it is computationally impossible, however, to protect a mobile agent from a malicious host [12]. Some researchers are looking at sociological (instead of the hard computational model) means of enforcing good host behavior [12].

4 Conclusion

Agent technology is a hot topic that is keeping many people both, in academia and industry quite busy. In this paper we have given a brief overview of software agents and their applications. We explained the characteristics of an agent and based on that we classified agents into six types: interface, collaborative, information, reactive, hybrid, and mobile. The mobile agent class was discussed in more details than any other class. This because in the author's view a useful mobile agent is inherently an information agent and should be collaborative and reactive. In other words, a mobile agent is different from a mobile object since the mobile agent has to exhibit some attributes of agency (e.g. autonomy, learning, co-operation, proactivity and reactivity).

5 References

[1] J. White. "Mobile Agents". In J. M. Bradshaw (editor), *Software Agents*. AAAI Press/MIT Press, 1997, pp. 437-472.

[2] P. Morreale. "Agents on the move". In *IEEE Spectrum*, April 1998, pp. 34-41.

[3] V. A. Pham, and A. Karmouch. “Mobile Software Agents: An Overview”. In *IEEE Communications Magazine*, July 1998, pp. 26-37.

[4] H. S. Nwana. “Software Agents: An Overview”. In *Knowledge Engineering Review*, Vol. 11, No 3, Oct/Nov 1996, pp. 205-244.

[5] S. Green, L. Hurst, B. Nangle, P. Cunningham, F. Somers, R. Evans. “Software Agents: A review”. *Technical Report*, Trinty College Dublin, Ireland, May 1996.

[6] S. Franklin, and A. Graesser. “Is it an agent, or just a Program?” A Taxonomy for Autonomous Agents”. In *Proceedings of the Third International Workshop On Agents Theories, Architecture, and Languages*, Springer-Verlag, 1996.

[7] P. Maes. “Intelligent Software”. In *Scientific American*, Vol. 273, No. 3, Sept. 1995, pp. 84-86.

[8] P. Maes. “Agents that Reduce Work and Information Overload”. In J. M. Bradshaw (editor), *Software Agents*. AAAI Press/MIT Press, 1997, pp. 145-164.

[9] P. Maes, and Schneiderman, B., "Direct Manipulation vs. Interface Agents: a Debate", In *Interactions*, Vol. IV Number 6, ACM Press, 1997

[10] B. Laurel. “Interface Agents: Metaphors with Character”. In J. M. Bradshaw (editor), *Software Agents*. AAAI Press/MIT Press, 1997, pp. 67-77.

[11] D. Chess, C. Harrison, and A. Kreshenbaum. “Mobile Agents: Are they a good idea”. In N. Vitek, and C. Tschudin (editors), *Mobile Objects Systems*, Springer Verlag Lecture Notes in Computer Science, July 1996, pp. 25-47.

[12] A. Rasmusson, and S. Janson. “Simulated Social Control for secure Internet commerce”. In *New Security Paradigms ’96*, ACM Press, Sept 1996.

[13] P. Maes (editor). “Designing Autonomous Agents: Theory and Practice from Biology to Engineering and back”. MIT Press.

[14] M. Graham, and P.R. Wavish. “Simulating and Implementing Agents and Multiple Agent Systems”. In *Proceedings of the European Simulation Multi-Conference*, Copenhagen, Denmark, June 1991.

[15] M. Wooldridge, and N. R. Jennings. “Intelligent Agents: Theory and Practice”. In *Knowledge Engineering Review*, Vol. 10, No 2, 1995, pp. 115-152.

[16] T. Koda and P. Maes, "Agents with Faces: The Effects of Personification of Agents”, In *Proceedings of HCI '96*, London, UK, August, 1996.

[17] Y. Lashkari, M. Metral, P. Maes. "Collaborative Interface Agents". In *Proceedings of the Twelfth National Conference on Artificial Intelligence*, Vol. 1, AAAI Press, Seattle, WA, August 1994.

[18] Jacques Ferber. “Reactive Distributed Artificial Intelligence: Principles and Applications”. In G.M.P. O’Hare, and N.R. Jennings (editors), *Foundations of Distributed Artificial Intelligence*, John Wiley & Sons, Inc., 1996, pp. 287-317.

[19] B. Moulin, and B. Chaib-Draa, “An Overview of Distributed Artificial Intelligence”. In G.M.P. O’Hare, and N.R. Jennings (editors), *Foundations of Distributed Artificial Intelligence*, John Wiley & Sons, Inc., 1996, pp. 3-55.

[20] E.H. Durfee, and T.A. Montogomery. “Coherent Cooperation among Communicating Problem Solvers”. In *IEEE Transactions on Computers*, Nov 1987, pp. 1275-1291.