INTELLIGENT SOFTWARE AGENTS: ANATOMY AND APPLICATIONS

F. Baig

Pakistan Television (PTV), Lahore Centre.

ABSTRACT: Intelligent agent is a software product which can work autonomously and intelligently on behalf of human workers. It has problem solving methods and knowledge about one particular domain. This paper provides overview of different types of intelligent agents, frameworks used for them and applications of these agents. Five types of intelligent agents are discussed: Mobile agents, Distributed agents, Multiple agents, Collaborative agents and Social agents. Agent frameworks are also explained which are used for the creation of agents, communication between them and discovery of agents. Agent based systems can work all the time. Besides, they have ability to take decisions intelligently. Hence, there are some useful applications of agent based systems in different areas of life in which manufacturing, process control, information management, e-commerce, entertainment, health care and e-education are included.

Keywords: Intelligent agent, mobile agent, distributed agent, multiple agent, collaborative agent, social agent.

INTRODUCTION

In different fields of life, a variety of human agents such as booking agents or sales agents are working. Similarly, there are also computer based intelligent agents to do work in one specific field. As robot is a hardware device, intelligent agent is a software product or we can say it is a software robot or simply softbot.

"An agent is anything that can be viewed as perceiving its environment through sensors and acting upon that environment through effectors" (Russell and Norvig, 1995), just like robots which can sense through sensors like camera and effect with various motors as effectors. Same things can be attached with software agents. These agents can work autonomously and intelligently on behalf of human workers. They contain problem solving methods and knowledge about one particular domain in which they work. More than one agent can also do work together in one specific environment by communicating with each other using communication languages like KIF (Knowledge Interchange Format) (Genesereth and Fikes, 1992) and KQML (Knowledge Query and Manipulation Language) (Finin *et al.,* 1992) developed by DARPA (Defense Advanced Research Projects Agency).

The following properties defined by Nwana and Azarmi (1997), are specifically considered when designing the multi-agent system:

Interactivity: the agent should interact with its environment when representing the individual or the entity and during this phase it should be able to interact with other agents.

Autonomy: the agent should be semiautonomous. This means they do not need direct supervision, but there are different degrees of autonomy, so the agent will always be under the control of individual or entity that it represents.

Proactivity: the intelligent agents will be proactive. This means that they have goals or explicit objectives and act consequently and in an autonomous manner to achieve them.

Learning: the intelligent agents should code knowledge about the represented entity and the environment where they carry out their functions. This knowledge is dynamic because it changes with time. The agent must learn from its environment and from interaction with other agents and update its knowledge base with these changes.

TYPES OF INTELLIGENT AGENTS

Croft (1997) defines following five types of intelligent agents according to their functionalities.

Mobile Agents: These agents are also known as

traveling agents because they can move from host to another host within a network (Kotz & Gray, 1999). In the client-server architecture, the agent as a client requests and server sends data to client. In some cases, processed data is also sent back to server. If huge amount of data is needed to move then significant bandwidth is required. Through mobile agents bandwidth performance can be improved by moving the agents to where the data reside instead of moving the data to where the agents reside (Croft, 1997).

Distributed Agents: "Load-balancing" can be achieved by these agents (Croft, 1997) which, are geographically distributed across the network in which every agent has limited amount of resources, processing power and problem solving capabilities. In this way agents can perform those tasks together which are beyond the capabilities of any one agent (Ndumu *et al.*, 1999).

Multiple Agents: These agents work at same place for one complicated huge task. This huge task is broken into sub-tasks and each sub-task is assigned to a concerned agent. There is no collaboration among such agents but the end result is based upon the successful operations of all agents (Croft, 1997).

Collaborative Agents: Such agents share information with each other for a common purpose (Croft, 1997). According to Allen, Blaylock and Ferguson (2002), collaborative agents must have the capability to:

1. Discuss and negotiate goals;
2. Discuss options and decide on courses of action, including assigning different parts of a task to different agents;
3. Discuss limitations and problems with the current course of action, and negotiate modifications;
4. Assess the current situation and explore possible future eventualities;
5. Discuss and determine resource allocation;
6. Discuss and negotiate initiative in the interactions;
7. Perform parts of the task, and report to others to update shared knowledge of the situation.

Social Agents: These agents can be helpful for small routine works, for example, such agents can inform you about new email received in your account. Similarly, they can recall you about small activities related to office work or studies etc. Such types of agents represent themselves as human-like (Parise *et al.*, 1996) and may have characteristics of "continuous speech, gestures, and facial expressions" (Croft, 1997).

AGENT FRAMEWORKS

Frameworks mainly contribute in creation of agents, communication between them and discovery of agents. JATLite, JADE and SoFAR are famous agent frameworks.

JATLite (Java Agent Template, Lite) is a collection of Java classes and programs and it is used to create systems comprise of intelligent agents distributed over internet. It provides a communication mechanism among these distributed agents using communication language, KQML. JATLite defines an agent as part of a community which performs computation using typed messages. It provides a robust infrastructure in which message routing and buffering service are provided to agents. Each agent is registered with an Agent Message Router (AMR), which acts like an email server and forwards buffered messages and files among all agents. This message router buffers the messages if the connection can not be made with the receiving agent (Jeon, Petrie & Cutkosky, 2000).

JADE (Java Agent Development Framework) is an open source middleware framework that is used to develop efficient distributed multi-agent environments. The communications model used by JADE is based on FIPA's ACL (Agent Communication language). It provides interoperability so that agents can interoperate with any kind of agents whether they confirm the same standard or not. It is easy to use and hides the complexity behind the set of APIs. It provides mechanisms for finding both local and distributed agents and starting communications with them (Bellifemine *et al.*, 2003). There is also a JessAgent written for JADE that allows programmers to take advantage of full capabilities of the JESS expert system (JESS, 2004) within their programs.

SoFAR (The Southampton Framework for Agent Research) is a lightweight multi-agent framework for the deployment of Distributed Information Management (DIM) agents (Moreau *et al.*, 2000). These agents run within a Java Virtual Machine (JVM) environment which is called a platform. In this platform there is a registry agent which

maintains the information about services provided by all other agents in that specific distributed environment. If an agent requires specific services then it will first communicate with the registry agent then registry agent will inform that which agent is offering those specific required services and where that agent is located. Then first agent will directly communicate with that agent offering those specific services. SoFAR agents use ontologies to communicate with each other. This communication model is based upon communication language KQML.

APPLICATIONS OF INTELLIGENT AGENTS

Intelligent agents are best substitute of human workers because it is not possible to work all the time. There are number of areas where these software robots can be used as problem solvers.

Manufacturing: Manufacturing involves supply chain management, planning, scheduling and controlling activities. Every process is consisted of sub processes or functions which can be assigned to agents separately. In this situation, multiple or collaborative agents can be used. Multi-agent based YAMS (Yet Another Manufacturing System) designed for product manufacturing where each agent has a separate collection of plans (Parunak, 1987).

Process Control: Different types of processes can be automatically controlled by agents. For example, in air traffic control system, when an airplane enters the airspace of airport then an agent can be associated with that to guide about landing. During this process that agent may exchange information with other agents in collaborative environment. Some other process control systems include automated warehouse (Kim *et al.,* 2004) and transportation management (Davidsson *et al.,* 2004) etc.

Information Management: In daily life, we need information for different purposes like studies or office work, but among huge amount of information only tiny portion is relevant to us. Hence "information filtering" is a big problem. Another problem is information gathering. Some social agents have been designed for these purposes like "Maxims", an electronic mail filtering agent, "Newt", an Internet news filtering agent (Maes, 1994) and "Letizia", a web browsing assistant (Lieberman, 1995).

E-commerce: This application area includes finding cheap products by comparing prices offered by different companies, electronic biding, and taking decisions about buying and selling of shares in the stock market etc. Usually mobile agents are used in such domain. Pivk and Gams (2000) have described some intelligent agents for e-commerce: "PersonaLogic" which returns a list of products that satisfy all of the consumer's hard and soft constraints, by which they are ordered; "BargainFinder", uses parallel search architecture. "It submits consumers' query in parallel to a group of on-line vendors by filling out the form at each site. It parses the query results after filtering out the header, trailer, and advertisements to find each vendor's price for the required product and prepares a summary to the consumers" (p. 55) for comparing prices; "Kasbah" is an on-line, multi-agent consumer-to-consumer transaction system. If any user wants to sell or buy something then he creates an agent and sends it to centralized agent marketplace for negotiations with other agents on behalf of the user. But it follows "user-specified constraints, such as an initial asking (or bidding) price, a highest (or lowest) acceptable price, and a date by which to complete the transaction" (p. 55). Pivk & Gams also discussed some other agents for e-commerce which are "Firefly", "Jango", "AuctionBot", "Tete-a-Tete" and "AuctionWeb".

Entertainment: In this field interactive games and theater are included. Computer games provide such immersive environments which are usually not in access in real life. Some times we become a part of national football team or some times we are in the past and fighting with dinosaurs. But to make such environments more attractive, there is a need to make them more interactive. Li, Musilek and Wyard-Scott (2004) have focused on this most important element with the help of agents and fuzzy logic. Similarly, Kelso, Weyhrauch and Bates (1993) have presented the Oz system architecture to create "interactive drama" based on believable agents. In such type of drama, we can interact with characters inside the simulated environment.

Health Care: Like Tele-Medicine and Bio-Technology, Medical Informatics is also becoming a popular application of computer science. Agent based problem solving approach can also be applied in this area. Huang, Jennings and Fox (1995) describe prototypical agent based distributed health care system in which agents contain a knowledge base system, a user interface and communication manager having massage passing functionality of agents. In the absence of

expert doctors we can provide health care and patient mentoring facilities. This system is based upon KADS model of expertise and the architecture is implemented in PROLOG.

E-education: Intelligent agent can be used as a personal digital assistant of a teacher. Such a social agent can provide information about those students who have not submitted their assignments or have not taken online quiz. This agent can be configured to automatically send emails to those students (Jafari, 2002). Lin *et al.* (2001) have presented a framework for designing and developing agent based online learning system. The architecture of this framework is consisted of three types of agents: course designer (professor) agents, learning agents (student-side agents) and server-side agents. There are many issues involved in e-education in which agent based systems can be used. For example, it is very difficult to observe the communication among the students. Jaques and Oliveira (1998) have presented a multi-agent based architecture which is able to monitor communication tools like chat or newsgroups etc. This system can provide complete report to the instructor about a student that whether he is taking part in online discussion or not. It is also possible to find out that which topics are discussed in one particular group and which groups of students interact with each other intensively.

CONCLUSIONS

It is not possible for humans to work 24 hours a day in one field. Experts are also not available every time and every where. That is why people are working for the automation of the processes. Intelligent agents can play a best role in this regard. They can be equipped with experts' knowledge for decision making besides they can work all the time on behalf of the humans. In this way not only efficiency but productivity can also be increased.

REFERENCES

Allen, J., N. Blaylock and G. A. Ferguson. Problem solving model for collaborative agents. Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent Systems: part 2, Bologna, Italy. pp. 774 – 781. ACM Press, New York (2002).

Bellifemine, F., G. Caire, A. Poggi and G. Rimassa. JADE - A white paper. Telecom Italia EXP Magazine, 3(3): 6-19 (2003).

Croft, D. W. Intelligent software agents: Definitions and applications. http://www.alumni.caltech.edu/~croft/research/agent/definition/ (1997).

Davidsson, P., L. Henesey, L. Ramstedt, J. Törnquist and F. Wernstedt. Agent-based approaches to transport logistics. 3rd International Joint Conference on Autonomous Agents and Multi Agents Systems (AAMAS), Workshop on Agents in Traffic and Transportation, New York, USA (August 19-23, 2004).

Finin, T. W., J. Weber, G. Widerhold, M. Genesereth, R. Fritzson, D. McKay, J. McGuire, R. Pelavin, S. Shapiro and C. Beck. Specification of the KQML Agent-Communication Language (Technical Report EIT TR 92-04). Enterprise Integration Technologies, Palo Alto, California, USA (1992).

Genesereth, M. R. and R. E. Fikes. Knowledge Interchange Format Version 3.0 Reference Manual. (Technical Report Logic-92-1). Stanford Logic Group, Stanford University, Stanford, California, USA (1992).

Huang, J., N. R. Jennings and J. Fox. An agent-based approach to health care management. Int. J. of Applied Artificial Intelligence, 9(4): 401-420 (1995).

Jafari, A. Conceptualizing intelligent agents for teaching and learning. Educause Quarterly, 25(3): 28-34 (2002).

Jaques, P. A. and F. M. D. Oliveira. Software agents for analysis of the interactions in a distance learning environment. The 3rd World Conference of The WWW, Internet & Intranet (WebNet'98), Orlando, Florida, USA (November 7-12, 1998).

Jeon, H., C. J. Petrie and M. R. Cutkosky. JATLite: A Java Agent Infrastructure with Message Routing. IEEE Internet Computing, 4(2): 87–96 (2000).

JESS. Jess, the rule engine for the Java platform. http://herzberg.ca.sandia.gov/jess/ (2004).

Kelso, M. T., P. Weyhrauch and J. Bates. Dramatic presence. Presence: The J. of Teleoperators and Virtual Environments, 2(1):1-15 (1993).

Kim, B., S. S. Heragu, R. J. Graves and A. S. Onge. Intelligent agent based framework for warehouse control. Proceedings of the 37th Annual Hawaii

International Conference on System Sciences (HICSS'04), Big Island, HI, USA. pp. 30070.1. IEEE Computer Society, Washington, DC, USA (2004).

Kotz, D. and R. S. Gray. Mobile agents and the future of the internet. ACM Operating Systems Review, 33(3): 7-13 (1999).

Li, Y., P. Musilek and L. Wyard-Scott. Fuzzy logic in agent-based game design. Proceedings of Annual Meeting of the North American Fuzzy Information Processing Society, Banff, Alberta, Canada. pp. 734- 739. IEEE Computer Society, Washington, DC, USA (2004).

Lieberman, H. Letizia: An agent that assists web browsing. Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence (IJCAI`95), Montreal, Québec, Canada. pp. 924-929. Morgan Kaufmann publishers Inc, San Mateo, CA, USA (1995).

Lin, F., P. Holt, L. Korba and T. K. Shih. A framework for developing distance learning systems. International Conference on Advances in Infrastructure for Electronic Business, Science, and Education on the Internet (SSGRR 2001), L'Aquila, Italy (August 6-12, 2001).

Maes, P. Agents that reduce work and information overload. Communications of the ACM, 37(7): 31-40 (1994).

Moreau, L., N. Gibbins, D. DeRoure, S. El-Beltagy, W. Hall, G. Hughes, D. Joyce, S. Kim, D. Michaelides, D. Millard, S. Reich, R. Tansley and M. Weal. SoFAR with DIM Agents An Agent Framework for Distributed Information Management. In J. Bradshaw and G. Arnold, (Eds.), The fifth International Conference on the Practical Application of Intelligent Agents and Multi-Agent Technology (PAAM 2000), Manchester, UK. pp. 369-388. The Practical Application Company Ltd, UK (2000).

Ndumu, D. T., H. S. Nwana, L. C. Lee and J. C. Collis. Visualising and debugging distributed multi-agent systems. Third Annual Conference on Autonomous Agents, Seattle, Washington, USA pp. 326-333. ACM Press, New York, USA (1999).

Nwana, H. S. and N. Azarmi (Eds.). Software agents and soft computing: Towards enhancing machine intelligence, concepts and applications. Lecture Notes in Computer Science (Vol. 1198). Springer-Verlag, London, UK (1997).

Parise, S., S. B. Kiesler, L. S. Sproull and K. Waters. My partner is a real dog: Cooperation with social agents. Conference on Computer Supported Cooperative Work, Boston, MA, USA. pp. 399-408. ACM Press, New York (1996).

Parunak, H. V. D. Manufacturing experience with the contract net. In M. N. Huhns (Ed.), Distributed Artificial Intelligence. Pitman Publishing, London (1987).

Pivk, A. and M. Gams. Intelligent agents in e-commerce. Electrotechnical Review, 67(5): 251-260 (2000).

Russell, S. J. and P. Norvig. Artificial Intelligence: A Modern Approach. Prentice Hall, New York (1995).