An Overview of Mobile Agents in Mobile Computing

CSE-6345: Term Paper

Anitha Chennamaneni

Abstract

Mobile agent paradigm is an emerging and exciting paradigm for mobile computing applications. Reasons are the inefficiencies associated with more traditionally distributed systems such as client-server applications in terms of latency, bandwidth, vulnerability to network disconnection, mobility etc. Mobile agent technology helps design wide range of adaptive, flexible applications with non permanent connections by adding mobility to code, machine based intelligence, improved network and database possibilities. This paper provides a comprehensive overview of mobile agents in mobile computing. Mobile agents, their usage, applications, architectures, languages, existing technologies and implementation challenges such as security, portability, scalability, standardization etc. are addressed

Introduction

As the trend towards wide-area open networks like the Internet and intranets grows larger, an increasing number of people are expected to use partially connected mobile devices such as laptops, mobile phones, personal digital assistants(PDA) home and business computers etc. to realize the benefits of having their electronic work available from anywhere and at any time. However, developing distributed applications that can function in a mobile computing environment is difficult for various reasons. First, mobile devices are not permanently connected to the network and often disconnect from it for long periods of time. Second, even when connected, the connection often has low bandwidth, high latency and prone to sudden network failures. Third, every time the mobile device reconnects, the network address assigned to it may change. Any distributed application that need to effectively use internet resources from a mobile platform must deal with the environmental changes and intolerant network conditions. Mobile agents because of their features such as autonomy, mobility, asynchronous communication, flexible query processing, intelligence, cooperation, reactivity etc. not only support mobile computers and disconnected operations, but provide a convenient, efficient and

robust programming paradigm for implementing distributed applications. This paper will provide a comprehensive overview of mobile agents in mobile computing. In particular, the paper will address what mobile agents are, their usage, advantages, applications, existing technologies, languages, architectures and implementation challenges such as security, portability etc.

Mobile Agents

A mobile agent is a program that is autonomous and can move through a heterogeneous network under its own control, migrating from a host to host and interacting with other agents. It decides when and where to migrate. It can execute at any point or suspend its execution, move to another host and continue its execution on that host.

Mobile agents have certain features such as autonomy, mobility, goal driven, temporarily continuous, intelligence, cooperation, learning, reactivity etc. Because of these features, they are well adapted to the domain of mobile computing. For instance, a mobile agent can move from a PDA to Internet to collect interested information for the user. Since it is on the network and does not have to transfer the multiple requests/responses across the low bandwidth connection, it can access necessary resources efficiently. Further, sudden connection losses will not affect the agent since it is not in continuous contact with the mobile device. An agent can perform its tasks even if the mobile device is disconnected from the network. Upon the reconnection of mobile device to the network, agent will return to it with results. Alternatively, a network application can dispatch a mobile agent onto the mobile device. The agent acting on behalf of the application interacts with the user regardless of whether or not mobile device is connected.

Mobile agents simplify the development, testing and implementation of distributed applications because of their ability to hide the communication channels and show the computation logic. They can distribute and redistribute themselves throughout the network and can act as either clients or servers depending on their goals. They can also increase the scalability of the applications because of their ability to move work to an appropriate location.

Advantages of Mobile Agents

Mobile agents have several advantages over traditional client/server model. Some of these include

Reduction of network traffic

In distributed systems, performing a simple job involves multiple interactions, resulting in increased network traffic. In mobile agent paradigm, the objective is to move the computation to the data rather than the data to the computation, thus consuming fewer network resources and thereby increasing efficiency.

Overcome network latency

Management of critical real time systems with substantial size networks creates latencies, which are unacceptable. Deployment of mobile agents will overcome this problem since these agents can execute locally upon the central controller's directions.

Encapsulation of protocols

Often, as new communication protocols which improve efficiency and security emerge, businesses need to upgrade their protocols, other wise they turn into a legacy problem. This whole process is cumbersome. Mobile agents will resolve this problem, as they can migrate to a remote host and establish channels based on proprietary protocols

Asynchronous and autonomous execution

For jobs that require a continuous open connection, deployment of mobile agents will result in cost savings. Mobile agents with embedded tasks can be dispatched into the network where they operate independently and asynchronously.

Dynamic adoption

Mobile agents can perceive the surrounding environment and can act dynamically.

Seamless system integration

Both from the perspectives of hardware and software, networking computing are heterogeneous. Mobile agents can provide seamless system integration since they are dependent only on those environments in which they execute.

Robust and fault-tolerant

Since mobile agents can act and react dynamically in presence of unfavorable conditions, it is easy to build a robust and fault-tolerant distributed system with mobile agents.

Some of the alternative techniques such as RPC, proxy servers etc can also provide many of the above advantages. However, each of these techniques is only suited for certain applications. In contrast, a mobile agent, which has a single unified framework is suitable for a wide range of distributed applications, thus making it an appealing solution for distributed applications.

Applications of Mobile Agents

Several applications benefit from the use of mobile agent technology. Some of these include

Electronic Commerce

Many commercial transactions require access to resources in real time. The ability of a mobile agent to personify their creators intentions and to act and negotiate on behalf of them makes it well suited for electronic commerce.

Personal Assistance

An agent can act as a personal assistant to the user and perform tasks for user on a remote host regardless of whether or not user is connected to the network. For instance, to schedule a meeting, a user can dispatch a mobile agent onto the network to interact with

agents belonging to other users. The agent can negotiate with other agents the convenient time for all of the users and can schedule a meeting.

Secure brokering

Mobile agent technology is an attractive solution to brokering, particularly in the context of untrustworthy collaborators. In such a situation, the interested parties can let the agents meet and negotiate at a mutually agreed impartial secure host and form alliance

Distributed information retrieval

Mobile agent technology provides efficient information retrieval. When dealing with large amounts of data, rather than moving all the data to search engine to create search indexes, user can simply dispatch mobile agents to remote sources to create those indexes locally and to ship them back later to its origin.

Telecommunication networks services

Mobile agents provide an effective and flexible solution to the management of advanced telecommunication services by providing dynamic network reconfiguration and user customization

Workflow applications and groupware

Mobile agents because of their features such as autonomy, mobility etc, provide autonomy to the workflow item and support the information flow between co-workers

Monitoring and notification

As a local representative for remote services, an agent can perform tasks on behalf a user irrespective of whether or not user is connected to the network. For instance, a user can dispatch a mobile agent to the internet to monitor the stock prices and to notify him/her only when certain thresholds are reached

Information dissemination

Mobile Agents, exemplifying the Internet Push Model can distribute information such as news and software updates for vendors. The agents carry the upgrades and installation procedures directly to the user's personal computer and with out any user's intervention updates and manages the software on the computer

Parallel processing

Mobile Agent technology can provide administration for parallel processing tasks. If a computation requires large amount of processing power so as it to distribute it among multiple processors, mobile agents can help get the processes out there.

Configuration of Mobile Agents

The tasks that an agent can accomplish depend on how the agent is configured in the first place. Some of the configurations of an agent include

Simple reflex agent - In this case, agent does not have memory. It simply perceives the current situation, finds a rule that matches the situation and executes it

A reflex agent with internal state - Agent perceives the current situation and based on its perception and stored internal state finds a rule that matches the situation and executes it

An agent with explicit goals: - Agent selects those actions that help it achieve goals. Compared to simple reflex agent, this agent is more flexible

A utility-based agent – While achieving its goal, this agent also maximizes some performance measure.

Architectures

The domain's characteristics determine what type of an agent system to be employed. These characteristics include the number of agents needed, the assigned time, communication and failure costs, involvement of the user, uncertainty of the environment and the probability of arriving at the goals dynamically. A system can have single or multiple agents.

Single Agent System

As the name suggests, a single agent system employs only one agent. This type of a system can become more complex than multiple agent systems, if the job assigned to an agent is complex. For instance, in multiple agent systems, control can be distributed and a single agent does not have to complete the given job. On the other hand, in a single agent system, an agent can be unnecessarily burdened if it has to complete all the tasks. Single agent architecture is better suited for those systems whose domain requires centralized control.

Multi-agent systems

Deployment of multi-agent architecture benefits a system in many ways. In these systems, various independent agents handle separate tasks. Provision of parallel processing capabilities and the presence of redundant agents increases system's operations and robustness. Even if one or more agents fail, a system can still work, since agents share responsibilities. Another advantage of multi-agent system is scalability. It is easier to add new agents to a multi-agent system than to single agent system.

Programmers can easily decompose a system into multiple tasks and assign these tasks to different agents. Hence from the programmer's perspective, it is relatively easy to program in these systems. Multi-agent system architecture suits those systems such as e-commerce, where criterion changes across agents or over time.

Multi-agent systems can be of different forms

Homogenous non-communicating multi-agent system

In such a system, all agents are configured with same goals, knowledgebase, possible actions and decision procedures. The differences lie in of how they receive the inputs, where they are located and how they take actions. Although these agents are similar, their knowledge of other agent's internal state is limited. Therefore, they cannot predict other agent's action. Most of the information filtering agents (Meta-Crawler) employ this type of architecture.

Heterogeneous non-communicating multi-agent system

The agents employed in this type of architecture are configured with different goals, domain models, actions, decision procedures etc. These agents can be either benevolent or competitive, i.e. they can cooperate with each other to accomplish their mutual goals or they can actively obstruct each other's actions. Cooperation is a significant issue in that it helps the cooperating agents build trust for each other. In a heterogeneous system, with out communication, the agents may not know the goals, knowledge base and decision procedures of other agents. Therefore, they will have to model themselves through observation.

Heterogeneous communicating multi-agent system

In this type of architecture, the employed agents differ in terms of their goals, domain models, actions, decision procedures etc. However, communication among these agents helps in effective coordination and resource allocation. Agents that have same goals but different abilities can be organized into a team. In such a team, each agent must be assigned a role, especially if they are specialized. The organization of teams can be along functional lines or formed dynamically for the duration of a task. The architecture for an e-commerce business needs to be flexible. It needs to have agents that are trustworthy and cooperative with the requisite knowledge to accomplish the tasks within the larger problem-solving framework. This can be accomplished by employing heterogeneous communicating multi-agent architecture.

Languages

A number of new and open technologies such as distributed objects, Java and extensible markup language (XML) can be used to implement mobile agents.

One of the popular languages for implementing mobile agents is Java. Infact, Concordia, Odyssey and Voyager are all implemented in Java. Some of the features of Java such as multi-platform support, object serialization, networking support which includes sockets, URL communication, distributed object protocol called remote method invocation (RMI) etc makes it extremely suitable for mobile agent technology. Applets can be used to launch and receive mobile agents. XML documents can be used to publish anything on the internet. It is one of the most popular accepted foundation layer on which

to build. It can be used to encode data with meaningful structure and semantics that any agent with proper authorization can easily access, understand and interpret.

Notable Mobile Agent Systems

Java applets are the most known examples of mobile code. However, java applets are not really mobile agents because they only migrate once and that too upon the user's request before they get executed. Mobile agents are much more powerful than that as they can migrate and execute at their own will. True mobile agent systems include Telescript, IBM Aglets, Odyssey, Concordia, Voyager etc, Agent Tcl, Tacoma, Mobile Service Agents, Ara etc. Currently, Telescript agents are used in managing networks, active e-mail, electronic commerce, business process management etc. A Telescript agent in network management may bring a software upgrade onto a user machine, install the code with out any intervention from user and disappear. A telescript agent in e-commerce may leave the user's mobile device, roam on internet, acting on user's behalf, collect the material and return to the users mobile device with the results. Tacoma is largely used in distributed weather simulation systems where immense volumes of data is handled. Mobile Service Agents are primarily used in computing applications where there is a need for the application to move to the user's machine. One example of that is electronic conference proceedings, where an agent is sent to the users machine. User participates in the proceedings by interacting with the agent. Agent Tcl is used mostly in information retrieval applications such as searching medical records, three dimensional drawings of mechanical parts etc and also in workflow applications such as performing a multi-step task at multiple locations by interacting with user .

Challenges in implementing Mobile Agents

Security

Security is one of the major issues that need to be considered in the implementation of mobile agents. One of the properties of a mobile agent is that it can roam on an inter-galactic network and can execute its code on a foreign server system. This property also makes it vulnerable to malicious attacks from other agents and servers. There are two

broad areas in mobile agent security: protection of host nodes from malicious agents and protection of agents from destructive hosts

Protection of hosts from malicious agents

Since mobile agent system is an open system, it can easily be attacked. Attacks can be in the form of leakage, tampering, stealing of resources and vandalism. The host executes mobile agent's code. Therefore a mobile agent has access to the resources of a host. This access gives the mobile agent the power to attack other local agents, generate viruses, worms or deny the services to other agents etc. Several researchers in this area have provided only a partial solution. Some of the solutions provided include authentication, verification, authorization, digital signatures, provision of hooks where an encryption system may be added, enforcement of access restrictions such as sandbox architecture, proof carrying code or some other restrictions like CPU or memory usage, . However, to achieve high level security, one needs to forfeit some level of flexibility or incur expenses. System designers need to balance these trade-offs carefully.

Protection of Agents from malicious hosts

It is much more difficult to protect an agent from malicious hosts than to protect a host from malicious agents. Since hosts execute the mobile agents, they can see the agent's code and data. Hence it makes easier for the host to tamper the agent's code or terminate them. Some of the ways to protect the agents from the host include limiting the amount of confidential data supplied to the agent, routing them in secure zones, use of cryptographic algorithm or tamper proof hardware, enforcing good host behavior etc. However, employing some of these techniques may compromise the concept of open mobile agent system.

Portability and Standardization

Mobile agent technology allows program to migrate freely from one host to another among heterogeneous machines. The code compiled into some sort of platform independent representation, such as java byte codes is either executed inside an interpreter or compiled into native code, when it arrives at the target machine. This code

if not, portable across mobile-code systems, will limit the usage of mobile agents. Standardization is required to make this code portable across mobile-code systems. The OMG MASIF standards currently addresses only cross-system communication and administration leading to a state where an agent is forced to migrate to a near by machine that runs the right agent system rather than to the desired machine. Standards for specific execution environments as well as the format and state of the encoded migrating agent need to be addressed

Performance and scalability

For portability and security reasons, most current agents are written in a slow interpreted language. As a result, these agents save network latency and bandwidth at the expense of high loads on service machines. For instance, while mobile agents execute faster in network inconsistencies, the opposite is not true. When there are no network disconnections, mobile agents usually take much longer time to perform a task than their traditional counterparts, since the time that is saved from avoiding intermediate traffic is much less than the time involved in slow execution and migration overhead. With the surfacing of just-in-time compilation code such as java, software fault isolation etc some of the problems with the slower execution of mobile agent code are alleviated. However, several other problems involving migration overhead, slow execution etc still remain to be addressed

Conclusion

Mobile agent paradigm is considered to be an effective paradigm in the area of distributed programming. Although current trends in internet technologies indicate widespread usage of mobile agents in near future, several technical and non-technical challenges such as security still remain and need to be resolved. However, once these challenges are met, and some internet sites accept mobile agents, the usage of mobile agents will spreads widely, thereby revolutionizing mobile computing.

References

1. Elmarie Bierman & Elsabe Cloete, Information Security and Risk Management: Classification of malicious host threats in mobile computing, Proceedings of the 2002 annual research conference of the South African institute of computer scientists and information technologists on Enablement through technology
2. Akhil Sahai, Mobile Agents for Location Independent Computing, Proceedings of the 1999 ACM symposium on Applied Computing
3. Joris Claessens, Bart Preneel & Joos Vandewalle, How can mobile agents do secure electronic transactions on untrusted hosts? A survey of the security issues and the current solutions, ACM Transactions on Internet Technology
4. Danny B. Lange, Mobile Objects and Mobile Agents: The Future of Distributed Computing
5. David Wong, Noemi Paciorek, and Dana Moore, Java-based Mobile Agents, Communications of the ACM Vol.42 No.3(pp92-102)
6. Pattie Maes, Robert H.Guttman and Alexandros G. Moukas, Agents that Buy and Sell, Communications of the ACM Vol.42 No.3(pp81-91)
7. S.Bandyopadhyay, Krishna Paul, Evaluating the performance of mobile agent-based message communication among mobile hosts in large ad hoc wireless network, August 1999, Proceedings of the 2nd ACM international workshop on Modeling, analysis and simulation of wireless and mobile systems
8. Akhil Sahai & Christine Morin, Mobile agents for enabling mobile user aware applications, Proceedings of the second international conference on Autonomous agents
9. Paolo Bellevista, Antonio Corradi & Cesare Stefanalli, A mobile agent infrastructure for the mobility support, Proceedings of the 2000 ACM symposium on Applied Computing
10. Sheng Zhong & Yang Richard Yang, Verifiable distributed oblivious transfer and mobile agent security, Proceedings of the 2003 joint workshop on Foundations of mobile computing
11. Jaco van der Merwe and S.H. von Solms, Electronic commerce with secure intelligent trade agents, Computers & Security Vol.17 No 5(pp 435-446)

12. David Kotz, George Cybenko, Robert S. Gray, Guofei Jiang, Ronald A. Peterson, Martin O. Hofmann, Daria A. Chacón, Kenneth R. Whitebread & James Hendler, Performance analysis of mobile agents for filtering data streams on wireless networks, April 2002, Mobile Networks and Applications
13. Luc Moreau, Agents, interactions, mobility and systems: A fault-tolerant directory service for mobile agents based on forwarding pointers, March 2002, Proceedings of the 2002 ACM symposium on Applied computing
14. Mieso K. Denko, The use of mobile agents for clustering in mobile adhoc networks, September 2003, Proceedings of the 2003 annual research conference of the South African institute of computer scientists and information technologists on Enablement through technology
15. Pattie Maes, Robert H.Guttman and Alexandros G. Moukas, Agents that Buy and Sell, Communications of the ACM Vol.42 No.3(pp81-91)
16. Danny B.Lange and Mitsuru Oshima, Seven good reasons for mobile agents, Communications of the ACM Vol.42 No.3(pp88-89)
17. Howard Smith and Kevin Poulter, Share the ontology in XML-based Trading Architectures Communications of the ACM Vol.42 No.3(pp110-111)
18. Catholijn M. Jonker and Jan Treur, A Re-usable Broker Agent Architecture with Dynamic Maintenance Capabilities, Autonomous Agents '99(pp376-377)
19. Moses Ma, Agents in E-commerce, Communications of the ACM Vol.42 No.3(pp79-80)
20. Maria Gini, Agents and other Intelligent Software for e-commerce, http://www-users.cs.umn.edu/~gini/csom.html
21. John Collins, Ben Youngdahl, Scott Jamison, Bamshad Mobasher, Maria Gini, A Market Architecture for Multi-Agent Contracting, Autonomous Agents 98, ACM(pp285-292)
22. Hyacinth S. Nwana, Jeff Rosenschein, Tuomas Sandholm, Carles Sierra, Pattie Mass, Rob Guttmann Agent Mediated Electronic Commerce:Issues, Challenges and some View Points(pp189-196), Intelligent Systems Research