Available at: www.researchpublications.org

Interaction of Mobile Agents and Web Services

Satish B. Belsare
Smt. R.S. College of Arts, Comm. and Sci Anjangaon Surji
satishbelsare@gmail.com

Swati. S. Sherekar
Shri Sant Gadge baba Amravati University, Amravati
ss_sherekar@rediffmail.com

Vilas M. Thakare
Shri Sant Gadgebaba Amravati University, Amravati
vilthakare@yahoo.co.in

ABSTRACT
Mobile Agent approach has emerged as a promising way to address the new challenges in the design and implementation of distributed systems. Web services are widely used in current web-based business applications; the specification represents an open standard for distributed service. Mobile agents provide very specific advantages with respect to dynamic service, execution, as well as the extension of server-sided services with client-side intelligence and functionality. The main objective of this paper is the interaction of mobile agent and web services with respect to the Client-Server paradigm which facilitate flexible behaviors of mobile agents in web service applications.

Keyword
Mobile Agent, Web Services.

1. INTRODUCTION
Mobile Agents (MAs) have recently emerged with the goal of allowing application developers to easily manage and control the properties and behaviors of mobile systems, especially in novel wireless and ubiquitous scenarios [1] [2]. The primary consequence is that one MA platform tends to offer middleware facilities scarcely compatible with other MA systems. MA tools are sometimes difficult to employ by common users who have to overcome a significant knowledge gap before properly exploiting MA-specific functions. The paper proposes a solution to open up the usage of MA systems via the bridge of the emerging Web Services (WS) standard technology to achieve interoperability.

In fact, mobile agents have several advantages in the development of various services in smart environments in addition to distributed applications [3].

• **Reduced communication costs:** Distributed computing needs interactions between different computers through a network. The latency and network traffic of interactions often seriously affect the quality and coordination of two programs running on different computers.

• **Asynchronous execution:** After migrating to the destination-side computer, a mobile agent does not have to interact with its source-side computer. Therefore, even when the source can be shut down or the network between the destination and source can be disconnected; the agent can continue processing at the destination. This is useful within unstable communications, including wireless communication, in smart environments.

• **Direct manipulation:** A mobile agent is locally executed on the computer it is visiting. It can directly access and control the equipment for the computer as long as the computer allows it to do so. This is helpful in network management, in particular in detecting and removing device failures. Installing a mobile agent close to a real-time system may prevent delays caused by network congestion.

• **Dynamic-deployment of software:** Mobile agents are useful as a mechanism for the deployment of software, because they can decide their destinations and their code and data can be dynamically deployed there, only while they are needed. This is useful in smart environments, because they consist of computers whose computational resources are limited.

• **Easy-development of distributed applications:** Most distributed applications consist of at least two programs, i.e., a client-side program and a server side program and often spare codes for communications, including exceptional handling. However, since a mobile agent itself can carry its information to another computer.

2. MOBILE AGENTS
Mobile agents are the technology that makes it very much easier to design, implement, and maintain distributed systems. Mobile agents reduce the network traffic, provide an effective means of overcoming network latency, and perhaps most importantly, through their ability to operate asynchronously and autonomously of the process that created them, and help to construct more robust and fault-tolerant systems [4].

2.1 Classification
There are different types of agents as follows:

■ Collaborative agents
- These emphasize autonomy and collaboration with other agents to perform their tasks.
- They may need to have "social" skills in order to communicative and negotiate with other agents.

■ Interface agents
- Emphasize autonomy and learning in order to perform useful tasks for their owners.

■ Mobile agents
- Mobile agents move from system to system.

■ Information/Internet agents
- Focus on helping us to cope with the sheer "tyranny of information" in the internet page.
- Help to manage, manipulate or collate information from many distributed sources.

■ Reactive agents
- A reactive agent consists of many agents,

a) Each of which has a very simple stimulus-response type behavior.

b) A single agent typically has no clue about the actual task to be performed, but the collective action of the group has an emergent behavior which causes the required task to be accomplished.

■ Hybrid agents

• Hybrid agents are those which combine more than one philosophy within the same agent.

■ Smart agents

• SMART Agents are a new form of software agent that interfaces with other agents forming an artificial intelligence system. The acronym "SMART" stands for "System for Managing Agents in Real Time"

2.2 Mobile Agent Behavior

Mobile agents are processes (e.g. executing programs) that can migrate from one machine of a system to another machine (usually in the same system) in order to satisfy requests made by their clients as shown in fig1.

Before the mobile agent paradigm appeared, many approaches have been proposed and developed for communication between client and server such as, Message Passing (MP), Remote Procedure Call (RPC) and Remote Evaluation (REV). In RPC, the client sends data as parameters to a procedure that resides at the server. The procedure will be executed on the server and the results will be sent back to the client. The REV is a different architecture from RPC. Instead of calling a remote procedure at the sever side, the procedure itself will be sent from the client to the server to be executed and returns the result. Briefly, in RPC the data is transmitted between the client and server in both directions. In REV, code is sent from the client to the server, and the data is returned. In contrast, a mobile agent is a program (encapsulating code, data, and state) sent by a client to a server [5] [4].

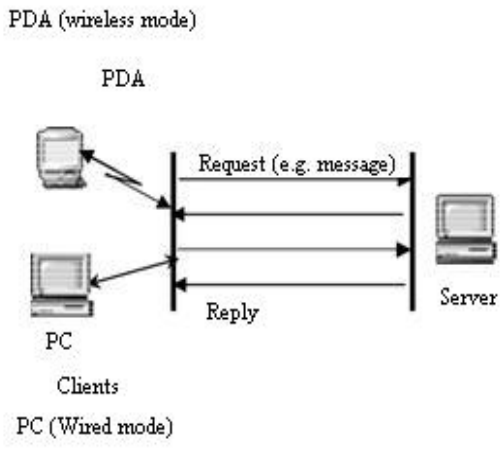

Fig1: Client-server paradigm

2.3 Mobile Agent Components

A mobile agent contains the following 3 components:

• Code - the program (in a suitable language) that defines the agent's behavior.

• State - the agent's internal variables etc., which enable it to resume its activities after moving to another host.

• Attributes - information describing the agent, its origin and owner, its movement history, resource requirements, authentication keys etc. Part of this may be accessible to the agent itself, but the agent must not be able to modify the attributes. The following fig2 shows the mobile agent component and mobile agent paradigms.

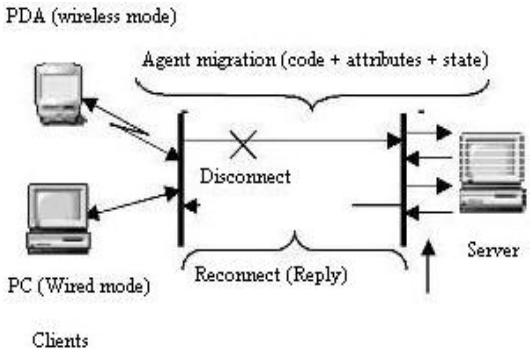

Fig 2: Mobile agent paradigm

2.4 Mobile Agent Services

Mobile services differ from traditional services in their ability to provide service offerings regardless of temporal and spatial constraints. They are also different from traditional interpersonal services that deliver face-to-face, or from other types of e-services, such as wireless online services, where the service delivery linked to a specific fixed local area network or specific location. Mobile services have some special characteristics in comparison to other types of services [6] [8].

The key differences related to spatial and temporal components of service usage. For example, if one wants to meet a bank teller, she has to visit the bank location at a certain appointment time. These restrictions are present to some extent even with many electronic services. Even though online banking over a DSL Internet connection. Even though online banking services are available 24 hours a day thus overcoming the problem of temporal availability, a fixed location is still needed for the DSL line, which is a spatial restriction [6] [9]. Mobile services, used with handheld mobile devices, overcome both spatial and temporal constraints an additional unique dimension of mobile services is the potential for individual personalization of service offerings.

Mobile handsets are multi-purpose private computers. This provides an attractive setting for electronic service delivery. Evaluation of mobile services as perceived by end-users is therefore challenging. Some papers claim that the opportunity cost of time is what matters. Some analyses also recognize the dimension of level of effort in using the service.

3. WEB SERVICES

A Web service is a method of communication between two electronic devices over the World Wide Web. A Web service is a software function provided at a network address over the web or the cloud, it is a service that is "always on" as in the concept of utility computing.

The term Web services describes a standardized way of integrating Web-based applications using the XML, SOAP, WSDL and UDDI open standards over an Internet protocol backbone. XML is used to tag the data, SOAP is used to transfer the data, WSDL is used for describing the services available and UDDI is used for listing what services are available. Used primarily as a means for businesses to communicate with each other and with clients, Web services allow organizations to communicate data without intimate knowledge of each other's IT systems behind the firewall. Unlike traditional client/server models, such as a Web server/Web page system, Web services do not provide the user with a GUI. Web services instead share business logic, data and processes through a programmatic interface across a network. The applications interface, not the users. Developers can then add the Web service to a GUI (such as a Web page or an executable program) to offer specific functionality to users. Web services allow different applications from different sources to communicate with each other without time-consuming custom coding, and because all communication is in XML, Web services are not tied to any one operating system or programming language. For example, Java can talk with Perl; Windows applications can talk with UNIX applications. Web services do not require the use of browsers or HTML. Web services are sometimes called application services. The architecture of web services is shown in fig3.

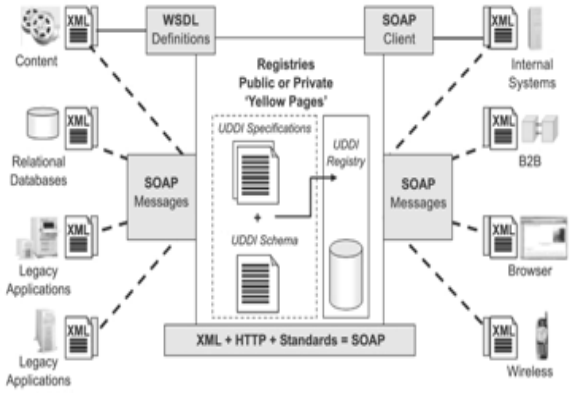

Fig3: Architecture of Web Services

4. INTERACTION OF MOBILE AGENT AND WEB SERVICES

In the last few years, Researchers have investigated the possibility of integrating Mobile Agent (MA) systems and Web Services (WS) components. One of the most relevant differences between MA systems and WS is the nature of the interactions between their components. While WS adhere to a typical synchronous paradigm in which a client requests service execution to a server endpoint, MAs interact with clients by following a typical asynchronous behavior: after creation, MAs can migrate between network nodes and interact with their needed resources, with no necessity to maintain continuous connectivity with the associated client. The functions of Web Services and Mobile Agent Systems as follows [7] [1]:

4.1 Mobile Agents to Web Services (MA2WS)

WSMI proxies play the role of static WS clients/servers [10] [11]:

- receive WS requests from MAs;
- deliver requests to WS providers;
- receive service responses from WS providers;
- Forward responses to the current location of MA requestors, by possibly following different delivery models.

Different types of request delivery channels are provided to better fit MA and/or proxy needs, as shown in Fig4. The first uses HTTP and performs a typical synchronous blocking interaction: MAs send their requests via SOAP messages and wait until proxies return back receipt confirmations, again via SOAP messages. After confirmation, proxies start operating, by forwarding requests to actual WS providers. The second model, instead, uses the TCP communication protocol and provides a typical asynchronous non-blocking interaction: MAs send requests but proxies do not send any confirmation; MAs are not blocked and proxies can forward requests immediately after their reception.

4.2 Web Services to Mobile Agents (WS2MA)

WS2MA exposes a set of services via a WS interface to provide external users with a standardized way to access relevant MA system functions.

• **browse descriptive MA services**. It include lists of all agents available for invocation in the specified topology, while lists of all agents visible to token-identified user and hosted by places defined in the specified service topology.

• **drill-down descriptive MA services**. They include services to monitor the current MA position and the current MA status. For synchronous MAs, there is also the service that returns the MA activity results.

• **Active MA services**. It creates an MA instance in the specified place (by starting its execution if specified), passing a user-defined initialization string. It is also possible to specify the type of visibility for bounded visibility MAs and MA semantics for synchronous agents, e.g., the mode exploited to return activity results.

• **browse descriptive topology services**. They comprise two services: one is the lists of all domains composing the MA system topology, while another is the lists of all places in a specified domain. Only globally recognizable users can invoke the first service, while users invoking the second must be recognizable either globally or at least by the specified domain.

• **drill-down descriptive topology services**. They include the Place Description and Domain Description services that describe topology entities defined in the MA system.

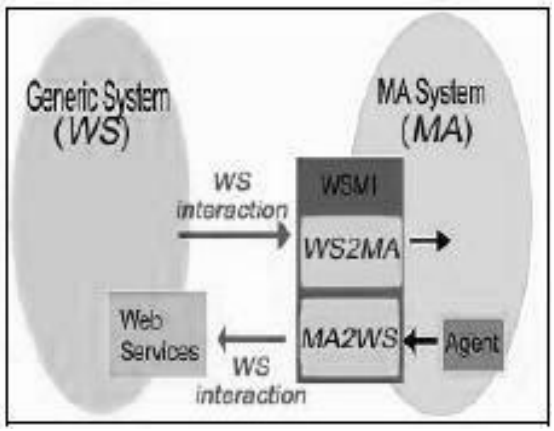

Fig4: Interaction of WS2MA and MA2WS.

• **Aggregate services**. Two service aggregates are defined. The first returns information about the current status, position, and activity results of a specified MA, while the second returns high-level (browse services) topology information and detailed (drill-down services) topology information. These services perform a breadth-first access to domain topology trees to obtain detailed descriptions about domains/places.

5. CONCLUSION

Mobile Agents are better support to the client-server paradigm which include the several advantages in the interaction of various web services such as reducing network load, overcome network latency, encapsulate protocols, execute asynchronously and autonomously, adapt dynamically, naturally heterogeneous and robust and fault-tolerant. The main conclusion of this paper is Mobile Agents are interacting with Web Services very easily in both ways like Mobile Agents to Web Services and Web Services to Mobile Agents which increase the performance of the network and the mobile agent behavior. This provides dynamic and flexible services for the users. There are different ways of improving the performance of mobile agent behavior. In the future, researcher needs to focus on the aspect of mobile agent technology and its services using temporal mining pattern to provide better mobile applications and services.

REFERENCES

[1] N. M. Karnik, A. R. Tripathi, “Design Issues in Mobile Agent Programming Systems”, IEEE Concurrency, Vol. 6, No. 3, pp. 52-61, July-Sep. 1998.

[2] Paolo Bellavista Antonio Corradi University of Bologna, Cesare Stefanelli University of Ferrara “Mobile Agent Middleware for Mobile Computing”, 0018-9162, March 2001.

[3] Ichiro Satoh: "Selection of Mobile Agents", Proceedings of 24th IEEE International Conference on Distributed Computing Systems (ICDCS'2004), pp.484-493, IEEE Computer Society, March 2004.

[4] Syed Adnan. John Datuin. Pavana Yalamanchili, “A Survey of Mobile Agent Systems “, CSE 221. June 13, 2000.

[5] S.M.Dejan, W. LaForge, D. Chauhan, “Mobile objects and agents (MOA)”, Proceedings of USENIX COOTS'98, Santa Fe, New Mexico, USA, April 1998.

[6] Saputra, R.E.; Wahjuni, S.; , "Mobile agent implementation in location-based services," Advanced Computer Science and Information System (ICACSIS), 2011 International Conference on , vol., no., pp.47-50, 17-18 Dec. 2011.

[7] B. Emako, R. H. Glitho, S. Pierre, “A Mobile Agent-Based Advanced Service Architecture for Wireless Internet Telephony: Design, Implementation, and Evaluation”, IEEE Transactions On Computer, Vol.52, No. 6, pp. 690-705, June 2003.

[8] Singh, R.; Dave, M.; , "Rescuing data of mobile agents blocked by malicious hosts in e-service applications," Multimedia, Signal Processing and Communication Technologies (IMPACT), 2011 International Conference on , vol., no., pp.24-27, 17-19 Dec. 2011.

[9] Gnanasekar, J.M.; Pillai, V.D.; , "A Novel Software Environment for Developing Migrating Internet Applications Based on Fusion of Mobile Agent, Web Services and BPEL Technologies," Services Computing (SCC), IEEE International Conference on , vol., no., pp.651-652, 5-10 July 2010.

[10] J.L.Chen," Resource allocation for cellular data services using multiagent schemes", IEEE Trans. Syst. Man Cybern. Vol. 31, Pages 864-869, 2001.

[11] C.Y Chang, M.S Chen, "Integrating web caching and web prefetching in client-side proxies", Vol. 16, page(s), 444-455, IEEE press,2005.

[12] P. Dasgupta, N. Narasimhan, L.E. Moser, P.m. Melliar-Smith, “Mobile agents for networked Electronic Trading”, IEEE Transactions on Knowledge and Data Engineering, Special Issue 6, pp. 509-525, July/August 1999.

[13] Chao-Lin Wu, Chun-Feng Liao, and Li-Chen Fu,”Service-Oriented Smart-Home Architecture Based on OSGi and Mobile-Agent Technology”, IEEE, Vol. 37, No. 2, pp. 193-205, MARCH 2007.