International Journal of Multidisciplinary and Current Research *ISSN: 2321-3124*
Research Article *Available at: http://ijmcr.com*

Intelligent Middleware Agent for Mobile Computing

Sameer

Shah Satnam Ji Boys' College, Sirsa, India

Received 12 Oct 2017, Accepted 15 Dec 2017, Available online 28 Dec 2017, **Vol.5 (Nov/Dec 2017 issue)**

Abstract

Today there is a significant amount of efforts being put on to the development of Mobile-code technologies which already address the problems that stem from changing the allocation of executing entities. This has led to the development of mobile agents. Guidelines drawn from these solutions help identify and fulfill mobile computing requirements. Secure and Open Mobile Agent's user virtual environment, mobile virtual terminal, and virtual resource management service layer provides a coordinated and flexible middleware that application designers can use to design and deploy Internet services that operate in an environment where users, terminals, resources, and services are all mobile.

Keywords: *Mobile Computing, security, virtual environment, Resource management.*

1. Introduction

The integration of the Internet with telecommunication networks promises a distributed computing infrastructure that provides globally available services. The Internet's diffusion permits an almost ubiquitous availability of attachment points, allowing users access to its information services irrespective of their location. In addition, advances in cellular telecommunications and device miniaturization let increasing numbers of portable devices connect to the Internet. Mobile computing requires an advanced infrastructure that integrates suitable support protocols, mechanisms, and tools. This mobility middleware should dynamically reallocate and trace mobile users and terminals and permit communication and coordination of mobile entities. In addition, open and untrusted environments must overcome system heterogeneity and grant the appropriate security level. Solutions to these issues require compliance with standards to interoperate with different systems and legacy components and a reliable security infrastructure based on standard cryptographic mechanisms and tools. Many proposals suggest using mobile agent technology middleware to address these issues.[l] A mobile agent MA) moves entities in execution together with code and achieved state, making it possible to upgrade distributed computing environments without suspending service. Finally, study tells about user virtual environment, mobile virtual terminal, and virtual resource management.

User virtual environment provides users with a uniform view of their working environments independent of current locations and specific terminals. Mobile virtual terminal extends traditional terminal mobility by preserving the terminal execution state for restoration at new locations, including active processes and subscribed services. Virtual resource management permits mobile users and terminals to maintain access to resources and services by automatically requalifying the bindings and moving specific resources or services to permit load balancing and replication.

2. Mobile agents in mobile computing

Mobile computing benefits from the synchronicity between user requests, terminal operations and their execution. For example, wireless connections impose strict constraints on available bandwidth and communication reliability of minimizing connection time for wireless device support. The MA paradigm does not need continuous network connectivity because connections last only long enough to inject agents from mobile terminals into the fixed network. With autonomous agents, users can access services even if the terminal disconnects because the agents deliver the results upon reconnection. Mobility middleware's location awareness facilitates service-specific optimization and allows users to adapt to local resources. Mobile users can change location and dynamically tailor mobility-enabled applications to the properties and characteristics of their network connections and hardware devices. Location awareness facilitates

Corresponding author's ORCID ID: 0000-0000-0000-0000
DOI: https://doi.org/10.14741/ijmcr/v.5.6.44

allocation visibility up to the application level supporting dynamic quality of service (QoS) adaptation to local needs.[2] hi addition, MA simplifies dynamic personalization by following user movements and tailoring service depending on personal preferences. Mobility stresses the importance of flexible and extensible middleware. The dynamic distribution of code typical of MA platforms requires new components and protocols to adapt to evolving service and user requirements. Mobile agents provide flexibility by moving code and preserving the state the computation produces. Mobility raises significant security issues for authentication of mobile users and terminals, authorizatkm to access system resources, and communications' secrecy and integrity assurance. After the pioneering work of IBM Aglets, recent MA platforms provide flexible mechanisms and policies to grant the most suitable security level. For example, many MA systems integrate with public key infrastructures, simplifying authentication of mobile users and terminals. To address similar problems, MA research promotes interoperable and standard interfaces.

Secure & open ma-based middleware

The secure and open mobile agent distributed programming framework (9] is a Java-based platform that provides a layered service infrastructure for designing, implementing, and deploying MA-based Internet applications. As Figure 1 shows, secure and open MA's architecture consists of four layers. The mobility middleware layer implements user virtual environment, mobile virtual terminal, and virtual resource services.

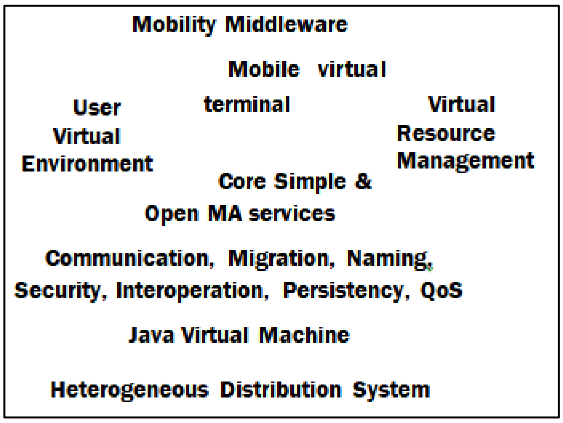
Mobility Middleware
Mobile virtual terminal
User Virtual Environment
Virtual Resource Management
Core Simple & Open MA services
Communication, Migration, Naming, Security, Interoperation, Persistency, QoS
Java Virtual Machine
Heterogeneous Distribution System

Figure 1 Simple & Open MA Architecture management

The core services layer includes communication, migration, naming, security, interoperability, persistency, and QoS adaptation services. The architecture's other two layers are a Java virtual machine and a heterogeneous distributed system. Secure and open MA offers locality abstractions to describe any kind of interconnected system, from simple intranet LANs to the Internet. Each node provides at least one environment for agent execution called place—an agent execution environment. Secure and open MA groups several places into domain abstractions that correspond to network localities. In each domain, a default place controls interdomain routing and integration with legacy components via Corba. The mobile place enhances the place locality abstraction with specific functions for automatic reconfiguration when changing domains.

A. *Secure and Open MA mobility services*

The secure and open MA's persistency service lets application designers and system administrators suspend agent execution by storing the agent's state on disk. The persistency service uses OMA's migration service to serialize both agent code and data, saving the information in persistent storage. Persistency minimizes the consumption of system resources while agents wait for a disconnected resource. In addition, persistency provides fault tolerance by duplicating and storing agent copies before starting critical operations. Mobile virtual terminal employs persistency to freeze and wake up agents and messages when user and terminal is connects and reconnects occur.

Secure and open MA naming derives from care-of mechanisms for locating mobile agents and places. The care-of for any mobile agent to be traced should be located where it was first created (agent home). Similarly, the care of for any mobile place is located at the instantiation domain's default place (place home).

Secure and open MA's middleware transparently updates agent homes at their migration and places homes at their connection or disconnection. Secure and open MA mobile agents and places have Guids independent of their current position. Guids consist of the corresponding home's identifier associated with a number unique in the home locality.

For example, a mobile place owns a Guid of the form Domain ID, progNumber where DomainID is the address of its place home. This solution permits immediate identification of the home, without querying the naming service.[3] In addition to providing basic naming mechanisms, secure and open MA middleware integrates a discovery protocol and an LDAP-based directory service. The discovery service provides the default solution for resource naming within a Secure and open MA domain.
A broadcast protocol registers and deregisters resources at the discovery server located at the default place.

The expected low frequency of resource migration and locality resource access suggest the choice of discovery. To date, secure and open MA has used a proprietary discovery protocol, but we are implementing an SLP- compliant solution.[4)

All entities that need global visibility register to the LDAP-based secure and open MA directory service. The LDAP directory server keeps its entity names and coordinates with other servers to maintain global consistency and to resolve external names. After migration, resources can override the default discovery solution and register with the secure and open MA directory for wider accessibility. Secure and open MA mobile agents implement the user virtual environment, mobile virtual terminal, and virtual resource management services. This facilitates distribution of the mobility middleware and simplifies enforcement of balancing and replication policies. In addition, the Java-based implementation of foe mobility middleware overcomes platform heterogeneity and applies to foe open Internet

Summary and conclusions

Currently available mobile-code technologies only partially address foe problems raised by changing the allocation of executing entities. The implementation of fois mobility middleware confirmed that a layered and modular MA-based service infrastructure can support a wide range of mobile computing requirements.

References

[1] E. Kovacs, K. Rohrlc, and M. Reich, “Integrating Mobile Agents into the Mobile Middleware,” Proc. Mobile Agents Int’l Workshop, Springer-Verlag, Berlin, 2005, pp. 124-135

[2] D. Kotz et al., “Agent TCL: Targeting the Needs of Mobile Computers,” Internet Computing, July/Aug.1997, pp. 58- 67

[3] P. Bellavista, A. Corradi, and C. Stefanelli, “Protection and Interoperability for Mobile Agents: A Secure and Open Programming Environment,” IEICE Trans. Comm., May 2000, pp. 961-972.