IMPLEMENTING MOBILE AGENT FOR LOCATION-BASED PRINTER SERVICES

Rendy Eka Saputra and Sri Wahjuni

Laboratory of Net-Centric Computing, Faculty of Mathematics and Natural Science, Bogor Agricultural University, Jl. Raya Dramaga, Bogor, 16680, Indonesia

E-mail: rendy@live.com

Abstract

Most devices in ubiquitous network environment have limited memory and processing power, thus they can't provide all services even when in suitable locations. Mobile agent technology could be implemented to improve resource efficiency. The goal of this paper is to implement mobile agent technology for location-based printer service. The system consists of four parts: (1) mobile agent server, (2) location-based service server, (3) printer service, (4) client. It is observed that mobile agent offers many advantages for ubiquitous and mobile computing settings.

Keywords: *location-based services, mobile agent, printer service, ubiquitous computing*

Abstrak

Perangkat dalam lingkungan *ubiquitous* memiliki memori dan daya proses yang terbatas, sehingga mereka tidak dapat menyediakan layanan secara utuh bahkan dalam lokasi yang tepat sekalipun.Teknologi *mobile agent* dapat diterapkan untuk meningkatkan efisiensi sumber daya. Tujuan dari *paper* ini adalah untuk melihat penerapan teknologi *mobile agent* pada lokasi berbasis layanan *printer*. Sistem ini terdiri dari empat bagian: (1) *mobile agent server*, (2) lokasi yang berbasiskan layanan *server,* (3) layanan *printer*, (4) klien. Hal ini menunjukkan bahwa *mobile agent* menawarkan banyak keuntungan untuk pengaturan komputasi *ubiquitous* dan *mobile*.

Kata Kunci: *lokasi berbasikan layanan, mobile agent, layanan printer,ubiquitous computing*

1. Introduction

According to Mark Weiser [1] and referenced by Ichiro Satoh [2], a goal of ubiquitous computing is to provide various service by making multiple computers available throughout the physical environment, but, in effect, making them invisible to the user. Another purpose of ubiquitous computing is to integrate physical world with cyberspace. In fact, today technology is able to detect location and position of someone or an object. Context-awareness, particularly user-awareness and location-awareness, become a very important feature of the services currently used in everyday life [2].

According to Ichiro Satoh [2], ubiquitous computing devices is not suitable to provide services that have multiple purposes and personalized services, because most of the devices tend to have limited storage capacity and processing power thus unable to maintain a variety of software and profile databases on the users.

However, Sawal [3] noted that the use of mobile agent technology could make resources used more efficiently. Resources can be more efficient because mobile agent could be implemented in a model where a program code sent to a data location or a node, that program code is a mobile agent. The code only gathers required data to be sent back to client. A mobile agent uses the network only for a relatively short period during this migration [4]. This process could reduce the use of resources.

To develop software with mobile agent technology JADE (Java Agent Development Framework) can be used. Jade is one the best agent environment. Jade is open source, follow the standard FIPA (Foundation for Intelligent Physical Agents) and works on many operating systems [5].

The goal of this paper is to implement mobile agent technology for location-based printer service. This section presents background information about concept used in this research.

This paper is the extended version from paper titled "Mobile Agent Implementation in Location-based Services" that has been published in Proceeding of ICACSIS 2011.

To outline the purpose of this paper, researcher presents a system scenario. A user goes into a building and for some reason he/she needs to print a document. On local area network (LAN) of the building there is a location-based services and printer service available free. So that user can connect to LAN and it will be served by an agent using mobile agent technology. Then the user can print his/her document with help of the agent.

Agent is basically a special software component that has a clear autonomy right that provides an interoperable interface to an arbitrary system. Mobile agent from the viewpoint of a distributed system is software that has unique characteristics, among others, can transfer program code, data and pointer in between machines on the network. To achieve this, mobile agents are able to suspend their execution at any time and to continue once resident in another location [6].

Agent has several properties [6] that are, first, autonomous is agent can operate without human intervention or others and have control over the action and the state itself. Second, social is agent can cooperate with humans or other agents in order to achieve its tasks. Third, reactive is agent perceives its environment and responds in a timely fashion to changes that occur in the environment. Fourth, proactive is agent does not simply act in response to its environment but it is able to exhibit goal-directed behaviour by taking initiative. Figure 1 is the design of a distributed system in which mobile agent technology implemented.

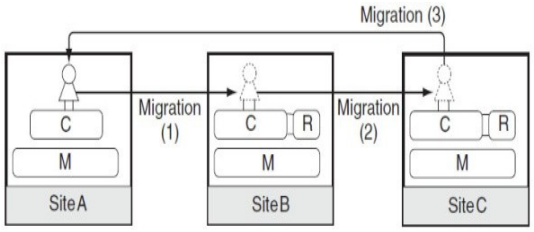

Figure1. Design of distributed system with mobile agent [7].

Program code (C) is a mobile agent that can move from one host to another host using resources (R) in machine (M) of the host. Mobile agent technology has several advantages in ubiquitous and mobile computing settings [2] that is, each mobile agent can move from computer to another under his own control. When a mobile agent move to another computer, code and state of the agent transferred to the destination. Each agent only needs to be on the device at the time when the device is required to offer the services provided by that agent. Therefore, mobile agent can save limited resources of computing devices.

Since each mobile agent is programmed entity, framework enables application-specific services, including the user interface and application logic, to be implemented within mobile agents. It then separates application-specific services from itself. Therefore, it can be a general infrastructure for a variety of location-aware services.

Using mobile agent in distributed systems also has several advantages [4] that are, first, a mobile agent is capable of autonomously achieving the goal it was created for on behalf its owner. It does not need intermediate involvement of the owner. This is important property if no reliable connection is possible between an agent and its owner. It results in a system that is much more robust than conventional distributed systems. Second, the services and resources that can be used by an agent change dynamically on large distributed systems. Existing services and resources may become unavailable, whereas new ones may come into existence during the lifetime of an agent. The autonomy of an agent enables them to choose the best alternative at the time they need it. Third, the mobile agent architecture has no central bottleneck components, which makes it very scalable. Fourth, the stationary components (service providers) can be simplified to only provide basic functionality. The program-logic required to refine the basic functionality into application specific functionality can be made part of the agent. By simplifying the stationary components, also their maintainability and genericity is increased. Fifth, a mobile agent uses the network only for a relatively short period during its migration. It is therefore capable of dealing with slow and intermittent available network links. Sixth, if the size of the program-logic needed for processing is small compared to the data that has to be processed, it makes sense to move the program-logic toward the data, instead mobbing the data towards the program logic. The required bandwidth can thus be reduced considerably. Seventh, because of their built-in "intelligence", agents can cope with syntactic and semantic inconsistencies that inevitably occur in large multi-vendor systems.

Java agent development framework (JADE) is a software development framework aimed at developing multi-agent systems and application conforming to FIPA (Foundation for Intelligent Physical Agents) standards for intelligent agents [8]. JADE provides a service platform called AMS (Agent Mobility Service), which implements the intra-platform mobility. It provides the ability for agents that can move from a container into another container within the same platform, so that JADE supports the development

of systems that require mobile agent technology. Figure 2 represents the main architectural elements of JADE.

JADE programmed with Java programming language. A JADE based application made up of several components, called agents that have a unique name. Agents stay on top of a platform that provides their basic service. A platform consists of one or more containers. Each container can have zero or many agents. Container can be executed on different hosts so as to make distributed platform. Main container is the first container started and all other container need to connect and register with main container [3].

JADE platform implements all the Massage Transport Protocol (MTP) defined by FIPA. As for internal communication protocol JADE is using JADE IMTP (Internal Message Transport Protocol), this protocol used for communication between agents on the same platform. By default JADE uses java RMI (Remote Method Invocation) for IMTP implementation.

An agent can have a task or process that specifically need to done by the agent, in the JADE framework the tasks assigned to an agent called behaviour. Behaviour presents agent task. A mobile agent can have more than one behaviour and agent can run more than one behaviour at the same time [8].

GSM Association defines LBS (Location-based Services) as services that use the location of the target for adding value to the service, where the target is the "entity" to be located. Another abstract definition given by *3rd Generation Partnership Project* (3GPP): an LBS is a service provided by a service provider that utilizes the available location information of the terminal [9]. So it can be concluded that the location-based services (LBS) utilize the location information of an entity, process it and provide the relevant services from the user side.

Analysts and researchers have taken several approaches to classifying LBS applications. A major distinction of services is whether they are person-oriented or device-oriented [10]. First, person-oriented LBS comprises all of those applications where a service is user-based. Thus, the focus of application use is to position a person or to use the position of a person to enhance a service. Usually, the person location can control the service (e.g., friend finder application). Second, device-oriented LBS applications are external to the user. Thus, they may also focus on the position of a person, but they do not need to. Instead of only a person, an object (e.g., a car) or a group of people (e.g., a fleet) could also be located. In device-oriented applications, the person or object located is usually not controlling the service (e.g., car tracking for theft recovery).

In addition to the above classification, two types of application design are being distinguished: push and pull services [10]. Push services imply that the user receives information as a result of his or her whereabouts without having to actively request it. The information may be sent to the user with prior consent (e.g., a subscription-based terror attack alert system) or without prior consent (e.g., an advertising welcome sent to the user open entering a new town). Pull services, in contrast, mean that a user actively uses an application and, in this context, "pulls" information from the network. This information may be location-enhanced (e.g., where to find the nearest cinema).

Technologically, the realization of LBS can be described by a three-tier communication model that can be seen in figure 3 including a positioning layer, a middleware layer and application layer. Application layer comprises all of those services that request location data to integrate it into their offering (e.g., a friend finder). Middleware layer can significantly reduce the complexity of service integration because it is connected to the network and an operator's service environment once and then mitigates and control all location services added in the future. The positioning layer is responsible for calculating the position of a mobile device or user.

Physical location can be broken down to the following subcategories that are relevant for creating and using LBS [9] that are, descriptive locations, spatial locations, network locations.

A descriptive location is always related to natural geographic objects like territories, mountains, and lakes, or to man-made geographical objects like borders, cities, countries, roads, buildings, and rooms within a building.

Strictly speaking, a spatial location represents a single point in the Euclidian space. It is usually expressed by means of two- or three-dimensional coordinates, which are given as a vector of numbers, each of it fixing the position in one dimension.

Network locations refer to the topology of a communications network, for example, the Internet or cellular systems like GSM or UMTS. These networks are composed of many local networks, sometimes also referred to as subnetworks, connected among each other by a hierarchical topology of trunk circuits and backbones.

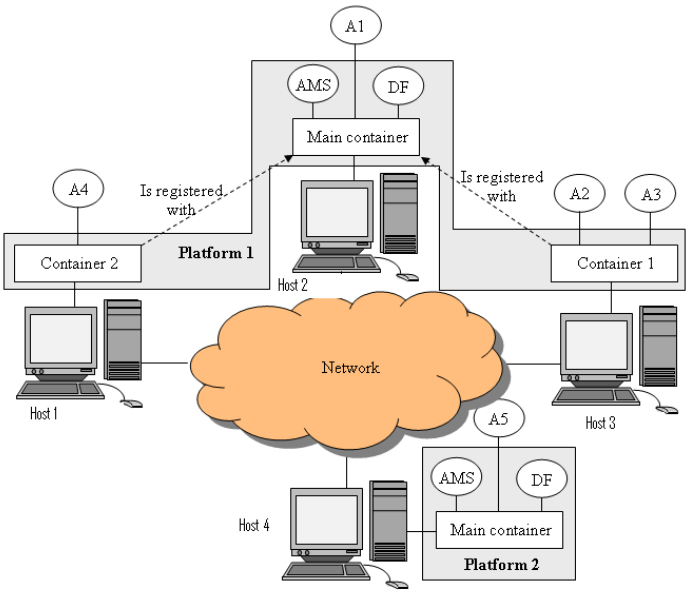

Figure 2. JADE architecture [11].

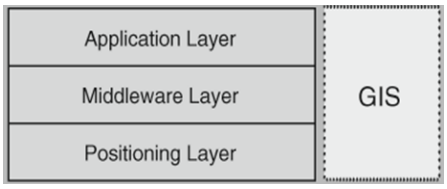

Figure 3. General LBS communication model [10].

2. Methodology

In accordance with the scenario of the system, Local Area Network (LAN) will be used. Network port that will be used is port 1099 using Transmission Control Protocol (TCP). Port 1099 is the default port for Java Remote Method Invocation (RMI) which is used by JADE platform for inter-platform communication.

The system consists of four main actors: Mobile Agent Server (MAS), Location-Based Service Server (LBSS), printer service and client. Figure 4 shows the design of network topology.

Agents design is grouped by container where the agents live. Agents in MAS are agent from JADE framework where the first container (main container) is started. The following is agents that live in MAS and their functions are, first agent name is AMS (Agent Management System) and its functionis the agent who supervised the entire platform. Every agent in the entire platform must register with AMS when started. Second, agent name is DF (Directory Facilitator) and its function is the agent that implements yellow pages service, which used by other agent to register its services or search for services. Third, agent name is RMA (Remote Monitoring Agent) and its function is allows platform administrators to manipulate and monitor platform with graphical user interface.

System has LBSS that helped to manage services list, service resource and provide service for connected client. The following is agents that live in LBSS and their functions are, first, agent name is Routing Agent and its function is to forward document sent by client to the intended service. In addition this agent also responsible to divide work if more than one container provides same type of service, the division of work is done using round-robin algorithm.

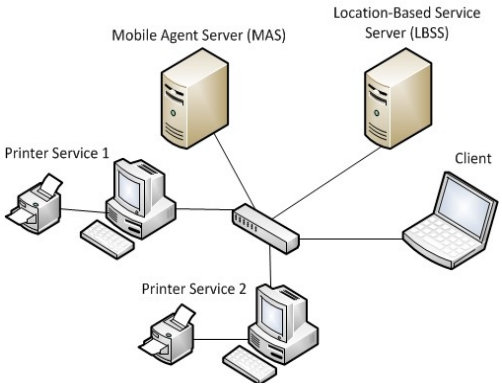

Figure 4. Design of network topology.

Second, agent name is Watcher Agent and its function is to find service available in the platform and maintain up to date list of service to be used by Routing Agent. This agent also may provide confirmation to the client whether there is a service platform. Third, agent name is Provider Agent and its function is to provide services to client. When

there is a new client connected to the platform, this agent will make mobile agent that will visit and serve the client.

As the system provides printer service, an agent especially designed to handle the service is needed. The agent will live in container in a machine that connected to printer service. The following is agent that lives in every container that provide printer service that agent name is Printer Agent and its function is to print document sent to it and send confirmation to client if the document received.

Design of the system is done using UML (Unified Modelling Language) modelling. One of the UML modelling used in the process designing the system is sequence diagram. Sequence diagram showing the flow of the system can be seen in figure 5. Sequence Diagram shows the object interaction in the system, namely: MAS, LBSS, printer service and client. Each object has one or more agents that can communicate with each other, using IMTP protocol that JADE implemented.

3. Results and Analysis

Programming language used in the development of location-based printer service system using mobile agent technology is Java. System architecture can be seen in figure 6. In the system there are two servers, Mobile Agent Server (MAS) and Location-Based Service Server (LBSS). MAS is machine that runs the main container, which is the first container started and used as a bootstrap point for JADE platform. All containers that run after the main container will be registered to main container at MAS when first started.

A container will be run on LBSS in charge of managing the services that available in the platform, from registration of new service, manage service list, delivering service to client and ensure the service requested by client are met. In the scenario there is print service available on the network. If there are two containers providing print service on the platform, LBSS will divide the work using round-robin algorithm.

Client requesting the service does not need an agent when started, there will be special agent with mobile agent created by LBSS that will visit and serve client. Implementation is successfully done on a virtual network using VirtualBox with bridged networking. A total of two virtual machines act respectively as MAS and LBSS, two virtual machines as print service and a virtual machine as a client. All machines are connected with VirtualBox bridged networking, so it seemed to physically connect using network cable.

System need to be tested to check whether they have met all system requirements. There are two types of test that are connection and functional. Connection test intended to ensure that all containers are connected and communicate with each other. If a container fails to connect to the platform then the test fails and troubleshooting process need to be taken. Functional test is intended to test mobile agent technology in its work deploying location-based services. The test will check every agent on the system whether they have worked well in accordance with its duties and function. Table I shows connection test result on system, all containers are able to connect to main container on MAS and communicate with each other.

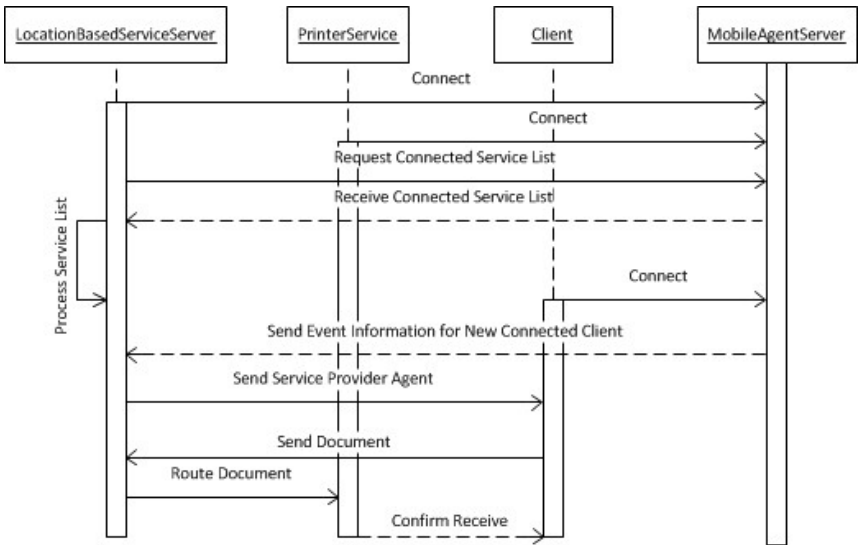

Figure 5. Sequence diagram of the system.

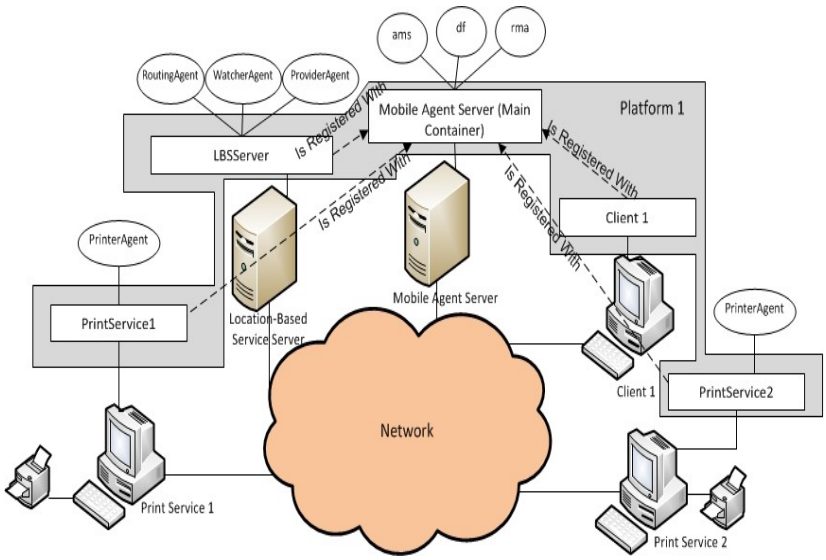

Figure 6. System architecture.

Functional testing will be grouped by where the agent belongs to MAS functional testing, LBSS functional testing, print service functional testing, client functional testing.MAS contains DF, AMS and RMA agent which is special agent from JADE, those agent run automatically when main container started. MAS is responsible to manage agent and run white page service and yellow page service on platform. MAS functional testing result shown on table II.

Provider agent on LBSS is responsible to create mobile agent to serve client. Function of Provider Agent which has been successfully tested is shown at table III. Routing agent function on LBSS is to forward document sent by client to printer service and divide work if there are more than one container registered as printer service. Work is divided using round-robin algorithm. Result of testing can be seen in table IV.

Watcher agent on LBSS is responsible to find service that registered on platform and maintain latest service list to be used by Routing Agent. This agent also able to confirm whether service is available on the platform when agent needs it. Agent watcher test result can be seen in table V.

In this research, service that provided by LBS is print service. To provide necessary services, an agent is needed to receive document for printing and sent confirmation to client that document is successfully received. The result of functional testing of Printer Agent can be seen in table VI.

Client in this system is special object, because all agents in the system primary purpose is to serve client. Client does not require an agent at run time, but there will be an agent from LBSS that will visit and serve client. The result of client functional testing can be seen in table VII.

TABLE I
CONNECTION TESTRESULT

Test scenario	Results
Run LBSS.	Connected
Run *print service*.	Connected
Run *client.*	Connected

TABLE II
MAS FUNCTIONAL TESTING RESULT

Function	Results
Each agent can register with AMS.	Succeed
AMS sends event information to ProviderAgent when there is a new client connected.	Succeed
DF implement yellow page service.	Succeed
DF provides service list for WatcherAgent.	Succeed

TABLE III
PROVIDER AGENT FUNCTIONAL TESTING RESULT

Function	Status
Receive event information from AMS when a new client is connected.	Succeed
Create and send a mobile agent to serve new client.	Succeed

TABLE IV
ROUTING AGENT FUNCTIONAL TESTING RESULT

Function	Status
Receive service list from Watcher Agent.	Succeed
Receive document from client.	Succeed
Forward document to print service.	Succeed
Implement round-robin algorithm as a method of work division.	Succeed

TABLE V
WATCHER AGENT FUNCTIONAL TESTING RESULT

Function	Status
Search for service provider listed in yellow pages MAS.	Succeed
Managing latest service list.	Succeed
Provide latest service list to Routing Agent.	Succeed
Confirm availability of services.	Succeed

TABLE VI
PRINT SERVICE FUNCTIONAL TESTING RESULT

Function	Status
Register as print service.	Succeed
Receive document from LBSS.	Succeed
Send confirmation to client.	Succeed

TABLE VII
CLIENT FUNCTIONAL TESTING RESULT

Function	Status
Requesting confirmation availability of services.	Succeed
Sending the document.	Succeed
Receive document delivery status.	Succeed

4. Conclusion

Researcher have implemented mobile agent technology on location-based printer services using JADE as framework. There are advantages and disadvantages of the application of mobile agent in this research. Advantages of mobile agent technology on location-based services are provide another solution to deploy services in an environment that consists of devices with limited resources, service management is easier and cheaper because it does not need additional infrastructure on addition of new service, easier maintenance, the system consists of an agent that is easy to debug and agent components can easily be replaced without affecting other components in the system and high scalability, system model scalable for many kind of network type. Disadvantages of mobile agent technology on location-based services are especially in the case of developing this system, two computer additions are required. These two computers will act as Mobile Agent Server and Location-Based Service Server respectively and mobile agent security is still untested.

References

[1] M. Weiser, "The computer for the 21st century," *Scientific American,* vol. 3, pp. 3-11, 1991.

[2] I. Satoh, "Location-based services in ubiquitous computing environtments,"*International Journal on Digiral Libraries,* vol. 6, pp. 280-291, 2006.

[3] K.M.S. Lubis, "Implementasi Teknologi Mobile Agent pada Sistem Pemantauan Jaringan Komputer,"B.S Thesis, Department of Computer Science, Faculty of Mathematics and Natural Sciences, Bogor Agricultural University, Indonesia, 2010.

[4] M. Delmeijer, E. Rietjens, M. Soede, D. Hammer, & A. Aerts, "A Reliable Mobile Agents Architecture" *InProceedings First International Symposium on Object-Oriented Real-Time Distributed Computing (ISORC 98)*, pp. 64-72, 1998.

[5] A. Pircanescu, C. Badica, & M. Paprzyki, "Developing a JADE-Based Multi-Agent E-Commerce Environment" *In IADIS AC'05: International Conference on Applied Computing*, p. 8, 2005.

[6] F. Bellifemine, G. Caire, & D. Greenwood, *Developing Multi-Agent Systems with JADE*, Wiley, Chichester, 2007.

[7] P. Braun & W. Rossak, *Mobile Agent: Basic Concepts, Mobility Models, and the Tracy Toolkit*, Elsevier Inc., San Fransisco, 2005.

[8] F. Bellifemine, G. Caire, T. Trucco, & G. Rimassa, Jade Programmer's Guide, JADE, http://jade.tilab.com/doc/programmersguide.pdf,2007, retrieved January 5, 2012.

[9] A. Küpper, *Location-Based Services : Fundamentals and Operation*, John Wiley & Sons Lt, Chichester, 2005.

[10] J. Schiller & A. Voisard, *Location-Based Services*, Elsevier Inc., San Fransisco, 2004.

[11] D. Grimshaw, JADE Administration Tutorial, JADE, http://jade.tilab.com/doc/tutorials/JADEAdmin/jadeArchitecture.html, 2010, retrieved January 5, 2012.