Aglets: Programming Mobile Agents in Java

Danny B. Lange¹, Mitsuru Oshima¹, Günter Karjoth², and Kazuya Kosaka¹

¹ Tokyo Research Laboratory, IBM Japan Ltd.
² Zurich Research Laboratory, IBM Swiss Ltd.

Abstract. Mobile agents are programs that can be dispatched from one computer and delivered to a remote computer for execution. Arriving at the remote computer, they present their credentials and obtain access to local services and data. They also provide a single uniform paradigm for distributed object computing, encompassing synchrony and asynchrony, message-passing and object-passing, and stationary objects and mobile objects. In this paper, we describe our Java-based mobile agents called Aglets and present its programming interface, called Java Aglet API.

1 Introduction

Java³ and applets [1] have revolutionized the Web, and executable content has become a common term in the Web glossary. Applets are essentially sets of program code that can be downloaded, instantiated, and executed in Web browsers. Recently, this concept has been matched by the introduction of the servlet [2]. A servlet moves program code in the opposite direction to an applet; that is, it allows the client program to upload additional program code to a server. The servlet’s code is then instantiated and executed in the server.

The aglet represents the next leap forward in the evolution of executable content on the Internet, allowing program code to be transported along with state information. Aglets [4] are Java objects that can move from one host on the Internet to another. That is, an aglet that executes on one host can suddenly halt execution, be dispatched to a remote host, and resume execution there. When the aglet moves, it takes along its program code as well as its data. Conceptually, the aglet is a mobile agent because it supports the ideas of autonomous execution and dynamic routing on its itinerary.

The Java Aglet API (J-AAPI) [3] is a proposed industry standard for Java-based mobile agents. J-AAPI was developed by a research team at the IBM Tokyo Research Laboratory in Japan in response to a call for a uniform platform for mobile agents in heterogeneous environments such as the Internet.

The notable thing about J-AAPI is that it extends the renowned write once, run everywhere capability of Java to agents. That is, once you have written an aglet, it will run on every machine that supports J-AAPI. You do not have to concern yourself with the underlying hardware or operating system, or with the nature of the particular implementation of J-AAPI on the host where your aglet is running.

³ Java is a trademark of Sun Microsystems, Inc.
In this paper, we describe the technical advantages of mobile agents, give an overview of Aglets, and explain how to program mobile Internet agents by using J-AAPI. Next, we briefly describe the Aglets Security Model, sketch some usage scenarios, and relate Aglets to other work. Finally, we draw some conclusions.

2 So Why Mobile Agents?

Today, the central paradigm linking all distributed object technologies is a synchronous message-passing paradigm whereby all objects are distributed, but stationary, and interact with each other through message-passing. This paradigm is incomplete, and needs to be enhanced in some fashion with additional paradigms such as asynchronous message-passing, object mobility, and active objects.

Mobile agents can provide a single uniform paradigm for distributed object computing, encompassing synchrony and asynchrony, message-passing and object-passing, and stationary objects and mobile objects. Mobile agents, like elementary particles in physics, are fundamental to distributed object computing. Along with mobility, agents have the following unique and important computational characteristics:

Object-passing: When a mobile agent moves, the whole object is passed; that is, its code, data, execution state, and travel itinerary are all passed together.

Autonomous: The mobile agent contains sufficient information for it to decide what to do, where to go, and when to go.

Asynchronous: The mobile agent has its own thread of execution and can execute asynchronously.

Local interaction: The mobile agent interacts with other mobile agents or stationary objects locally. If needed, it can dispatch messenger agents or surrogate agents, which are all mobile agents, to facilitate remote interaction.

Disconnected operation: The mobile agent can perform its tasks whether the network connection is open or closed. If the network connection is closed and it needs to move, it can wait until the connection is reopened.

Parallel execution: Multiple mobile agents can be dispatched to different sites to perform tasks in parallel.

The technical advantages of mobile agents are many, and there is no single alternative to all of the functionality they provide [11]. In addition to supporting existing network services, mobile agents are also likely to make possible new services and thus new businesses.

3 J-AAPI and the Aglet Object Model

This section provides a quick review of elements of the Java Aglet Application Programming Interface (J-AAPI).

J-AAPI is a proposed standard for interfacing aglets and their environment. This is the API you will be using to create aglets. It contains methods for