

Abstract

As a result of the diffusion of information system with the appearance of the low cost but high performance computers, two or more computers are available for each of us nowadays. Increasingly, computers are installed in public spaces for temporary use in addition to the computer that belongs to a person or an organization. However, users have no right to install their required applications in such computers because of the security reasons. Therefore, limited applications are available in such a computer, for example web browser, even if the number of public computers is increasing. If users require specific applications, they may need to carry their own personal computer with them at all times. In a word, the computers that are installed in public spaces are not effectively used.

In addition, the increase of the number is not only the applications but also the functions within an application. Besides, users usually use only a part of functions in spite of a large number of functions. It can cause a waste of computer resources and difficulties of application management for individual users to install there required application in public computers even if it is possible. However, it is impossible to install nothing but parts of an application separately because of its monolithic structure even though the application may use dynamic link libraries.

On the other hand, the spread of the network infrastructure such as Access Points of wireless network allows both public computer users and personal computer users to be able to connect to the network. It is, however, easy to imagine that there will be a large number of illegal accesses from the network if the network opens to everyone without any authentications. Therefore, the system requires to define users if they are authorized or not, and prevent malicious computers to access with spoof IP/MAC addresses.

Based on the background which is described above, this thesis presents the way of achieving effectual use of computer resources by running applications that each user requires with a need functions without relying on computer environment (hardware or OS). In addition, we would like to provide a framework that executes the application while downloading necessary functions from network. We would also like to apply this execution method to the application management system including install, execute and update the application. Furthermore, the reduction of starting time is explained and required resources of well designed applications by using the framework is compared with the traditional method that downloads entire program of the application before the execution. In order to download required functions (programs), we extend the dynamic loading mechanism of Java language. In addition, we provide a system that can distinguish authorized users and prevent illegal accesses from malicious computers by introducing user authentication to provide network connectivity the framework requires. Moreover, we confirm the efficiency of the system from benchmark tests and the working result.

These framework/system enables an authorized user to execute his/her required

applications while downloading necessary functions on demand from network with any available computers.