Supporting Adaptive Application Mobility

Francis M. David, Bill Donkervoet, Jeffrey C. Carlyle, Ellick M. Chan, and Roy H. Campbell

University of Illinois at Urbana-Champaign, Urbana IL 61820, USA
{fdavid,donkervo,jcarlyle,emchan,rhc}@uiuc.edu
http://choices.cs.uiuc.edu/

Abstract. Application mobility has the potential to enhance user experience in ubiquitous computing environments by providing a flexible and reusable solution to managing applications across myriad computing devices, especially when applications adapt to the characteristics of individual devices. Using example scenarios, we argue that application mobility is a better solution to the problem of accessing remote applications than schemes like remote desktop which only export displays. Our mobile application framework provides the opportunity for applications to better adapt their user interface to the new environment. This ability is enhanced through the use of the Model-View-Controller design pattern. Our framework also uses discovery mechanisms to find potential migration targets. Lost applications are recovered through a simple scheme called homing. A preliminary implementation for our framework is based on the JADE mobile agent platform.

1 Introduction

We live in a world where we are constantly surrounded by computers. Libraries, airports and coffee-shops have wireless networking or public computer terminals which provide Internet connectivity. Networked mobile computing devices like laptops, PDAs and smart phones are also widely available. With the constant availability of computing devices, it has become desirable to access our data and work from anywhere. We can achieve this to some degree using tools such as remote desktop, VNC [1] and XMOVE [2] which export user interfaces through a network; however, these solutions are not perfect and are deficient in several aspects.

Let us consider the remote desktop solution to accessing applications available on your office PC through your laptop when you are away from the office. This would require that both your office PC and your laptop are connected to the Internet. This requirement holds even if the applications you are accessing don’t have any use for a network. Interactivity is also poor because of network latency issues. If your applications were mobile, you could move the applications to your laptop at any time and work with them without sacrificing response times even when you are disconnected from the network.

Let us consider another scenario. You are editing a file on your office PC and would like to continue editing the file on a PDA during your commute back home.
Normally, you would close the file on your PC, save the file to your PDA and start editing it using a PDA application. Wouldn’t it be nice if the application jumped from your PC to the PDA? And automatically adjusted itself to the smaller screen size? And reduced the use of energy intensive graphical effects to save battery power? After you finish editing the file, you start watching an on-demand movie on the PDA. When you walk into your home, it would be extremely convenient if the movie player migrates to your home computer and starts to use your big plasma screen on the wall after being transformed into a high-bandwidth application. As you are watching the movie, your boss calls your smart phone. You cannot hear the phone ringing over the sound of the movie and miss the call. Ideally, a remote notification application should have been placed on the home computer by your smart phone when you walked in. The incoming phone call would have then triggered this application to take appropriate action like pausing the movie. From these examples, it is easy to see that when applications are mobile, they can present themselves to users on a more convenient device in a form more suited to that device. This enhances user experience and increases productivity.

Virtualization technology presents a different solution to the problem of pervasive application and data access. Products such as VMware’s VMotion and several research projects propose the use of virtual machine migration to manage the mobility of data and applications. One drawback of this approach is that this scheme does not work across heterogeneous platforms. Yet another drawback is that migrating a complete virtual machine consumes more time and bandwidth than just selective application migration.

We believe that a framework which supports application mobility provides a flexible and reusable mechanism to manage applications in a world where we are surrounded by computing. We discuss our design for supporting mobility and the rationale behind the choices we make in Section 2 and describe a preliminary implementation in Section 3. We highlight other related projects in Section 4 and conclude in Section 5.

## 2 Supporting Adaptive Application Mobility

Our design is based on the functionality already provided by frameworks like JADE that support mobile agents. We can leverage existing research and work in this area in order to support application mobility. In this section, we describe modifications and extensions to the basic migration and naming functionality provided by agent frameworks.

### 2.1 Target Discovery

Discovery is used to continuously check for potential migration destinations. In the case of mobile devices, the framework can discover new targets as the user moves around. In the case of fixed devices, the mobility framework can be initially configured with a fixed set of target hosts and discovery can be used