A Mobile and ubiquitous system for the new millennium. Doing new things with ancient technology from the 70s!

Demonstration Proposal

Gorka Guardiola Múzquiz and Katia Leal Algara
{paurea,kleal}@lsub.org
Laboratorio de Sistemas — Universidad Rey Juan Carlos
Madrid, Spain.

Introduction

We have written a complete system, Plan B [3], that relies on volatile late binding and a distributed file system to provide all system services. Our aim is to demonstrate that our approach makes it is easy to build applications which work on highly dynamic and heterogeneous environments, and can adapt to environment changes. For the purpose of the demonstration, we focus on our user interface and command services. The application used in the demonstration, which controls X10 sensors and actuators, shows how our approach works well both for mobile and for ubiquitous environments.

Our user interface service exports widgets as if they were files. By doing so, we have a portable way to program and control user interfaces. When combined with late binding, this technique permits applications to adapt to environment changes and to effectively use the set of distributed resources. For example, as it will be shown in the demonstration, our applications can easily relocate, split and merge their user interfaces using different screens serviced by heterogeneous platforms. This can be done also in a transparent way when it is necessary.

The file system interface, together with late binding, decouples our applications from the services they use. For example, we can zip (or tar) a user interface, send it by mail to another place, extract the interface from the archive and keep on using it. Such degree of decoupling makes it easier to work on a mobile environment. Furthermore, since the interface for the service is a file system, interoperability is quite easy and does not require middleware layers (most platforms know how to share or remotely use files). We can run our services on different operating systems and use them to control devices over a plethora of computing platforms (iPAQs, PCs, phones, etc.), and we can move and tolerate disconnections.

What has been said works for most services in Plan B, although in this demonstration we consider just user interfaces together with remote execution and control of user programs.

Description

The demonstration consists on the presentation of one prototype of our system using part of the equipment we will bring to the workshop, including laptops, a Pocket PC, and a Mobile phone. We can setup point to point links, but Wireless LAN availability would lead to a better setup.

We will first remotely start our example application using the Plan B command service [1]. The application permits the control and monitoring several X10 devices (using also a file system interface). Once it has started, we will archive the user interface in a file, send it from the PC to the iPAQ, and extract it there. The interface will be used while the application is disconnected, to demonstrate disconnected operation. Later, we will reconnect the application to see how the operations on the interface are performed.

To further demonstrate our system, the command service will be used to redirect the interface from one machine to another while the application runs. Emphasis will be made in that this can be done even with simple standard shell commands, without needing middleware services or complex tools. The control for both the UI and the X10 devices can be done using portable standard file commands like cp and rm or standard file system calls like read and write.

Contributions

The main contribution is that we show that by using old tools like file systems and late binding we can build distributed applications for highly dynamic and heterogeneous environments.

One lesson learned will be that exporting services like the user interface or the command execution service as a file system, enables us to control and use our services using legacy software that is present on every computing platform. We will show how this also solves important problems like protection, access control, naming, and heterogeneity in a simple and well understood way.

Another lesson learned will be that by using late binding, we can also adapt the application to the environment, and change the environment if we need to.

As in the case of the user interface, we export X10 devices as a file system. What we show on the demo is an emulation for this file system. We have a production system used daily that uses real X10 devices. We might include it in the demonstration provided that we could use some power sources and lamps. Nevertheless, as the demo will hopefully show, another benefit of using files is that we can easily emulate the service for debugging or demonstration purposes.

Project description and author information

Plan B [2] is an operating system built from the beginning for use in highly distributed and dynamic environments, like those available in mobile computing and in ubiquitous systems. These are the principles Plan B is built on: (1) All resources (processes, devices, etc.) are perceived as a single abstraction. (2) The system operates on both local and remote files through the same protocol. (3) Volatile late binding is combined with per–application name spaces as the means for adaptation. More information and papers about the project can be found at [3].

Although our group, the *Laboratorio de Sistemas* at Rey Juan Carlos University, is very young, its members have years of experience in the field of adaptable and distributed operating systems, including their design and implementation. For example, the 2K operating system built at the University of Illinois at Urbana–Champaign included work from our group members; the Off++ microkernel and part of the distributed file system for ubiquitous environments (a predecessor of the Context File System they are using in Gaia). We have also been (and still are) active members of the Plan 9 from Bell Labs community. Right now we are working on the Plan B system, which is the subject of this demo proposal.

References

1. K. L. Algara, F. J. Ballesteros, G. G. Múzquiz and E. S. Salvador, Plan B's Personal Command Module. Commanding user activities in mobile and ubiquitous environments, *Submitted for publication to WMCSA*, 2004.
2. F. J. Ballesteros, G. G. Muzquiz, K. L. Algara, E. Soriano, P. H. Quirós, E. M. Castro, A. Leonardo and S. Arévalo, Plan B: Boxes for network resources, *Submitted for publication. Also in http://lsub.org/ls/export/box.html*, 2004.
3. Plan B web site., *http://lsub.org/planb*, 2001.