

Title	状況に適應するアプリケーションの構築法に関する研究
Author(s)	赤木, 敏和
Citation	
Issue Date	1997-03
Type	Thesis or Dissertation
Text version	author
URL	http://hdl.handle.net/10119/1059
Rights	
Description	Supervisor:中島 達夫, 情報科学研究科, 修士

A Framework for Building Adaptive Applications in Mobile Computing Environments

Toshikazu Akagi

School of Information Science,
Japan Advanced Institute of Science and Technology

February 14, 1997

Keywords: Mobile computing, Real-Time Mach .

With the advance of technology, the portable computer have come smaller and powerful. We can carry the computer to everywhere and can use every time with the network connection. We call these environments as “mobile computing environments”, and it enables very useful variety of computing. These ordinary mobile computers have PC card slots and batteries, so we can use many types of devices by inserting PC card devices and can use computers continuously without AC adapters. By using PC card devices, device configurations of the computer is dynamically change with inserting device or removing. And such mobile computers also have the APM (Advanced Power Management) BIOS. APM enables to controlling power of CPU, disks and any of devices of the mobile computer. It also have managing function of batteries. By using APM, we can know about remaining batteries, status of charging and availability of AC adapters.

By using these devices, the state of environments of computer is dynamically change. It means that device configurations, remaining batteries and availability of AC adapter be affect the execution of applications. It is very useful that applications dynamically change behavior to adapt to the environment.

In this paper, we propose a framework for building adaptive applications in mobile computing environments.

First, we present the modification of Real-Time Mach Microkernel for supporting PC card devices and APM device. It enables that operating systems can change dynamically device configurations by inserting a PC card or removing a PC card, and we can get status of battery or availability of the AC adapter. So we also propose *Event Manager* that deliver events generated in the kernel to user-level servers. And propose *Environment Server* architecture that deliver events needed by the application to applications by filtering. In this architecture, the filter of event can describe with the Tcl scripting language. By inserting filters to the *Environment Server*, applications can receive events to change behavior. Finally, we describe some adaptive applications that use *Environment Server*.

Environment Server provides a good abstraction for event types, resource names and control of system components. We can use path names to specify events, resource control when writing applications. With the using path names, we can access uniform to all of the system resources. For example:

```
/localhost/pccard/event
/localhost/pccard/event/inserted
/localhost/apm/status/batt
/localhost/apm/control/suspend
```

Path names are construct from the host name, the component name, event types, status types and controls. And an every path object has a script written by Tcl scripting language, and with casting to path objects applications can refer the resource. To get events, application install a scripts specified a path name. Applications can also make a conditions with specify a event and some resources in the scripts. For Example, we can describe in scripts as follows:

```
path  : /localhost/pccard/event/inserted
script: "if { [cast /localhost/apm/status/ac-online] == 1 } {\n
        set ret \"card inserted and ac-online\"\n
      }\n"
```

When we specify these scripts, application gets notification when a card was inserted and a battery status is AC on-line.

Environment Server architecture will provide these advances.

- It enables deliver events efficiency from kernel to user-level server.

- It can describe event filters by Tcl Scripting Language.
- By using *Environment Server*, applications can refer the system resource.
- And also, *Environment Server* provide a good abstractions to applications with accessing event types, resource names.
- *Environment Server* user libraries provide good way to access the server.

These character are able to advantage of building adaptive applications that using events, referencing resources and controls. And it also provides to applications a good abstraction of events, resources and control.

The outline of this thesis is as follows:

- In chapter 1, we introduce the outline of the this research.
- We discuss the problems that why need adaptive applications and how to develop adaptive applications in such as mobile computing environments in chapter 2.
- And we discuss work related to writing adaptive software using system events, providing abstraction of resources to applications, and developing adaptive applications, in chapter 3.
- We propose a framework for building adaptive applications in mobile computing environments in chapter 4. It Also describe about modifications to Real-Time Mach Microkernel to support PC card devices and APM device, *Event Manager* that enables deliver events to user-level programs and describe the overview of *Environment Server* architecture.
- And then, we describe about *Environment Server* that provide the event delivering mechanisms and abstractions of resources to applications, in chapter 6. We also describe about the script for writing event filters, interfaces to applications and user libraries to support writing applications.
- In chapter 6, we show some examples using *Environment Server* to build applications. And we discuss about effects of our framework.
- Finally, we will conclude this thesis in chapter 7.

In this research, we develop a prototype system and applications as follows:

- Modifications to Real-Time Mach Microkernel supporting PC card devices and APM device.
- *Event Manager*, it extensions kernel function.
- *Environment Server* software.
- And some of example applications(cardmonitor, esh, webcontrol.)