

An Implementation of Mobile Desktop Manager for Mobile Web Devices

Kyung Hee Lee*, Hadipurnawan Satria*, Chaedeok Lim*, Kwan-jong Yoo**

*Embedded SW Platform Research Team, ETRI, Korea

**Chungnam National University, Korea

kyunghee@etri.re.kr

Abstract—A number of mobile devices such as smart phones, smart TV's, portable media players, etc. are introduced these days. They are implemented on their own operating systems so that they cannot provide compatibility of application services. To solve the problem among application services we suggest an architecture that uses a web browser as a basic application platform. The proposed architecture uses Java Script language. With this architecture application developers can build mobile services with ease as well as they can provide compatibility of the application services on the devices.

Keywords—Java Script, Mobile, Desktop, User Interface

I. INTRODUCTION

Most personal computers have desktop managers. Desktop manager is a kind of application program run on an operating system and it can install, delete and execute application programs. Mobile devices also provide mobile desktop manager to get user input and to run mobile applications. These mobile devices have different mobile desktop managers according to their operating systems, manufacturers, and device models. If applications can be executed on the devices, we have to port the applications for their operating systems and system libraries. This interoperability problem makes us difficult to reuse the application source codes. To cope with the problem, Java was introduced for interoperability among operating systems and devices. Java uses JVM (Java Virtual Machine) that is a kind of virtual machine to run Java applications. Java can be one of the solutions to the problem, but the devices have to have its JVM and there can be a royalty problem. Since most mobile devices have web browsers for internet users, we choose Java Script as a solution. Java Script can be run on most web browsers. For these reasons, we implemented mobile desktop manager based on Java Script.

Java Script supports only low level graphic functions for application developers so that it is difficult to build graphic applications. For this difficulty we add user interface library

that is written in Java Script. This Java Script user interface library is a kind of Java Script application and provides graphic widgets such as button, list, and window for user interface application programmers. They can make applications with the widgets in Java Script user interface libraries. We built mobile desktop manager that uses the user interface libraries and we made the mobile desktop manager process not only mouse click events but also touch events such as touch and drag, touch-down, and long press events. And we also added the functionalities such as install, deletion, run and termination of applications that application manager has to have.

In this paper, we suggest software architecture for the compatibility of mobile desktop application services. We describe an ordinary graphic user interface systems in Chapter II, propose the new architecture of mobile desktop system manager in Chapter III, and conclude in Chapter IV.

II. OVERVIEW OF GRAPHIC USER INTERFACE SYSTEMS

This chapter describes the overview of ordinary user interface systems for mobile devices.

A. Architecture of Window System

The target system of this paper is a mobile web device that is a kind of embedded systems and has restriction such as processing power restriction of central processing unit or physical memory size, etc. Typical mobile handsets can be PMP (Portable Multimedia Player) and smart phones.

We chose Linux for the referential operating system. Linux is a kind of open source software and it does not have standard graphic window system and graphic library system.

Window system is a software component that processes basic Input/output mechanisms such as graphic output, audio output, text output, mouse event input, and keyboard event input.

There can be two types of graphic window systems. The one is server-client type and the other is library type. The server- client type is run in the form of a process and it provides inter-process communication interface to interact with application programs. The library type is run by direct linking to them so that direct call mechanism is used to interact with them.

B. Ordinary Architecture of User Interface Systems

There can be few graphical requirements for simple mobile phones but there can be high quality 3D graphic requirements for specialized mobile handsets such as 3D game devices and web devices. In case of smart phones, there can be requirements of stylish user interface and high quality graphic due to digital convergence, improvement of computing power for mobile devices, etc.

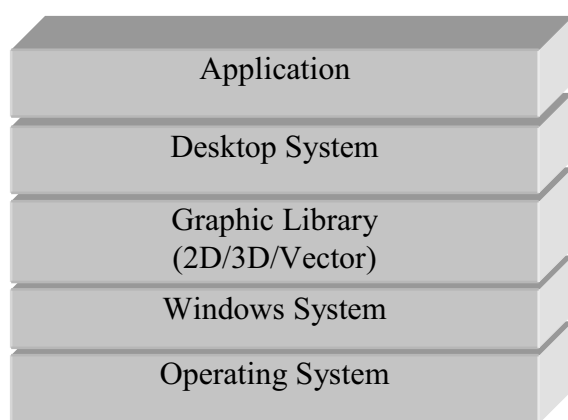


Figure 1. Ordinary Software Architecture of Graphic Systems

2D graphic library is a software component for 2 dimensional graphic processing such as drawing lines, circles, rectangles, etc. 3D graphic library is a software components for representing, recording, rendering 3 dimensional data. Vector graphic library is a kind of 2D graphic library. 2D graphic library is based on representing all pixel information of lines, circles, etc but vector graphic library is based on mathematical information of objects such as coordinates of start point and end point and thickness of the line.

In ordinary systems, 2D, 3D, and vector graphic libraries are installed and executed separately.

Embedded systems such as mobile web devices can have dependency or constraints on hardware resources unlike personal computers. As an example, a certain mobile device may neither support graphic acceleration function nor floating point operation for graphic processing.

Desktop System is a software component that manages application services and provides basic functions for them. It is a software component that has functionalities to show wall paper image, icons for applications and provides mechanism

and interface to install, run and control the application services.

C. Problems of Ordinary User Interface Systems

Most of applications have dependency on their operating systems and graphic libraries. If a user changes the operating system of a device, then he/she may get a new version of the application for the target operating system. As in the case of the compatibility of the operating systems, graphic library can affect the compatibility of the application services. To cope with this problem, the application service providers have to provide different versions of application services for each operating systems and graphic libraries.

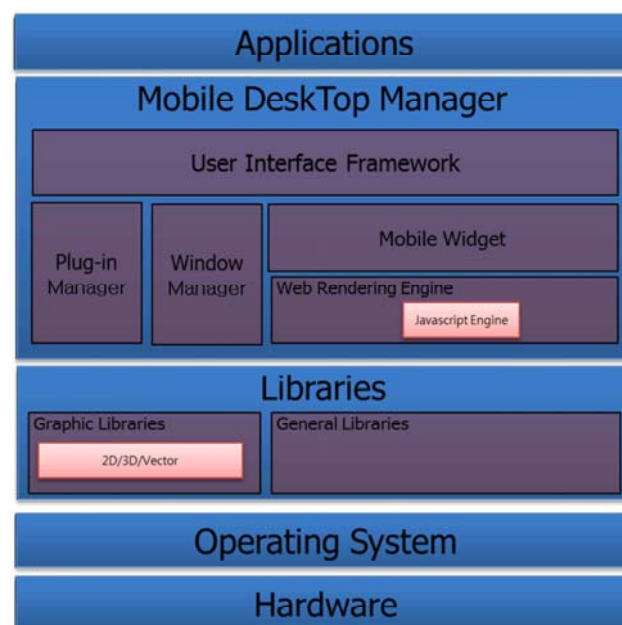


Figure 2. Software Architecture of the Mobile Desktop System

III. AN IMPLEMENTATION OF THE MOBILE DESKTOP SYSTEM

In this chapter, we suggest a mobile desktop system for application services compatibility for different operating systems and graphic libraries.

A. Java Script Language

Java language can be a solution to cope with the compatibility problem. But we have to install Java Virtual Machine for Java language and there can be loyalty problem.

Most web browsers support Java Script language and most mobile devices such as smart phones, portable media players, etc. have web browsers. Our target device is a mobile web device so that we choose Java Script language.

B. Architecture of Mobile Desktop System

We design our mobile desktop system with Java Script engine. Figure 2 shows our architecture of the mobile desktop system. We use Linux as an operating system and 2D/3D/Vector graphic libraries.

Mobile Desktop Manager is composed of three major software components. They are User Interface Framework, Mobile Widget, and Web Rendering Engine. Plug-in Manager and Window Manager are sub-components of Mobile Desktop Manager. Mobile Desktop Manager provides a touch-based screen keyboard component for text input from users. Mobile Desktop Manager also supports basic applications such as an image viewer, a media player, a calculator, a clock and a system configuration program, etc.

Web Rendering Engine is typically a component of a web browser. The Web Rendering Engine parses HTML (Hyper-Text Mark-up Language), interprets Java Script language, and displays text and graphic images to users.

TABLE 1. MOBILE WIDGETS

Category	Widgets	Category	Widgets
Button	- Button - Radio Button - Toggle Button	Indicators	- Progress Bar - Progress Pill - Progress Slider
Container	- Drawer - Scroller	Text Input	- Text Field - Password Field - Rich Text Field
Picker	- Date Picker - Integer Picker - Time Picker - File Picker - People Picker	Viewer	- Image Viewer - Web Viewer
Selector	- Check Box - List Selector - Slider	List	- List - Filter List - Widget List
Dialog	- Dialog - Alert Dialog - Error Dialog	Menu	- App Menu - Command Menu - View Menu - Sub Menu

C. Mobile Widgets

Mobile Widget is a kind of system library for graphical user interfaces. It has widgets to support application programmers to build Java Script-based application services. Table 1 shows the elements of Mobile Widgets for button, container, picker, selector, dialog, indicator, text input, viewer, list and menu objects.

Mobile Widget is a package of Java Script library and support compatibility among various operating systems and web browsers. This is the goal of Mobile Widget.

Due to the progress in virtual machine for Java Script language, computing power of central processing units, capacity of memory devices, and speed of web browsers, Java Script-based application development is one of efficient and easy methods for web based application programs.

Mobile Widget also supports event processing mechanisms for touch based mobile devices. The touch-based events are described below.

- Touch-down event
- Touch-Release event
- Long press event
- Touch & drag (8 directions)
- Swipe event (8directions)
- Kinetic scroll (UP/DOWN/LEFT/RIGHT)

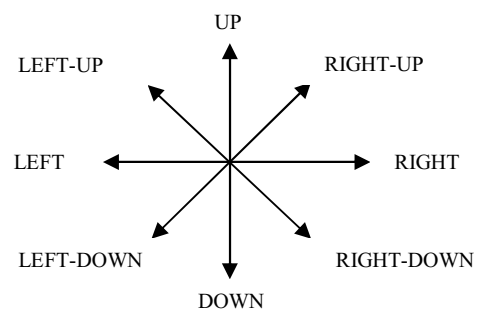


Figure 3. 8 directions

D. User Interface Framework

User Interface Framework includes Plug-in Manager, Window Manger. User interface Framework has functionalities for application install & deletion, application launching & termination, switching among running applications, and management of window resources. User Interface Framework support multi-tasking that several applications can be run simultaneously.

Plug-in Manager supports the following functions.

- Data sharing mechanism among applications
- Message passing mechanism among applications
- Background image management

- Local data base management

Window Manager has the following functions.

- Region control
- Top window control
- Privilege control for window device

The basic layers of Window Manager are Top, Middle, and Bottom Layer. Top Layer is used for event indication such as battery status and network status. Middle Layer is used for application programs. Bottom Layer is a region for static icons of frequently used applications. In Middle Layer users can locate application icons and remove icons with combination of touch events.

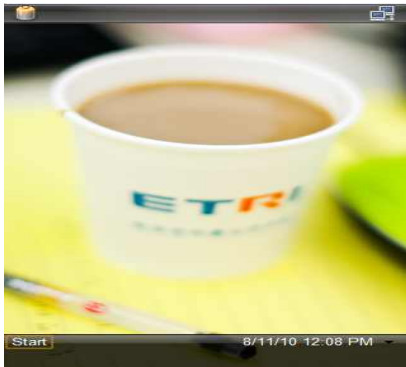


Figure 4. An Example of the Mobile Desktop System

IV. CONCLUSIONS

There are no standard operating system and graphic library for embedded systems such as mobile web devices. This problem makes it difficult to reuse application without changes among the mobile web devices.

To cope with the problem and to meet application compatibility, we suggest Java Script based Mobile Desktop Manager. It can be run on a web browser so that applications of the Mobile Desktop Manager can run on most devices that have web browsers.

Even though computing power of central processing unit is increased, there can be requirement of execution speedup. For this issue, we are going to optimize the code of Java Script Engine.

In addition, we have a plan to provide application development tools on Mobile Desktop Manager for application programmers.

ACKNOWLEDGMENT

This work was supported by the IT R&D program of MKE. [2009-S-029-01, Development of Embedded SW Common Platform for Information and Industrial Devices]

REFERENCES

- [1] Ilhami Visne, et al, "RGG: a general GUI Framework for R scripts," *BMC bioinformatics* vol. 10, pp. 10, 2009.
- [2] S. Kim, et al, "Modeling and Implementation of Multilingual Meta-search Service using Open APIs and Ajax," *Journal of Korean Industrial Information Systems Society* v.14 no.5, pp. 11-18, 2009.
- [3] Sagar, et al, "The Oea framework for class-based object-oriented style JavaScript for web programming," *Computer standards & interfaces* v.31 no.5, pp.894-905, 2009.