Mobile Device Programming and its Applications

Dr. SRN Reddy, Asstt Prof, IGIT, GGSIPU, Delhi (rammallik@yahoo.com)
Prof. Nupur Prakesh, Principal, IGIT, GGSIP U, Delhi (nupurprakash@rediffmail.com)
M.Reddi Prasad Reddy,Research scholar, CDAC, Noida (reddy.mrp@gmail.com)

Abstract

Mobile Phones have become an integral part of our day to day life. The Mobile Phones are not only being used for voice communication and also for data communication but mobile computing. The hardware features of mobile devices support interface with external world with enhanced computational power, multimedia and networking support, large memory with various sensors and increased battery life. There are several Mobile Real-time Operating Systems (MRTOS) available in the market which can be chosen as per requirement. In this chapter, the architecture of the most widely used MRTOS is presented along with the case study of the S60 (Symbian Operating System) from Nokia. The various types of mobile application development environments are described in this chapter and the choice of the development environment is made by the application developers and the programmers based on the comparative study. The mobile phone programming supports various applications like dialing a number, recording the audio and video etc. The various SDKs, Python interpreters and its corresponding script shells required for programming various mobile devices is discussed in detail.

1. Introduction

The phenomenal growth in wireless communication services like wireless web browsing/ mobile internet, real time mobile multimedia streaming and interactive applications have accelerated the growth of reliable ubiquitous computing. The penetration of mobile phones is manifolds higher than that of the land line fixed phones, desktop computers and the mobile computers. Mobile devices are widely used for data communication along with voice communication these days. This integration is due to the several advanced hardware and software features built within these devices in the last five years. Loaded with advanced features, these devices are now called Smart Phones and used as palmtop computers. The hardware features support, enhanced computational power, multimedia interfaces, multiple network interfaces, and large memory with reduced battery consumption.

In this chapter the merits and demerits of various types of mobile application development environments are discussed along with programming of device peripherals. The development environment is selected by the developer based on the features like portability, efficiency, ease of development etc. A mobile device can be programmed as a generic instrument to develop various types of applications like dialing a number, recording audio and video clippings, etc.

The actual mobile devices can be used for demonstrating concepts of communication and mobile computing in real time environment, rather than a simulated environment. The chapter presents a complete overview of various types of mobile architectures,

programming environments and applications.

The seasoned section of the chapter explains the brief architecture of Symbian OS and its features. The various programming environments for the Symbian OS platform is given in section 3. In section 4, the salient features of python mobile programming language is explained. The generic mobile programming environment is discussed in section 5. The step by step procedure for creating a mobile applications and running it on SDK using a in real mobile device is explained in section 6. Finally section 7 gives the details of the sample programs, future work and conclusions

2. Symbian OS architecture

The embedded software in a mobile phone enables the use of hardware effectively and helps in scheduling the multiple tasks in real-time mode. This software is called the Mobile Real-time Operating System (MRTOS). There are several MRTOS available in the market which can be chosen as per requirement. For example, Symbian IOS, Maemo (Meego), Android, Windows CE to name few [1]. In this section, the architecture of the widely used MRTOS, is presented along with the case study of the S60 (Series 60, Symbian Operating System) from Nokia.

2.1. Symbian S60
S60 is a Symbian OS software platform for mobile phones and is one of the oldest MRTOS used in smart phones world over. It was created by Nokia and has been used by several mobile device manufacturers including Siemens mobile, Lenovo, LG Electronics, Panasonic etc. S60 consists of a suite of libraries and standard applications, such as telephony, personal information manager (PIM) tools, and multimedia players. S60 is available in various editions with various features [2,3]. The architecture of the Symbian S60 is shown in figure 1. Symbian OS is the advanced open source operating system adopted the several leading mobile phone manufacturers. It is designed for the specific requirements of advanced 2.5G and 3G mobile phones. Symbian OS combines the power of an integrated applications environment with mobile telephony. Symbian OS supports a wide range of device categories with several user interfaces, including Nokia S60, UIQ and the NTT DoCoMo common software platform for 3G FOMATM handsets.

Key features of Symbian OS: Symbian OS supports several features such as Platform security, Comprehensive Java support, hard real-time capabilities, Support for the latest hardware, Complete messaging capabilities, Rich multimedia capabilities, Powerful graphics, Broad support for communications protocols and Optimized for mobile phones. The layered architectural view of Symbian OS is shown in figure 3. It consists of five layers mainly[2].

UI Framework Layer: UI Framework layer provides the framework and libraries for constructing a user interface, including the basic class hierarchies for user interface controls and other frameworks and utilities used by user interface components.

Application Services Layer: The Application Services layer provides support independent of the user interface for applications on Symbian OS.

OS Services Layer: The OS Services layer is, in effect, the 'middleware' layer of Symbian OS, providing the servers, frameworks, and libraries that extend the bare system below it into a complete operating system. These services are generic operating system services, communication services, multimedia and graphics services, and connectivity services.

Base Services Layer: Base Services layer provides the lowest level of user-side services like the File Server and the User Library.

Kernel Services and Hardware Interface Layer: The lowest layer of Symbian OS is the Kernel (real time) Services and Hardware Interface Layer. It contains the operating system kernel and the supporting components.
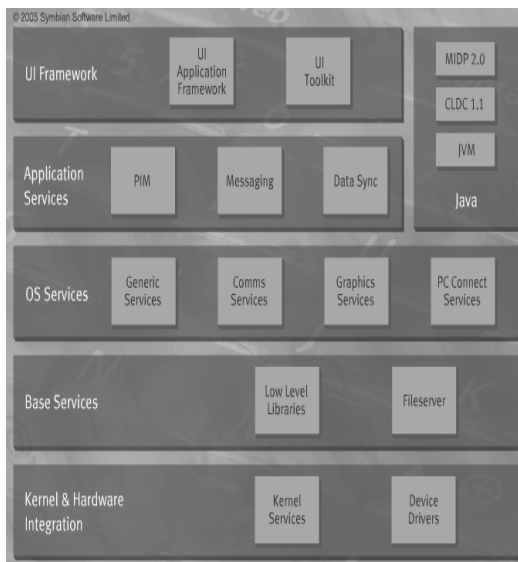

Figure 1. Overview of Symbian Architecture[2]

3. Programming Environments

Four different programming environments are widely used in Symbian platform for application development. Table 1 shows the comparison between these programming environments. In this chapter python application programming environment is discussed in detail because it is very easy to use, interpreted and easy to develop the applications using the mobile device itself rather than simulator.

Parameters	Application Development			
	Python	Java	Symbian C++	Qt
Ease of development	Very Easy and more productive	Easy but not productive	Difficult and productive than Java	Similar to C++ and easier than Symbian C++
Portability	Portable: Needs Separate Interpreter for each platform	More Portable: Needs JVM for each platform	Not portable	Portable: Write once and compile for all
Efficiency	Less efficient	Reasonable	Most efficient	More efficient than Java and Python
Open source	100%	yes	No	yes
Community Support	Strong	Very Strong	Not much	Strong
Documentation	Very good	Very good	Not much	Good

Table 1: Comparison between different programming environments

4. Why Python

Python language has several salient features compared to other languages.

Simple: Python is a simple, easy to learn, interpreted, object oriented high-level language.

Open Source: Python is open source software. One can freely distribute copies of the software, read the source code, make changes to it and even it is free to use for commercial purpose as well.

Portable: Due to its open source nature, Python has been ported to

many platforms like Linux, Windows, Macintosh, Solaris, Palm OS, QNX, VxWorks, Windows CE, Pocket PC, and Symbian S60 to name a few [1].

Embeddable: You can embed Python in your programs written in other languages like C, C++ to give 'scripting' capabilities.

Extensive Libraries: The Python Standard Library is huge. It is distributed with regular expressions, documentation generation, threading, databases, web browsers, CGI, FTP, email, XML, HTML, WAV files, cryptography, GUI etc.

Other Features: It supports dynamic typing, static scoping and is well tested. In python, everything is an object and it is having strong community support.

5. Generic Mobile Programming Environment

The selected SDK (MRTOS+ IDE (Integrated Development Environment) + Emulator) (in this case Nokia SDK) [4] is installed in the normal host environment also known as development workstation shown in figure 2. There exists several Nokia SDKs for Desktops [4]. This environment is used to develop the mobile applications by using the host environment and simulating it on the given emulator (SDK). Once the application is successfully tested with the emulator (SDK), it is transferred to the corresponding mobile device to test it in the real environment as shown in the figure 2. The current SDK (Nokia Qt SDK) supports not only Symbian platforms but also desktop (Windows), Maemo etc. which makes development of mobile applications easy for

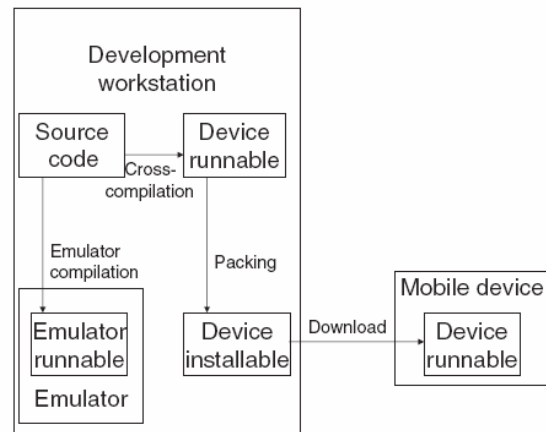different platforms by using a single SDK [5].

Figure 2. Generic Mobile Programming Environment [6]

6. Creating and Running the Mobile Application

The following steps are used to create and run an application on mobile device.

Creation of the source file: Any available text editor can be used to create a python source file which is saved as text file with .py extension.

Transferring the py source file: With the help of PC Suite or by Bluetooth or USB the source file is transferred from desktop to Mobile Phones as shown in figure 2. An open source or commercial editor can be installed in the mobile devices to create source file on the mobile itself. For example, PED[7], a text editor, focused on writing, executing and debugging Python code directly on the mobile device. Additional features like comfortable built-in Python Shell makes it a true mobile Python IDE.

S60 Edition	Python interpreter and script shell files for S60 [3, 8]
2nd Edition	PythonForS60 1 4 0 2ndEd.SIS
	PythonScriptShell 1 4 0 2ndEd.SIS
2nd Edition Feature Pack 2	PythonForS60 1 4 0 2ndEdFP2.SIS
	PythonScriptShell 1 4 0 2ndEdFP3.SIS
2nd Edition Feature Pack 3	PythonForS60 1 4 0 2ndEdFP3.SIS
	PythonScriptShell 1 4 0 2ndEdFP3.SIS
3nd Edition	PythonForS60 1 4 0 3rdEd.SIS
	PythonScriptShell 1 4 0 3rdEdselfsigned.SIS

Table 2 selection of python interpreter and its script shell for different Symbian mobile devices

Running the Program: Python interpreter must be installed in the mobile device as shown in figure 3 or installed on the desktop in order to run the python source file. Since, python is an interpreted language, hence, its interpreter must be preinstalled. Different Symbian mobile devices use different Symbian S60 SDKs as depicted in figure 3. The selection of the python interpreter and script shell required for different Symbian devices is shown in table 2[4]. For example to run a python source file in Nokia E63 mobile device, *PythonForS60 1 4 0 3rdEd.SIS (*python interpreter*)* and *PythonScriptShell 1 4 0 3rdEdselfsigned.SIS (*Python script shell, needs to be installed, which provides the environment to run the python source files)* by referring to the table 2 and figure 3. Similarly a suitable python interpreter and its corresponding script shell must be installed based on the mobile device. Run script option is selected to run the python source file. Interactive console option is used to execute the python

program line by line. The result of the execution or error message will be available on the display of the screen of the mobile device immediately.
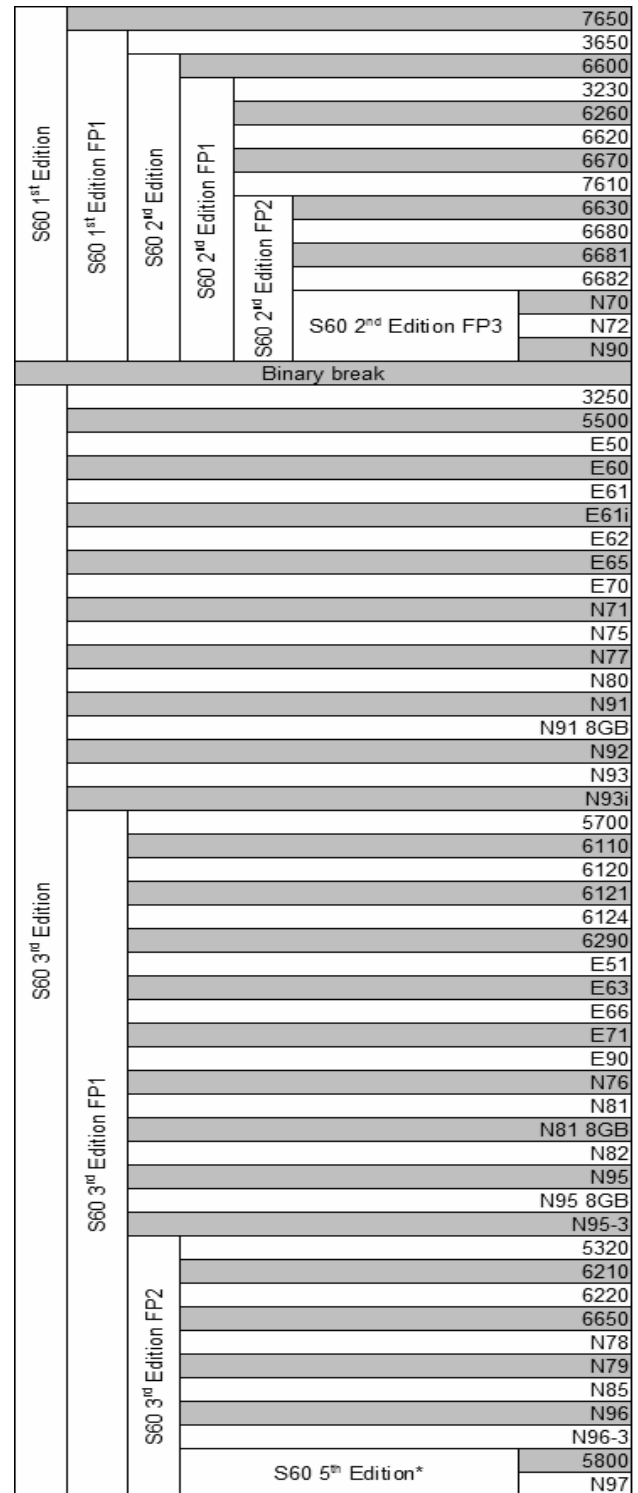

Figure 3 Mobile device and its supported SDKs[9]

7. Sample Programs

The following six sample python programs are given for Symbian s60 based mobile devices. Several application programs and relevant software required for different devices can be found in [3].

```
Program for Text to speech Conversion

import appuifw, audio
data= appuifw.query(u"Type a word:",
"REDDY")
audio.say(data)
Program for Taking a Photo

import camera
photo = camera.take_photo()
photo.save("E:\\Images\\myphoto.jpg")

Program for Dialing a Number

import telephone, e32
telephone.dial('9810101742')
e32.ao_sleep(10)
telephone.hang_up()

Program for playing a song
import audio
sound=audio.Sound.open("E:\\Sounds\\m
ysound.mp3")
def playMP3():
sound.play()
playMP3()
Program to Measure the Battery Level
import sysinfo
print "Battery level is: %d" %
sysinfo.battery()

Program to read SMS inbox

import inbox, appuifw
box = inbox.Inbox()
for sms_id in box.sms_messages()[:4]:
msg = box.content(sms_id)
appuifw.note(msg)
```

8. Conclusion
In this chapter the Symbian OS architecture and its different programming environments are presented. Python language has been chosen for developing the mobile applications. The required SDKs, python interpretors and its script shell for various mobile devices is also illustrated. The mobile devices can be used in new applications like Health care, Education, Wireless Sensor networks for the benefit of society at large.

References:
1. http://www.emich.edu/compsci-/projects./Master_Thesis_-_Benjamin_Speckmann.pdf
2. S60: http://en.wikipedia.org/wiki/Symbian_OS
3. https://garage.maemo.org/projects/pys60/
4. Nokia SDK: http://www.forum.nokia.com/info/sw.nokia.com/id/4a7149a5-95a5-4726-913a-3c6f21eb65a5/S60-SDK-0616-3.0-mr.html
5. http://www.forum.nokia.com/Library/Tools_and_downloads/
6. Tommi Mikkonen, "Programming Mobile Devices: An Introduction for Practitioners", John Wiley & Sons Ltd,2007.
7. PED : http://handheld.softpedia.com/dyn-postdownload.php?p=65727&t=0&i=3
8. https://garage.maemo.org/projects/pys60/
9. http://wiki.forum.nokia.com/index.php/image:sdk.png
10. J Scheible and Ville Tuulos John, "Mobile Python Rapid Prototyping of Applications on the Mobile Platform" Wiley India Pvt. Ltd, 2008.

Profiles of Authors

Brief Profile About self Dr. (Mrs.) Nupur Prakash is presently working as Principal of

Indira Gandhi Institute of Technology (IGIT), a constituent college of GGS Indraprastha University and a girl's engineering college. Prior to this assignment she was Additional Director (IT) at Institute of Chartered Accountants of India, Delhi. She has worked as Professor and Dean in the University School of Information Technology at GGS Indraprastha University, Delhi. She has also held the post of Principal, Indira Gandhi Institute of Technology, for four years (2002-06).

She received her B.E. (Electronics & Communication Engineering) and M.E. (Computer Science & Technology) degree from University of Roorkee (now I.I.T. Roorkee) in 1981 and 1986 respectively. She was awarded with the Chancellor's Medal for being the best outgoing student of 1981. She has also worked as a scientist at CSIO, Chandigarh (1983-85) on microprocessor based projects.

She has served the Department of Computer Science & Engineering at Punjab Engineering College, Chandigarh as Head and Assistant Professor (1991-2000) and has traveled to Australia, Malaysia, Dubai, USA, Tokyo, Italy, New Zealand and Austria to deliver invited talks, attend workshops and present papers at several International Conferences.

She possesses a work experience of around 29 years in research and academic administration. Her major areas of interest are **Mobile Communication, Network Security, Cryptography, Secure Wireless Communication** and **Natural language Processing**. She has authored around 100 research papers and articles for various national and international journals/conferences. She is a life member of professional bodies like Computer Society of India (CSI) and Associate member of Institute of Electronics and Electrical Engineers (IEEE), USA.

Dr. S. Ramanarayana Reddy has received his PhD in Embedded

Systems Design and M. Tech in CS, from Jawaharlal Nehru University, New Delhi in 2009 and 2002 respectively. He is working as Asst. Professor (from 2002) in the Dept. of Computer Science, IGIT, IP University, Delhi. His research interest includes embedded systems, Mobile Architecture and programming, real-time systems and publishes more then 20 research papers that include international journals, conferences. He is teaching the embedded systems course for M.tech and B.Tech students since 2002. He has 14 years experience that includes industry, consultancy and teaching. He is a member of CSI, VLSI Society of India and AMIE. He has conducted several training programs and workshops in the area of embedded systems and Mobile Architecture and Programming and handling the sponcered projects from industry. He has conducted full day tutorials on "Embedded Systems Design", and "Mobile Architecture and Programming" in few conferences and workshops. rammallik@yahoo.com

Mr. M. Reddi Prasad Reddy

M. Reddi Prasad Reddy is a Project associate at I.I.T. Delhi. He obtained his B.Tech in Information Technology from Sreenivasa Institute of Technology and Management studies, Chittoor in 2009 and M.Tech in VLSI Design. from School of Electronics, CDAC NOIDA in 2011. He has published one article in e-magazine at SITAMS, Chittoor and one paper in international conference. His areas of interest are Mobile computing, Cryptography, Mobile Education and Grid computing. He has given/attended many seminars in faculty development programs, workshops and training programs.