Understanding and Developing Applications for Maemo Platform

Speaker:
Leandro Melo de Sales
leandro@embedded.ufcg.edu.br
About me

- PhD candidate at Federal University of Campina Grande, Paraíba, Brazil; Professor in Computer Science at Federal University of Alagoas

- Have been working for embedded systems:
  - Location Based System
  - Universal Plug and Play
  - Mobile Payment
  - VoIP, DCCP protocol and Linux Kernel
  - Maemo PC-Connectivity
Summary

- Introduction
- Maemo timeline
- Maemo development
- Examples & Applications
- Conclusion
Intro: What is maemo platform?

- Maemo is a software platform developed by Nokia for smartphones and Internet Tablets.
- Based on the Linux.
- Maemo is mostly based on open source code with many open source projects such as the Linux kernel, GTK and Qt.
Intro: presentation goal

Provide to the attendees the current maemo state-of-art in terms of development tools for maemo platform

Expected result: allow attendees to become able to develop their own maemo applications and also introduce the next step of a near future: the MeeGo platform.
Maemo timeline: Nokia 770

- **1st version**: November, 2005
- OS2005 - v1.1; OS2006 - v2.0-2
- Processor: 252Mhz TI-OMAP
- Memory: 64MB RAM, 128MB Flash
- Bluetooth and IEEE 802.11g
- Display: 800x600 4.13in diagonal

**Tools:**
- Opera web browser
- Flash 6
- E-Mail & RSS
- PDF reader
- Music & video player
Maemo timeline: Nokia n800

- **2nd version**: January, 2007
- OS2007 - v3.0-2; OS2008 - v4.0-1
- Processor: 330-400Mhz TI-OMAP
- Memory: 128MB RAM, 64GB Flash
- Bluetooth, IEEE 802.11g & USB 2.0
- Two slots SD (micro/mini)SD, MMC

**Tools:**
- Mozilla-based Micro-B browser
- Flash 7
- E-Mail & RSS
- PDF reader
- Music & video player
- Camera 640x480 VGA
- Gizmo / Skype
- FM Radio Tuner
Maemo timeline: Nokia n810

- 3rd version: October, 2007
- OS2008 - v4.0-1
- Processor: 400Mhz TI-OMAP 2420
- Memory: 128MB RAM, 64GB Flash
- Storage: 256MB + 2GB Flash
- Bluetooth 2.0, IEEE 802.11g & USB 2.0
- Two slots SD (micro/mini)SD, MMC
- WiMax Edition
  - April, 2008
  - Production canceled in Jan, 2009
Maemo timeline: Nokia n810

Major changes from N800

- Sliding, backlit keyboard
- Front-facing webcam (replacing pop-out rotating device)
- Ambient Light Sensor
- Integrated GPS
- No longer has an FM tuner
Maemo timeline: Nokia n900

- 4th version: November, 2009
- Maemo 5 Linux
- Processor: 600Mhz TI-OMAP 3430 Cortex A8
- Memory: 256MB RAM, 768 Swap space
- Storage: 256MB + 32GB eMMC Flash
- Bluetooth 2.1, IEEE 802.11g & USB 2.0
- One slot microSD microSDHC
- Tools:
  - Mozilla-based Micro-B browser
  - Flash 9.4 & RSS
  - Phone Application
  - VoIP: SIP, Skype, GTalk
  - OVI Maps
Maemo timeline: Nokia n900

Major changes from N810

- New user interface with multiples environments support
- Graphics: PowerVR SGX 530 GPU with OpenGL ES 2.0
- Camera: 5.0MP, Carl Zeiss Tessar lens (rear camera) 0.3MP (640×480) (front camera)
- Connectivity: GSM 850/900/1800/1900
  - GPRS 107/64 kbps DL/UL
  - EDGE 296/178 kbps DL/UL
  - UMTS 900/1700/2100
  - WCDMA 384/384 kbps DL/UL
  - HSPA 10/2 Mbps DL/UL
  - Bluetooth 2.1
  - FM receiver / Transmitter
  - Infrared transceiver
- Battery: 1320 mAh and 2400mAh
Maemo timeline: maemo + moblin = MeeGo

5th version: May, 2010?
Maemo Platform - Architecture

- Based on open source components
  - Linux
  - Hildon
  - LibOSSO
  - GTK+
  - D-BUS
  - Qt
  - Bluez
  - Gstreamer
  - ...
Maemo development

- Development is similar to Desktop
- A subset of Linux libraries ported to maemo
- Virtual environment based on Scratchbox
- GNU Toolkit (gcc, gdb, make, autotools, ...)

Leandro Melo de Sales, Msc
Federal University of Campina Grande
Maemo development

Supported programming languages

- Shell scripts (remember, it is Linux!)
- C / C++ / Qt
- Python
Maemo development - C / C++

- Code the program in the desktop
- Execute the program in the device
- Connect to device
- Copy the program to device
- Cross-compile to ARM
- Start scratchbox
- Execute the program in the device
- Connect to device
- Copy the program to device
- Cross-compile to ARM
- Start scratchbox
Maemo development - Python

- Alternative to C / C++
- Simple and easy to learn
- No need of scratchbox
Maemo development - C / C++

1. Code the program in the desktop
2. Start scratchbox
3. Cross-compile to ARM
4. Execute the program in the device
5. Copy the program to device
6. Connect to device
7. Execute the program in the device
8. Copy the program to device
9. Connect to device
10. Copy the program to device
11. Execute the program in the device
Maemo development - Python

1. Code the program in the desktop
2. Execute the program in the device
3. Connect to device
4. Copy the program to device
Maemo development - Python

The development of applications for mobile devices is becoming more simple.
Maemo development - Tools

- Scratchbox + vi

- Maemo PC-Connectivity (MPC)

- IDE Integration / ESBox / PluThon
  http://maemo.org/development/documentation/ide_integration

- Qt Creator
  http://qt.nokia.com/products/developer-tools

- Maemo Application Development and Debugging Environment (MADDE)
  http://wiki.maemo.org/MADDE
Maemo development - Tools

Scratchbox + maemo sdk
- The basic maemo development environment
- It allows compilation in C and C++
- It provides python VM
- See example...
Maemo development - Tools

Maemo PC Connectivity (MPC)
- The default maemo development environment consists of two parts:
  - The host PC environment with a maemo SDK installation
  - Tablet (device)
- Components are usually developed as far as possible using the maemo SDK (on the host PC)
- Only after everything seems to function is testing done with the true Tablet environment
- It must be easy to run, test and debug components using the Tablet

http://maemo.org/development/documentation/pc_connectivity/
**Maemo development - Tools**

**IDE Integration / ESBox**
- ESbox is an Eclipse-based product that helps programmers to develop applications for Maemo platform using Scratchbox.
- It supports C/C++ and Python programming languages with source editing, code completion, build, launch, debug, profiling, and packaging support.

http://esbox.garage.maemo.org/
Maemo development - Tools

IDE Integration / ESBox (contd.)
- Maemo development - Tools
  - IDE Integration / ESBox (contd.)

[Image of a computer screen showing a MaemoPad window and an ESbox development environment]
Maemo development - Tools

IDE Integration / PluThon
- PluThon is an Eclipse based product that provides support for developing Python applications for maemo
- It does NOT require Scratchbox
- Developers run and debug applications directly on a maemo device, speeding up the development time

http://pluthon.garage.maemo.org/
Maemo development - Tools

IDE Integration / PluThon (contd.)
Maemo development - Tools

IDE Integration / PluThon (contd.)
Maemo development - Tools

Qt Creator

- Cross-platform integrated development environment (IDE) for developing Qt applications
- It provides:
  - An advanced C++ code editor
  - Integrated GUI layout and forms designer
  - Project and build management tools
  - Integrated, context-sensitive help system
  - Visual debugger
  - Rapid code navigation tools
  - Supports multiple platforms
  - Recent/preliminary support for Symbian platform
  - Support for maemo platform through MADDE

Maemo development - Tools

QtCreator (contd.)
Maemo development - Tools

MADDE

- Maemo Application Development and Debugging Environment
- It is in technical preview status
- It provides:
  - Command-line cross-compiling
  - Multi-platform support (Linux (32-bit/64-bit), Windows, Mac OS X)
  - Configurable for different targets & toolchains
  - Client for the device to simplify the development process

- It will [probably] become part of the development environment for MeeGo
- http://wiki.maemo.org/MADDE
**Maemo development - Libraries**

- **Hildon**
  - A *framework* for developing GUIs
  - Provides a set of dialogs, widgets and themes based on GTK
Maemo development - Libraries

LibOSSO
- Access some system and device functions
- Date/time management
- Display management (turn on/off)
- Battery level
- Auto-saving / state-saving
- Notification dialogs
Let's go to examples...

```python
# Create application menu
menu = gtk.Menu()
items = [gtk.MenuItem("Open...")],
       gtk.MenuItem("Save"),
       gtk.MenuItem("Exit")]

# List of callbacks to handle menu events
callbacks = [self.open, self.save, self.exit]

# Connecting callbacks to each items of the menu
for i in range(len(items)):
    items[i].connect("activate", callbacks[i])
    menu.append(items[i])

# Set the menu of the window
self.window.set_menu(menu)
```
Let's go to examples... (contd.)

```python
# Open existing file
def open(self, widget, *args):
    # Create a file chooser dialog to select which file to open
    dlg = hildon.FileChooserDialog(self.window,
                                    gtk.FILE_CHOOSER_ACTION_OPEN)
    response = dlg.run()
    # If the user clicked in ok, we get the selected filename, get
    # the content of the selected file and put it into the textbuffer
    # of the textview object
    if response == gtk.RESPONSE_OK:
        self.filename = dlg.get_filename()
        infile = open(self.filename, 'r')
        if infile:
            string = infile.read()
            self.textbuffer.set_text(string)
            infile.close()
    dlg.destroy()
```
Let's go to examples... (contd.)

```python
# This function creates a libOSSO info_print object
# and shows it, returning the result

def show_infoprint(obj_name, message):
    osso_c = osso.Context(obj_name, "0.0.1", False)
    note = osso.SystemNote(osso_c)
    note.system_note_infoprint(message)
```

Hello World!!!
Welcome to the Bossa Conference.
Examples of complete apps / libs

- LibGPSBt - provides API to get access to the geographical coordinates
Examples of complete apps / libs

Carman - application to get information from cars through bluetooth: velocity, engine rotation, fuel level, ...
Examples of complete apps / libs

- Canola - multimedia player
Next step: MeeGo Platform

MeeGo is an open source, Linux project which brings together
  – Moblin project, by Intel
  – Maemo project, by Nokia

Integration between them into a single open source activity integrates the experience and skills of two significant development ecosystems

Many of the presented development tools will be available in the near future

Support platform for a variety of devices: netbooks, pocketables, in-vehicle, connected TV and smartphones
Resources

- http://maemo.org/
- http://maemo.org/development/
- http://garage.maemo.org/
- http://meego.com/
Conclusion

- Understanding and developing applications for maemo platform
- Platform that is getting more and more popular, providing interesting environment for mobile application development
- It still getting improved and supported by Nokia and the open source community
- The latest big news:
  - Release of maemo 5 and Nokia N900 device
  - Merge(Maemo, Moblin) = Meego...
- See you in the demos session
  - Evolution of maemo platform
Thank you!

Question and discussions