

Openmoko

Development environment

Neng-Yu Tu

2008/09

Outline

- Introduction
- 2007.2
- 2008.8
- Development
- Sample/Demo

Introduction

Introduction

- Openmoko development
 - GTK
 - Om2008.8 (Qt/x11)
 - FSO (python)
 - Others
- Community Image
 - Debian
 - Gentoo
 - Qtopia
 - Others

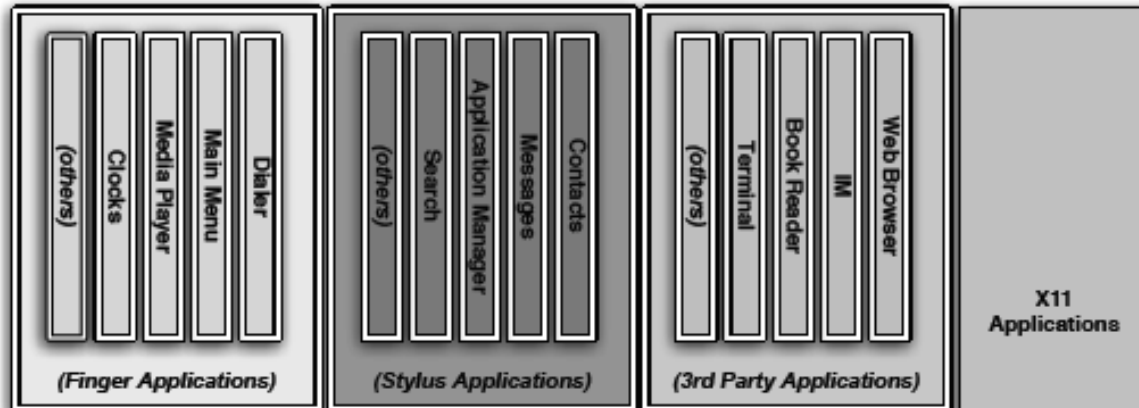
Popular Distro

- Current
 - GTK (2007.2)
 - <http://downloads.openmoko.org/releases/Om2008.4/>
 - QT/X11 (ASU)
 - <http://downloads.openmoko.org/releases/Om2008.8-update/>
 - *Qtopia
 - <http://www.qtopia.net/modules/mydownloads/singlefile.php?lid=83>
 - FSO
 - <http://wiki.openmoko.org/wiki/FSO>
 - *Debian
 - <http://wiki.debian.org/DebianOnFreeRunner>
- Others

2007.2

2007.2

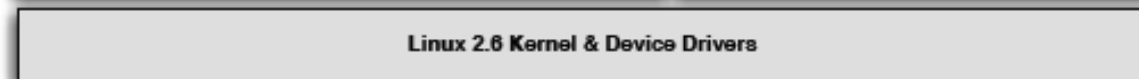
GPL



LGPL



GPL



(Target Board)



(PC)

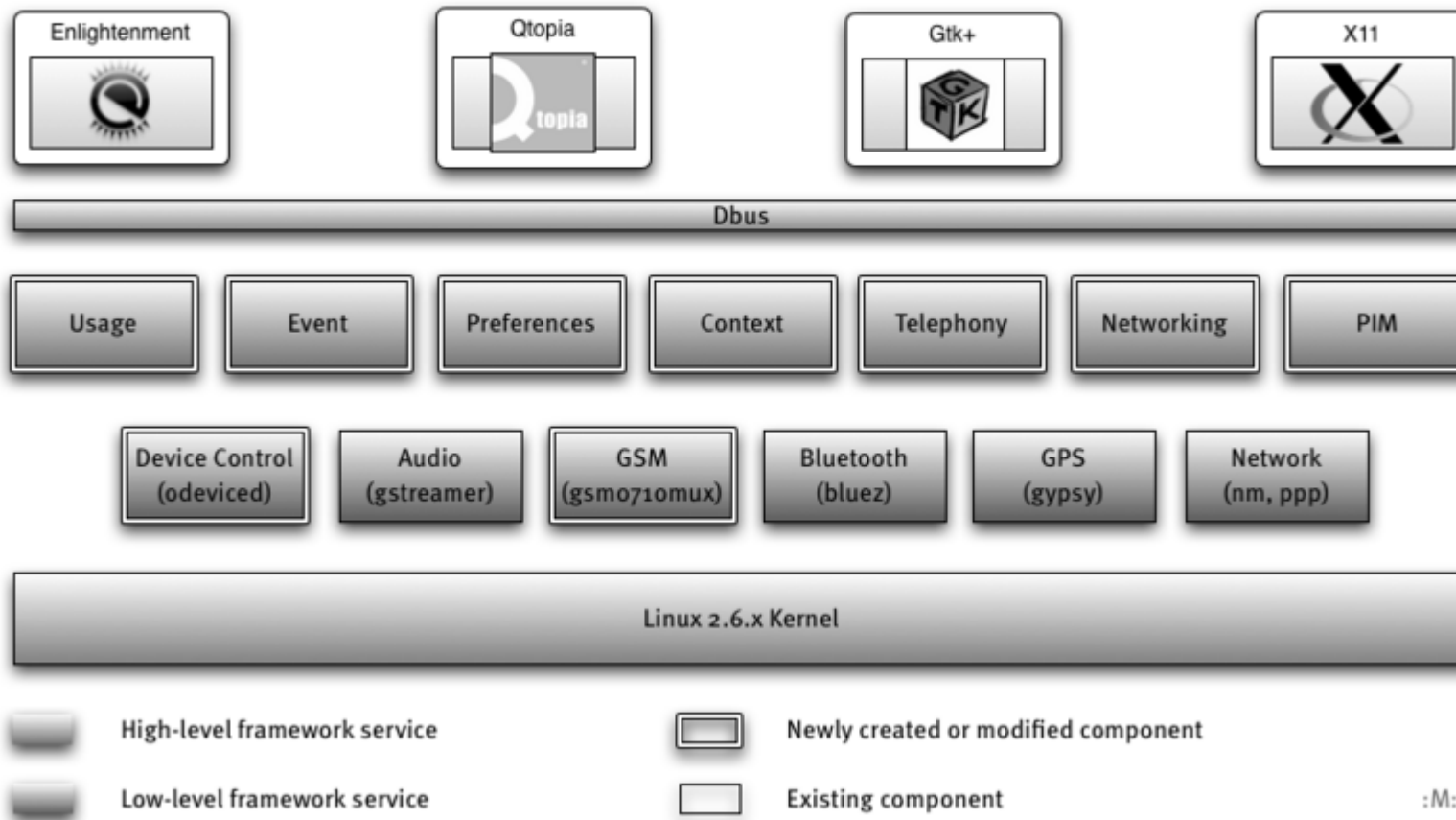
2007.2

- OM2007.2
 - same foundation as OM2007.1
 - leaner
 - cryptic
 - GTK based
 - Migrated to 2008.8 and FSO

2008.8

Architecture

Openmoko 2008 Software Architecture



ASU

- Qtopia-based + X11
 - Mature "standard" applications (from Qtopia) plus our own goodies (built around EFL, Enlightenment Foundation Libraries)
 - "Illume" (Enlightenment module: Application launcher, keyboard, applets, etc.)
- Application
 - Installer (opkg)
 - "Splinter" (GPS maps and annotations - POI)
 - Setting (configuration - suspend time, change ring profiles, select Wifi)

FSO

- FSO (@ milestone 3 now)
 - Not a GUI per se but middleware/framework (hw abstraction, GSM, GPS, PIM services, etc.)
 - Dbus-based
 - Proof-of-concept EFL-based UI on top of it (might get scrapped)
 - Language-agnostic (as long as you speak dbus)
 - Basis for future development
 - Python is recommended language for GUI apps
- Localtion
 - <http://downloads.openmoko.org/framework/milestone3/>

Development

How-to start?

1. What is your needs
2. Check wiki
3. Ask question in community
4. Select images
5. Install toolchain or mokomkefile
6. Start development
7. Join community, find asnwer by yourself

Development Resources

- General
 - [wiki www.openmoko.org](http://wiki.openmoko.org)
- Shop
 - www.openmoko.com
- Mailing lists
 - <http://lists.openmoko.org/mailman/listinfo/>
- Sources
 - git.openmoko.org
 - svn.openmoko.org
- Schematics/mechanical CAD files
 - downloads.openmoko.org
- Production test
 - git.openmoko.org

Firmware Update

- DFU-util
 - Device Firmware Update (USB standard implementation)
- Install

```
svn co http://svn.openmoko.org/trunk/src/host/dfu-util/  
cd dfu-util  
./autogen.sh  
./configure  
Make
```
- Boot from SD card
 - http://wiki.openmoko.org/wiki/Booting_from_SD

Partitions

- GTA02 have following partitions
 - U-boot (u-boot)
 - U-boot environment (u-boot_env)
 - Kernel (kernel)
 - Splash
 - Factory
 - Rootfs
- Command
 - `./dfu-util -a 1 -R -D /path/to/openmoko-devel-image.jffs2`
 - `./dfu-util -a u-boot -R -D /path/to/openmoko-devel-image.jffs2`

Toolchain

- Wiki
 - <http://wiki.openmoko.org/wiki/Toolchain>
- How to install toolchain
 - Put in the `/usr/local/openmoko`
- Meta toolchain
 - More flexible
 - Small footprint

MakeFile

- Function
 - Automatic download all sources
 - Automatic download required tools
- Mokomakefile
 - <http://wiki.openmoko.org/wiki/MokoMakefile>
- Others
 - fsomakefile

U-boot

- Download Source
 - `git clone git://git.openmoko.org/git/u-boot.git`
- Toolchain
 - <http://wiki.openmoko.org/wiki/Toolchain>

`make ARCH=arm`

`CROSS_COMPILE=/usr/local/openmoko/
arm/bin/arm-angstrom-linux-gnueabi- u-
boot`

Kernel

- Download source
 - `git clone git://git.openmoko.org/git/kernel.git linux-2.6`
- Tree
 - Development tree
 - Stable tree
 - Function specific tree
- Build
 - `git clone git://git.openmoko.org/git/kernel.git linux-2.6`
 - `cd linux-2.6`
 - `git checkout -b mystable origin/stable`
 - `cp defconfig-gta02 .config`
 - `./build`
- `make ARCH=arm`
`CROSS_COMPILE=/usr/local/openmoko/arm/bin/arm-angstrom-`
`linux-gnueabi- ulmage`

QEMU

- Function
 - Work without device
 - Accelerate development
 - Good for high level development
- Install
 - Install from Mokomakefile

Application

- Source
 - <http://downloads.openmoko.org/sources/>
- Development process
 - Download proper image
 - Select language RAD tool
 - Deployment

Samples/Demo

Openmoko Sample

- Location

- <http://wiki.openmoko.org/wiki/Mokomakefile>

- ```
cp -r /usr/local/openmoko/source/openmoko-sample2 ~/
```

- Remember to set the proper environment variables (again with "sh" or "bash") for openmoko:

- ```
./usr/local/openmoko/arm/setup-env  
om-conf openmoko-sample2
```

- Optionally now you can modify the source code in openmoko-sample2/src. Before the next step, go into the sample directory.

Openmoko Sample

```
cd openmoko-sample2
```

- If you are using an older version of the toolchain, you may have to create the makefile by running `./autogen.sh`. Otherwise, to build the application from the source code just type:

```
Make
```

- If there are errors (i.e. "You need to install gnome-common from the GNOME CVS") deal with them. Also see "Troubleshooting" section at the end of this page for known issues.
- If you want to install this project on host for staging usage later, a shared library, for example, you can do the following to install it into a given configured prefix.

```
om-conf --prefix=/usr/local/openmoko openmoko-sample2  
cd openmoko-sample2  
make install
```

Python

```
import etk

#create a button (not yet on any window)
b = etk.Button(label="Hello")

#create a (nonvisible) window and put the button on the window
w = etk.Window(title="Hello", child=b)

#create a silly callback function
def hello(target):
    print 'Hello World'
    etk.main_quit()

#make the button call the callback when pressed
b.on_clicked(hello)

#make the window display
w.show_all()

#start processing screen events
etk.main()
```

Now, “Free Your Phone.”

Thanks for Your Time.