Martin Kolman, Faculty of Informatics, Masaryk University

TIZEN

maemo.org

webOS

MeeGo

Mobile Linux distributions

SHR

mer

QtMoko

slides: http://www.modrana.org/om2012

Android

first device: **T-Mobile G1 (HTC Dream)**
released: **October 2008**

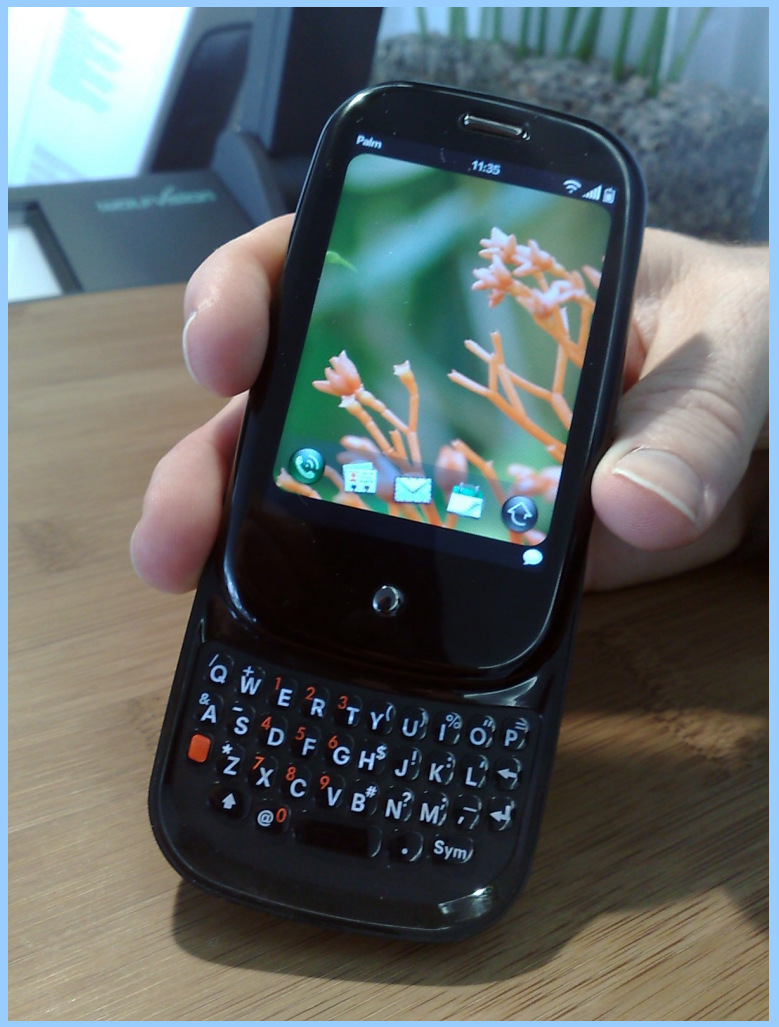
Android

- the most used Linux based mobile “distribution”
- developed by Android Inc., which was bought by Google in 2005
- uses (a patched) Linux kernel and mostly custom userspace
- the primary development language is a custom version of Java running on the Dalvik VM
  - programs written in this language are not compatible with “normal” Java SE
- a native SDK (called NDK) was later added
- versions 1-2 targeted mainly mobile phones, version 3 targeted tablets and the current Android 4 “Ice Cream Sandwich” targets both

Android

- software is distributed using Google Play (previously called Android market) software catalog, but many similar third-party stores exist
  - software is distributed as *APK* packages, which are based on the *JAR* file format
- Android doesn't use the X Server and has its own graphics system
- Android is open source, most of the code is licensed under the Apache license, with the rest (including Kernel changes) being under GPL v2
  - while the code is open, the development is not
  - Google also already once withheld the source code release (for Android 3)
  - some Google provided applications (such as Google Maps) are closed source

WebOS



first device: **Palm Pre**
released: **June 2009**

WebOS

- initially developed by Palm, then bought by HP (with Palm)
- OpenEmbeaded is used to build the WebOS images
- doesn't use an X Server, runs directly on the framebuffer
- primarily based on Web technologies – HTML & javascript
  - in form of the older Mojo and newer Enyo GUI frameworks
- a native SDK (called PDK) was released later
  - this SDK also contains an official SDL port, used mostly by games
- WebOS 1-2 targets mobile phones, WebOS 3 is running on the Touchpad tablet
  - parts of the WebOS 3 GUI use Qt & QML, which is unfortunately not accessible to third party application

WebOS

- applications are distributed using the built-in application catalog or the community-supported *preware* repository
  - the *ipkg* package management system is used
- future outlook is uncertain since HP decided to abandon future WebOS development in July 2011
  - this decision might have been at least partially reversed
  - this was also the cause for all the Touchpads sold for 100 & 150 $
- the whole codebase is currently being open-sourced
  - the target date for a fully open source WebOS is September 2012

Maemo

first device: **Nokia N770**
released: **November 2005**

Maemo

- originally a Linux-based operating system for Internet Devices – small tablets using wifi for Internet connectivity
- based on Debian, using Debian packages
  - older Maemo versions (up to 4) had no central repository for community applications
- uses X server and a GTK extension called Hildon for its GUI
- contains many non-open components
  - this is seriously hindering community OS maintenance

Maemo 5

first device: **Nokia N900**
released: **November 2009**

Maemo 5

- first Maemo version for mobile phones
- Hildon was upgraded with Clutter for animation & transition support
- Open GL ES support
- has a central repository for community applications called Extras
  - there is a community QA in place: applications first go to Extras-devel, from there they go to Extras-testing
  - after getting enough positive votes from community testers, they are promoted to the main Extras repository (which is enabled by default on all devices)
- Ovi Store also supplies Maemo 5 applications

Maemo 5

- has nice IM integration & sharing framework
- GTK, Clutter and Qt are officially supported
  - there are also Python bindings for most of the main OS components
  - there are many community supported libraries, toolkits and software packages
- last big official update from Nokia, called PR 1.3, was issued in 2010
  - since then the community started its own update initiative called CSSU
    - contains a newer version of the Qt libraries and many bugfixes
    - hindered by many closed-source drivers & system components

MeeGo 1.2 Harmattan
(Maemo 6)

first device: **Nokia N9** released: S**eptember 2011**

MeeGo 1.2 Harmattan *(Maemo 6)*

- MeeGo in name only
- while it contains some MeeGo components, most of the stack was recycled from Maemo
- its graphical interface is called “Swipe UI” and is based on the Qt-based MeeGo Touch framework and QML
  - still uses the X Server
- Nokia store (formerly Ovi store) serves as the main repository for both commercial and community applications
  - there is also the Apps for MeeGo community repository
  - unfortunately, both repositories currently don't have support for handling third-party libraries and other dependencies
- Python bindings for Qt (PySide) are available by default and applications written in Python can be included in the Nokia store

MeeGo

first device: **?**
released: **never**

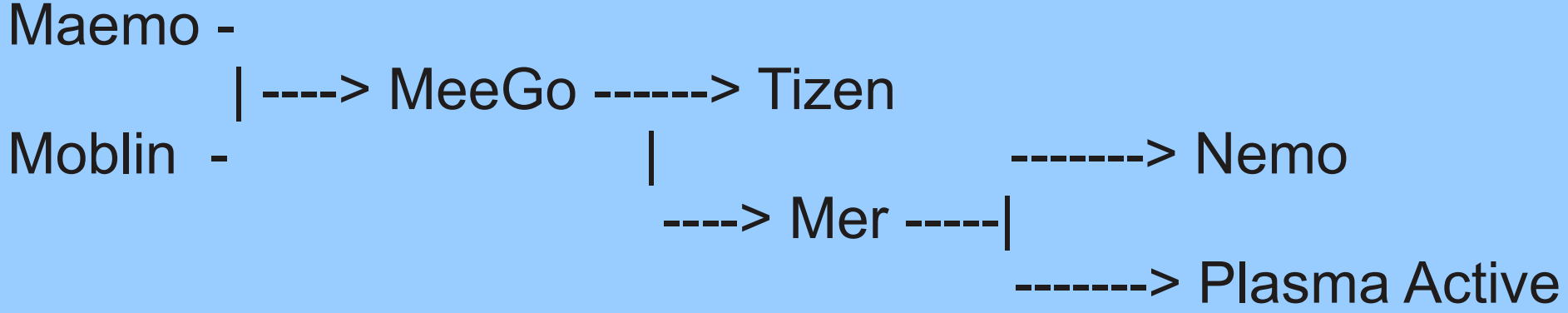
MeeGo

- MeeGo was a cooperative effort between Nokia, Intel and other partners, with the MeeGo trademark administered by the Linux Foundation
  - the initial plan was to merge Intel's distribution for netbooks called Moblin with Nokia's Maemo mobile operating system
- Qt was to be the main GUI toolkit
  - there were separate reference GUIs for Netbooks, Tablets, interactive TVs and in-vehicle use (IVI)
- RPM was selected as the package format
- the OBS (Open Build System) was selected for building packages and installation images
- MeeGo was to be meritocratically governed

MeeGo

- in 1.3, the last actively developed MeeGo version, SystemD was integrated
- on 11th February 2011 Nokia announced a change in strategy from MeeGo to Windows Phone and in September 2011 the Tizen project was announced
  - as a result, MeeGo is currently dead
  - no real MeeGo mobile device was ever released

MeeGo family tree



Tizen

first device: **?**
released: **not yet**

Tizen

- the official MeeGo successor
  - reuses very little of the work done for MeeGo, if any
- the main actors are Samsung and Intel
- based mainly on HTML 5 technologies and Enlightenment Foundation Libraries (EFL)
- uses X Server
- preliminary SDK is already out, no hardware announced so far

Mer

first device: n**/a**
released: n**/a**

Mer

- the de-facto MeeGo successor
  - reuses most of the MeeGo core components
- Mer is openly developed and meritocratically governed
- it is actually just a core without a default GUI or device adaptations
  - it can be used for building modern mobile distributions
- contains the X Server, SystemD, QT libraries, D-BUS support, Connman, Gstreamer and others
- uses the RPM package management system
- packages & images are built with OBS

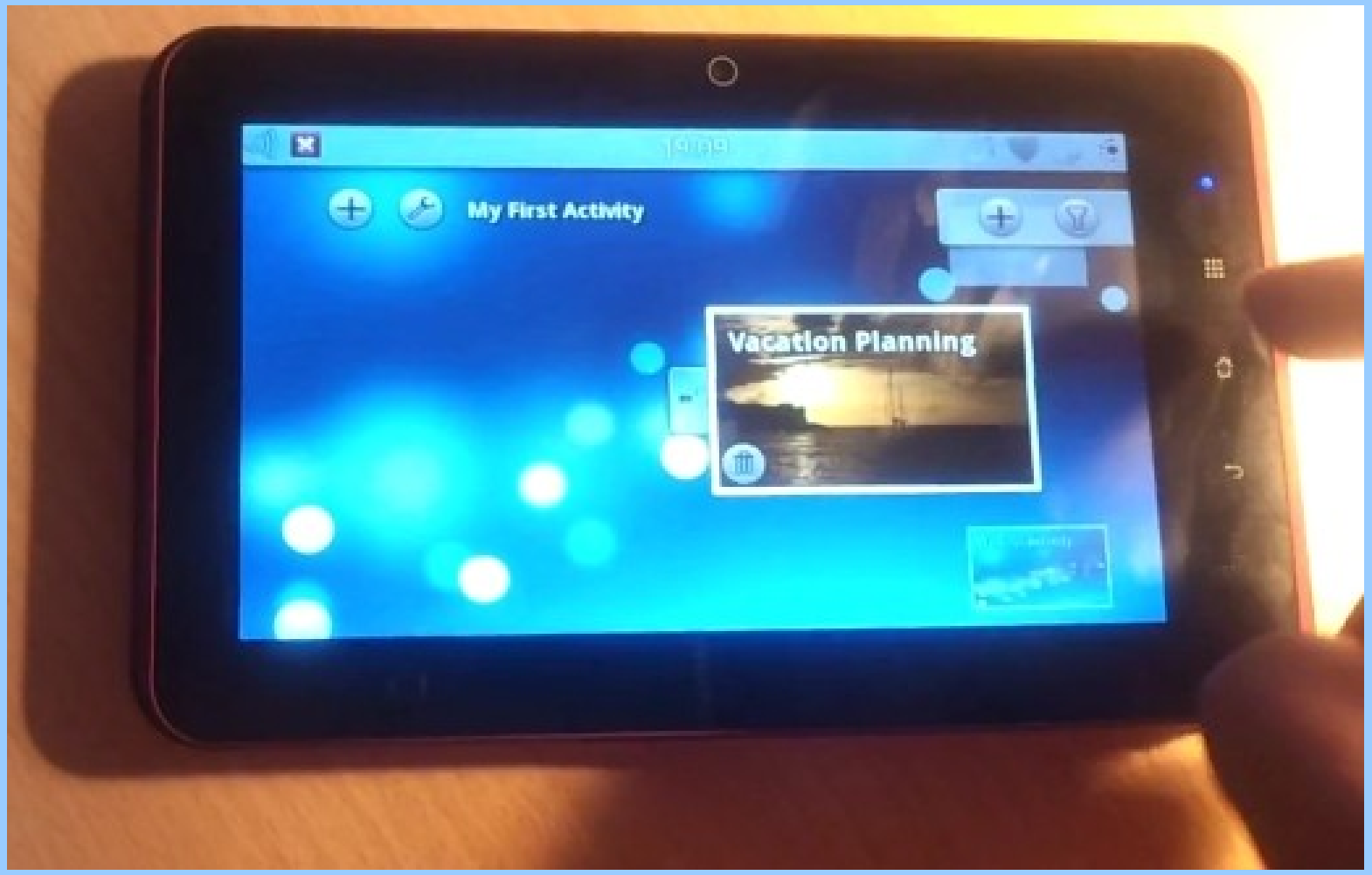
Nemo

first device: **N900,N9**
released: n**/a**

Nemo

- a mobile distribution based on Mer targetings mobile phones
  - currently running on the N900 and N9
- the GUI uses the MeeGoTouch framework and QML
- already almost usable

Plasma active



first device: ~~**Spark**~~ **Vivaldi tablet**
released: **May 2012 ?**

Plasma active

- uses Mer as its core
- UX based on Qt and various KDE technologies
- uses X Server for graphics
- first device, the Vivaldi tablet (formerly called Spark) is already available for pre-order
  - hardware similar to the Zenithink C71 Android tablet
  - 1 GHz Cortex-A9 processor, Mali 400 graphics, 512MB of RAM, 4GB of storage, 800 x 480 pixel display, 2 USB ports and an HDMI connector

OpenMoko Linux

first device: **Neo FreeRunner**
released: **June 2008**

OpenMoko Linux

- the original software distribution for the Neo FreeRunner
- based on the Ångström distribution
- developed between 2007 and 2009
  - last version released in June 2009
  - further FreeRunner OS development was taken over by the community (SHR, Qt Moko, …)
- used the X Server and a mix of various GUI libraries (EFL, GTK, Qt) during its lifetime

SHR

- SHR is originally based on the last OpenMoko software release
- the primary target is the Neo FreeRunner
  - other devices are also supported (in various states of usability)
    - Nokia N900, Palm Pre/Pre Plus/Pre 2, GTA04 and others
- SHR packages & images are built with OpenEmbedded
- its graphical environment is called Illume 2 and is based on Enlightenment Foundation Libraries (EFL)
  - other GUI toolkits (GTK, Qt (?)) are also available
  - X Server is used
- SHR recently migrated from init to SystemD

SHR

- SHR uses the opkg package management system, software packages are built using the OpenEmbedded *bitbake recipes*

Qt Moko

- a Linux distribution for the Neo FreeRunner based on Debian and Qt Extended (formerly Qtopia)
  - due to the Debian ancestry, it uses the Debian packaging system
  - the currently used Qt version is quite old (no QML support)
- active incremental development
- fully usable
- it is possible to start an X Server and run non-Qt programs
- support for the GTA04 is a work in progress

Why so many ?

- different goals
  - commercial
  - community driven
- old & stable vs new & unstable
  - old but in a usable state and deployed on many devices
  - a cutting-edge work-in-progress
- device oriented
  - device independent
  - for mobile phones
  - for tablets
- design philosophy
  - PC in the pocket
  - embedded mobile device

Openness

- Android - code open, development not
- WebOS - going open source
- Maemo 5 & MeeGo 1.2 Harmattan – partially open
  - contains large chunks of closed code

- Mer & derivates - open
- SHR & Qt Moko - open

- main problem – closed source, binary-only drivers

How to develop for all this

- native code is portable, platform-specific frameworks & languages are not
  - Dalvik only on Android, Mojo & Enyo only on WebOS
- porting libraries and CLI applications is relatively easy
- portable graphics & multimedia are more difficult
  - Qt might be a good bet due to good availability
    - Maemo 5 & 6, Android, Mer, WebOS, PC...
  - SDL has also good availability, but is more low-level
    - Maemo 5 & 6, Android, WebOS (official port), PC...

What language ?

- C/C++ is generally supported by the native SDKs
  - arcane toolchains make for an interesting experience when compiling complex projects
  - most libraries, including GUI toolkits, are written in C/C++, so interfacing with them is easy
  - platform-specific tweaks might be necessary

What language ?

- Python has a very good availability on mobile platforms
  - Maemo 5 & 6 (official port), Android, WebOS, SHR & Qt Moko,...
    - someone just needs to compile the C-based Python runtime once *vs* compiling own C/C++ code on every platform
  - there are PySide (Qt bindings), PyGame (SDL bindings) and PyGTK to choose from for GUI development
    - and others like EFL or Clutter Python bindings
  - applications can be easily tweaked on the go
  - broad standard library & many useful third-party modules
  - fast enough :)

Misc tips & tricks

- backend & GUI separation is a must
- on-device application testing is indispensable
  - a very rapid change & test-on-device cycle is possible with Python & rsync
- even when using the same GUI toolkit on multiple platforms, platform-specific tweaks might still be needed
  - both hardware & UX specific
- watch out for shell script compatibility !
- mobile GUI development is not easy
- test early, release often
- user feedback is very important !

Future outlook

- open source WebOS
  - but will there ever be any new hardware ?
- Plasma active Vivaldi tablet shipping
- a Meltemi device from Nokia shipping (?)
- a Mozilla B2G device shipping (?)
- a Tizen device (probably from Samsung) shipping (?)
- Ubuntu on mobile phones (?)
- more open source GPU drivers for mobile devices
  - closed GPU drivers are the main obstacle for community OS development & maintenance

Thank you !

Questions ? :)

Want to contact me ? :)
Martin Kolman
email**: martin.kolman@gmail.com**
jabber**: m4rtink@jabbim.cz**
github**: https://github.com/M4rtinK**