Jalimo
Unifying mobile development

Sebastian Mancke

tarent

Mobile platforms

S60

openmoko

android

maemo™

iPhone

Windows Mobile®

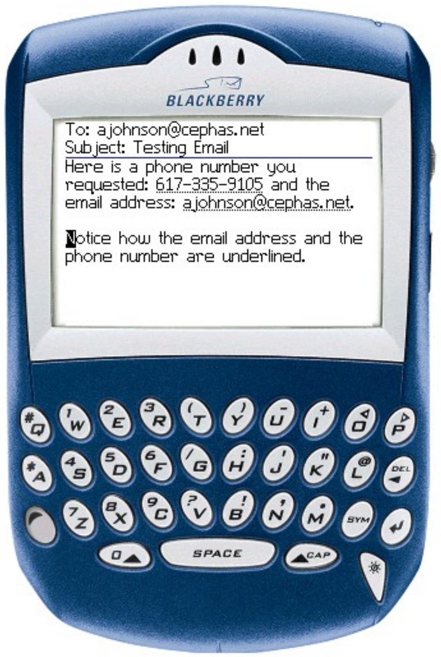
BlackBerry®

Windows Mobile

- Very Closed
- Main language: Visual C++
- Supported developing:
  - Visual Basic
  - .NET
  - ASP.NET
- Supported devices: many phones
- Company behind: Microsoft

BlackBerry

- Very Closed
- Main language: Java (J2ME) + proprietary extensions
- Supported devices: BlackBerry
- Company behind: RIM



To: ajohnson@cephas.net
Subject: Testing Email
Here is a phone number you requested: 617-335-9105 and the email address: ajohnson@cephas.net.

Notice how the email address and the phone number are underlined.

iPhone

- Very Closed
- Proprietary marketing model
- Based on BSD + iPhone Framework
- Main language: Objective C
- Unofficial gcc based SDK available
- Device: iPhone (1 model)
- Company behind: Apple

Symbian S60

- Proprietary, but open for development
- Based on Symbian
- Main language: Symbian C++
- Supported frameworks:
  - Open C (Posix porting layer)
  - Java (J2ME)
  - Python
  - Adobe Flash Lite / Web Runtime
- Supported devices: many phones
- Company behind: Nokia

Android

- Free & Open? Not known jet!
- Based on Linux + Android runtime
- GUI Toolkits: Android
- Language: Java only
- Written from scratch
- Supported devices: emulator + HTC (announced)
- Company behind: Google + Open Handset Alliance

Maemo/ITOS

- 98 % Free & Open
- Based on Linux, DBus and X11
- GUI Toolkits: GTK/hildon, QT (soon)
- Main languages: C, Python, C++
- Based on Debian (forked)
- .deb based packaging
- Supported device: n810 (1 model)
- Company behind: Nokia

OpenMoko

- 100 % Free & Open
- Based on Linux, DBus and X11
- GUI Toolkits: GTK, QT, EFL
- Main languages: C, Python
- Based on OpenEmbedded
- .ipk based packaging
- Device: Neo freerunner (1 model)
- Company behind: FIC/OpenMoko

Biggest problems in mobile development

S60

openmoko

ANDROID

maemo™

iPhone

Windows Mobile®

BlackBerry®

Problem 1: Too many restrictions

- Only small control over the system
- Often only limited APIs are available
- Features are locked, signing processes are forced
- The core components are not replaceable

Free platforms change this:

openmoko maemo™ android

JALIMO

Problem 2: Too many platforms

- Large number of different platforms
- Few standards for cross platform development
  - J2ME
  - HTML/Web Applications
- J2ME often relies on proprietary extensions
- Some platforms cover only one device

solution:

- Cross platform development standards
- Or: Focus on widespreaded platforms

Problem 3: Different development approach

- Experienced developers want to reuse their knowledge
- Companies don't want to hire additional staff for mobile development
- Development Environments should be the same in mobile and desktop development
- Applications/frameworks should be reused
- Multi tier applications should use a homogeneous software stack if possible

Solved by:

openmoko maemo™ Windows Mobile

JALIMO

Problem 3: Different development approach

- Experienced developers want to reuse their knowledge
- Companies don't want to hire additional staff for mobile development
- Development Environments should be the same in mobile and desktop development
- Applications/frameworks should be reused
- Multi tier applications should use a homogeneous software stack if possible

Solved by:

For a small group of target developers, only!

What's wrong with J2ME?

Targets much of the problems, but ..

- Has too much restrictions
- Differs to much from usual Java
  - Completely different APIs
  - No code reuse
- Even if it is standardized:
  - Every manufacturer has different implementations
  - Applications have to be device specific
- Not powerful enough for much application types

What's about Android?

Targets much of the problems ...

- has only few restrictions
- uses real Java
- promises wide availability

... but ...

- still differs from usual Java
  - special APIs
  - code reuse only below the GUI
- is not standardized
  - is not designed for integration in other platforms
  - does not integrate other approaches

What's about Maemo & OpenMoko?

They do a lot of things right ...

- eliminate restrictions
- use real desktop toolkits

... but ...

- each platform has only one device
- no API standardization (between those platforms)
- focus only on a very special developer community

What's about QT?

QT has the chance to become the solution!

- Good, powerful Toolkit
- Wide availability
  - Windows, Windows Mobile
  - Linux, OpenMoko
- Good Java bindings: Jambi

Acquired by Nokia

- Soon supported on Symbian and Maemo

What is jalimo?

- Project to bring free full Java to mobile and embedded platforms.
- Composition and maintainment of a complete solution stack for its target platforms.
- Support for the development lifecycle to target mobile devices.
- Current targets: maemo, OpenMoko

What is jalimo not:

- No additional mobile platform!
- No additional JVM!

Why we are doing jalimo?

- Small devices have special Java requirements
- Integration in mobile windowing systems
- Most free linux mobile distributions are not aware of Java
- Java developers need special support when they target Linux mobile devices
- Java developers need a maintained platform to rely on – especially commercial ones

Why we are doing jalimo?

- Tarent has employed 56 people (~40 Java developer)
- Most of our projects use java on the server side

We want:

- One technology among our applications!
- Use the same staff for mobile and server side development!

Why we are doing jalimo?

FREEDOM

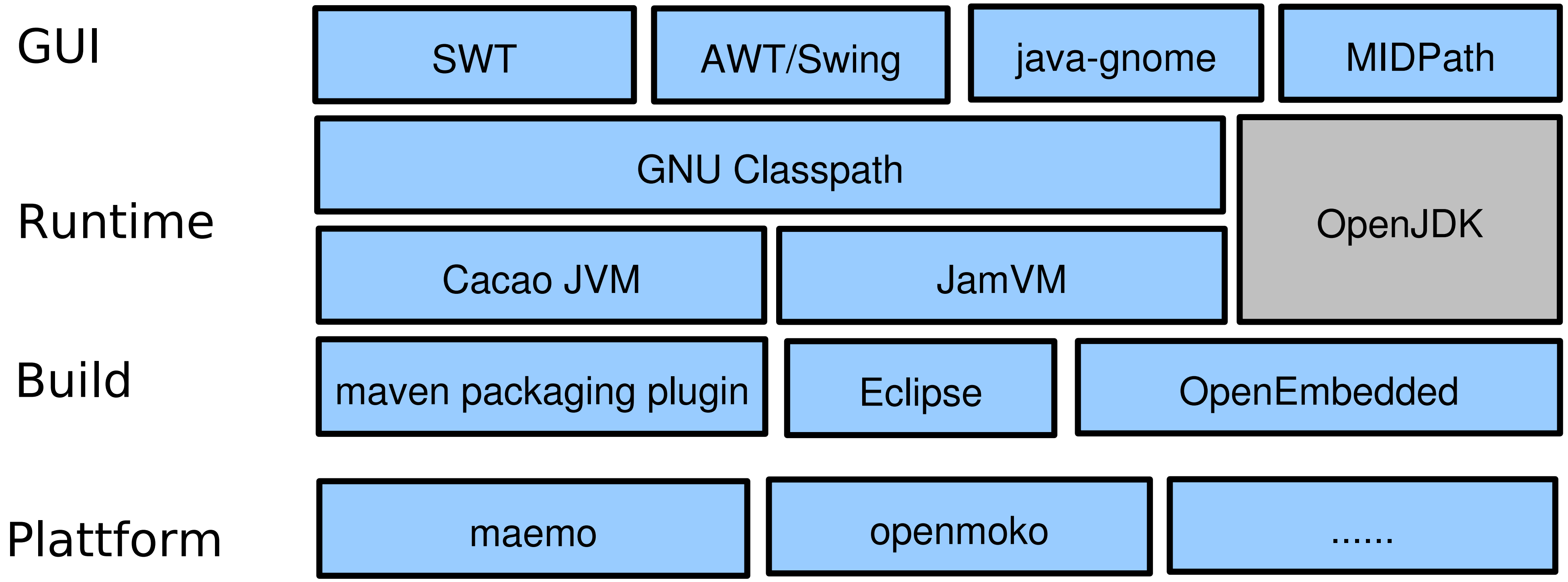
and

FUN!

Components we use

- Mostly J2SE 1.5 focused
- Different alternatives for different requirements
- The goal is to add more alternatives
- Focus: CacaoJVM, GNU Classpath, Eclipse SWT
- Also: jamvm, swing, java-gnome, midpath
- Also: Mysaifu JVM + eSWT on Windows Mobile
- Additional libraries: java-dbus, scio (more coming)
- Toolchain support: OpenEmbedded, maven-pkg-plugin

Parts


GUI
SWT
AWT/Swing
java-gnome
MIDPath
Runtime
GNU Classpath
OpenJDK
Cacao JVM
JamVM
Build
maven packaging plugin
Eclipse
OpenEmbedded
Plattform
maemo
openmoko
......

So, what is possible?

- Run nearly all Java 1.5 applications on maemo and OpenMoko
- Choose between SWT, Swing or java-gnome GUI
- Run J2ME applications
- Integrate into the target system, using dbus
- Consume web services using scio
- Package your application as .deb or .ipk using maven
- Porting our stack to every Linux Embedded platform, very fast

OpenEmbedded Toolchain

- OpenEmbedded infrastructure
  - self-hosting toolchain
  - builds jalimo packages for arbitrary distributions and hardware. From a single source!
- OpenEmbedded build recipes
  - „swt“, „dbus-java“, ...
  - in OpenEmbedded upstream
  - and Jalimo svn-overlay
- Repositories
  - public repositories for maemo and OpenMoko
  - Integration in OpenMoko, Angstroem, ...

maven packaging plugin

- Maven2 is currently the mainstream build tool for Java
- Maven allows dependency definition at artifact (.jar-file) level
- Packages Maven2 projects for specific distributions
- On the fly dependency translation
  - maven dependencies => platform dependencies
- Supported Platforms
  - Maemo Chinook (.deb)
  - OpenMoko (.ipk)
  - Debian (.deb)

What we are missing most?

- A VM which is as fast as cacao with the startup of jamvm
- Debugging support (JVMTI)
- Really fast Swing implementation for OpenMoko
- More free phones!

What we are missing most?

OUTDATED

- A VM which is as fast as cacao with the startup of jamvm
- Debugging support (JVMTI)
- Really fast Swing implementation for OpenMoko
- More free phones!

JIT-Cache

- Robert Schuster implemented caching of native code for the CacaoJVM (will be released soon)
- Allows fast startup of applications
- Integration into the build process using Qemu: precompiled binaries

Small goals

- More library code and integration
- Integrate SUN's Swing implementation
- Attract more developers
- Package the complete Eclipse RCP/eRCP
- Support QT-Jambi
- More applications

Big goals (whishes)

- Port Android APIs to other platforms
- SWT implementations for Android, iPhone
- Get additional VMs e.g. for Symbian S60

Resources

jalimo.org

- Documentation of how to install binary packages for Maemo & OpenMoko
- Simple development & packaging tutorial

mvn-pkg-plugin.evolvis.org:

- Project site and documentation (examples!) for packaging plugin

Thank you!