Mobile Development Environments

David Cheeseman

Outline

- Motivation for Investigating Mobile Development Environments
- Introduction to Mobile Environment
- Choices in Platform Design
- Detailed Description of Select Mobile Environments
  - OS Overview
  - Hardware Overview
  - Marketplace Environment
  - Development Freedom/Limitations
  - Personal Input
- Who I'm siding with and why.
- Comments, Questions

Motivational Interest

Initial Interest

- Tracker Project Research Platform (Systems Lab)
- Initially Symbian, transitioned to Android.
- Project Now Called DroidScanne

Continued Interest

- Mobile Productivity Apps
  - Scheduled Muting
  - Environment Aware Applications
- Mobile Gaming
  - Performance on Low-Power Devices
  - Environment Aware Games (GPS, AR Games)
  - Cross Platform Games (iPhone, Android, & More)

Introduction

Smartphones are quickly becoming ubiquitous and affordable.

- Ubiquity was predicted way back in the days of Java ME.
  - Abowd, G. D. and Mynatt - Charting past, present, and future research in ubiquitous computing.

Moore's law also applies to price, not just power.

- Battery life limits major performance increases.
- As phones powerful enough to run major OS's permeate the market, supply will drive down price.

Introduction Cont.

Side-Effects of Ubiquity?

- Different marketing and usage philosophies develop.
  - Hardware and Software Standards
  - Business vs. Pleasure Usage
  - Gadget vs. Complete Platform
  - Etc.
- Developers must choose 1 or more sides.
- Several philosophies conflict.

Introduction Cont.

Current Platforms (* - denotes ones I will cover)

- Java Enabled Phones
- Windows Mobile
- Blackberry
- Symbian
- iPhone OS*
- Palm WebOS
- Android*
- Windows Phone 7 Series*

Choices in Platform Design

- Restricted vs. Unrestricted Hardware
- Closed vs. Open Source
- Applications Model
- App Distribution Model

Restricted vs. Open Hardware

Restricted Hardware

Hardware is limited to a single manufacturer or hard requirements on hardware to run the platform are put in place.

- Pros:
  - Minimal Performance Level Ensured
  - Predictable Sensor/Input Access
- Cons:
  - No or Few 3rd Party Phones and/or Accessories
  - New sensors/features not always available until a change in the standard.
  - Early adopters can feel the pain on a change in the hardware standard.

Restricted vs Open Hardware

Open Hardware

No hard limits are placed on the hardware. OS is made available to any manufacturer.

- Pros:
  - Great variety in hardware and accessories!
- Cons:
  - No guaranteed access to sensors/inputs.
  - No guaranteed performance level.
  - OS should accommodate developers to detect above shortcomings of hardware.

Closed vs. Open Source

Open Source (Free and Open Source)

- Pros
  - Makes OS available without barrier to all developers/hardware manufacturers.
  - Improves security/performance through transparency.
- Cons:
  - Hard to control installable apps.
    - e.g. Rooting an Android device and installing unsigned packages.
  - People can branch the OS to suit their needs
    - e.g. AT&T branched Android to only allow apps it approves in the Android Marketplace. They're still rootable though.
    - e.g. Alternative interface designs and conventions. (HTC unlocking multi-touch)

Closed vs. Open Source

Closed Source

- Pros
  - Your performance/security tweaks stay secret.
  - Can license your OS to device manufacturers to make money.
  - Can license your API's to make even more money.
- Cons
  - No one besides you can comment on your performance/security tweaks.
  - No one besides you can catch performance/security holes.
- Counter Pro
  - You can use BSD or similar licensed open source code and still keep your source closed.

Application Model

Single or Multi-Process?

- Single Process
  - Ensures maximum usage of system resources for consistent behavior of an app.
  - No convention necessary for memory and resource management.
- Multi-Process
  - Allows for backgrounds services (chat clients, syncing services, background music, etc)
  - Allows App-to-App interaction (though I can't think of a usage case)..
  - Must choose convention on how to manage memory, processes, and system resources.

App Distribution Model

Marketplace Models

- Default Marketplaces
  - Allows control over available apps.
  - Arguably a requirement of today's mobile marketplace.
- 3rd Party Marketplaces
  - Allows for diverse marketplace standards.
  - e.g. SlideME Android Marketplace & Cydia for iPhone*
- Single Party Distribution
  - Allow individual developers self-publish their apps.

iPhone Platform

- Arguably the first general user oriented smartphone.
- 42+ million units sold to date.
- New mobile development paradigms created on the iPhone.
  - Single developer, millionaire returns.
  - Context aware applications and games.
- New legal issues.
  - Lawsuits against Palm and HTC for mobile UI patents.

Releases

- Original - June 2007
- 3G - July 2008
- 3GS - June 2009

Other Hardware:

- iTouch
- iPad

iPhone Platform

OS Overview

- Single Process Application Model
  - Prevents usable chat clients and background music.
  - Multi-Process Rumored for 4.0.
- Push Services for Application Notification
- Programmed in Objective C
- No Flash Support

Hardware Overview

- Very Restricted Hardware Deployment
  - iPhone, iTouch, and iPad only.
- Huge Market Presence
- Hardware/Pricing Updates Burned Many Early Adopters
- Consumer Hardware = Development Hardware

iPhone Platform

Marketplace Overview

- No 3rd Party Marketplaces.
- $99 to Access SDK/Publish Apps.
- Very profitable marketplace.
- Apps can and have been removed even after they've been approved.

Personal Input

- Apple's ability to pull apps seems is bad without a 3rd party alternative marketplace.
- Look, feel, and *usability of the iPhone is very good.
- Can't be ignored due to its pervasiveness in the market.
- The Apple 'cult' provides market stability.

Android Platform

- Linux Based Mobile OS

- 2003 - Android Inc. founded and starts Android OS.
- August 2005 - Google buys Android.
- November 2007 - Release announced in by the Open Handset Alliance.
- October 2008 - First Android phone released and entire Android source opened under the Apache Licence (Free and Open Source).

Android Platform

OS Overview

- Apps Programmed in Java
- Multi-Process
- No Multi-Touch (US Only, Apple's Fault?)
- Designed for Battery/Resource Efficiency
  - Every window is its own process or 'activity'.
  - OS removes processes from memory and saves its state for late resumption.
  - Background services can be suspended similarly.
  - OS allows apps to keep themselves from being paged out and the disabling of power saving conventions

Android Platform

Hardware

- Development Hardware
  - Buy a dev phone, root a consumer phone or use the emulation environment.
- Consumer Hardware (no real limit)
  - Huge selection of phones.
  - Netbooks and internet tablets.
  - iTouch/Zune HD competitive touch devices?

Marketplace Overview

- 3rd Party Marketplaces Allowed!
- Unsigned Installs Allowed (grassroots distribution).
- Most apps are free (only games seem to sell).
- No cost to publish apps.

Android Platform

Personal Input

- Perfect for mobile phone sensor network research.
- Development is out-pacing documentation.
  - e.g. No example documentation from Google on how to use Multicast networking in 1.6.
- Google might not officially enable multi-touch in US until Apple's done suing Palm and HTC.
  - Requires that I root a phone to develop multi-touch.
  - Apple could sue me for distributing a multi-touch apps.

Windows Phone 7 Series

- Very recent development.
- Announced at CES 2010 with Demo Hardware
- Is NOT Windows Mobile but an expansion of the Zune OS concept.
- Developer Information Announced at MIX 2010
- First phones slated for release this holiday season.
- App market already being built via an emulation environment.

Windows Phone 7 Series

OS Overview

- Apps Programmed in .Net
  - XNA or Silverlight API's Only
  - Designed to provide cross platform support with Xbox, Windows 7, Zune, and WP7.
- Single-Process Application Model
- UI divided into hubs.
  - Peope, Music & Videos, Marketplace, etc.
- Xbox Live, Office, Zune, and Netflix integration.

Windows Phone 7 Series

Hardware Overview

- Hard minimum requirements for licensing WP7.
  - 1GHz Minimum CPU!
  - Dedicated GPU required.
  - Multi-touch resolution minimum set.
  - Two standard screen resolutions.
  - Minimum Button requirements.
  - Camera and Accelerometers
- Very Few Customizable Hardware Options
  - Slide-Out Keyboard
  - Extra Buttons
  - Arguably will make it hard to distinguish between manufacturers.
- Phones are developer-unlocked remotely.

Windows Phone 7 Series

Marketplace Overview

- 3rd party marketplaces not allowed.
- SDK is free, but $99 to register as an *individual developer.
- XNA enables Xbox and PC Games to be playable on the phone.

Personal Input

- Gaming/Multimedia potential is HUGE!
- Won't compete with the general app marketplaces of Android and iPhone, mostly for gamers.
- NO COPY PASTE!?!

Picking Sides

Productivity and Web Applications

- iPhone
  - Huge market share, profitable app store, and they'll probably enable multi-process to compete with Android.

Research

- Android
  - Can create your own distribution for specialty sensors and persistent background processes are supported.

Games

- WP7 and/or All Platforms
  - WP7 is great combined with the Xbox Live marketplace, but cross-mobile-platform games have no marketplace limits.

Questions?
Comment?