1 Mobile and Ubiquitous User Interfaces

2.1 Mobile Computing

2.2 Input and Output on Mobile Devices

2.3 Design Guidelines for Mobile Devices

2.4 System Architectures for Mobile Devices

2.5 Example Applications

2.6 HCI and Ubiquitous Computing

Literature:

- Scott Weiss: Handheld Usability, Wiley 2002
- Jonathan B. Knudsen: Wireless Java - Developing with J2ME. Second Edition, Apress 2003. (Chapter 5 available online)
- Jonathan B. Knudsen, Sing Li. Beginning J2ME: From Novice to Professional. 3rd Edition, Computer Bookshops 2005
- http://code.google.com/android/

Developing Applications for Mobile Devices

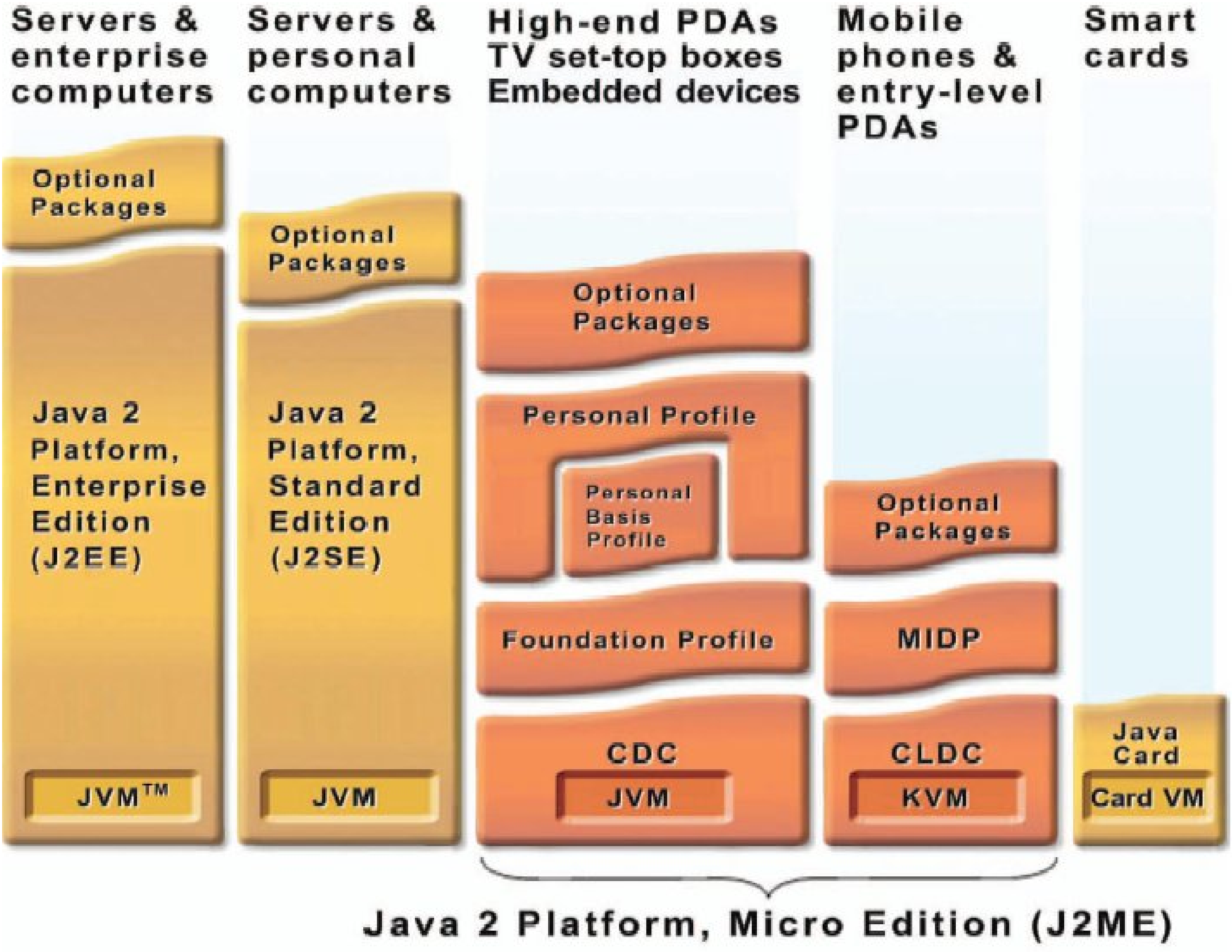
- Devices: Basic Phone, Extended Phone, Smartphone, PDA, Notebook
- Platforms (Mobile Phone, Smartphone)
  - Platform specific:
    - Symbian OS (C++, OPL)
    - Palm OS (C++)
    - Pocket PC
    - Vendor-specific
  - Platform independent: J2ME (Java 2 Platform, Micro Edition)
    - Supported by Motorola, Nokia, Panasonic, Samsung, Sharp, SonyEricsson, Toshiba, etc.
  - Android (Google, Open Handset Alliance)
    - technically Java, but not called Java for legal reasons
    - HTC, NTT DoCoMo, LG Electronics, Sprint, Motorola, T-Mobile, Samsung; eBay, Intel, Nvidia etc.

Acknowledgement for slides: Enrico Rukzio

Java on mobile devices: History

- 1990: Java started as an internal project at Sun Microsystems
- 1995: Initial release of JDK 1.0 (applets → servlets)
- 1999: JavaOne conference
  - Subdivision of Java in
    - » Java 2 Enterprise Edition (J2EE)
    - » Java 2 Standard Edition (J2SE)
    - » Java 2 Micro Edition (J2ME)
      (successor of Personal Java and Embedded Java)
- 2000/01 First mobile phones with support for J2ME

The Java universe

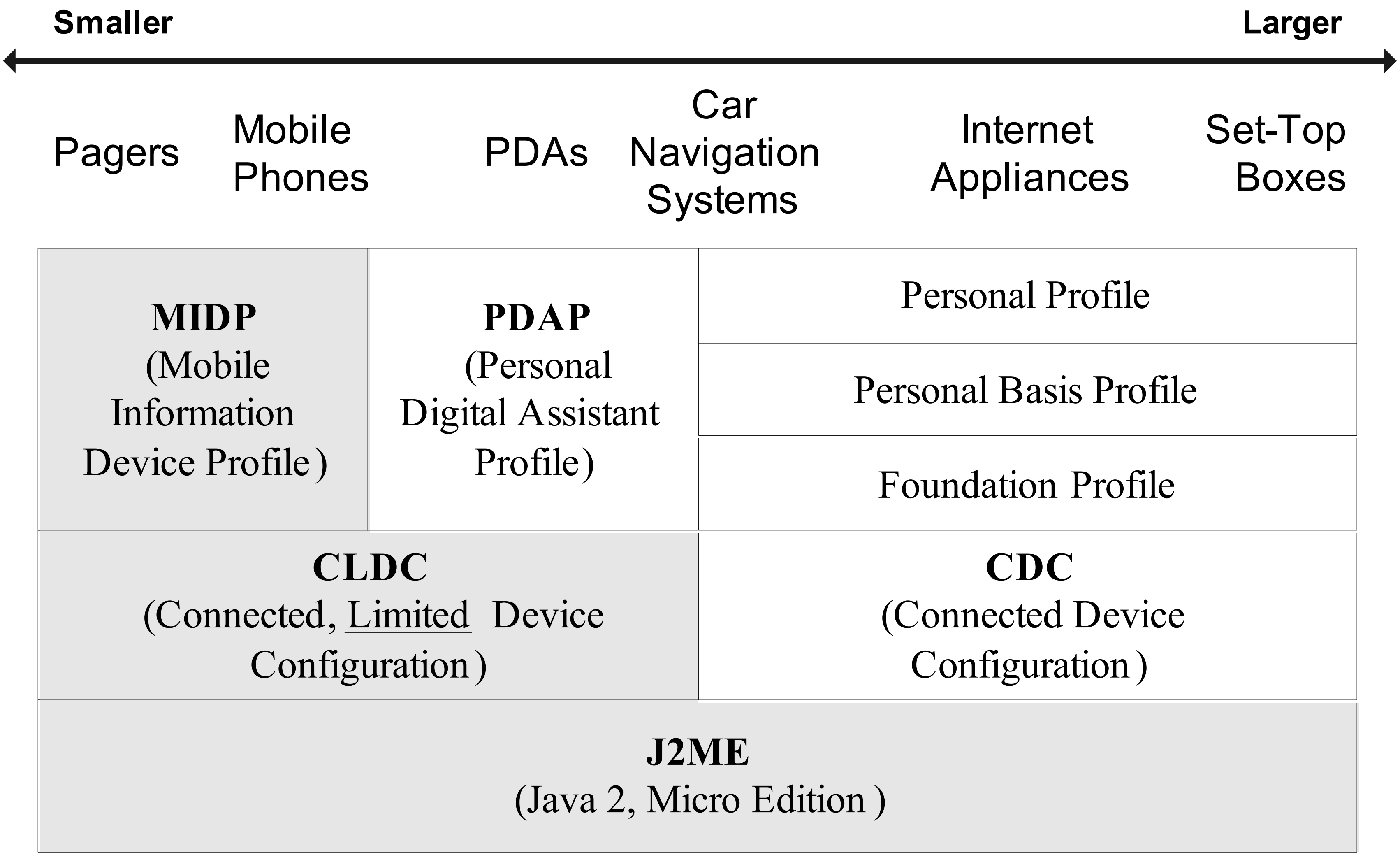

Servers & enterprise computers
Servers & personal computers
High-end PDAs TV set-top boxes Embedded devices
Mobile phones & entry-level PDAs
Smart cards
Optional Packages
Optional Packages
Optional Packages
Optional Packages
Java 2 Platform, Enterprise Edition (J2EE)
Java 2 Platform, Standard Edition (J2SE)
Personal Profile
Personal Basis Profile
Foundation Profile
MIDP
CDC
CLDC
Java Card
JVM™
JVM
JVM
KVM
Card VM
Java 2 Platform, Micro Edition (J2ME)

KVM = Kilobyte VM

J2ME: Basics

- J2ME: Java 2 Platform, Micro Edition
  - “Java for small devices”
  - 2005: 700 million mobile devices support J2ME
    - » More than number of desktop PCs supporting Java
- Stack
  - Configuration + profile + optional APIs
- Configuration: Specific kind of device
  - Specifies a Java Virtual Machine (JVM)
  - Subset of J2SE (Standard Edition)
  - Additional APIs
- Profile: more specific than configuration
  - Based on a configuration
  - Adds APIs for user interface, persistent storage, etc.
- Optional APIs:
  - Additional functionality (Bluetooth, Multimedia, Mobile 3D, etc.)

The J2ME Universe



Smaller

Larger

Pagers

Mobile Phones

PDAs

Car Navigation Systems

Internet Appliances

Set-Top Boxes

MIDP (Mobile Information Device Profile)

PDAP (Personal Digital Assistant Profile)

Personal Profile

Personal Basis Profile

Foundation Profile

CLDC (Connected, Limited Device Configuration)

CDC (Connected Device Configuration)

J2ME (Java 2, Micro Edition)

J2ME: CLDC

- *Connected, Limited Device Configuration* (JSR 139)
- For small devices (e.g. mobile phone, pager, PDA) with small screen size, limited memory, slow network connection
- For devices with 160 to 512KB (according to the specification) of memory for Java Platform
- JVM: KVM (“Kilobyte Virtual Machine”)
  - Not a full standard bytecode verifier
  - Adding native methods not allowed → not possible to access platform-specific functionality
- CLDC 1.0 / CLDC 1.1. (Floating point data types)

J2ME: MIDP 2.0

- MIDP 2.0 (JSR 118, based on CLDC)
  - MIDP 3.0 under development (JSR 271)
- *Mobile Information Device Profile* for mobile phones and pagers
- Device characteristics (according to the specification):
  - Min. 128KB RAM (Java Runtime Heap)
  - 8KB for persistent data
  - Screen: > 94*54 pixel
  - Input capacity, Network connection
- Advantages:
  - WORA (Write Once, Run Anywhere)
  - Security (Sandbox KVM)

J2ME: APIs in CLDC 1.1 + MIDP 2.0

MIDP 2.0

javax.microedition .lcdui
javax.microedition .lcdui.game
javax.microedition .media
javax.microedition .media.control
javax.microedition .midlet
javax.microedition .pki
javax.microedition .rms

CLDC 1.1

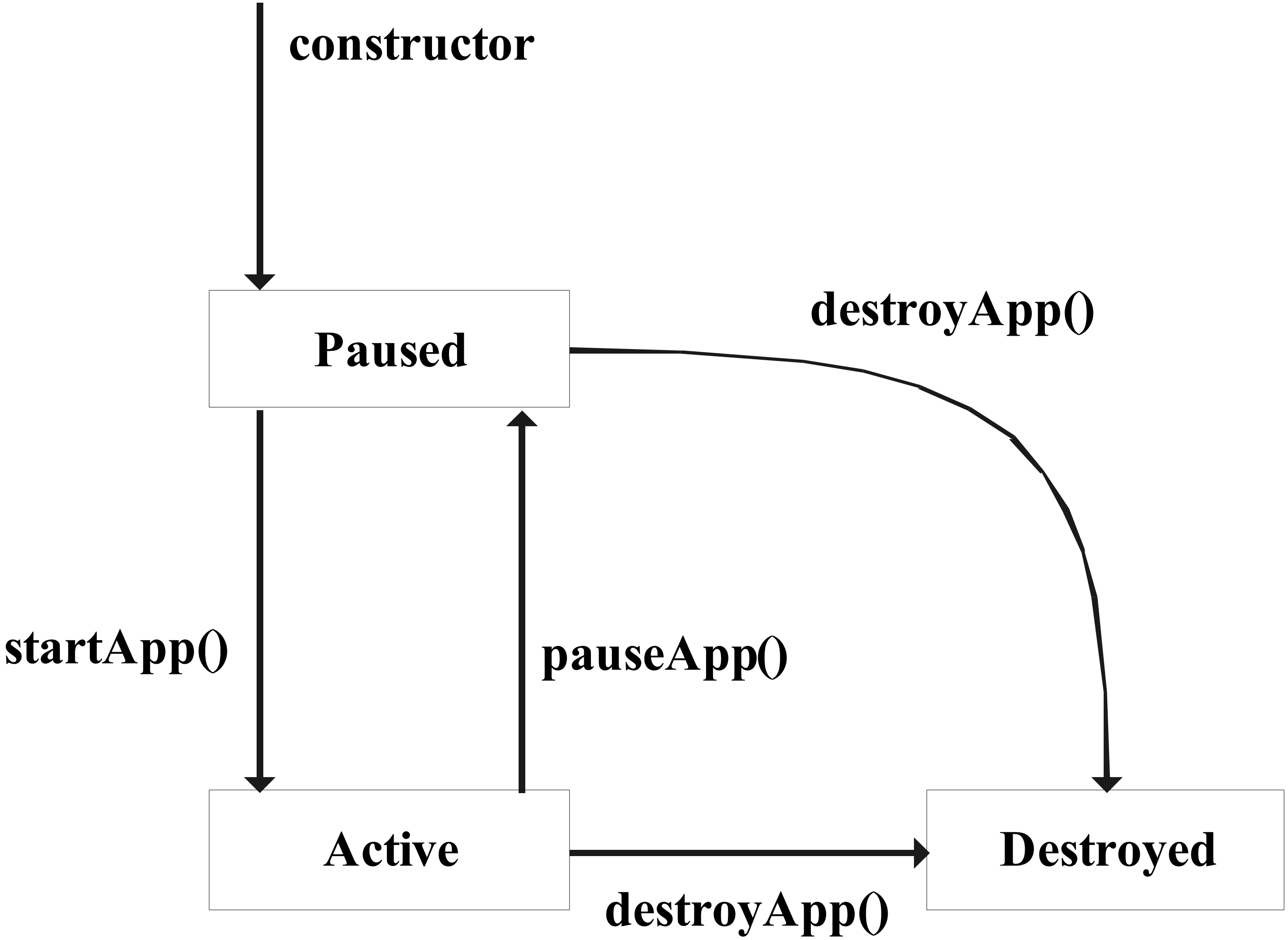
java.lang
java.lang.ref
java.io
java.util
java.microedition .io

APIs are restricted when compared with J2SE

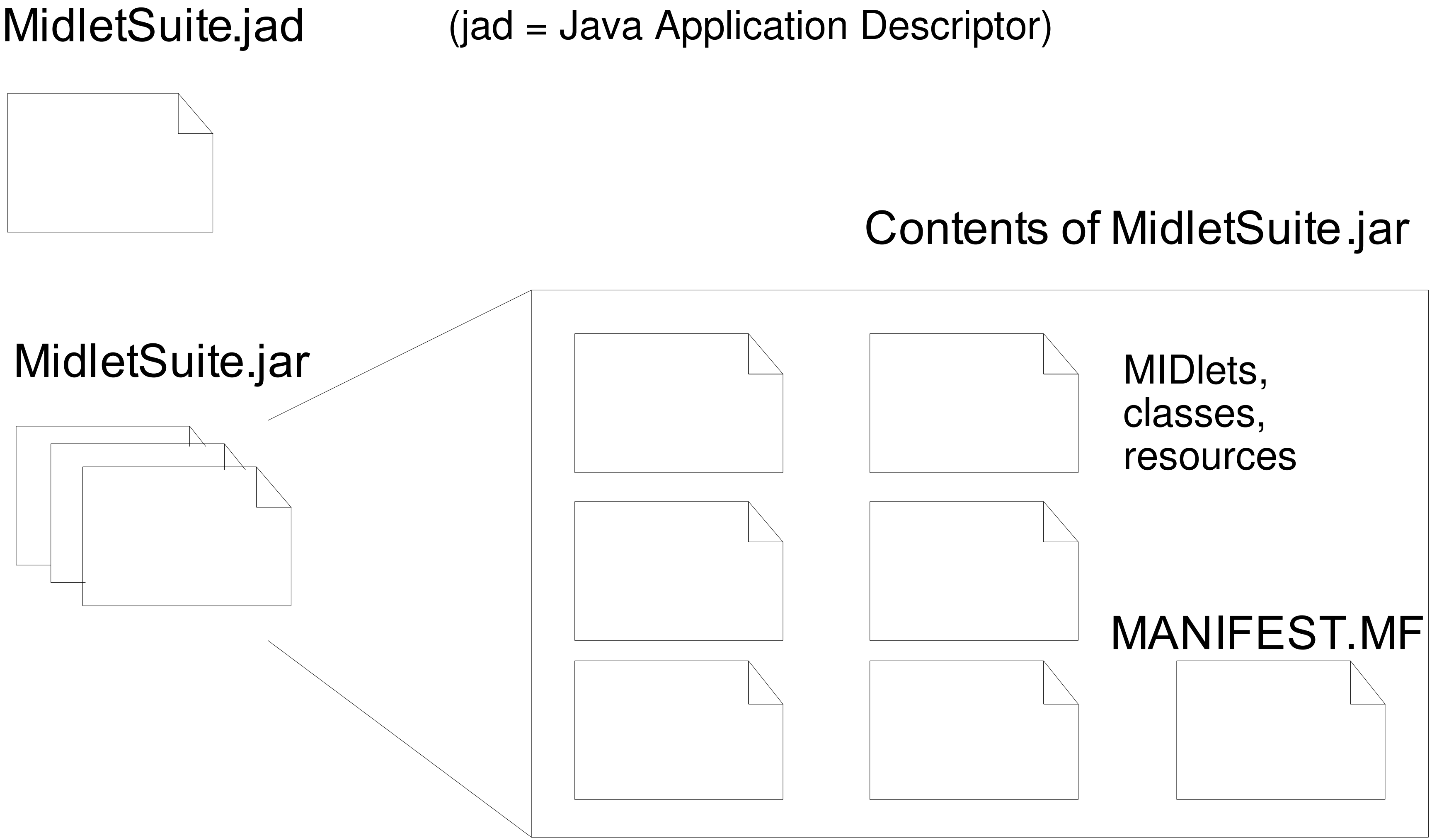
MIDlet

- MIDP applications are called MIDlets
  - Several MIDlets can be combined into *MIDlet suite*
- Every MIDlet is instance of javax.microedition.midlet.MIDlet
  - No argument constructor
  - Implements lifecycle methods
- Conceptually similar to Applets
  - Can be downloaded
  - Executed in host environment

MIDlet (MIDP Application): Life Cycle



Anatomy of a MIDlet Suite



MidletSuite.jad

(jad = Java Application Descriptor)

Contents of MidletSuite.jar

MidletSuite.jar

MIDlets,
classes,
resources

MANIFEST.MF

MIDP: User Interface

- Goal: Write Once, Run Anywhere
- Anywhere?
  - different screen sizes
  - resolution of screen
  - color or grayscale screen
  - different input capabilities
    - » numeric keypad
    - » alphabetical keyboards
    - » soft keys
    - » touch screens, etc.

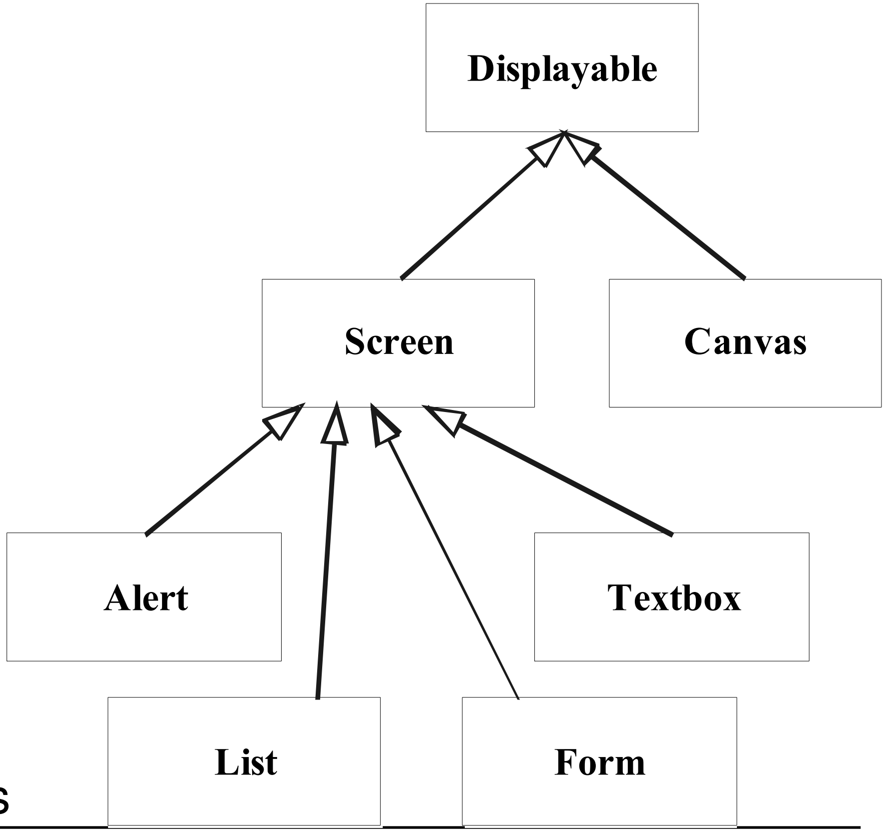
MIDP User Interface: Methodology

- Abstraction (→ Preferred Method)
  - Specifying a user interface in abstract terms
  - (Not:) “Display the word ‘Next’ on the screen above the soft button.”
  - Rather: “Give me a Next command somewhere in this interface”
- Discovery (→ Games)
  - Application learns about the device + tailors the user interface programmatically
  - Screen size → Scaling

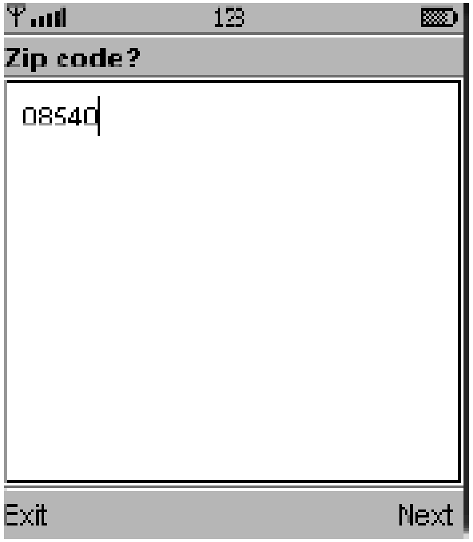
http://developers.sun.com/mobility/midp/chapters/j2meknudsen2/Chap5.pdf

MIDP User Interface: View from the Top

- User-interface classes **`javax.microedition.lcdui`**
- Device display represented by instance of *Display* class
  - Factory method: **`getDisplay()`**
  - Keeps track of what is shown (**`Displayable`** instances)
- Analogy: Easel (**`Display`**) and canvas (**`Displayable`**)
- **`Canvas`***:* Discovery method
  - Fine control
  - Special cases
    - For games: *GameCanvas*
- **`Screen`***:* Abstraction method
  - Standard user interface elements



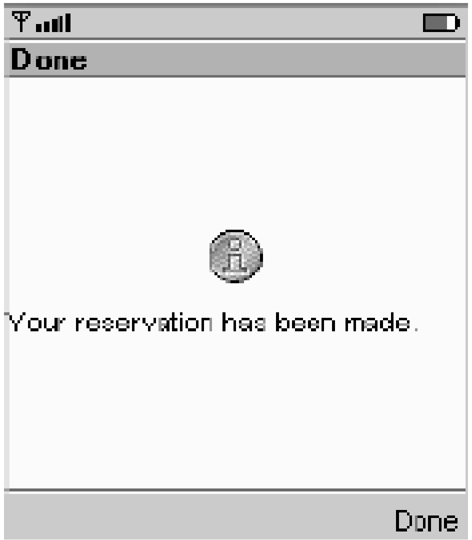
MIDP User Interface: Subclasses of *Screen*



Zip code?

08540

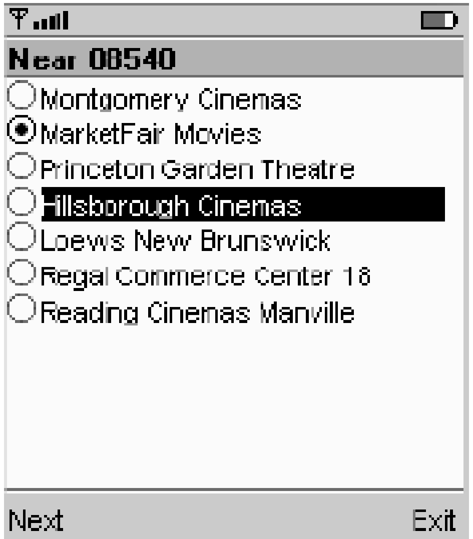
TextBox



Done

Your reservation has been made.

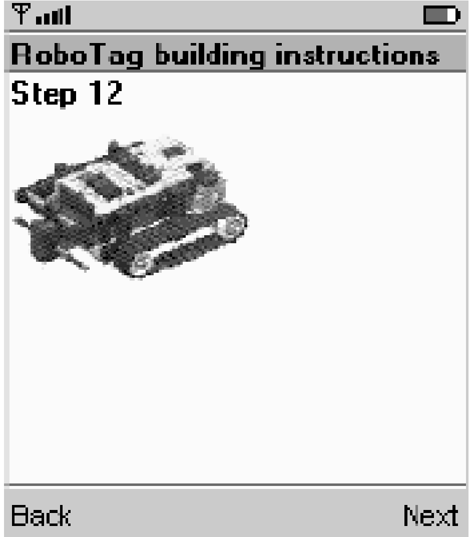
Alert



Near 08540

- ○ Montgomery Cinemas
- ● MarketFair Movies
- ○ Princeton Garden Theatre
- ○ Hillsborough Cinemas
- ○ Loews New Brunswick
- ○ Regal Commerce Center 18
- ○ Reading Cinemas Manville

List



RoboTag building instructions

Step 12

Form

MIDP User Interface: Making Things Visible

- To change the contents of the display:
  - Passing **`Displayable`** instances to **`Display`**'s **`setCurrent()`**
- To find out what is displayed:
  - **`getCurrent()`** (may not be shown)
  - **`isShown()`**
- Query methods for display capabilities
  - E.g. **`isColor(), numColors(), vibrate(),`** …
- Typical Sequence
  - Show a **`Displayable`**
  - Wait for input
  - Decide which **`Displayable`** should be the next one
  - Repeat

MIDP User Interface: Commands

- `Command`: Something the user can invoke
  - Similar to button
  - Programmer does not care about representation (keypad button, soft button, touch screen, …)
- Command constructor:
  - `Command(name, type, priority)`
- Every `Displayable` keeps a list of its `Commands`
  - `public void addCommand(Command cmd)`
  - `public void removeCommand(Command cmd)`
- Commonly used commands signified by “type” value:
  - `OK, CANCEL, BACK, STOP, HELP, SCREEN`
  - Examples:

```
Command c = new Command("OK", Command.OK, 0);
Command c = new Command("Launch", Command.SCREEN, 0);
```

- Responding to commands: `CommandListener`

MIDP User Interface: Simple Example

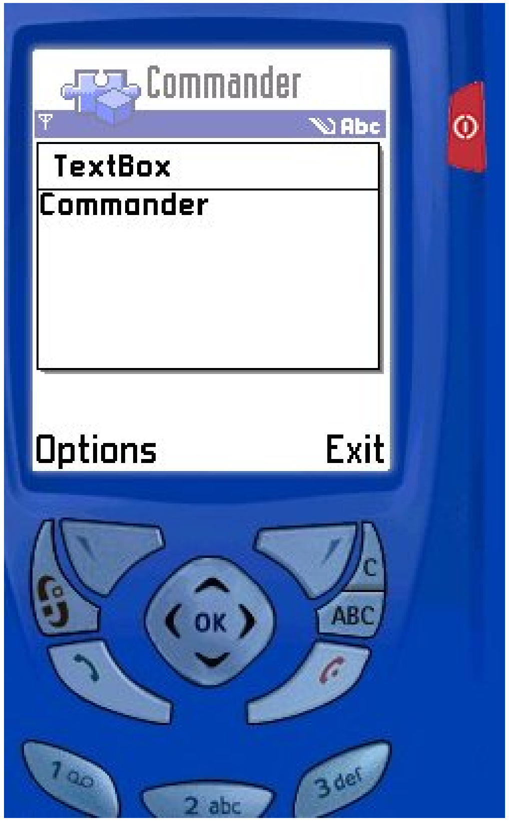
```
public class Commander extends MIDlet {

  public void startApp() {
    Displayable d =
       new TextBox("TextBox", "Commander", 20, TextField.ANY);
    Command c = new Command("Exit", Command.EXIT, 0);
    d.addCommand(c);
    d.setCommandListener(new CommandListener() {
      public void commandAction
               (Command c, Displayable s) {
        notifyDestroyed();
      }
    });

    Display.getDisplay(this).setCurrent(d);
  }

  public void pauseApp() {}

  public void destroyApp(boolean unconditional) {}
}
```

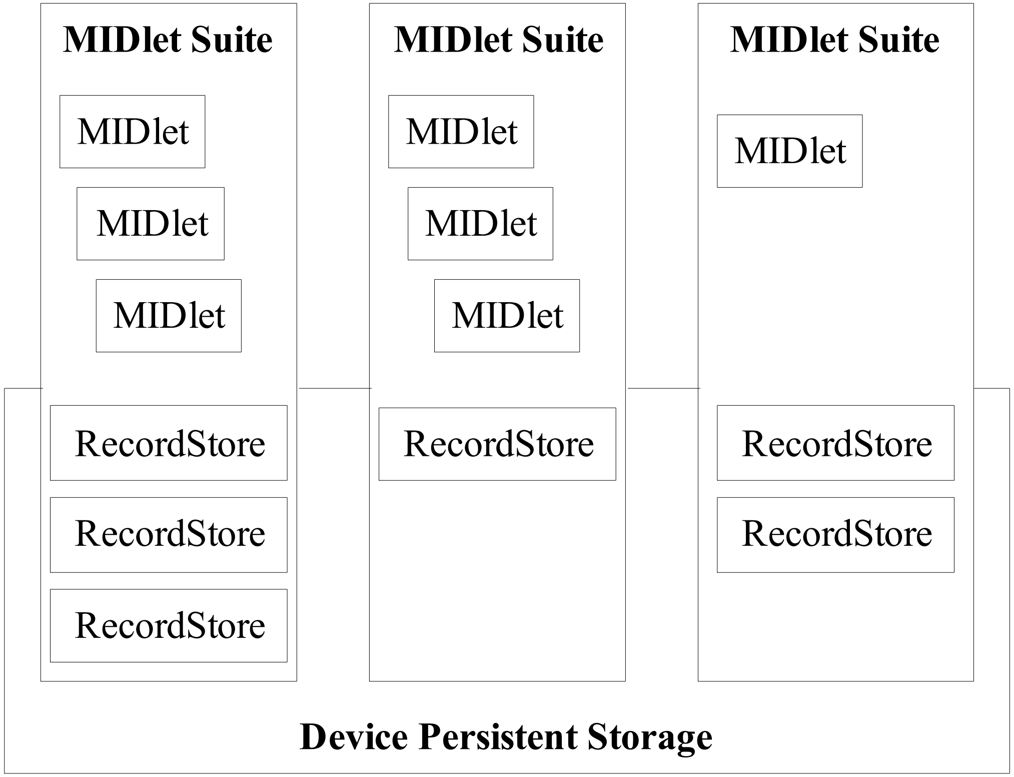


MIDP: Persistent Storage

- Goal: Write Once, Run Anywhere
- Anywhere?
  - – Device with Flash ROM
  - – Battery-backed RAM
  - – Small Hard Disk
  - → Abstraction is needed
- Record stores (small databases)
- Min. 8KByte

Persistent Storage: Records

- *Record store*
  - contains *records* (pieces of data)
  - instance of **`javax.microedition.rms.RecordStore`**
- Every MIDlet in a MIDlet Suite can access every Record Store
- Since MIDP 2.0: Access across Suite boarders possible

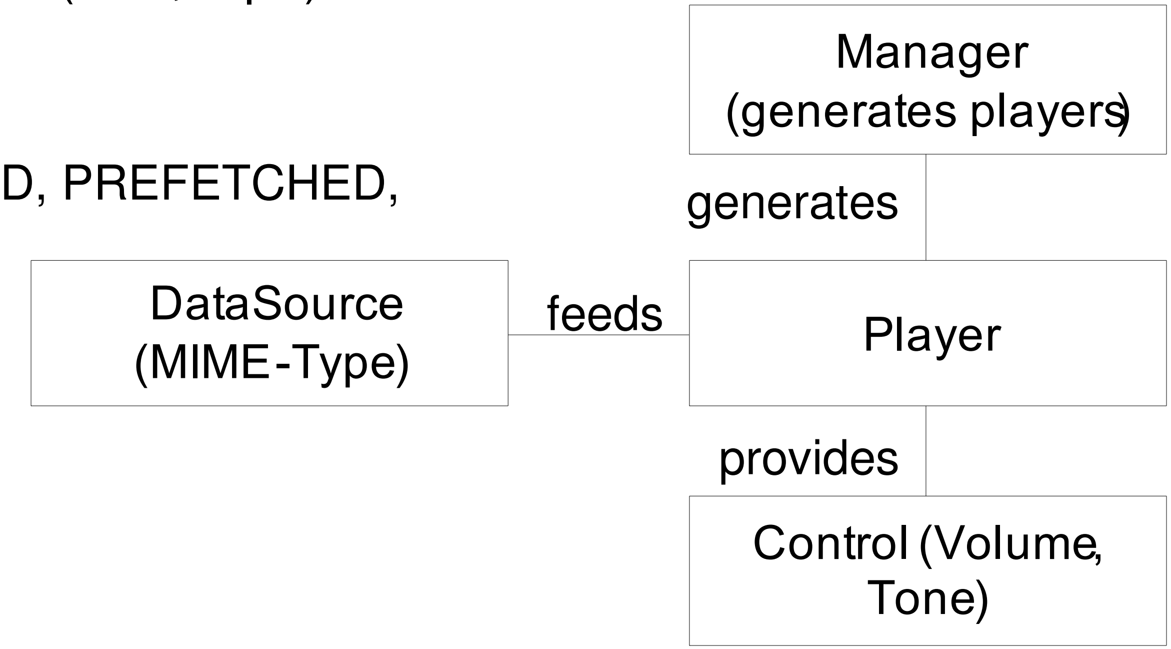


Connecting to the World

- Generic Connection Framework
- Extremely flexible API for network connections
- Contained in **`javax.microedition.io`**
- Classes based on **`Connection`** interface
  - **`HttpConnection`** (Get / Post) / **`HttpsConnection`**
  - **`SocketConnection`**
  - **`ServerSocketConnection`** (Responding to incoming connections)
  - **`SecureConnection`** (TLS or SSL socket)
  - **`CommConnection`** (SerialPort)
  - **`DatagramConnection`** (UDP DatagramConnection)

MMAPI (Sound, Music, Video)

- Mobile Media API – similar to JMF (Java Media Framework)
- General API for multimedia rendering and recording
- ABB (Audio Building Block) – play simple tones (MIDI – note, duration, volume) and sampled audio (wav, mp3)
- Player lifecycle:
  - States
    UNREALIZED, REALIZED, PREFETCHED, STARTED, CLOSED
  - Methods
    `realize()`, **`prefetch()`**, **`start()`**, **`stop()`**, **`deallocate()`**, **`close()`**



http://developers.sun.com/mobility/midp/articles/mmapioverview/

Further APIs (Examples)

- Wireless Messaging API (JSR-120)
- Mobile Media API (JSR-135)
- Bluetooth API (JSR-82 no OBEX)
- FileConnection and PIM API (JSR-75)
- Mobile 3D Graphics API (JSR-184)
- Location API (JSR-179)
- Web Services API (JSR-172)
- Advanced Multimedia Supplements (JSR-234)

- Further APIs (not JSRs): kXML, kSOAP, Parsing of GPS data, etc.

Selected Experiences from J2ME Development for Mobile Phones

- Phones are getting more powerful quickly
- Standards are being established (e.g. Series 60), but still:
  - Big differences between the emulators and the real phone.
  - Testing of applications on the mobile phone (!!!) is very important.
- Lack of memory and processing power is still a problem.
- Debugging on the mobile phone is a big problem.
  - No meaningful error messages.

Symbian Series 60 Phones

- Symbian:
  - Operating system for mobile devices
  - Derivative of the Psion operating system EPOC
  - 32-bit multitasking OS, mostly written in C++
  - Dealing with calls and messages coming in during application runtime
- Symbian Series 60 Phones
  - Smartphone standard platform
  - LG, Lenovo, Nokia, Panasonic, Samsung, …
- Software development for Series 60 phones, examples of languages:
  - OPL (similar to BASIC)
  - Visual Basic
  - Java
  - C++
  - Python

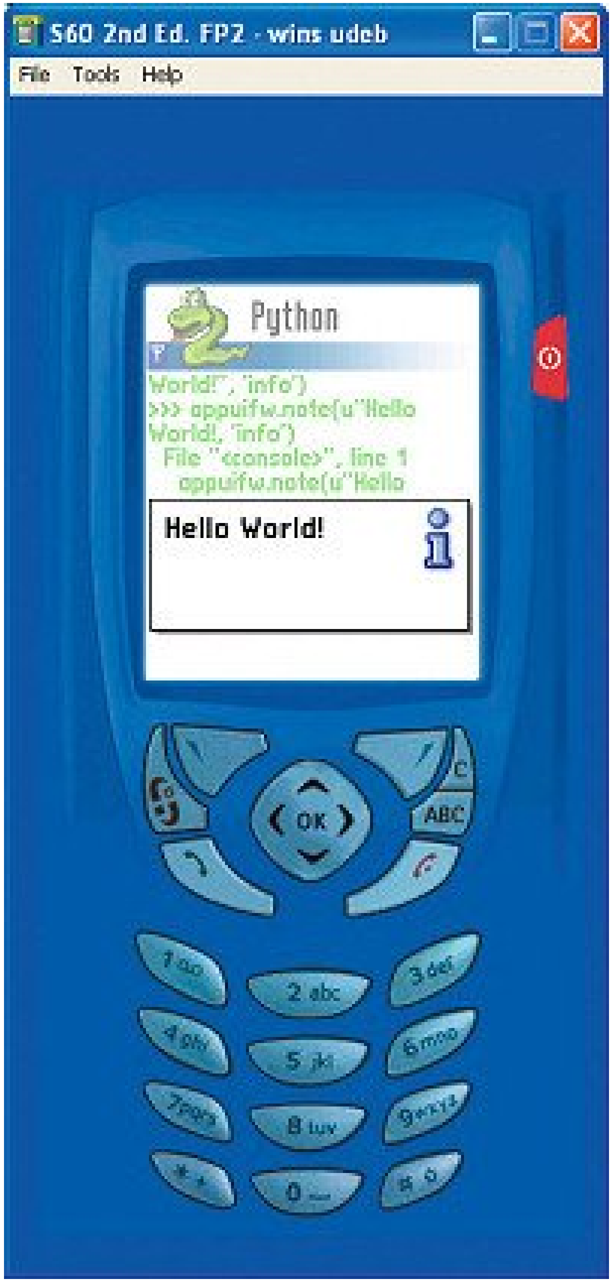
http://www.onlamp.com/pub/a/onlamp/2004/09/16/symbian_programming.html

Python for Series 60 Phones

- Python:
  - Open Source programming language (Guido von Rossum)
  - Interpreted, interactive, object-oriented
- Python for Series 60 phones
  - Python interpreter for Series 60 phones
  - Large parts of Python standard library
  - Smartphone-specific modules, e.g. GUI widgets, Bluetooth, GSM Location, SMS messaging, camera access, …
- Example:

```
import appuifw
appuifw.note(u"Hello World!","info")
```

http://www.forum.nokia.com/python
http://www.heise.de/mobil/artikel/74083

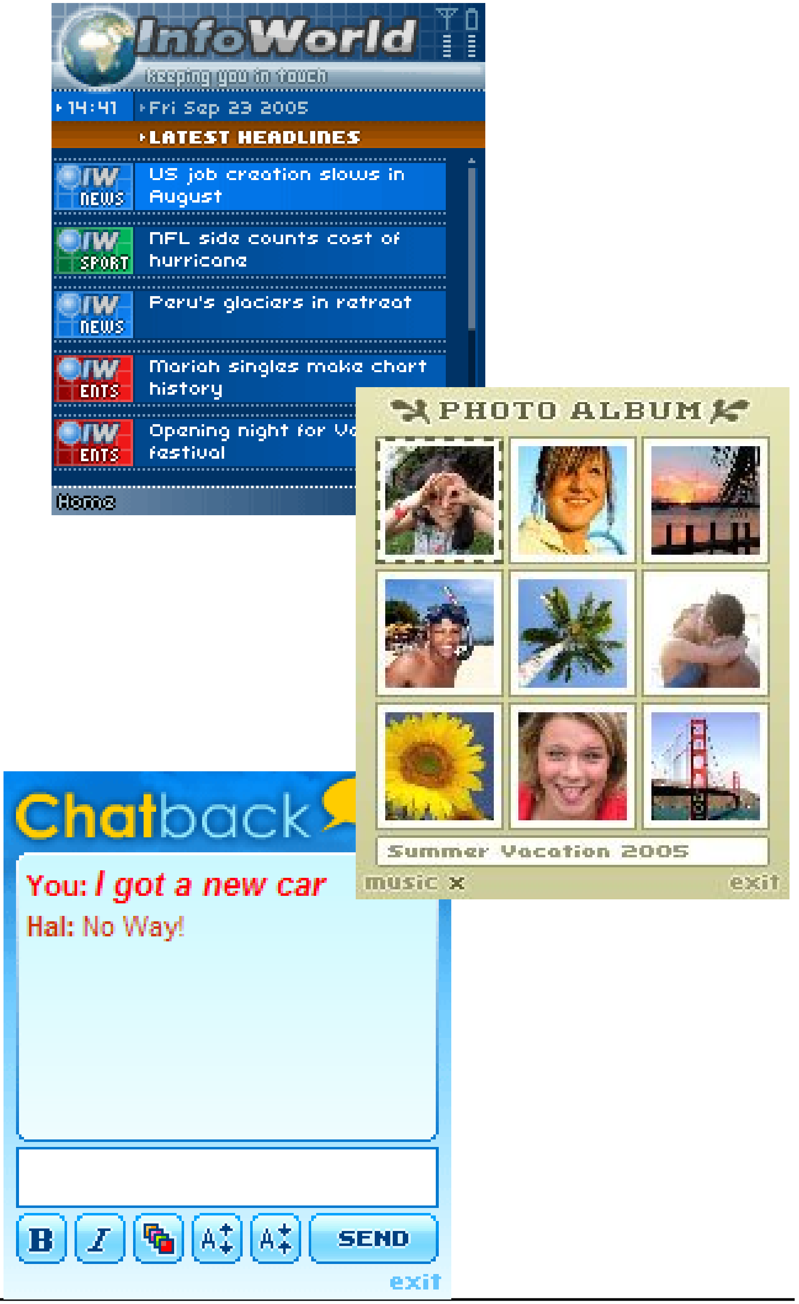


```
World!", 'info')
>>> appuifw.note(u"Hello
World!, 'info')
  File "<console>", line 1
    appuifw.note(u"Hello
```

Hello World!

Adobe Flash Lite (1)

- Player Flash Lite 2
  - Flash technology specifically developed for mobile phones and consumer electronic devices
  - Based on Flash Player 7
  - Pre-installed (Asia, Flash for i-mode)
- Authoring tool:
  Flash Professional 8 / CS3
- Example features
  - Dynamic XML data
    - » As in Flash player 7
  - Dynamic multimedia
    - » Loading of images, sound, video
  - Text enhancement
    - » User modifies text properties

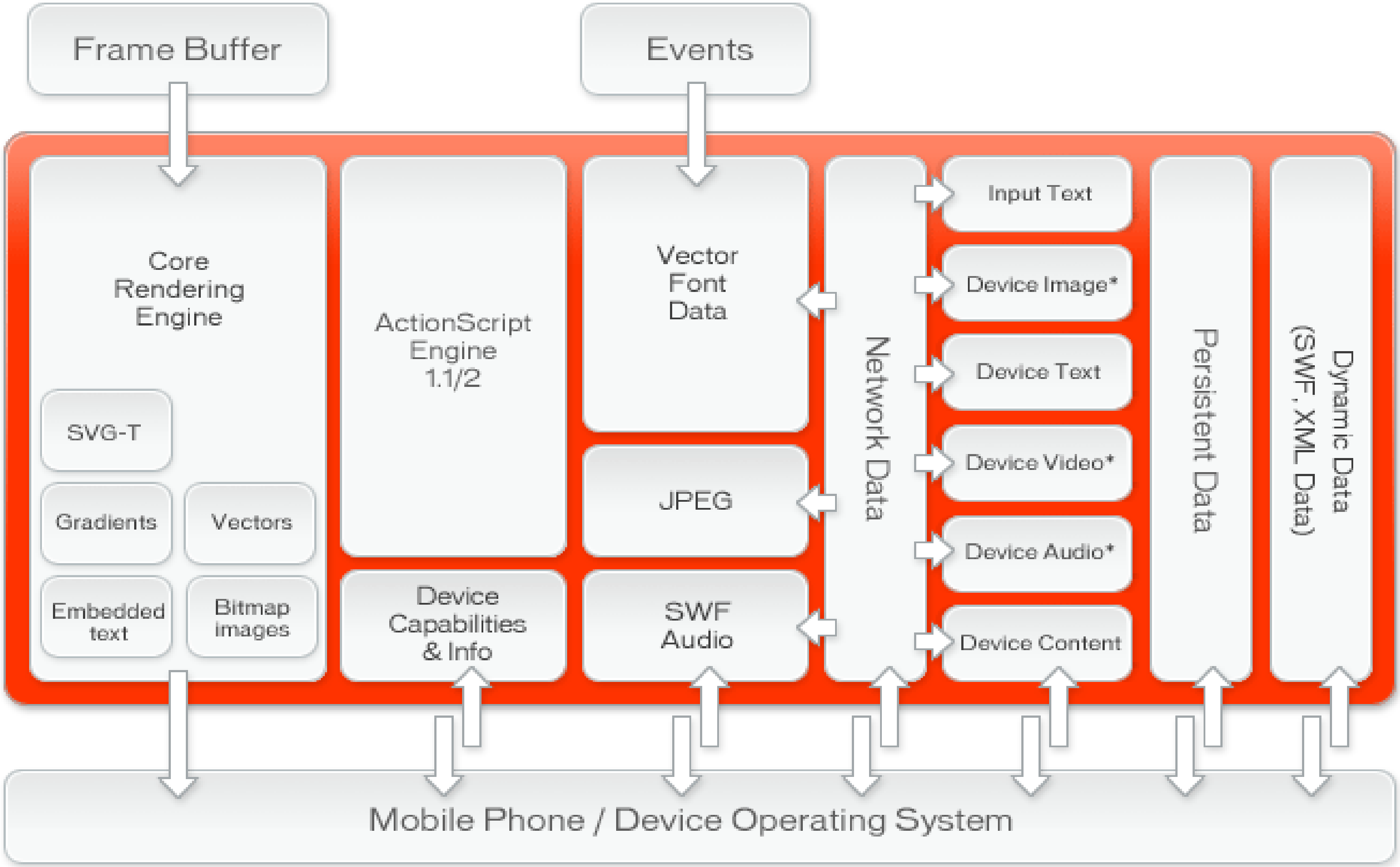


InfoWorld
keeping you in touch
14:41 Fri Sep 23 2005
LATEST HEADLINES
IW NEWS US job creation slows in August
IW SPORT NFL side counts cost of hurricane
IW NEWS Peru's glaciers in retreat
IW ENTS Mariah singles make chart history
IW ENTS Opening night for Ve festival
Home

PHOTO ALBUM
Summer Vacation 2005
music ✕ exit

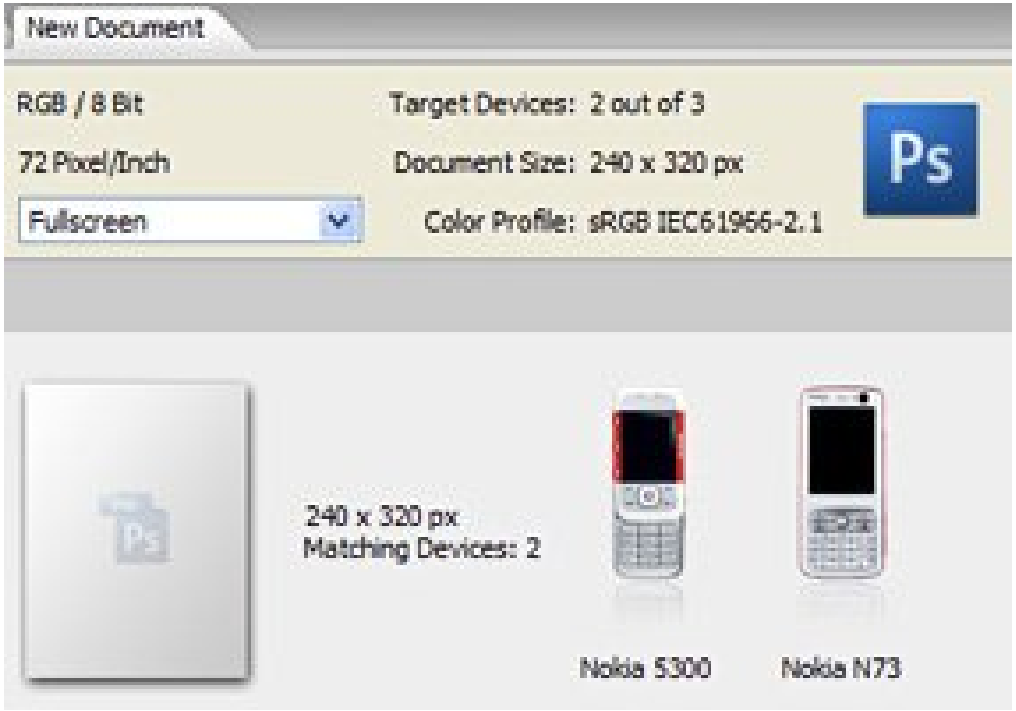
Chatback
You: I got a new car
Hal: No Way!

Flash Lite: Architecture



* Using on-device, platform-based codecs

Adobe Flash Lite (2)



New Document

RGB / 8 Bit

Target Devices: 2 out of 3

72 Pixel/Inch

Document Size: 240 x 320 px

Fullscreen

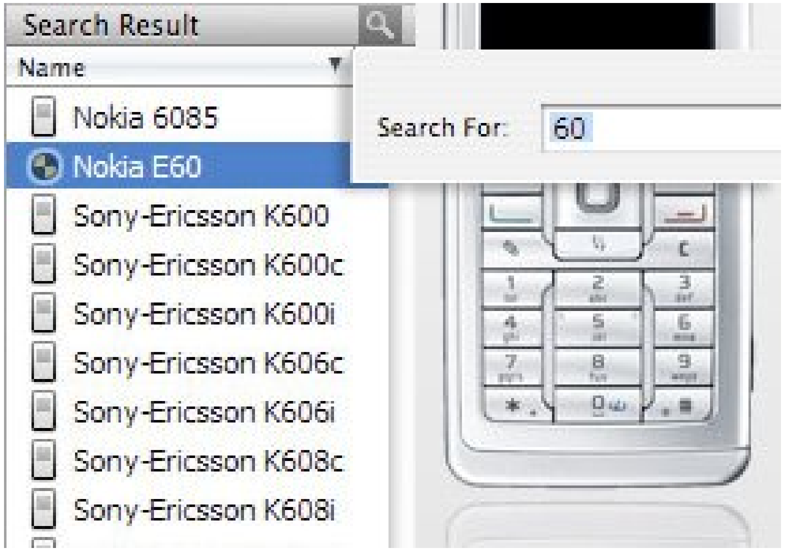
Color Profile: sRGB IEC61966-2.1

240 x 320 px
Matching Devices: 2

Nokia 5300

Nokia N73

Authoring with Flash tools,
Customized for mobile devices



Search Result

Name

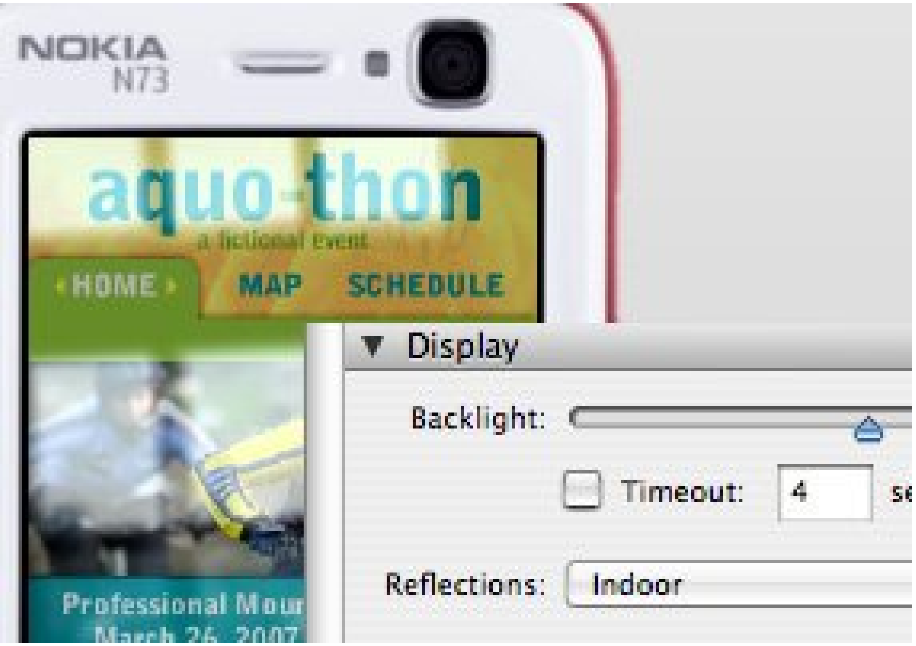
- Nokia 6085
- Nokia E60
- Sony-Ericsson K600
- Sony-Ericsson K600c
- Sony-Ericsson K600i
- Sony-Ericsson K606c
- Sony-Ericsson K606i
- Sony-Ericsson K608c
- Sony-Ericsson K608i

Search For: 60

Searching a device profile

Adobe Flash Lite (3)

Emulators

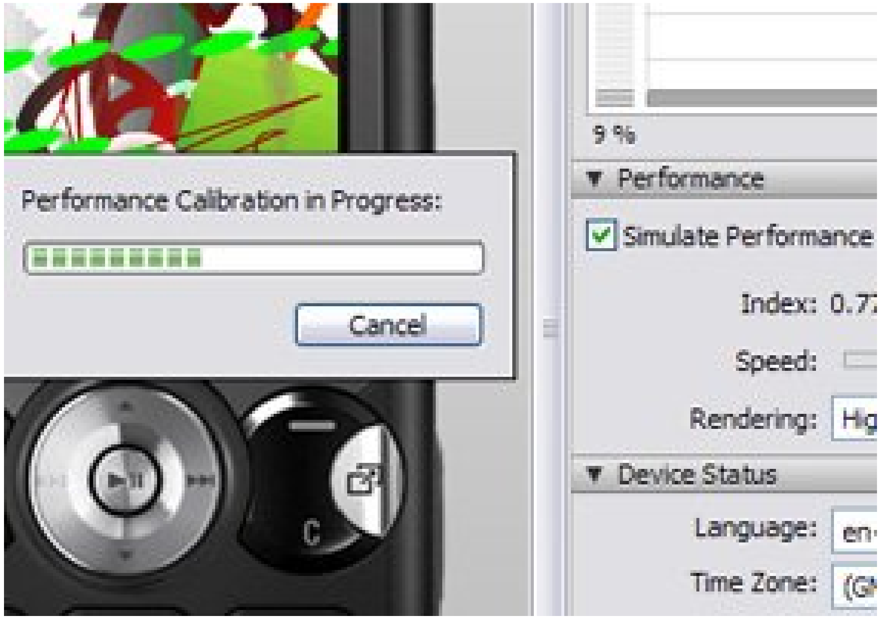


▼ Display

Backlight:

□ Timeout: 4 se

Reflections: Indoor


Performance Calibration in Progress:

9 %

▼ Performance

☑ Simulate Performance

Index: 0.7

Speed:

Rendering: Hig

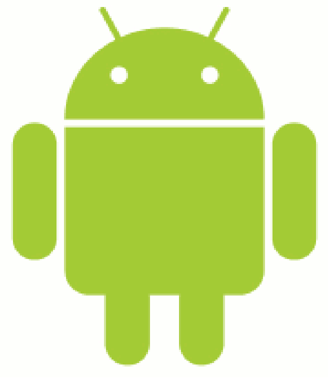
▼ Device Status

Language: en-

Time Zone: (G

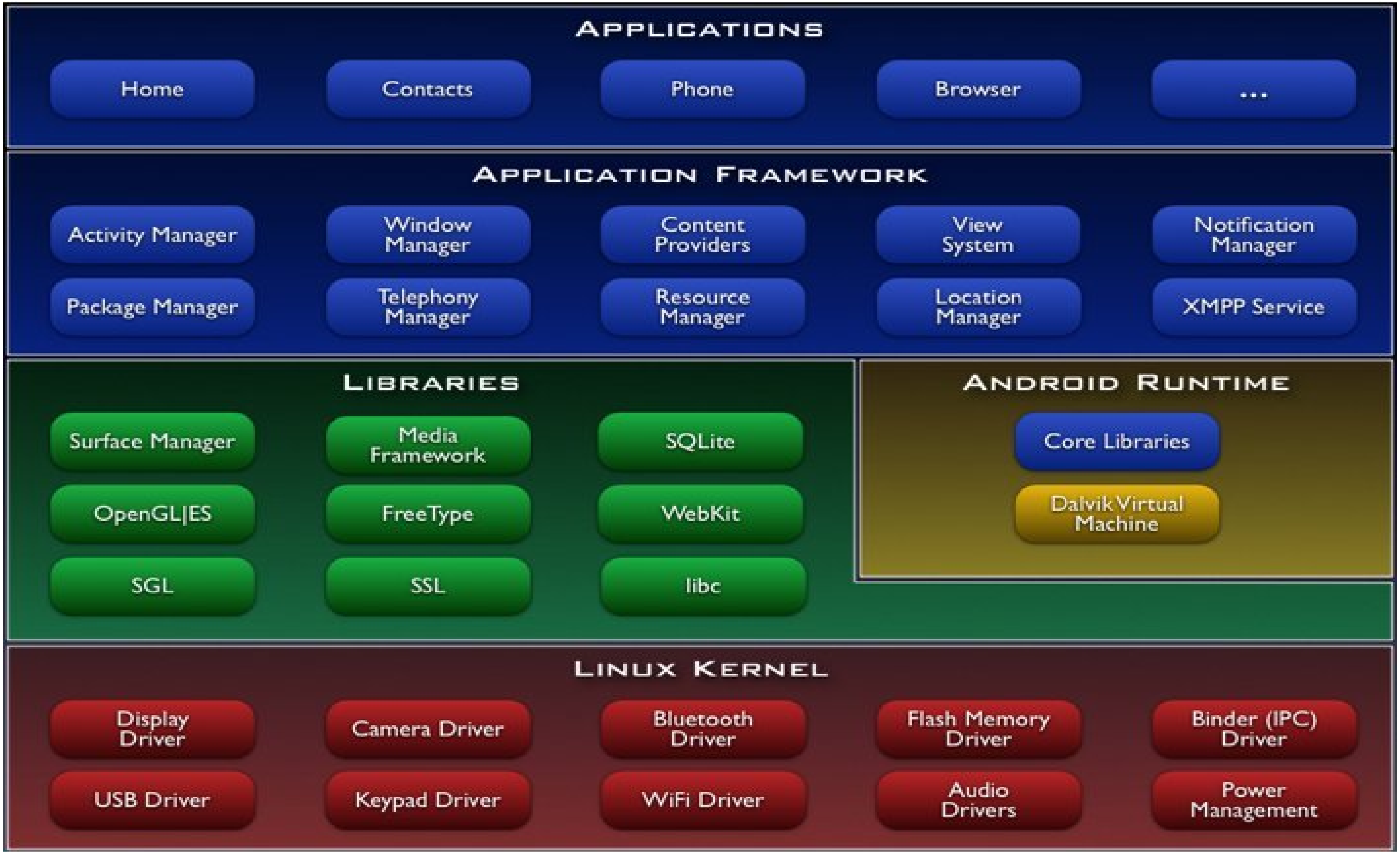
Android

- Created by Open Handset Alliance
  - More than 30 companies, led by Google
  - Includes mobile operators, chip makers, handset makers, software companies
  - Overall (official) goal: Improve mobile user experience
- Free, open mobile platform
  - Many parts open-source (Apache license)
  - Both Free and commercial software encouraged
  - Contrast to more closed, commercial J2ME world
- Applications can access same features as software shipped with device
- Stress on networked applications, sensor-equipped phones
- Claims easier development of applications than J2ME
- First version of SDK in late 2007 – first hardware expected in late 2008



http://code.google.com/android/
http://www.openhandsetalliance.com/
http://www.heise.de/newsticker/Ausblick-auf-Android--/meldung/108785

Android Architecture



APPLICATIONS

Home, Contacts, Phone, Browser, ...

APPLICATION FRAMEWORK

Activity Manager, Window Manager, Content Providers, View System, Notification Manager

Package Manager, Telephony Manager, Resource Manager, Location Manager, XMPP Service

LIBRARIES

Surface Manager, Media Framework, SQLite

OpenGL|ES, FreeType, WebKit

SGL, SSL, libc

ANDROID RUNTIME

Core Libraries

Dalvik Virtual Machine

LINUX KERNEL

Display Driver, Camera Driver, Bluetooth Driver, Flash Memory Driver, Binder (IPC) Driver

USB Driver, Keypad Driver, WiFi Driver, Audio Drivers, Power Management

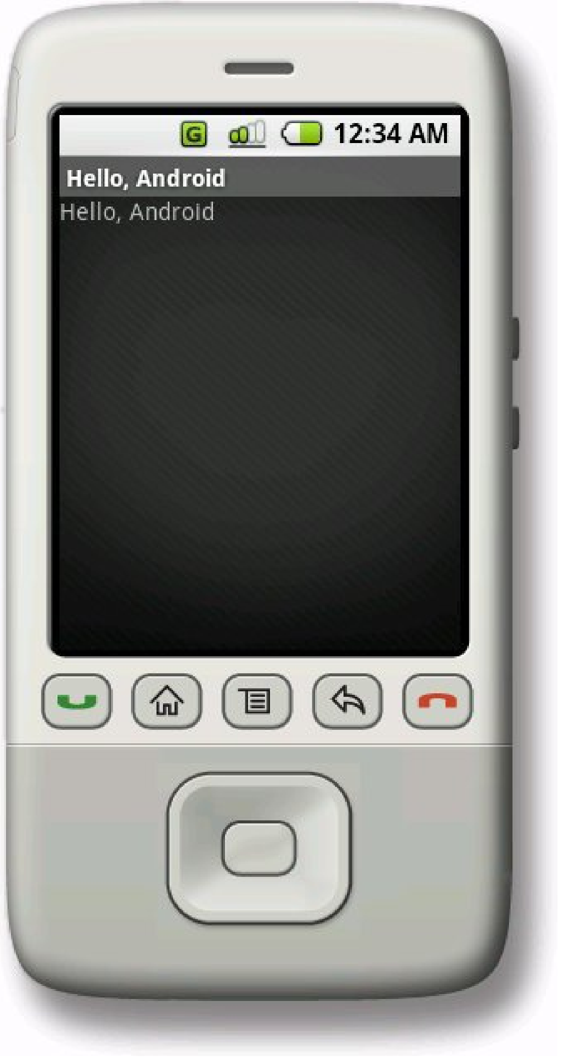
http://code.google.com/android/what-is-android.html

Android – Hello World

```
package com.android.hello;

import android.app.Activity;
import android.os.Bundle;
import android.widget.TextView;

public class HelloAndroid extends Activity {
   /** Called when the activity is first
       created. */
   @Override
   public void onCreate(Bundle icicle) {
       super.onCreate(icicle);
       TextView tv = new TextView(this);
       tv.setText("Hello, Android");
       setContentView(tv);
   }
}
```



Hello, Android

Hello, Android

1 Mobile and Ubiquitous User Interfaces

2.1 Mobile Computing

2.2 Input and Output on Mobile Devices

2.3 Design Guidelines for Mobile Devices

2.4 System Architectures for Mobile Devices

2.5 Example Applications

2.6 HCI and Ubiquitous Computing

Praktikum WS07/08

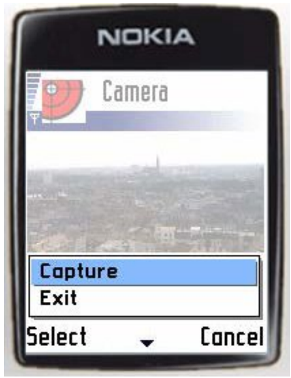
- Entwicklung von Mediensystemen (Mobile Endgeräte)
- Development of a mobile application within a team (idea, concept, implementation, evaluation)
- Supervisors: Alexander De Luca, Gregor Broll

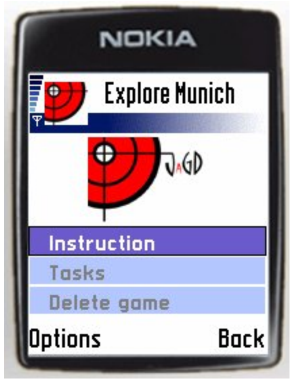
Hardware

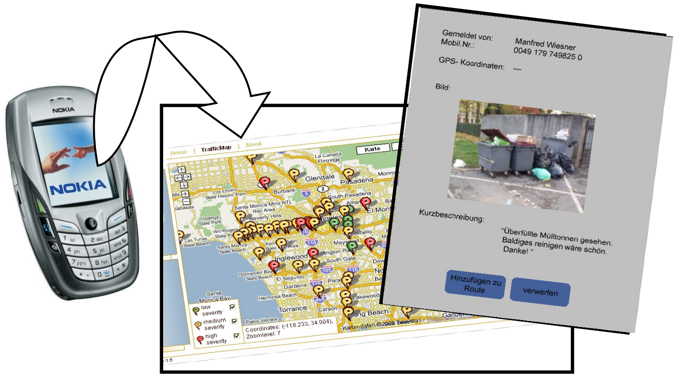
- Mobile Phones: Nokia N73 (3x), Nokia E61, Nokia N80, Nokia 6131 NFC, Nokia 5500 (2x), Nokia N90, Nokia N91, Nokia N70, Nokia 6630 (2x), Nokia 6600 (4x), Nokia 3220, Samsung SGH-E760
- GPS-receiver, NFC, visual tags
- SIM-Cards (O2, T-Mobile, Vodafone)
- Mobile Health Equipment (ECG-Reader, Pulse Oximeter)

Examples

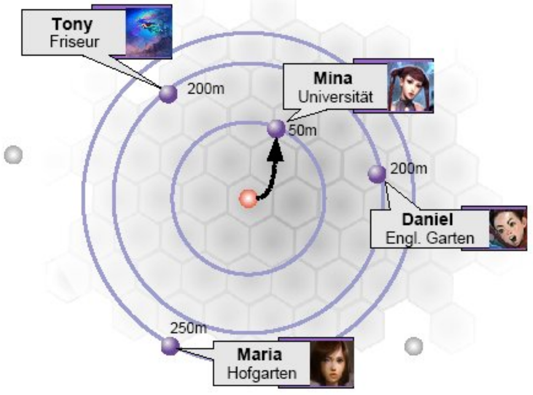
- Praktikum WS 04/05
  - 3 Anwendungen:
    - JaGD
    - Traffic Warden Support
    - Posters as Gateways
- Praktikum Mobile Productivity WS 06/07
  - Entwicklung von mobilen Anwendungen für blue-collar worker
  - 3 Anwendungen:
    - Mobile Inventory System
    - Mobile Product Evaluation and Comparison of Prices
    - Mobile Tagging Platform
- Praktikum SS 07
  - Running Project “Beeepr” Mobile Tagging
- Praktikum SS 08
  - Android: Mobile Health





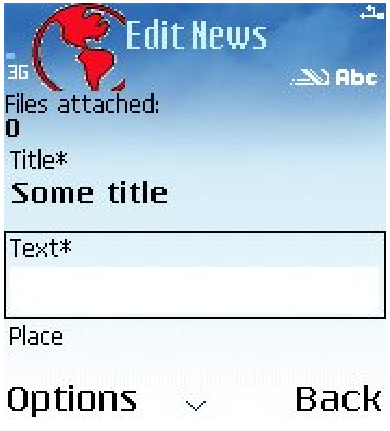


Gemeldet von: Manfred Wiesner
Mobil.Nr.: 0049 179 749825 0
GPS- Koordinaten: ---
Bild:
Kurzbeschreibung: “Überfüllte Mülltonnen gesehen. Baldiges reinigen wäre schön. Danke! ”



Mobile Reporter

- Mobile Blogging Platform
- Submit via SMS, MMS, E-Mail, MIDlet and Webinterface
- Available at Sourceforge

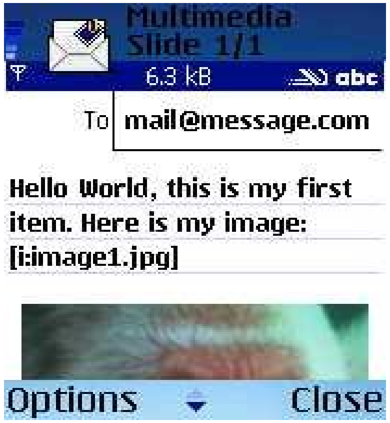


Files attached:
0

Title*
Some title

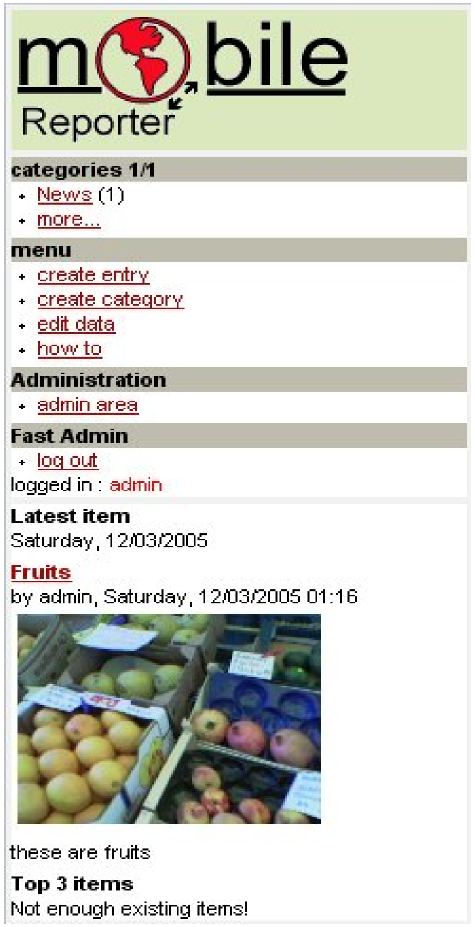
Text*

Place



To mail@message.com

Hello World, this is my first item. Here is my image: [i:image1.jpg]



categories 1/1
- News (1)
- more...

menu
- create entry
- create category
- edit data
- how to

Administration
- admin area

Fast Admin
- log out

logged in : admin

Latest item
Saturday, 12/03/2005
Fruits
by admin, Saturday, 12/03/2005 01:16

these are fruits

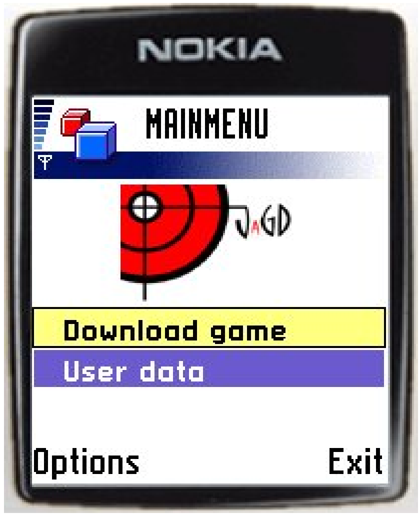
Top 3 items
Not enough existing items!

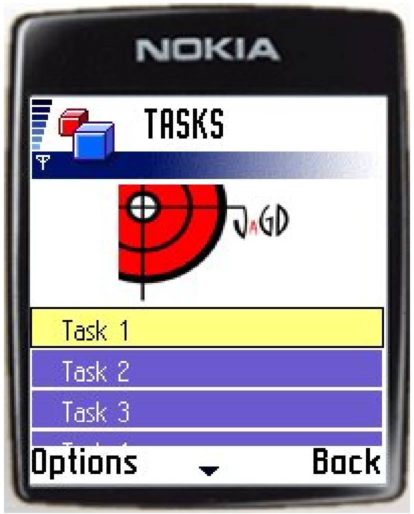
Alexander De Luca

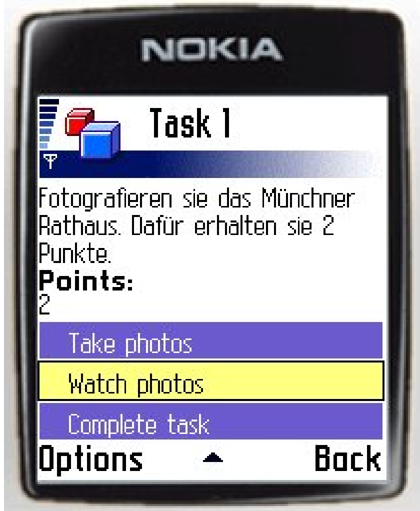
Mobile Photo Treasure Hunt

- Mobile Learning and Gaming Platform
- Online game editor
- XML game format
- Mobile phone application

Alexander
De Luca







Fotografieren sie das Münchner Rathaus. Dafür erhalten sie 2 Punkte.

Points:
2

1 Mobile and Ubiquitous User Interfaces

2.1 Mobile Computing

2.2 Input and Output on Mobile Devices

2.3 Design Guidelines for Mobile Devices

2.4 System Architectures for Mobile Devices

2.5 Example Applications

2.6 HCI and Ubiquitous Computing

Just a few trends…

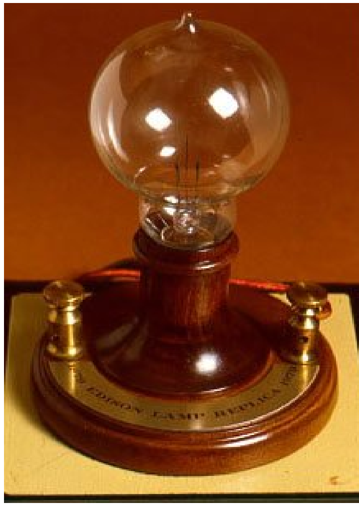
(Acknowledgement: Albrecht Schmidt)

Trends (1)

mobile communication is ubiquitous

- Terminals for mobile communication have advanced significantly over recent years
- **Infrastructure is ubiquitously deployed**
- Interesting developments happen beyond the classical handsets (when thinking of electricity it is not the advances in light bulbs that changed the world)

- How many handsets will a user have in 10 years time?
  →a guess 2-6 (some mobile phones, car phone, …)
- How many communicating appliances and devices will users have in 10-20 years time?
  → a guess 20+ (security system, TV, front door, dog collar, wrist watch, camera, headset, coffee machine, alarm clock…)



Trends (2)

mechanical and electro-mechanical systems
will be computer controlled

- Mechanical and electro-mechanical systems become computer controlled.
- User interfaces for mechanical and electro-mechanical systems have a tradition of being tangible.
- Many **design restrictions** due to mechanics **are gone** – novel interfaces (for the better or the worse) are possible and emerge.
- **Sensing of actions and reactions from users becomes an interface option.**
- Examples: automotive, industrial machinery, tools, buildings.

Trends (3)

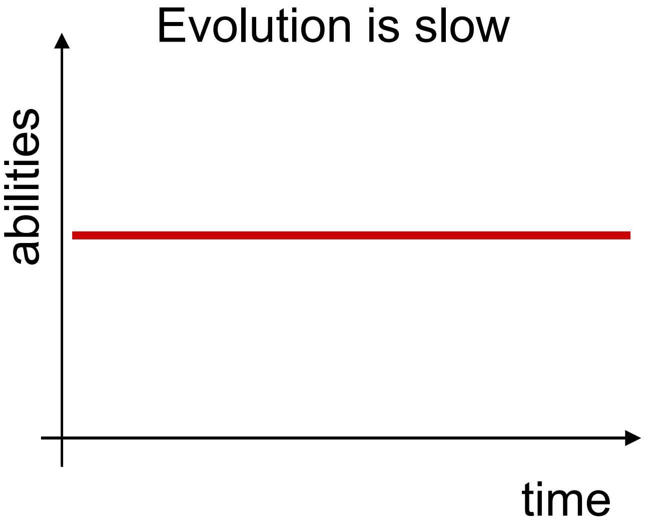
declining willingness for training

- An average person acts today as driver, telephonist, photographer, film-maker, and type setter without much training (many task with just one device – the phone).
- In a fast paced job market training to operate a system is a significant obstacle (and cost factor) for the introduction of new systems.
- Dangerous actions should be prohibited in the first place by the controls available in the user interface.
- User interfaces that have **clear affordances** and draw on the **prior knowledge** of potential users (“intuitive UIs” and “natural interaction”) reduce the need for leaning

Trends (4)

user’s abilities

- Abilities of un-augmented users in general do not change a lot over time, e.g.
  - ability to cope with cognitive load
  - willingness to cope with stress
  - time one can concentrate on a particular problem
- Abilities between individual users vary a lot
  - long term, e.g. physical and intellectual abilities
  - short term, e.g. effect of stress or fatigue
- Abilities of one individual users changes over time (e.g. getting old)



Human in the loop
Interactive systems for “augmenting the human intellect” as alternative to automation.

Trends (5)

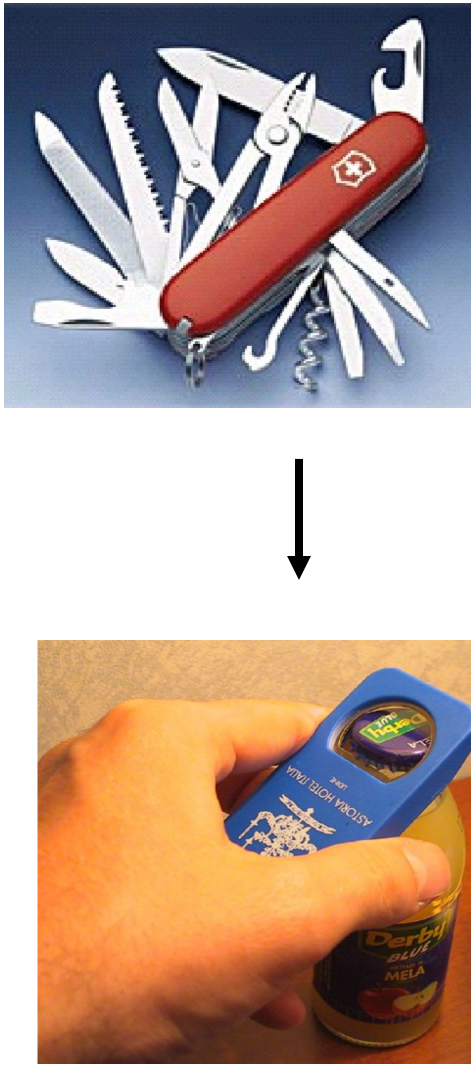
technology becomes widely available

- Technologies that may be today "specialist devices" become common in a *few* years
- Technologies that are shared now may become personal technologies
- Technologies that are expensive at one point are not even considered as additional cost in the future, e.g.
  - Video camera connected to a computer
  - Biometric authentication
  - Book printing on demand
  - Eye gaze tracking
  - 3D scanning and printing
  - Integrated production systems

Trends (6)

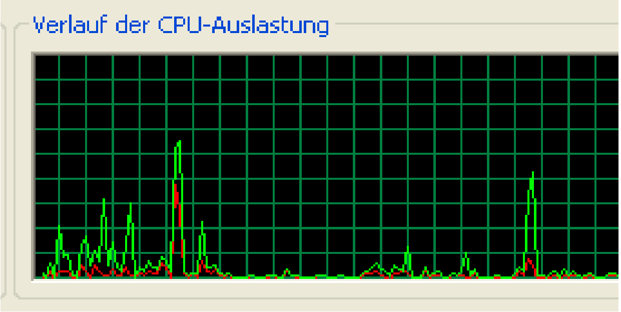
appliance computing

- Post-PC area
  - Specific tools that are designed to support a specific task
  - Not a all-round tool
  - Different tools for different tasks
- *"[…] the primary motivation behind the information appliance is clear: simplicity.* ***Design the tool to fit the task so well that the tool becomes part of the task****, ..."* (Don Norman)
- Context and adaptation to the real world is an option to overcome the multi-device dilemma



Trends (7)

computing, storage and communication are not the limit



- For personal computing there are few technical limitations
- Processing power is available
  - Already now desktop machines run with minimal processing power
- Massive amounts of storage are readily available
  - Phones with 4GB disk
  - Record everything you ever said on a hard drive
  - Have all movies ever produced in a single device
- Bandwidth (wireless and wired) is huge
  - While you tie your shoe laces you can cache all the latest 20 different news papers
  - While you wait for the bus you can transfer a complete movie

User interfaces and interaction for networked devices that are embedded into the users' lives.

- Anytime and everywhere
- Design restrictions are gone
- Sensing and actuators are part of the UI
- Must be obvious to use (affordances)
- Current cost of technology is not an issue

The interface between the user and the machine is most critical to create effective and efficient systems.