Tutorial: Advanced Topics in HCI (Mensch-Maschine-Interaktion II)

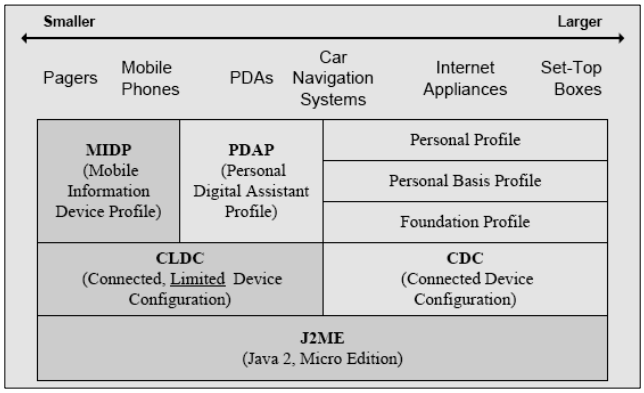
Sebastian Boring, Richard Atterer, Andreas Butz
Media Informatics Group
Ludwig-Maximilians-Universität
Munich, Germany

Java ME – Introduction

- Java ME:
  - Slim Java for mobile devices
  - Java ME stack:
    - Configuration + profile + additional APIs
  - Configuration:
    - JVM + minimal amount of functionality (e.g. CLDC 1.1)
    - Subset of Java SE
  - Profiles:
    - Enhance the configuration with functionality (e.g. MIDP 2.0)
    - APIs for user interface, persistent storage, etc.
  - Additional APIs for Bluetooth connections, Multimedia, etc.

Java ME – Introduction

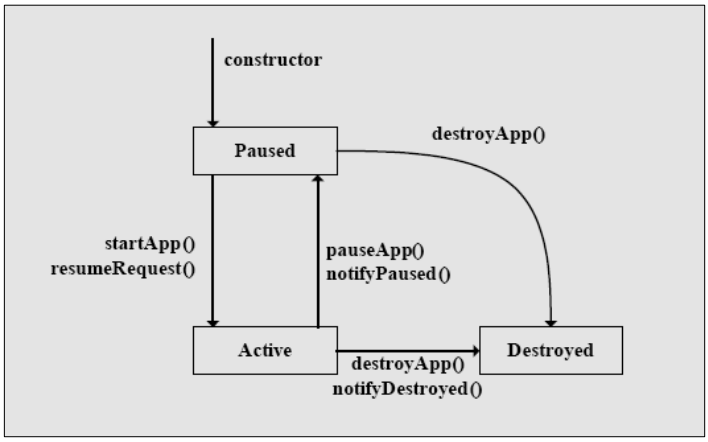
The Java ME Universe:



Java ME – MIDlets

- MIDlets:
  - MIDP applications are called MIDlets
  - Every MIDlet is an instance of:
    - `javax.microedition.midlet.MIDlet`
    - Constructor + Implementation of lifecycle methods
  - Conceptually similar to applets:
    - Can be downloaded
    - Executed in host environment

Java ME – MIDlet Lifecycle



5

Java ME – MIDlet Lifecycle (contd.)

- Lifecycle:
  - Application Manager (AM) controls Installation and Execution of MIDlets on the phone
  - Starting a MIDlet:
    - Constructor + `startApp()` are executed by AM
  - MIDlet:
    - Places itself in paused state (`notifyPaused()`)
    - Destroys itself (`notifyDestroyed()`)
  - One method for every state transition

6

Java ME – Build Cycle

- The Build Cycle:
  - Edit source code
  - Compile (like compiling Java SE apps)
  - Preverify:
    - Bytecode verification (ensures correct behavior + won't do "nasty" things) split into two steps
    - Lightweight second verification on the mobile device (standard verification is too memory intensive)
    - Special class format (adds 5% to normal class file size)
    - Normally not visible to the programmer

LMU 7

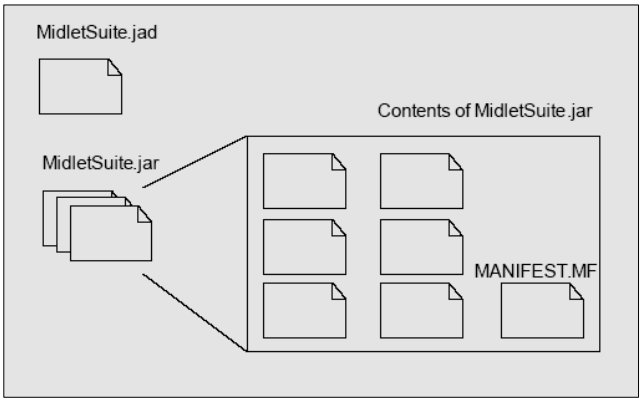
Java ME – Build Cycle (contd.)

- The Build Cycle:
  - Application Package, MIDlet Suite:
    - MIDlets + classes + resources + Manifest Information
      → Java Archive (JAR)
    - Manifest: describes content of archive (CLDC and MIDP versions, name, version, vendor, ...)
    - Application descriptor (`*.jad`):
      - Same information as in the manifest file, but external file
      - Normally only used for installation
  - Test or Deploy!

LMU 8

Java ME – Anatomy

Anatomy of a MIDlet Suite:



Java ME – User Interface

- MIDP: User Interface:
  - Goal: Wirte once, **run anywhere**
  - Run anywhere?
    - Different screen sizes
    - Resolution of a screen
    - Color, grayscale or monochrome screen?
    - Different input capabilities:
      - Numeric keypad
      - Alphabetical keyboards
      - Soft keys
      - Touch screens
      - Sensors, ...

Java ME – User Interface (contd.)

- User Interface: Methodology
  - Abstraction (Preferred Method):
    - Specifying a user interface in abstract terms
    - **Not:** Display the word "Next" above the left soft key
    - **But:** Give me the "Next" command somewhere in the interface
  - Discovery (Games):
    - Application learns about the device + tailors the user interface programmatically
    - e.g.: Screen size → Scaling

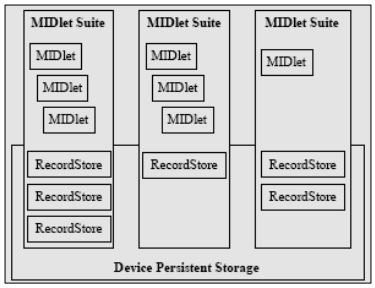
Java ME – Storage

- MDIP: Persistent Storage:
  - Goal: Wirte once, **run anywhere**
  - Anywhere?
    - Devices with Flash ROM
    - Battery-packed RAM
    - Small Hard Disc

  → **Abstraction needed!**
  - Record stores: Small databases (Minimum: 8 kByte)
  - Newer phones contain JSR-75 (File API) for direct access to the phone's file system

Java ME – Storage (contd.)

- Persistent Storage: Record Stores:
  - Record Stores:
    - 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 boundaries possible!



Assignment II – Task I

- Explore the basics of Java ME:
  - Make yourself familiar with the given source code
  - Add an input field and display the written string instead of "Hello World"
  - Allow the display of an image on the screen. Use the provided image "ex2.png"
- It is enough to let the code run in the emulator!
- Deadline: **Monday, May 26th, 2008**

Assignment II – Task II

- Develop a scenario that employs the following devices:
  - One large screen:
    - High resolution (up to HD)
    - Sound output available
    - *Optional:* Interactive surface
  - Multiple mobile phones:
    - High-quality screen with 240x320 pixels + audio output
    - Numeric key input (with joystick keys)
    - Built-in high-resolution camera (~ 5 Megapixels)
    - Built-in acceleration sensors

Assignment II – Task II (contd.)

- **First step:**
  - Create a scenario (i.e. an application idea) based on the given devices and their capabilities
  - Envision a good usage of the different screen sizes and types (public vs. personal screens)
  - Be as creative as possible!
- **Second step:**
  - Out of your scenario, create a paper prototype:
    - Walk-through of your application
    - Screen visualization without electronic help (only hand-drawings)
    - Visualize both mobile screens and large screen content
- Deadline: **Monday, June 2nd, 2008**

Submission of solutions

- Only one submission per group:
  - **Task I:** Monday, May 26th, 2008
  - **Task II:** Monday, June 2nd, 2008
- Next tutorials:
  - Monday, May 26th, 2008:
    - Discussion of Task II – Open question session
  - Monday, June 2nd, 2008:
    - Presentation of the results (Task II)