LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

LFE Medieninformatik • Gregor Broll, Alexander De Luca

Praktikum Entwicklung von Mediensystemen

Mobile Dienste für Studenten

Introduction – 10/23/2009

Outline

Outline:

- **Basic Information**
- **Organizational Stuff**
- **Technology**
  - SVN
  - Java ME
  - Resources and IDEs
- **Exercise 1**

Basic Information about the Practical

- **Design and development of mobile applications with Java ME**
- **This year: special focus on mobile services for students**
- **Two phases: single and group work phase**
- **Software development in teams**
- **Phase 1 – Individual Work:**
  - Exercise 1 and 2
  - Exercise 3 partially group work
- **Phase 2 – Project Work:**
  - Starting 11/13/2009
  - Project implementation
  - User study and evaluation

Mobile Services for Students

- **Practical is part of a greater effort at LFE Media Informatics to investigate mobile services for students**
  - Adaptation of existing services and information to mobile usage
  - Creation of new, more mobile services
- **Collaboration with LMU-IT (Herr Diekamp)**
  - Practical to develop prototypes that use real info and services
- **Practical runs in parallel with a diploma thesis**
  - Analysis of needs and requirements
  - Implementation and evaluation of mobile service prototypes
- **Several touching points between practical and diploma thesis**
  - Practical can use results of analysis
  - Practical to be involved with analysis (focus groups)
  - Otherwise independent from each other

Schedule

Date	Topics
23.10.2008	L1: Introduction to Java ME; Mobile Media API
30.10.2008	L2: Http and Record Stores
06.11.2008	L3: Brainstorming, Scenarios, Design, Focus Groups
13.11.2008	Project Phase starts
27.11.2009	Milestone Meeting 1
18.12.2009	Milestone Meeting 2
08.01.2010	Milestone Meeting 3 - Introduction to User Studies
22.01.2010	Milestone Meeting 4
12.02.2010	Final Presentation

Organizational Stuff I

- **4 SWS**
- **Weekly meetings**
  - Friday, 10:00 – 12:00
  - Room 107, Amalienstraße 17
- **Room for the practical parts:**
  - Medienlabor 103, Amalienstraße 17
  - Special accounts required
  - Open during regular working times (8:00 – 17:00)
  - 1 key for each group
- **Homepage:**
  - www.medien.ifi.lmu.de/lehre/ws0910/pem/
- **Bachelor**
  - 6 ECTS credits
  - Individual grades = group results + personal effort (interview)

Organizational Stuff II

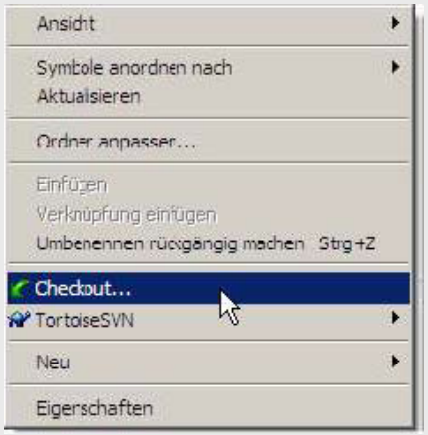
- **Needed Accounts**
  - Medienlabor-Kennung
  - Belegungsplan Medienlabor
  - SVN username
- **SVN**
  - svn://murx.medien.ifi.lmu.de/ws0910/pem/team[number] (e.g. svn://murx.medien.ifi.lmu.de/ws0910/pem/team1)
- **Teams**
  - Team 1:
  - Team 2:
  - Team 3:

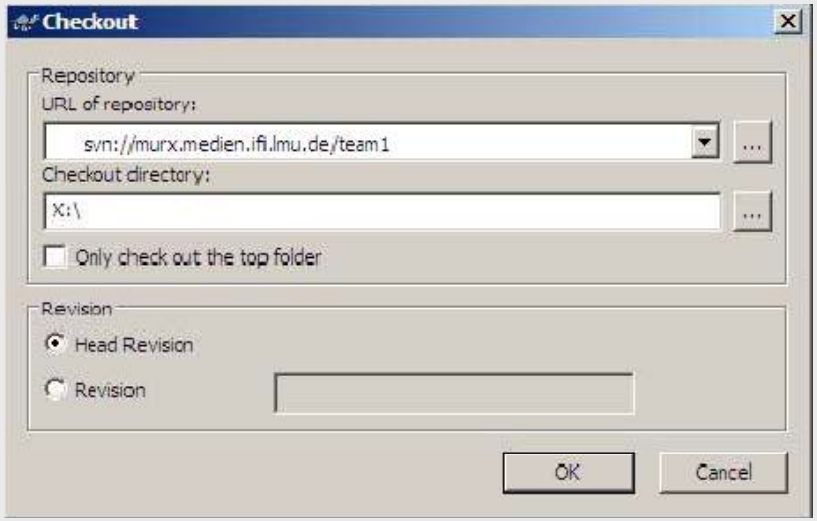
SVN - General

- Version control system
- Enables collective editing of shared source code
- Data stored in a „Repository“ which is accessed over the network
- Editing of local copies of the files
- Old version available on the server
- When possible, files will be merged automatically when edited by multiple users at the same time
- Similar to CVS

SVN – First Steps (using Tortoise SVN)

1. **Download a SVN Client like Tortoise SVN for Windows** http://tortoisesvn.net/
2. **Checkout your team repository (creates a local copy of the repository)**
3. **Create an empty folder, open it, right-click and choose „Checkout“.**





SVN – First Steps (using Tortoise SVN)

4. **Each time you start working perform the “Update“ command.**
5. **Each time you‘re done working perform a “Commit”. Both commands are located in the right-click menu.**
6. **Further functionalities are available in the right-click menu like “delete“, “rename“ and more.**
   Attention: Do not use the OS-functionalities for this functions.
   - OS-delete => deletes local version of files
   - SVN-delete => deletes file in repository upon next commit
7. **For further Information read the German SVN introduction by Richard Atterer, which can be found here:**
   http://www.medien.ifi.lmu.de/fileadmin/mimuc/mmp_ss04/Projektaufgabe/mmp-subversion.pdf

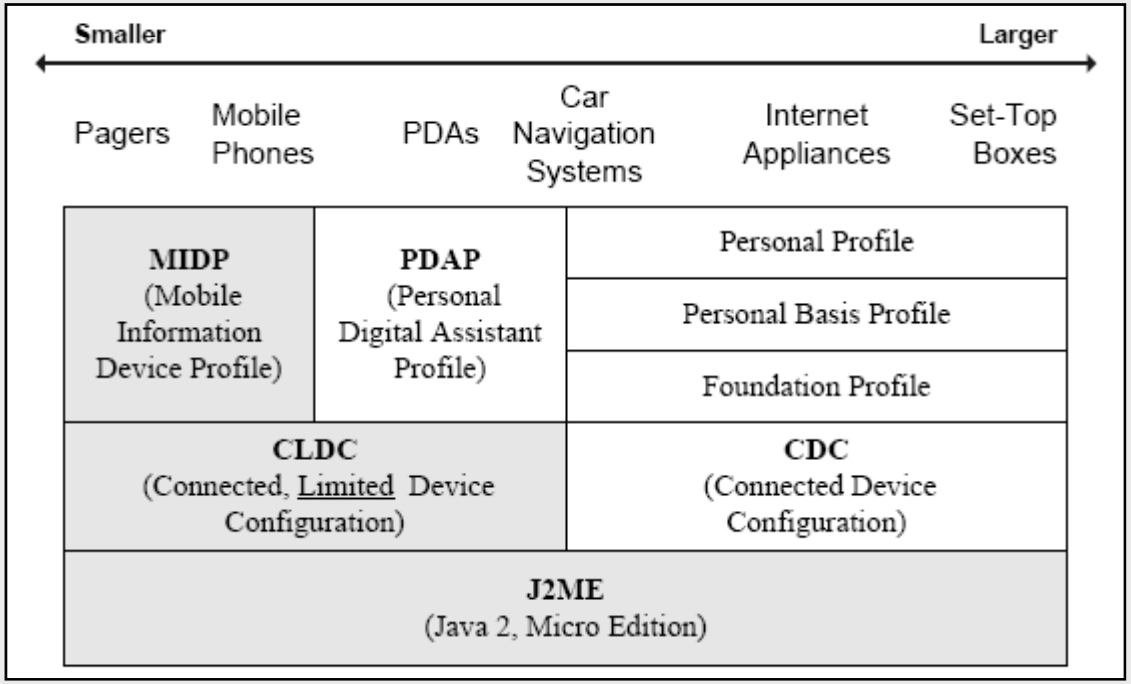
Technology – Java ME I

Java ME

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

Java Technology:
MIDP 2.0
CLDC 1.1
JSR 135 Mobile Media API
JSR 172 Web Services API
JSR 177 Security and Trust Services API
JSR 179 Location API
JSR 180 SIP API
JSR 184 Mobile 3D Graphics API
JSR 185 JTWI
JSR 205 Wireless Messaging API
JSR 226 Scalable 2D Vector Graphics API
JSR 234 Advanced Multimedia Supplements
JSR 75 FileConnection and PIM API
JSR 82 Bluetooth API
Nokia UI API

The Java ME Universe

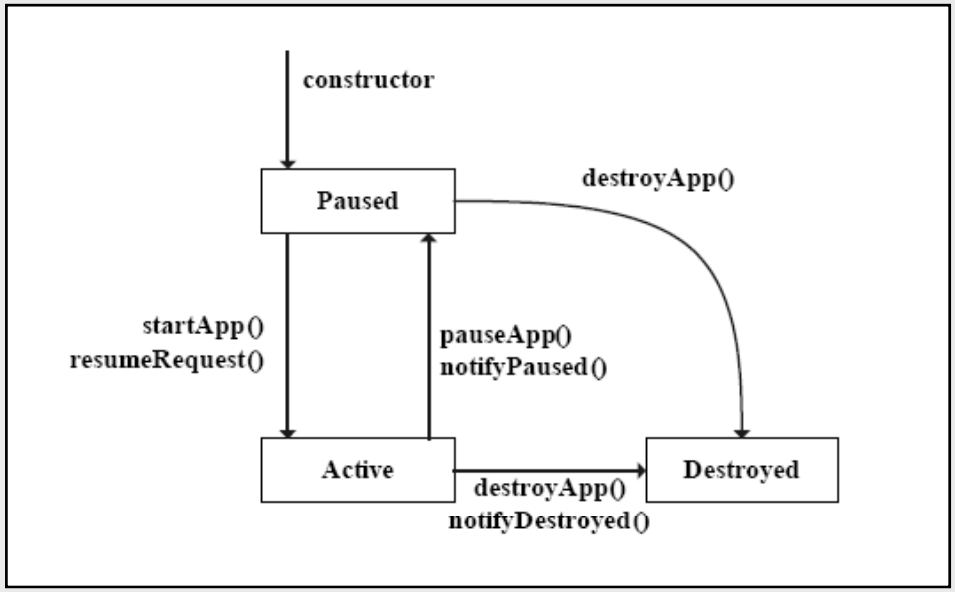


MIDlets

- **MIDP applications are called MIDlets**
- **Every MIDlet is an instance of javax.microedition.midlet.MIDlet**
  - Constructor
  - Implements lifecycle methods
- **Conceptually similar to Applets**
  - Can be downloaded
  - Executed in host environment

MIDlet Life Cycle

- **Application Manager on mobile device controls the installation and execution of MIDlets**
- **Start of a MIDlet: constructor + startApp() are executed by the Application Manager**
- **MIDlet**
  - Place itself in paused state (notifyPaused())
  - Destroy itself (notifyDestroyed())
- **One method for every state transition**

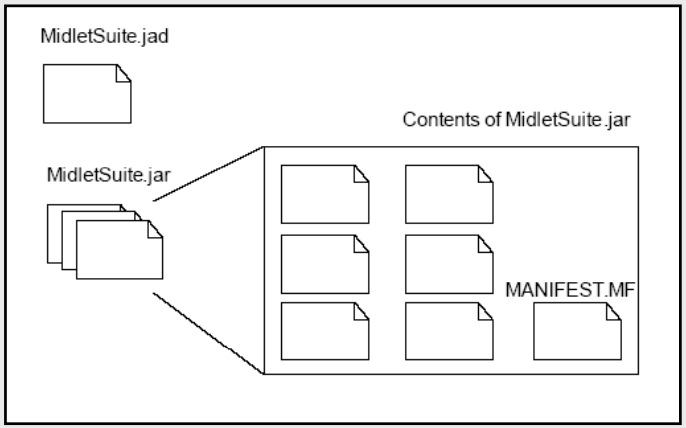


MIDlet Build Cycle 1/2

1. **Edit source code with IDE**
2. **Compile (like compiling normal java)**
3. **Preverify**
   - Bytecode verification (makes sure it behaves well + won't do nasty things) is split into two steps
   - Lightweight second verification on the mobile device (standard verification too memory intensive)
   - Special class format (adds 5% to normal class file size)
   - Normally not visible for the programmer

MIDlet Build Cycle 2/2

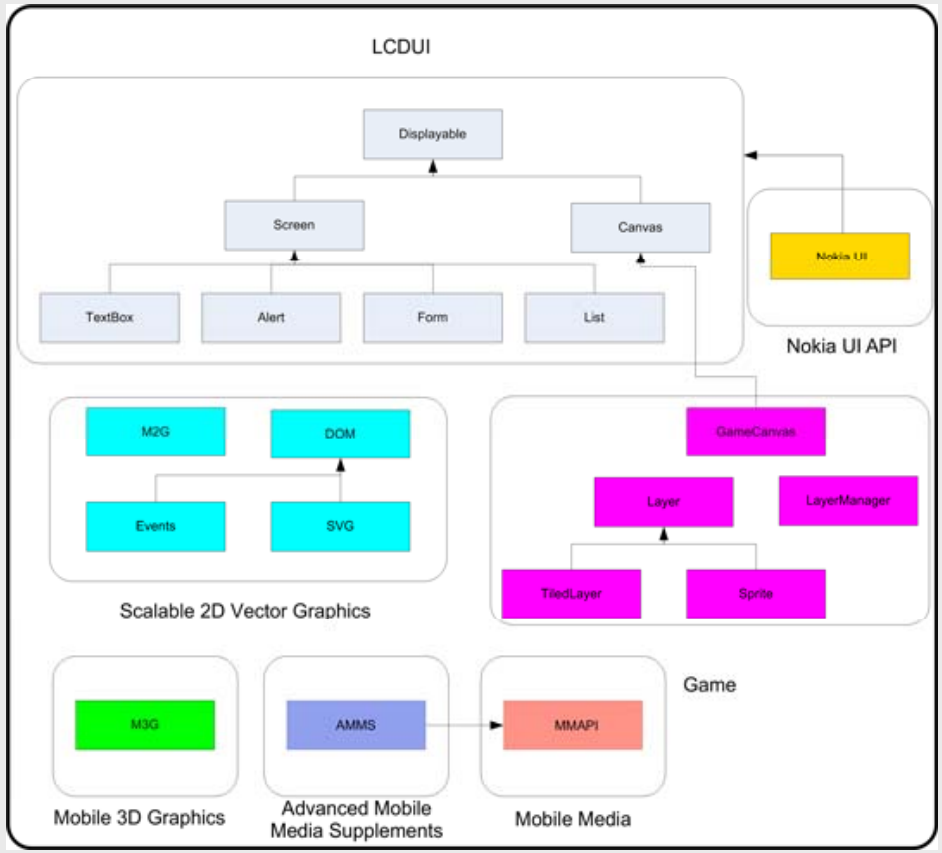
4. **(Application) Package, MIDlet Suite**
   - MIDlets + Classes + Ressources + Manifest Information => Java Archive (JAR)
   - Manifest: describes content of archive (versions of CLDC and MIDP, name, version, vendor)
   - Application Descriptor (*.jad)
     - Same information like manifest (+ MIDlet-Jar-Size, MIDlet-Jar-URL), but a external file
     - Normally used for installation
5. **Test or deploy on emulator or mobile phone**



MIDP: User Interface 1/2

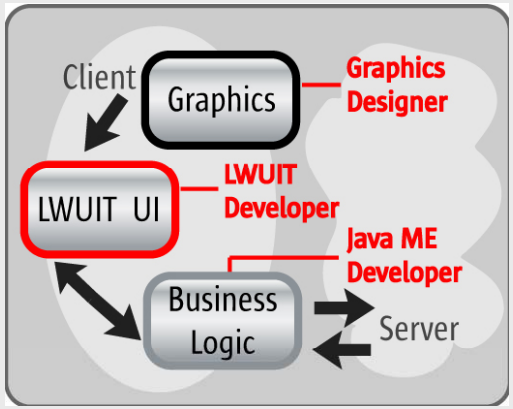
- **Goal: Write Once, Run Anywhere**
- **Anywhere?**
  - Different screen sizes, resolutions, color or grayscale
  - Different input capabilities (numeric keypad, alphabetical keyboards, soft keys, touch screens, etc.)
- **Abstraction (Preferred Method)**
  - Specifying a user interface in abstract terms
  - (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

MIDP: User Interface 2/2



LWUIT

- **Light Weight User Interface Toolkit**
- **Mobile UI library inspired by Swing**
- **Clear separation of model, view and control**
- **Integrated with applications during development**
- **Supported by CLDC 1.1 and MIDP 2.0**
- **Customizable and extendable**
- **Rapid development**
- **Features:**
  - Layout manager
  - Themes, Fonts, Look&Feel
  - Touch screen support
  - Animations, 3D, SVG



MIDP: Persistent Storage

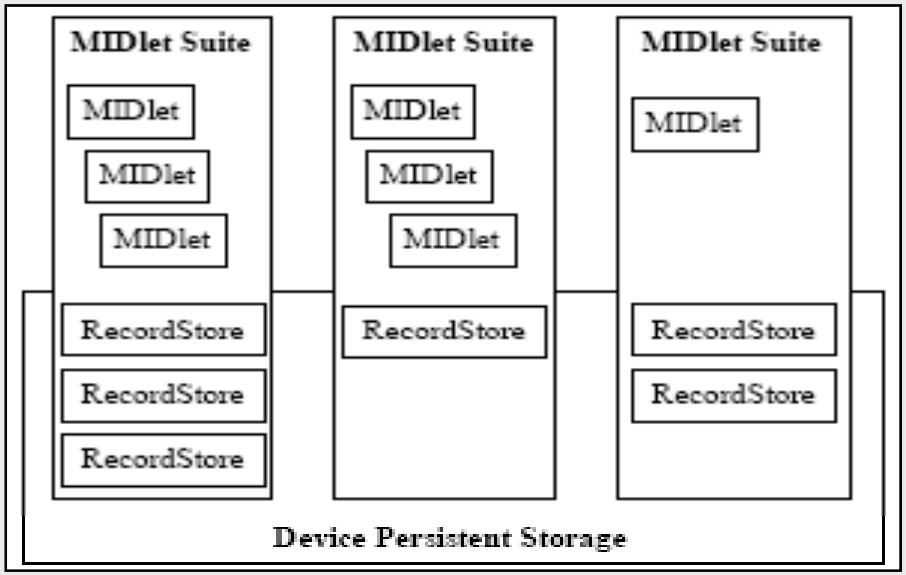
- **Goal: Write Once, Run Anywhere**
- **Anywhere?**
  - Device with Flash ROM
  - Battery-backed RAM
  - Small Hard Disk

=> Abstraction is needed

- **Record stores (small databases)**
- **Min. 8KByte (Nokia 6600: 'the only limitation is the amount of free memory')**
- **New Mobile Phone contain the File API, which allows direct access to the file system**

Persistent Storage: Record Stores

- **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 !!!



Basics:

- *Manager* **object is used to create a** *Player* **on a given data stream**
  - e.g. *Manager.createPlayer("capture://audio?encoding=amr")*
- *Player* **used for controlling streams**
- **Different player states:**
  - *Unrealized:* not enough information yet
  - *Realized:* all required information available but no resources used
  - *Prefetched:* player can be started
  - *Started:* player is running until the stream ends or the method stop()is called
  - *Closed:* all resources are freed and player stopped. Can be reached by calling close().

Device Abilities:

- **Supported content types:**
  - *Manager.getSupportedContentTypes(Stringp)* returns all available content MIME-types (e.g. image/gif) for a specific protocol (e.g. http) or all available types if argument is null
  - *System.getProperty(Stringproperty)* can be used to check for available features.
  - Examples:
    - »System.getProperty(“supports.audio.capture”)
    - »System.getProperty(“supports.audio.capture”)
    - »System.getProperty(“audio.encodings”)
    - »System.getProperty(“video.encodings”)

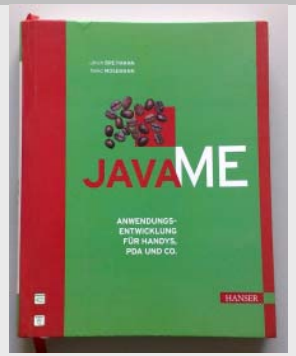
Java ME Resources

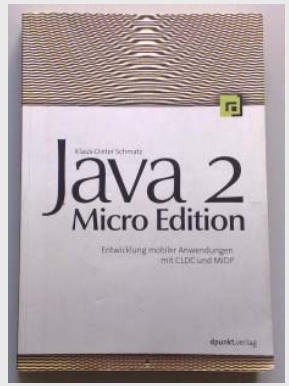
Web Links:

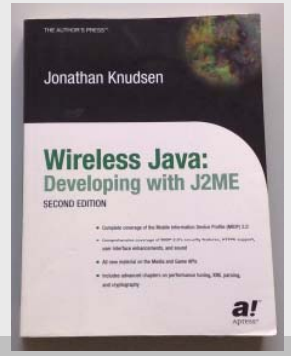
- SUN, Java ME
- Java ME References (APIs, Docs, Code Samples …)
- Nokia Series 60 Usability Guidelines
- Forum Nokia
- Java ME Developer's Library 2.0
- Forum Nokia Wiki

Books:

- 3 books about Java ME available in room 107
- Can be used for development, but must remain in the room
- „Java ME“ by Ulrich Breymann contains good references for solving exercise 1







Recommended IDEs

- **Netbeans (http://www.netbeans.org/index.html) + Mobility Pack (http://mobility.netbeans.org/)**
  - Much better Java ME support then Eclipse (e.g. graphical interface editor)
- **Eclipse (http://www.eclipse.org/) + Mobile Tools for Java (http://www.eclipse.org/dsdp/mtj)**
  - Maybe better for developers who are already familiar with Eclipse

Exercise 1

- **Introduction to basics of Java ME**
- **Introduction to the Mobile Media API (JSR135) and sensors**
  - Specification of an audio and video API for mobile devices
  - Optional specification => not supported by all mobile phones
- **Task:**
  - Create a camera application for taking pictures and displaying them

  or
  - Create a recording application for recording and playing sounds with the mobile phone's microphone
- **Attention:**
  - Capturing should always run in a dedicated Thread!!

Exercise 1 - II

Solution:

- Exercise sheet and material available on the PEM-website
- Midlet must work on a mobile phone (emulator not enough)
- Mobile phones can be lent from the university (Alexander De Luca, 5th floor)

Submission:

- Each student must submit his/her own solution via email to gregor.broll@ifi.lmu.de and alexander.de.luca@ifi.lmu.de by Monday, October 29th, 12 p.m.
- Create a zip-file named after you and insert a folder called *exercise1* containing your solution

Questions?

Have fun!