

CmpE 473

Internet Programming

Pınar Yolum
pyolum@cmpe.boun.edu.tr

Department of
Computer Engineering
Boğaziçi University

Chapter 13

Mobile Java

Mobile Internet

- Limitations of mobile devices
 - Small memory size
 - Frequent disconnects
 - Small displays and keyboards
 - Slow transmission
 - Low processing power
- Goal-oriented activities
 - Reserving tickets (Time constraints)
 - Web search (Precision constraints)

J2ME

- Java 2 Micro Edition
- Designed for mobile phones, PDAs, TVs, embedded devices
- Applications exploit devices' native capabilities
- Supports on-demand applications
- Contains configurations, profiles, and optional packages

J2ME Architecture (1)

- Configurations
 - Virtual machine and minimal set of class libraries
 - Contains minimal similar characteristics
 - Network connectivity
- Current configurations
 - Connected Limited Device Configuration (CLDC)
 - Connected Device Configuration (CDC)

J2ME Architecture (2)

- Profiles
 - Higher-level APIs that build on the configuration
 - Contain application life-cycle, user interface, device-specific properties
 - Cell phones generally have CLDC+ Mobile Information Device Profile (MIDP)
- Optional packages
 - Database connectivity
 - Wireless messaging
 - Web services

CLDC (1)

- Java language and virtual machine (subset of J2SE libraries)
 - Core Java libraries (java.lang.*, java.util.*)
 - Input and output (java.io.*)
 - Limited exception handling
 - No finalization of class instances
 - No daemon threads
 - No user-defined class loaders
- CLDC specific
 - Inside javax.microedition

CLDC (2)

- Supports dynamic downloading of Java applications
- Must read .class and .jar files
- Internationalization
 - Must support Unicode characters
- Networking
 - Network protocol implementations are not defined
 - Deferred to the profile layer

CLDC Security

- Low-level security
 - Malformed or malicious code does not crash the host device
 - Class file verifier checks bytecodes for malicious instructions
- Application-level
 - Access to resources (files or libraries) that it has privileges for
 - Required for protecting system classes (javax.microedition.*) and private data
- End-to-end
 - Encrypted path from host to destination
 - Not supported by CLDC

MIDP (1)

- Hardware Requirements (Minimum)
 - Screen-size: 96*54
 - (One-handed or two-handed) Keyboard or touch-screen
 - Memory
 - 128 KB non-volatile memory for MIDP
 - 8 KB non-volatile memory application created data
 - 32 KB JVM
 - Networking
 - Two-way, wireless

MIDP (2)

- Software Requirements (Minimum)
 - Kernel to manage the HW (should be able to run JVM)
 - No need to support processes (different address space)
 - Support read-write access to persistent storage
 - Timestamp for write
 - Support send and get messages
 - Capability to write a bit-mapped display
 - Capability to get input
 - Capability to manage application life-cycles

MIDP (3)

- Scope (APIs limited to those needed to ensure mobility)
 - Application (Control and execution)
 - User interfaces (Display and input)
 - Persistent storage
 - Networking
 - Timers
- Out-of-scope
 - System-level APIs: Power management or voice support
 - Application delivery: Application installation; storage; no guarantees for a file system
 - Security: Nothing more than the CLDC

Application Model

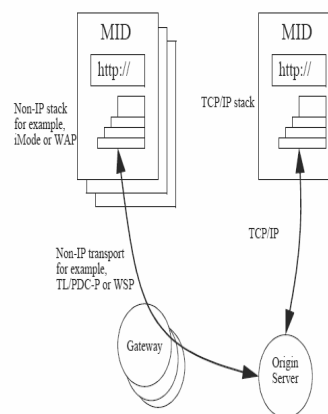
- Allow resources to be shared among MIDP applications (called MIDlets)
- MIDlet Suite Package
 - A manifest for the contents
 - Used by the application management software
 - Describes version, installation, data size, and so on
 - Public and private .class files
 - Resource files
- Each MIDlet extends the *MIDlet* class
 - Provides application lifecycle operations (for CLDC to start a MIDlet)

Spring 2005— Pinar Yolum

13

Networking

- Subset of HTTP protocol
- Implemented with TCP/IP or with non-IP protocols such as WAP and i-mode
- Must provide support to access HTTP servers



Spring 2005— Pinar Yolum

14

Persistent Storage (1)

- Record Management System (RMS)
- Manage consistency of the records
 - Reboots, battery changes, disconnected transactions
- Record Stores
 - Contain records that will stay persistent
 - Unique record store names
 - Multiple record stores allowed
- No locking operation

Persistent Storage (2)

- All record store operations are
 - Atomic
 - Synchronous
 - Serialized
- What happens if multiple threads access the same record?
- Records
 - Array of bytes
 - Uniquely identified based on the time of insertion

User Interface (1)

- AWT cannot be used as is
 - AWT relies on garbage collection (e.g., with events)
 - AWT supports overlapping windows; window resizing, etc.
 - AWT relies on pointer device
- MIDP UI API designed for MIDs
 - High-level (Abstract, portable)
 - Low-level (Concrete)

User Interface (2)

- Main abstraction: Screen
 - Predefined screen structures (e.g., List, Textbox)
 - Generic screens that applications build (e.g., Form)
- Controlled by the application manager
 - startApp: Create first screen
 - pauseApp: Pause threads
 - destroyApp: Kills the application

Event Handling

- High-level events for abstract commands
 - Employ unicast listeners for screens
 - Not all changes reported immediately (but before a second change)
- Low-level events for key presses and releases
 - Can query if *keyPressed* (*int keyCode*); and *keyReleased* (*int keyCode*)
 - Requires standard key code, ITU-T keypad: 0—9, *, #

M-Commerce

- Transactions initiated through a mobile host
- NTT DoCoMo's i-Mode (Launched in 1999)
 - Services provided by 3rd party providers (roughly 1000 content providers)
 - Service types include e-mail, ticket reservation, banking
 - Roughly 30 million users in 2001
 - Small monthly fee + traffic charges + offers billing services for service providers

Usability

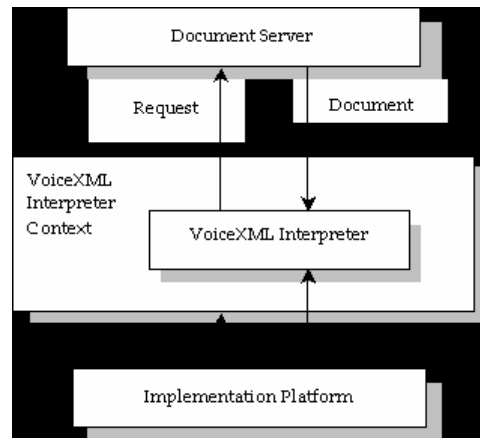
- Difficult to view and browse through too much content
- Content delivered slowly
- Examples:
 - Web search returns hundreds of pages
 - Amazon.com browsing yields many book titles
- Partial solution: Personalization
 - Model the user accurately
 - Display only part of the content that is relevant

Usability

- Use voice for input and output
- Enable speech recognition and synthesis for input
- Example
 - AOLbyPhone: Reads e-mail over the phone
 - Shop online with voice commands
- Underlying standard VoiceXML

VoiceXML

- Document server processes customer requests and generates VoiceXML documents
- VoiceXML interpreter sends necessary commands to the implementation platform



VoiceXML

```
<?xml version="1.0" encoding="UTF-8"?>
<vxml xmlns="http://www.w3.org/2001/vxml"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.w3.org/2001/vxml
  http://www.w3.org/TR/voicexml20/vxml.xsd" version="2.0">
  <form>
    <field name="drink">
      <prompt>Would you like coffee, tea, milk, or nothing?</prompt>
      <grammar src="drink.grxml" type="application/srgs+xml"/>
    </field>
    <block> <submit next="http://www.drink.example.com/drink2.asp"/>
    </block>
  </form>
</vxml>
```

Example from w3c.org

Business Models (1)

- Quick transactions
 - Checking bank balance
- Urgent transactions
 - Cancel a flight
 - Find driving directions
- Location-based transactions
 - Find nearest Chinese restaurant
- Context-based transactions
 - Buy a game when stuck in traffic
 - Dial a Coke (conducted at Helsinki airport)

Business Models (2)

- Mobile Enterprise Resource Planning (m-ERP)
- Mobile Customer Relationship Management
- Mobile Health Care
- Mobile Tracking

Enabling Technologies

- **Security and Privacy**
 - Guaranteeing end-to-end security
 - Protecting privacy of users
 - Most mobile carrier services can locate your phone
 - Target advertisements based on your location?
- **Supporting Services**
 - Methods for common billing for various services
 - Methods for easy payments