

Instant Messenger on Mobile Phone

- EECS 298 Project Report

EECS Dept, UCI
Hui Liu #72887237
Mingjie Lai #19429781

Dec 10. 2004

Outline

- **Motivations**
- **Related Technologies**
 - Mobile Networks
 - Mobile Terminal Platforms
 - Sony Ericsson T610
- **Design Description**
 - System Context View
 - Software Architecture
 - Design Consideration
- **Demo**
- **Conclusions**

Motivations

■ Why Mobile phone

- Typical Embedded, Mobile Device
- Mobile phone application platform: J2ME, CLDC, MIDP
- Mobile networks technology: GSM, GPRS
- Develop application on Embedded Platform

■ Why IM?

- Logic is Simple
- Networked Application on Heterogeneous Env.
- User Interactive

Networks Tech Intro

■ GSM

- TMDA

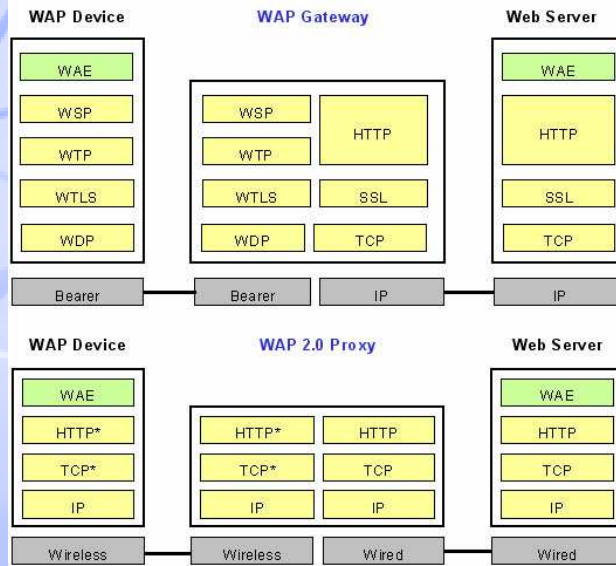
■ *Wireless Application Protocol, WAP*

- an application communication protocol
- used to access services and information for handheld devices such as mobile phones
- a protocol designed for micro browsers
- inherited from Internet standards
- uses the mark-up language WML (not HTML)

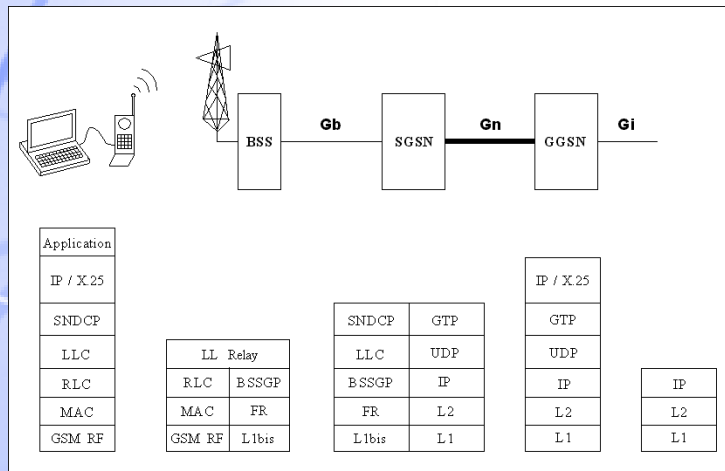
■ *General Packet Radio Service, GPRS*

- A radio technology for GSM networks that adds packet switching protocols.

Networks Tech Intro - WAP



Networks Tech Intro - GPRS



Mobile Terminal Platforms

■ J2ME

- For tight memory, processor-challenged devices
- *Connected Limited Device Configuration, CLDC*
- *Mobile Information Device Profile, MIDP*

■ Others

- BREW
- Symbian OS
- Windows CE

Mobile Terminal Platforms, cont.

■ Why defining other concepts, as configuration, profile In J2ME

- Configuration: for different kind of device
- Profile: more specific to CLDC

■ CLDC

- Defines the VM as KVM, for kilobytes memory devices

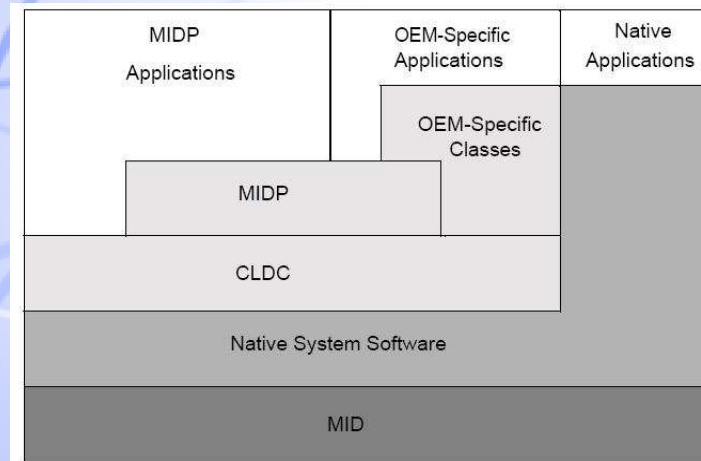
■ MIDP

- In addition to the CLDC, interfaces for accessing UI, network access, local data storage, etc.

■ Others in J2ME

- Consumer and Embedded Device Technologies, CDC, for network-connected consumer and embedded devices
- Foundation Profile, Personal Basis Profile, Personal Profile

Mobile Terminal Platforms, cont.



MIDP / CLDC Structure

MIDP Programming Introduction

- **Application Lifecycle**
 - Extends the abstract class *MIDlet*
 - Over-rides Method: *startApp*, *pauseApp* and *destroyApp*
- **Networks Interface**
 - *Http API* only
- **User Interface**
- **Event Handling**
 - *Callback*
- **Thread**

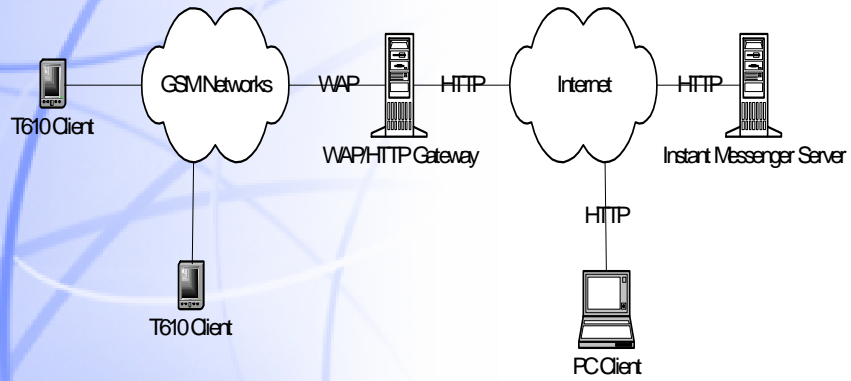
HW Spec: T610

Characteristic	Value
Network	GSM Triband 900/1800/1900
Infrared / Bluetooth	Yes / Yes
Infrared API	No
Java Bluetooth APIs	No
Display	128x160 pixels, 65,535 colors, STN
Memory	2MB dynamic (for the user) memory
Java Heap	256 KB
Native UI Heap	150 KB
Native Video Ram	80 KB
JAVA	J2ME/CLDC 1.0 /MIDP 1.0
WAP	2.0
GPRS	No

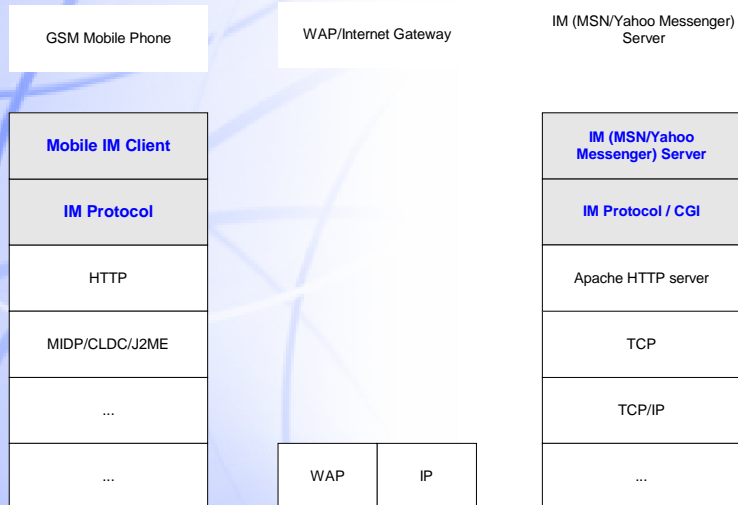
Design Description

- System Context View
- Software Architecture
- Design Considerations

System Context View



Software Architecture



Software Architecture: Cont.

■ IM Client

- Written in Java
- Running on MIDP 1.0
- Sony Ericsson T610

■ IM Server

- CGI perl program running on Apache http server

■ IM Protocol

- Defines the basic interaction/behavior between IM Client / Server
- Send/Retrieve (No data push)

Design Consideration 1

■ Hardware Limitation

- 2MB storage
- 256KB Java heap; 150KB UI Heap; 80KB Video Ram

■ Solutions

- Use Inheritance instead of creating new classes: Avoid duplication of available code and minimize program size.
- Be careful when using threads
 - Overhead of Context switch is expensive
 - Synch to guarantee thread-safe is also expensive
- Avoid the use of setter/getter methods, define data member as public.
- Avoiding Memory Fragmentation: memory holes
- Use short Class/method/package names

Design Consideration 2

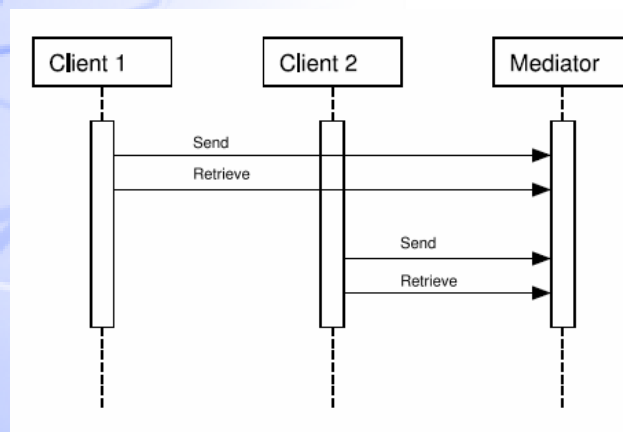
■ Networks Operation Problems

- Slow network connections.
- MIDP 1.0 Only provides HTTP API
- HTTP is not a connection-oriented protocol
- Could not set up peer to peer connection in the IM

■ Solutions

- Introduce a Mediator to allow clients 'talk' to each other
- Implements the Mediator using CGI perl on top of apache HTTP server
- Messages are polled from Mediator

Design Consideration 2, cont.



Design Consideration 3

■ IM protocol

- Most current IM as MSN etc have their specific protocol, and are not open
- Can't be implemented with HTTP
- Too complex

■ Solutions

- Define our specific protocol
- No authorization
- No client status as on-line/off-line
- Chat room

Demo

Conclusions

- **Mobile is getting powerful**
 - Can easily develop complex applications
- **Platform Tech is getting mature**
 - A lot of platforms: J2ME, others: Symbian, BREW
 - Very easy to develop applications on top of them
 - Not easy to debug
- **Current mobile networks can provide data communication service**

Questions & Answer (if possible)