

Department of Management and  
Technology

# Software Comprehension and Maintenance

Wireless IRC project  
Design of a new feature

# Wireless Irc's Design Presentation

- Brief explanation of Midlet Suites function
- Presentation of the tools used to build and run WRLIrc.
- Description of the program's packages and classes .
- Design of the autocomplete command and communication with the developer.

# WRLIrc is a Midlet Suite

- **Midlets** are **small** programs that are written in order to be used in **mobile Devices** .  
Midlets are build by **JVM** and are run and controlled by the **Application management Software** . The application management software is part of the mobile device's software that deals with creating, starting, pausing and deleting Midlets and allocating resources among the Midlets.It is responsible for managing the activities of multiple Midlets within a runtime Environment.
- What makes them different is the fact that they can exploit efficiently and **share the limited resources of a mobile device** and that they can conform to the security framework of the device.  
Midlets are grouped together in packages called Midlet Suites.

# WRLIrc is a Midlet Suite

A Midlet Suite includes:

- A **JAR** File that may include one or more **Midlets** , some other classes ,the resources ( text, image ,properties ,database and utf files) and a **manifest** File .
- A **JAD** (Java Application Descriptor ) File that includes all the information needed by the Application Management Software to **identify, retrieve and install** midlets.  
(Such information may be the name and the version of the application, the name of the vendor and a digital signature) .  
The descriptor allows the application management software on the device to **verify** that the MIDlet is **suited** to the device before loading the full JAR file of the MIDlet suite.

# J2ME Wireless Toolkit and Net and NetBeans Mobile

- In order to run a Midlet in a computer we need an emulator. An emulator is software that simulates the execution of the application on various mobile devices.
- J2ME Wireless Toolkit includes an emulator and a Development Environment that can be used to prepare class and jar Files and directories and to run class Files. As result, it simulates the whole procedure of executing Over The Air an application from a server to a device.

# WRLIrc Packages

The basic packages imported in WRLIrc are:

- `javax.microedition.midlet`
- `javax.microedition.lcdui`
- `javax.microedition.io`

# javax.microedition.midlet

A MIDlet is an abstract class. Every application must extend this class to allow the application management software to **control** it and to be able to **retrieve properties** from the **application descriptor** and notify and request state changes.

The methods of this class allow the application management software to **create, start, pause, and destroy a MIDlet.**

**Specifically a Midlet contains methods that:**

- Signal the MIDlet to enter a specific state (start,pause,destroy) or notify the application software that the Midlet has entered a state.
- Provide the MIDlet with a mechanism to retrieve named properties from the application management software.
- Requests that the device displays or installs the indicated URL.

# javax.microedition.lcdui

This package manages the user interface. It includes:

- A Displayable object that shows one or more components in the screen. Only one Displayable may be visible at a time, and the user can see and interact with only contents of that Displayable
- The class Display that acts as the display manager that is instantiated for each active MIDlet and provides methods to retrieve information about the device's display capabilities. A Displayable is made visible by calling the setCurrent() method of Display.
- A variety of components that can be displayed
  - List is used when the user should select from a predefined set of choices.
  - TextBox is used when asking textual input.
  - Alert is used to display temporary messages containing text and images.
  - A special class Form is defined for cases where screens with a predefined structure are not sufficient.



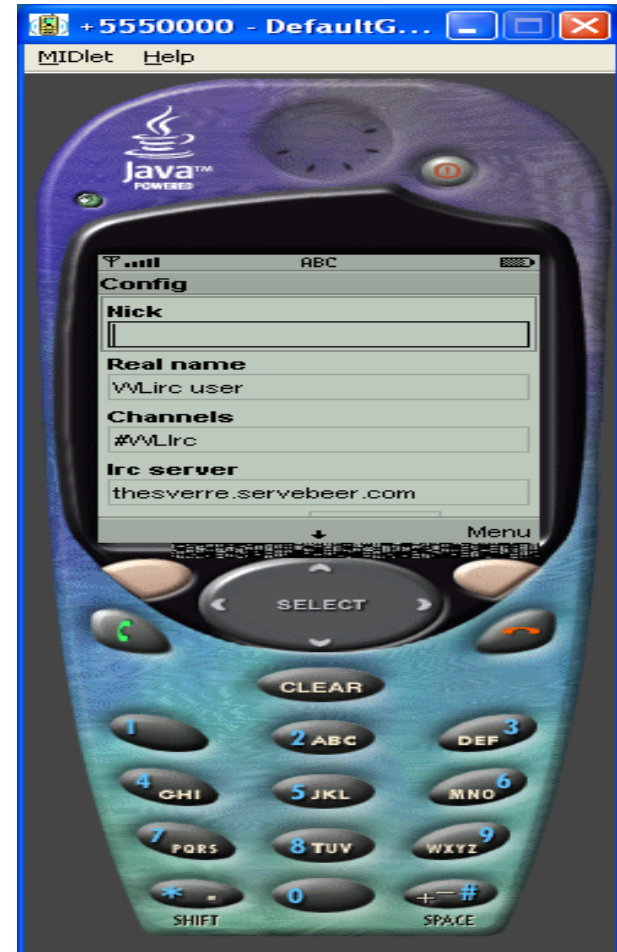
# WRLIrc's class Diagram

- **Class Hierarchy**
- java.lang.Object
  - Database
  - javax.microedition.lcdui.Displayable
    - javax.microedition.lcdui.Canvas
      - DisplayCanvas
  - Encoding
  - Listener (implements java.lang.Runnable)
  - Message
  - javax.microedition.midlet.MIDlet
    - WLIrc (implements javax.microedition.lcdui.CommandListener)
  - PollHttpIrc (implements Irc)
  - ScreenOutput (implements javax.microedition.lcdui.CommandListener)
    - Channel
    - Console
    - Private
  - ScreenOutput.ListListener
  - ScreenOutput.TextboxListener
  - SocketIrc (implements Irc)

# How does WRLirc work

- WRLirc's main classes are WRLirc.java and Screenoutput.java.
- When the Midlet is launched WRLirc.java is the first class to be executed. It displays on the screen the name of the project and a menu with the commands (connect, config, advanced, font, exit) and defines the results of the interaction with the user. If the user selects the connect command the class PollHttpIrc **will open a new connection.**
- If the user selects the config command the class WRLirc **will display a new form with some Textfields that are used for entering data.**

# How does WRLirc work



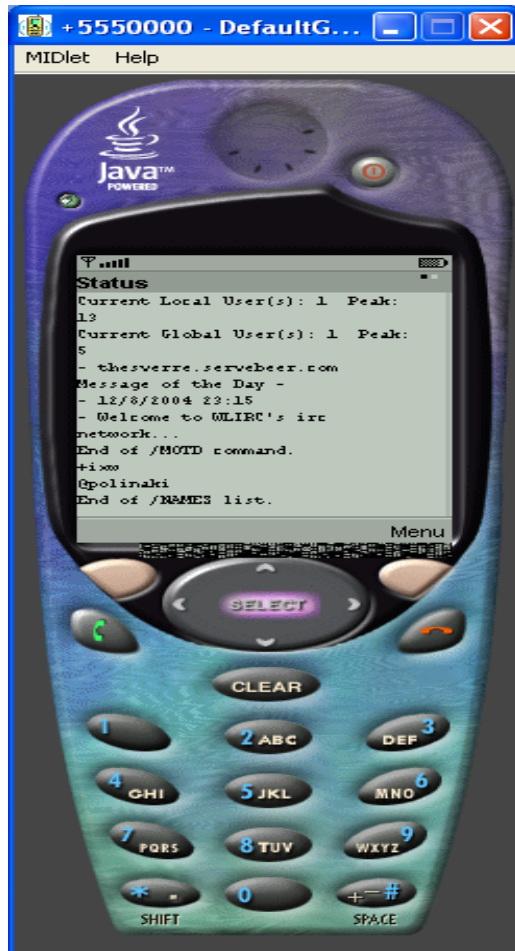
# How does WRLIrc work

- Class database. It is used to save all the data that must be reused when the application is closed and then reopened.
- It uses the class RecordStore. A record store consists of a collection of records which will remain persistent across multiple invocations of the MIDlet. The platform is responsible for making its best effort to maintain the integrity of the MIDlet's record stores throughout the normal use of the platform, including reboots, battery changes, etc.
- Record stores are created in platform-dependent locations, which are not exposed to the MIDlets. Records are saved in properties files that can be found in the resources directory.

# Design of adding the autocomplete command

- A vector called messages will be used to store the users' favourite messages.
- The menu displayed under the message textbox will have three more commands:
  - save,
  - delete,
  - autocomplete

# How does WRLIrc work



18/04/2005



Tzavala Polina

# Design of adding the autocomplete command

- The command save will add the content of the textbox as a new element to vector messages and will call the Database save method. This method replaces the third record of the Record Store with the vector messages. Before this happens the vector messages should be converted into a byte[]array.

# Design of adding the autocomplete command

- The autocomplete command will compare all the elements of the vector messages to the string entered into the textbox. Then it will display a list with all the messages that start with this String.



# Design problems and solutions

- I have tried to save messages in two different ways: by adding a new record to the RecordStore and by adding the vector to the existing record. When I tried to close and reopen the device ,all the records (even the configuration data such as the nickname) where destroyed .I have tried to delete the database file where the records are stored. Then the autocomplete command worked correctly.

# Communication with the developer

- The developer, Mr Sverre has suggested improving the autocomplete command so that it displays a list with all the saved messages. Then the list will decrease to the messages that match the String entered by the user. What's more he has suggested the following ToDo List:
  - make a historylist, like mirc's arrow up, arrow down.
  - - timestamps
  - - ignore list
  - - make the codebase smaller. Remove the textboxlistener class
  - - implement flood control
  - - nick list (move back, next button to a command instead of an element)