

Testing of MIDlet based, Adaptive, Context Aware Applications

Michele Sama

University College London

Dept. of Computer Science

Software Systems Engineering Group

Outline

- Introducing AdaptiveReminder
- Issues in running testing MIDlet
- Introducing TestingEmulator
- Emulator Demo
- AdaptiveReminder Demo
- Future works



Case study: Reminder for a meeting

- A few minutes before:
 - If the user is not in the meeting room.
 - If the user is not near his boss.
- Remind him to go to the meeting:
 - A silent reminder because he is in his office.
- Long before:
 - If the user is far from the building.
 - If his boss is not next to him.
- Remind him to move close to the meeting:
 - An invasive reminder because he is not in his office.

Formal specification

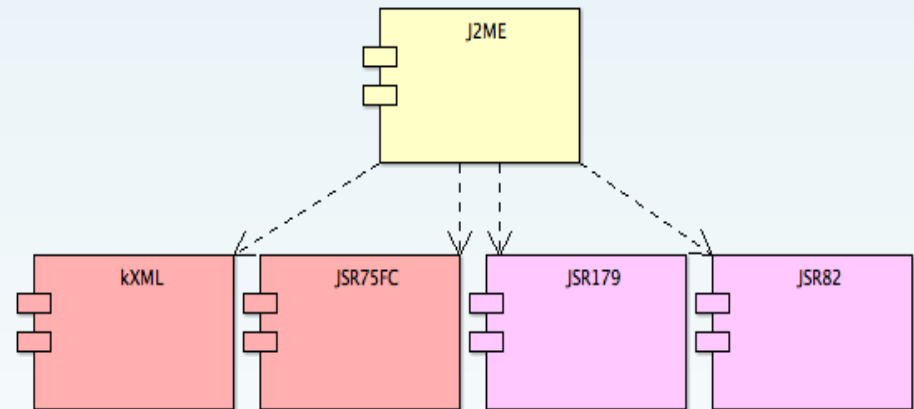
- Each condition can be represented with a tree of **Constraints** (a **Rule**).
- Each constraint is related to a specific query to a single **Context Condition**.
- Each context variable is monitored by a specific **Handler**.
- Each Handler notify all the **registered** Constraints in case of some context change.

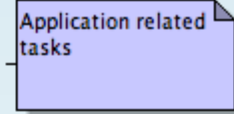
From a Sample to a Framework

- A constraint based system for event activation.
- A decision engine for Activities and Notification activation.
- Dynamic **allocation** and **deallocation** of ContextHandlers
- Dynamic **registration** and **deregistration** of Constraints

From Abstract to Technology

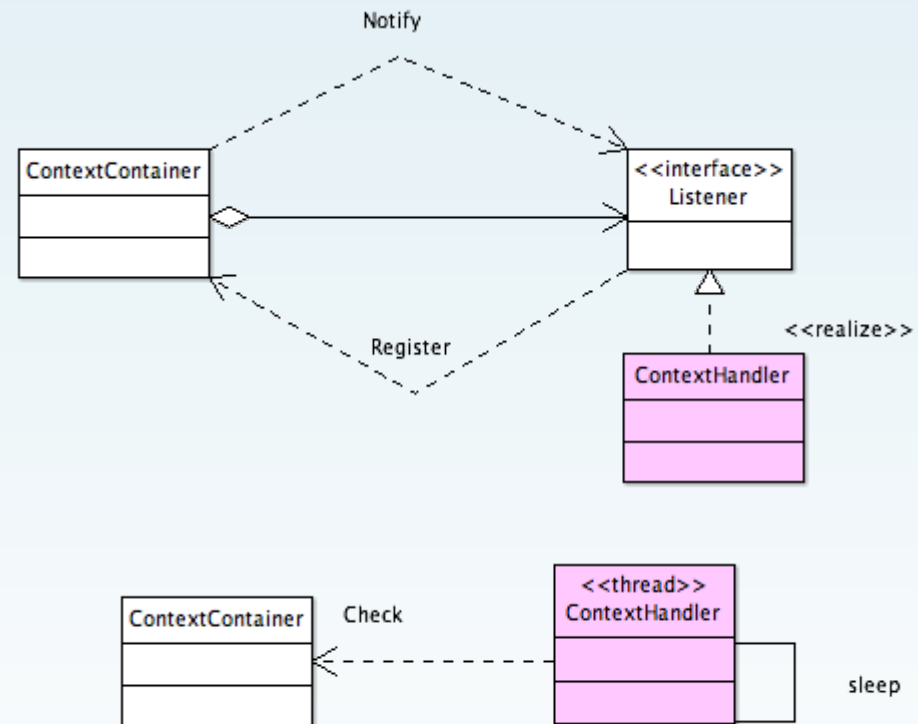
- J2ME
 - MIDlet
 - JSR82 Bluetooth
 - JSR179 Location
 - JSR75FC File connection
- XML
 - A file for each Activity
 - <http://kxml.sourceforge.net/>





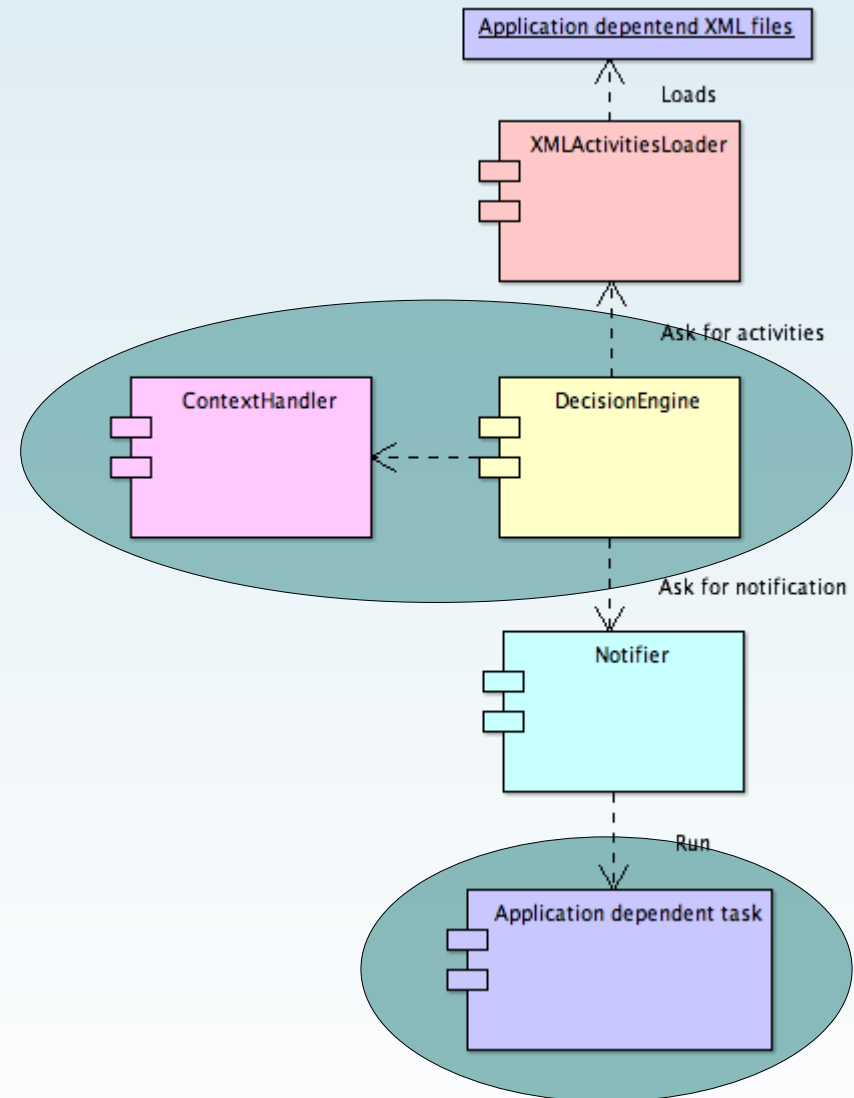
ContextHandlers implementation

- **Passive Handlers**
 - They receive notification from an event by the J2ME framework
- **Active Handlers**
 - Thread based, they refresh their status with a loop



Adaptation

- The evolution of the internal status of the application is related to the **activation flow** of the loaded Activities.
- The activation of loaded Rules start some **application components**.



Future Works

- Constraints based Rule life cycle:
 - A constraints tree for Rule **activation**.
 - A constraints tree for Rule **validation** (now).
 - A constraints tree for Rule **deactivation**.
- Use Notifiers as Activity loaders:
 - Activities can **load each others**;
 - **Pre and Post** conditional Activities;
 - **Flow** of activities.

Issues in developing, running and testing

- J2ME support for some context variables is **missing**
 - Signal level
 - Battery status
 - Connected networks
- Testability
 - Real device with **uncomfortable** support
 - Certificates
 - Available libraries
 - Emulator with no control of the context
- Impossibility to create a proper test

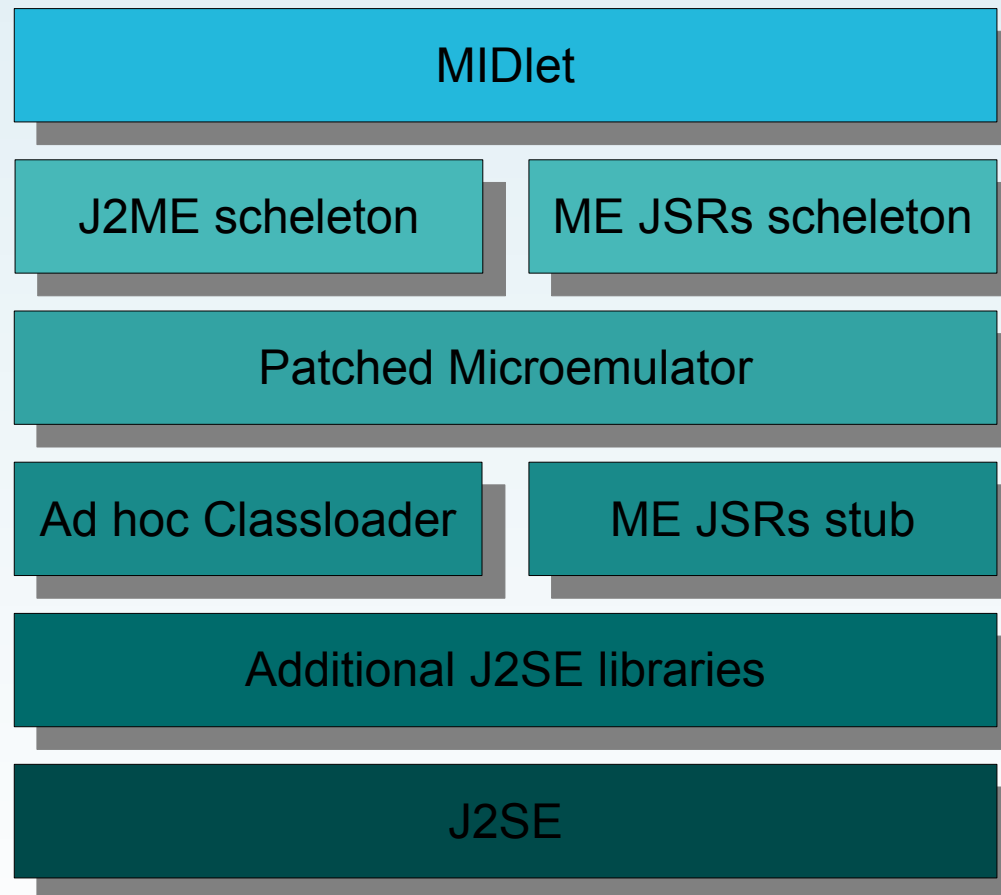
What do we need?

- From a device:
 - Access to context parameters.
 - Modify to security level
 - Support for standard framework libraries
(Device's implementation is partial or missing)
- From an emulator:
 - Possibility to **manipulate the environment**.
 - Execution of **testing script** not inside the application itself.
 - Direct access to application variables.
- An Emulator for testing!

The idea: Running J2ME over J2SE

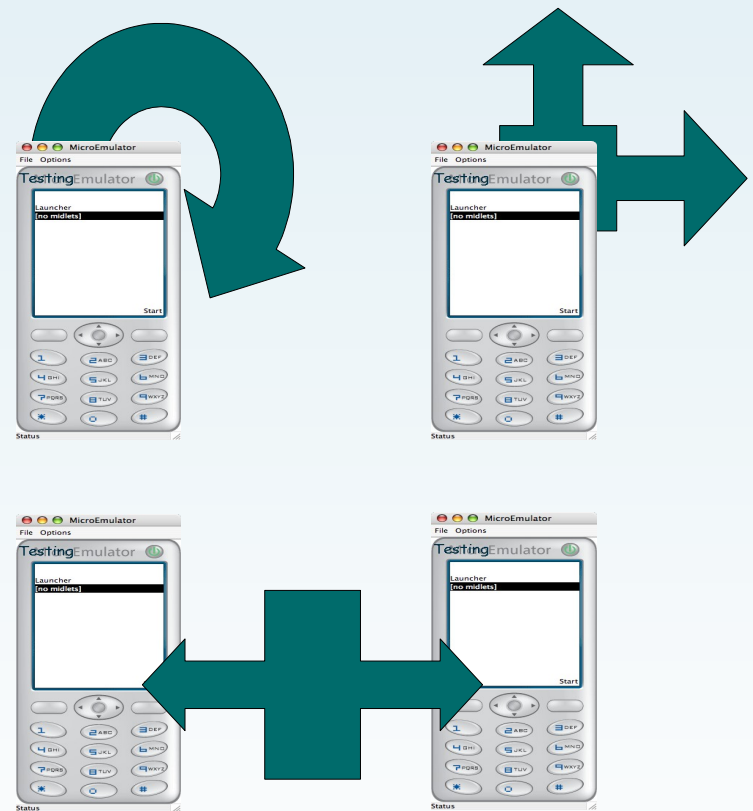
- J2ME is a **subset** of J2SE
 - CLDC classes provide a subset of the methods of the same classes of J2SE
 - Compile and preverify with J2ME but **run** with J2SE
 - The same virtual machine but different Classloaders
- J2ME provides additional device related libraries
 - A stub implementation with JavaBeans allows the emulator to **modify the status** of the emulated context

Execution stack



Running and testing capabilities

- 3 Possibilities:
 - Standalone with emulated context
 - Standalone using real devices of the host
 - Networked testing the cross-evolution of the application in different devices



Emulating devices without specific JSRs


- Possibility to choose:
 - Which library to load.
 - Which implementation of the same library.
- Emulation of device with some missing support.



- ☒ JSR82
- ☒ JSR75PIM
- ☒ JSR75FC
- ☒ JSR179

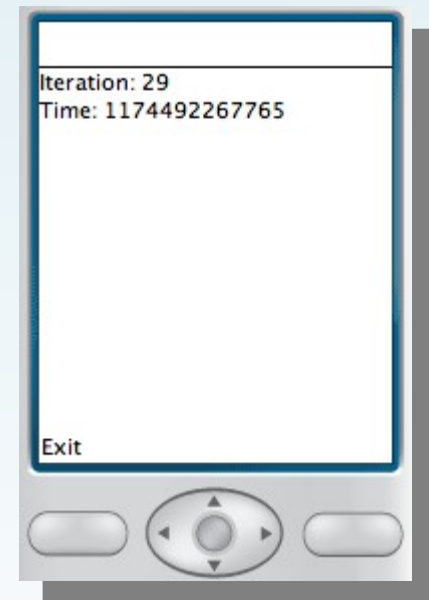
Emulating device time

- Time as a context
 - What happen when two different time based, collaborating applications run in devices with different times?
 - We need to:
 - Set the time
 - Stop the time
 - Handle delays

- 
- ☒ Time normal
 - ☐ Time stopped
 - ☐ Time stopped and delays handled
 - ☐ Time warped

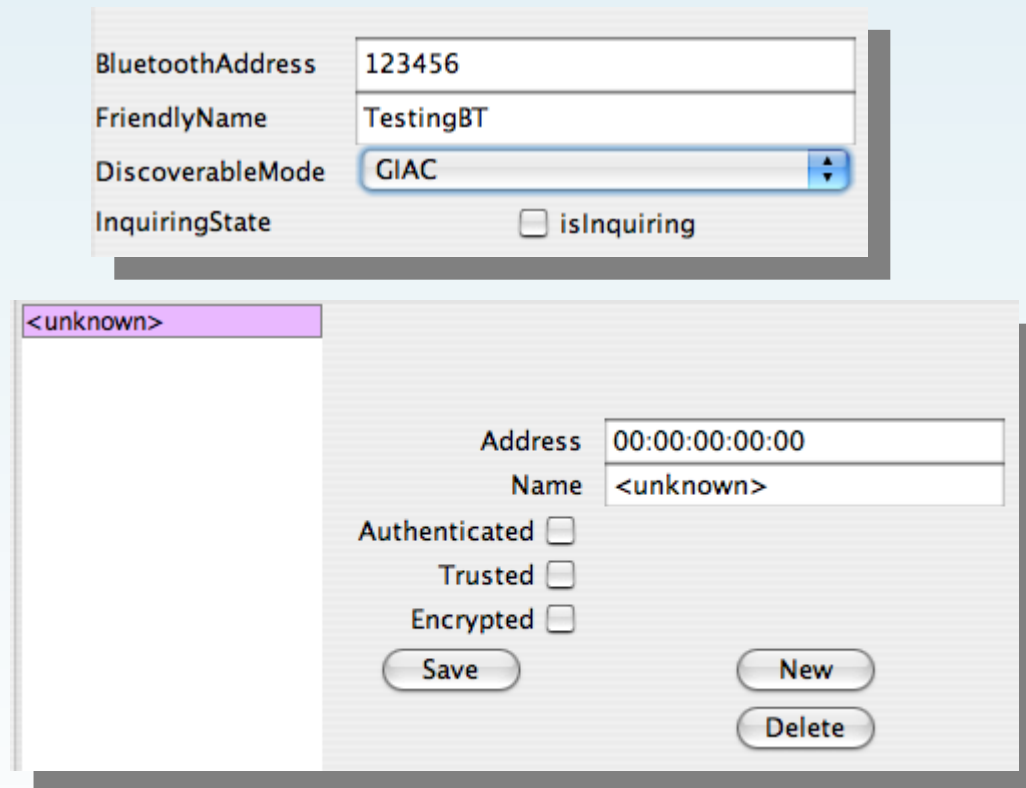
Time and Delays Demo

- DelayTest.jar
 - A Thread sleeps for 1000 msec and refresh the GUI with the new time.
 - Time is loaded from `System.currentTimeMillis()`
 - Delays are created with `Thread.sleep(msec)`



Emulating Bluetooth

- Bluetooth
 - JavaBeans implementation of the device
 - Runtime configuration of the Bluetooth environment



The screenshot shows a JavaBeans configuration window for Bluetooth emulation. It contains the following fields and controls:

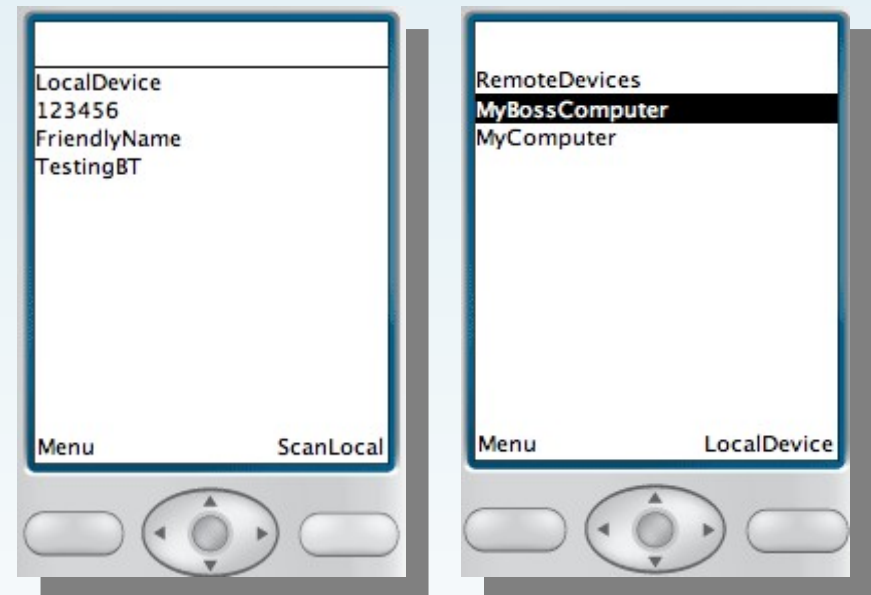
- BluetoothAddress:** A text field containing the value "123456".
- FriendlyName:** A text field containing the value "TestingBT".
- DiscoverableMode:** A dropdown menu currently set to "GIAC".
- InquiringState:** A checkbox labeled "isInquiring" which is currently unchecked.

Below these fields is a list of discovered devices. The first device is highlighted with a purple background and labeled "<unknown>". To the right of this list is a detailed configuration panel for the selected device:

- Address:** A text field containing "00:00:00:00:00".
- Name:** A text field containing "<unknown>".
- Authenticated:** A checkbox, currently unchecked.
- Trusted:** A checkbox, currently unchecked.
- Encrypted:** A checkbox, currently unchecked.
- Buttons:** "Save", "New", and "Delete" buttons are located at the bottom right of the panel.

LocalDevice and DiscoveryAgent Demo

- LocalDeviceScanner.jar
 - Displays local device attributes.
 - Scans for discovered remote devices.



Emulating a FileSystem for the device

- JSR75FC
 - DeviceRoots mapped into host directories
 - Runtime set-up of roots
 - Wrapping of JSR75FC file access with J2SE java.io.

RealPath	<input type="text" value="/Users/rax/j2meFakeRoot"/>	<input data-bbox="1651 714 1796 749" type="button" value="..."/>
LogicalPath	<input type="text" value="/NewRoot0"/>	
<input data-bbox="956 806 1091 849" type="button" value="Save"/>		<input data-bbox="1651 806 1796 849" type="button" value="Remove"/>
		<input data-bbox="1651 863 1796 906" type="button" value="Add"/>

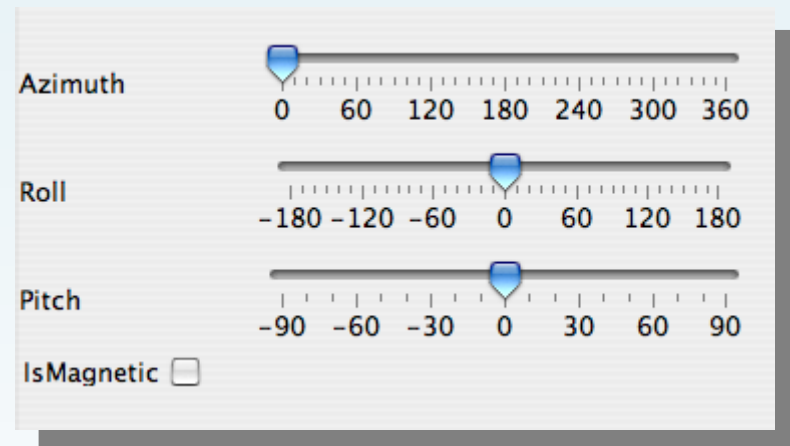
FileSystem Demo

- JSR75fcDemo.jar
 - Shows all the accessible device roots.
 - Browse file and folders.



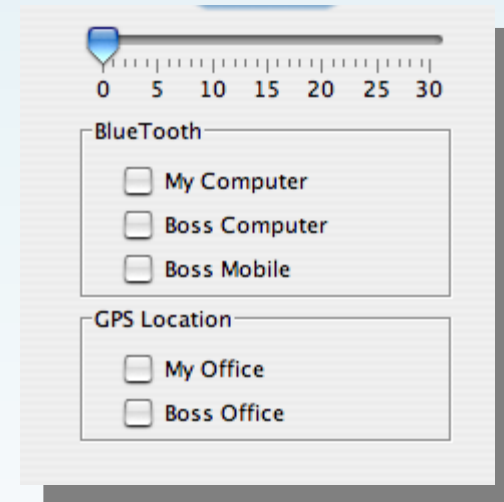
Emulating discovery of Location

- JavaBeans implementation
 - Device Orientation
 - Current Location
 - Localization method
 - LandmarkStores



Support for Scripting

- Support for dynamic loading of script with Reflection
- Access to all the context variables and JavaBeans of the emulator.

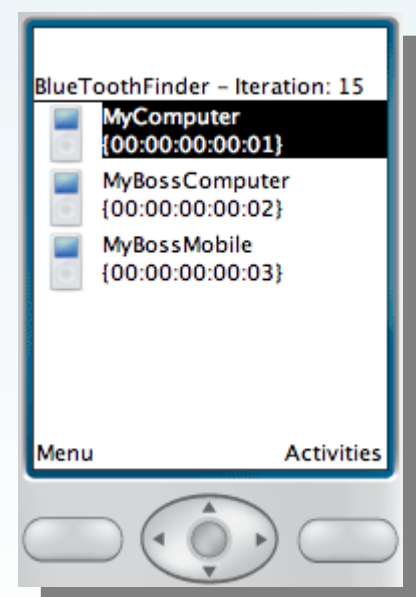
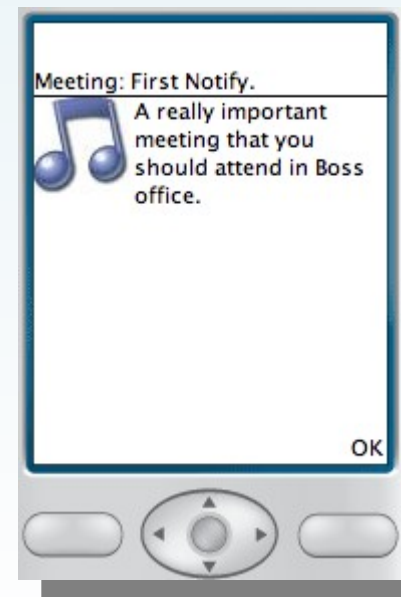
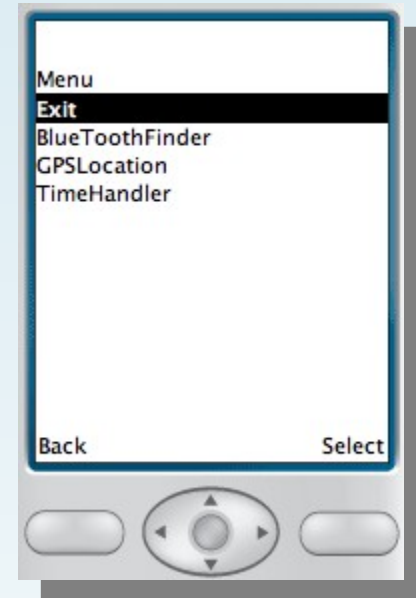
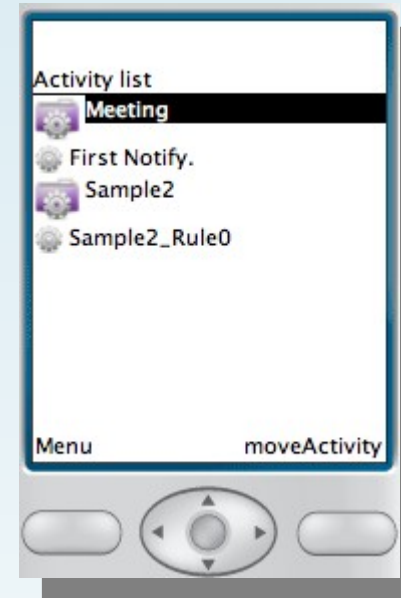


State of the art.

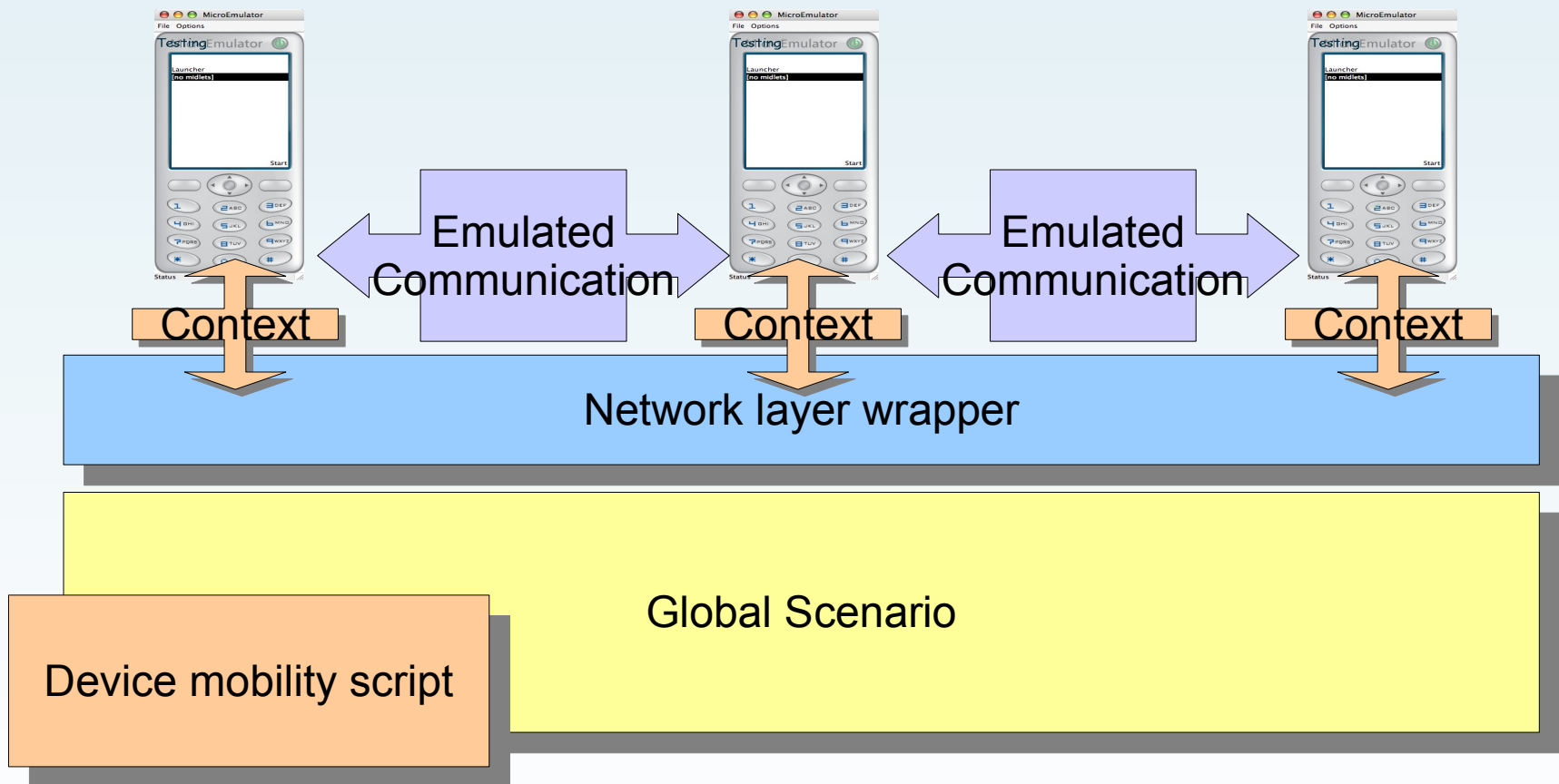
- To do
 - UEI Universal Emulator Interface
 - Adding missing JSR
 - Profiling
 - “Networking”
- Alternatives
 - PhoneME
 - Same stack of the device.
 - Open source.
 - Closed context.
 - WTK2.0
 - deprecated.
 - MPowerPlayer
 - Based on Microemulator.
 - Restricted support.

AdaptiveReminder Demo

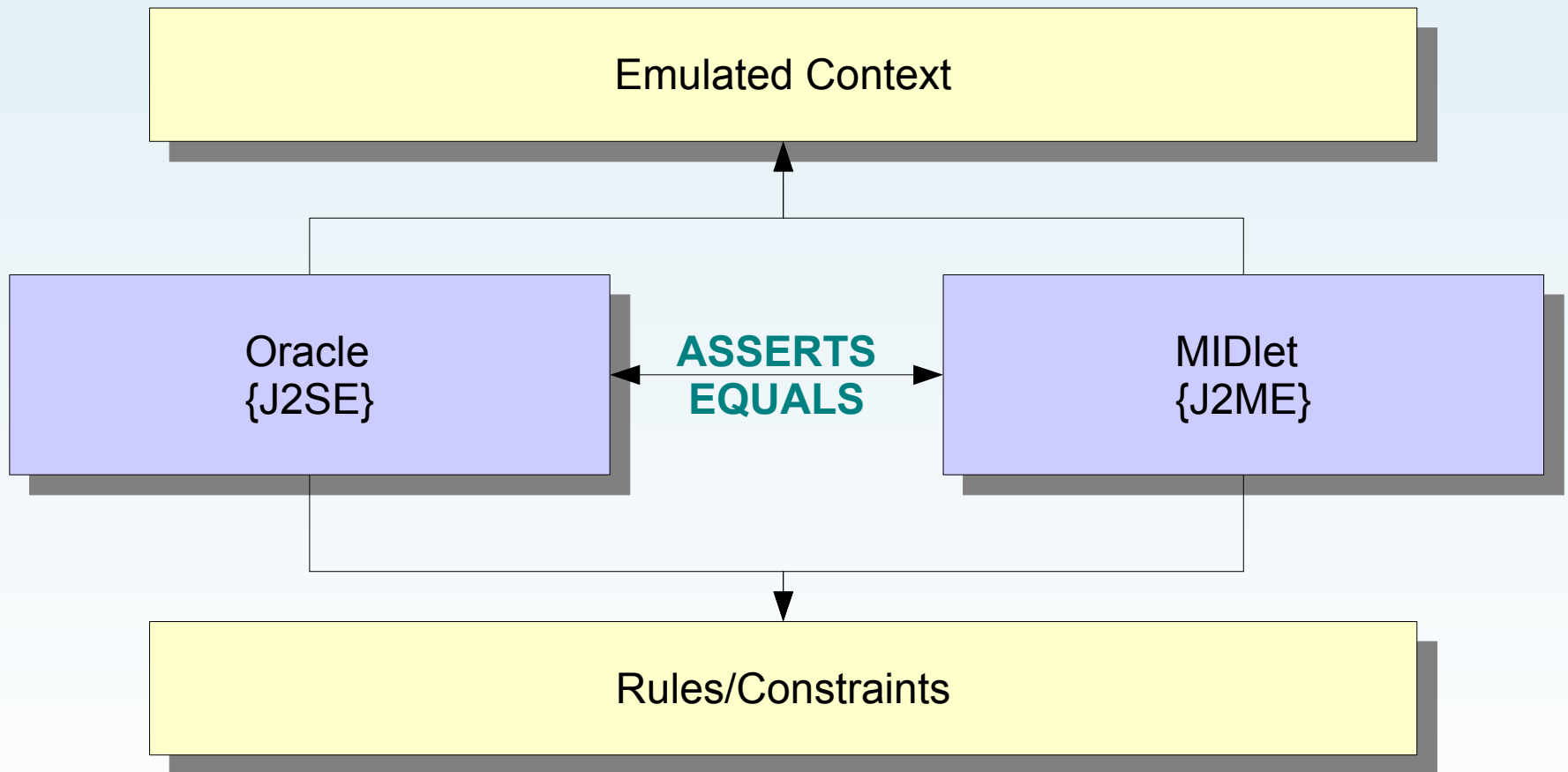
- AdaptiveReminder
 - Notification.
 - Garbage collecting.
 - Meeting.



Networking concept

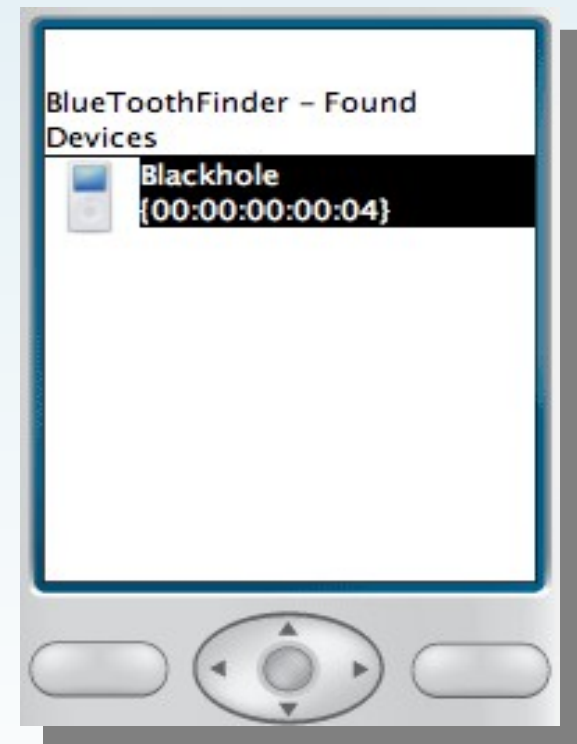


Oracle concept when testing J2ME with J2SE



Found and corrected failures: Frontend dispose

- After a notification a disposed Frontend is still displayed
 - All the reference to that class are removed but the component is still displayed.
 - The failure occurs because of a reference in the Display framework object.



How to contribute

- AdaptiveReminder
 - Creating more specific context handlers
 - Creating other case studies
 - Implementing an Oracle
- TestingEmulator
 - Adding missing JSR
 - JSR75PIM
 - Improve existing implementation of JSR
 - Network support for Mobility Script Support