EclipseWorld 2006

Eclipse on Cell phones!?
An Introduction to eRCP

Chris Aniszczyk <zx@us.ibm.com>
Mark Rogalski <rogalski@us.ibm.com>

Agenda

- **Demos!!!** (5-10 mins)
- **Background** (10 mins)
  - What is eRCP?
  - Community
- **Architecture** (15-20 mins)
- **Coding Exercises!** (30 mins)
- **Q&A** (10 mins)

Agenda

- **Demos!!!**
- **Background**
  - What is eRCP?
  - Community
- **Architecture**
- **Coding Exercises!**
- **Q&A**

Demos!

- **Windows Mobile device**
- **Nokia device**
- **What do you want to see?**

Agenda

- **Demos!!!**
- **Background**
  - What is eRCP?
  - Community
- **Architecture**
- **Coding Exercises!**
- **Q&A**

What is eRCP?

- Embedded version of RCP
- Utilizes RCP application framework model
- Reduces RCP size/function to fit on devices
- Pushes changes back to core components to enable running those components on J2ME CDC/Foundation Profile
- Adds component to enable application binary compatibility across a range of devices with different input mechanisms and screen types/sizes

Why is eRCP important?

- **Next generation of mobile Java development!**
- **Extensive rich UI (RCP) capabilities**
  - Integration of native platform look & feel
- **Higher level of device abstraction**
- **Delivers OSGi service oriented features to devices**
  - Dynamic install/uninstall/update
  - Service sharing
  - Heck, OSGi is just *cool*
- **Puts Eclipse programming model on devices!**

Community Involvement

- **1000+ downloads for EACH of the past few milestones**
- **Active newsgroup community**
- **Use of bugzilla for reporting/monitoring defects**
- **Web site updated with latest JavaDocs, How-To Docs, etc.**

What Have We Been Doing?

- **Monthly Milestone builds**
- **Improving quality in preparation for 1.0 release**
- **eUpdate is completed!**
- **Extending reach to more platforms (QTe)**

Platform Implementations

- **Windows Mobile 2003/2005**
- **Windows Desktop**
- **Nokia Series 60/80**
- **QTe**

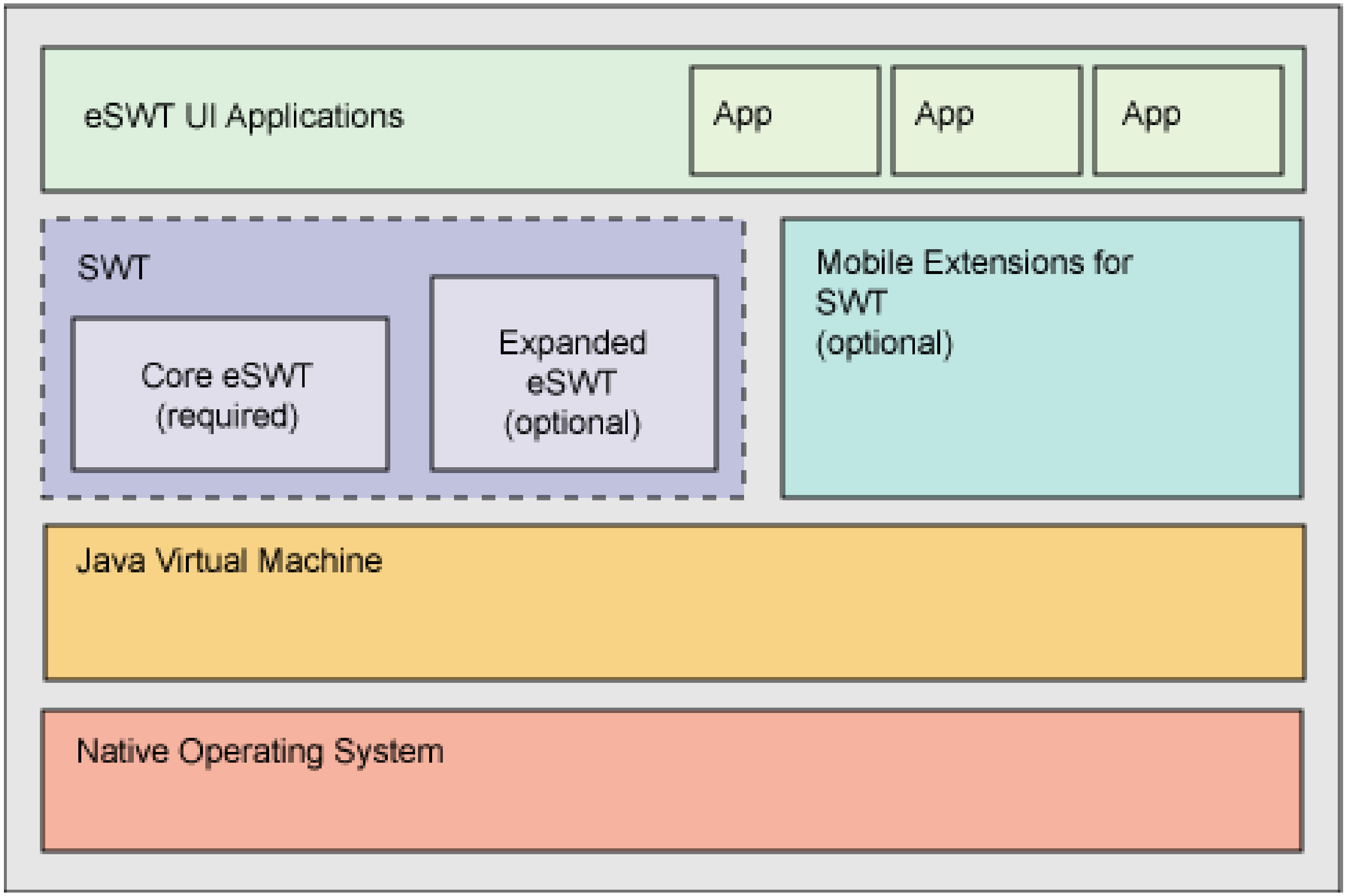
Agenda

- **Demos!!!**
- **Background**
  - What is eRCP?
  - Community
- **Architecture**
- **Coding Exercises!**
- **Q&A**

Architecture and Application Model

- **OSGi underpinnings**
- **eRCP applications run in a workbench similar to Eclipse IDE plugins**
- **Also supports stand-alone eSWT applications**
- **Applications provide a View *extension* which the eWorbench instantiates on demand**
- **Applications and services run in the same JVM**
  - Consumes fewer resources than separate JVMs
  - Allows sharing of services
  - Enables variety of life cycle choices

Architecture and Application Model



Existing RCP Components

- **Core Runtime**
- **OSGI Framework**
- **Commands**
- **Expressions**

RCP Subsets

- **eSWT**
  - Core – Base function and widgets
  - Expanded – Adds Browser, Table, Tree, more layouts
- **eJFace**
  - Subset provides Table/Tree Viewers, Resource handling
- **eUpdate**
  - Allows end-user to browse to update site and download features
  - Provides subset of API for applications to use update services
- **ui.workbench**
  - Size/Function reduced

New Components

- **Mobile Extensions**
  - For eSWT – provides device abstraction
  - For Desktop SWT – provides upward compatibility with desktops
- **eWorkbench**
  - Provides GUI for launching/switching among eRCP apps
  - Customizable by device provider to support devices features
- **microXML**
  - Provides full SAX and DOM support in very small footprint
- **Test Harness**
  - Provides test delivery to device, execution, and reporting

Device Adaptation (1/4)
"Devices are special…"

- **Programming for devices is special**
  - For example, screen *real-estate* is as expensive as California *real-estate* prices

Device Adaptation (2/4)
“Tips & Tricks”

- **Implicit normalization**
  - Automatically provides some level of device adaptation by giving applications indirect access to a device’s native widgets
  - eSWT widgets appear and behave similarly to widgets in native applications
  - End-users can *recognize* and interact with these widgets!
  - The developer gains these benefits simply by using eSWT

- **Explicit normalization**
  - Specific mechanisms that a developer is encouraged to use

Device Adaptation (3/4) “Display Normalization Tips”

- **Use flow based layouts**
  - Layouts position widgets independently of screen size

- **Don’t use absolute coordinates**
  - Display sizes and aspect ratios can vary considerably

- **Even though layouts help considerably in adapting to different screen sizes, well written programs also:**
  - Check if the computed layout is larger than the available screen size, and if so, add scroll bars to allow scrolling the content
  - Check for high aspect ratios which restrict layout or allow for additional content

Device Adaptation (4/4) “Input Normalization Tips”

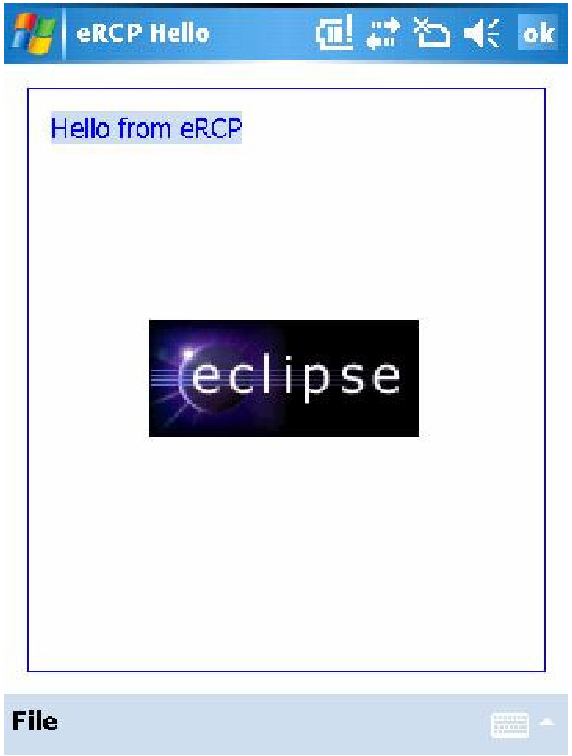
- **Use Mobile Extensions `Command` class**
  - Commands are an abstraction that the Mobile Extensions library maps to a device specific mechanism depending upon the device capabilities (usually menus or softkeys)
- **Use of buttons is discouraged!**
  - Many devices do not have pointers or touch screens, which means buttons must be navigated to using jog controls or arrow keys and then selected. Jogging through numerous widgets is more time consuming and may also be cumbersome depending on the device controls
  - If you really want buttons, then check via the MobileDevice class to see pointer selection is available
- **Use of menus is discouraged!**
  - Width of the menu bar may be limited on many devices
  - Menus may be difficult to navigate on non pointer devices
  - Most important, you may be bypassing the device’s most natural or efficient input mechanism

Agenda

- **Demos!!!**
- **Background**
  - What is eRCP?
  - Community
- **Architecture**
- **Coding Exercises!**
- **Q&A**

Exercise 1: Setup the Environment
"Crap, why doesn't this stuff work?"

- **Download**
  - Grab latest eRCP (Windows Desktop) latest
    - http://www.eclipse.org/ercp
  - Eclipse 3.2
    - http://www.eclipse.org
- **Run the hello application**



Hello from eRCP

Exercise 2: SWT/eSWT/eJFace
“Look mom, twins!”

```
public static void main(String[] args) {
   Display display = new Display();
   Shell shell = new Shell(display);
   shell.setText(“Hello World");
   Label = new Label(shell, SWT.CENTER);
   label.setText(“Howdy”);
   shell.open();
   while (!shell.isDisposed()) {
        if (!display.readAndDispatch())
             display.sleep();
   }
   display.dispose();
}
```

Exercise 3: eWorkbench (1/3)
“Your mobile device wants a workbench”

- **Create plugin class:**

```
public class AppPlugin extends AbstractUIPlugin {
  private static AppPlugin plugin;

  public AppPlugin() {
          super();
          plugin = this;
  }

  public void stop(BundleContext context) throws Exception {
          super.stop(context);
          plugin = null;
  }

  public static AppPlugin getDefault() {
          return plugin;
  }
}
```

Exercise 3: eWorkbench (2/3)
“Your mobile device wants a workbench”

- **Create a view class:**

```
public class NormalView extends ViewPart {
  Label label;

  public void createPartControl(Composite parent) {

        label = new Label(parent, SWT.NONE);
        label.setText("Hello World");
        label.setSize(parent.getSize());
  }

  public void setFocus() {}
}
```

Exercise 3: eWorkbench (3/3)
“Your mobile device wants a workbench”

- **Define your application:**

```
<extension
    point="org.eclipse.ercp.eworkbench.applications">
     <application
        id=“sample.app"
        name=“Sample Application">
        <views normal=“sample.app.views.normal"/>
     </application>
   </extension>
```

- **Define your view:**

```
<extension point="org.eclipse.ui.views">
  <view
        category="org.eclipse.ercp.eworkbench.viewCategory"
         class=“sample.app.views.NormalView"
         icon="icons/sample.gif"
         id=“sample.app.views.normal"
         name="Sample Normal View">
  </view>
</extension>
```

Exercise 4: Deployment
“Updating is easy, I swear!”

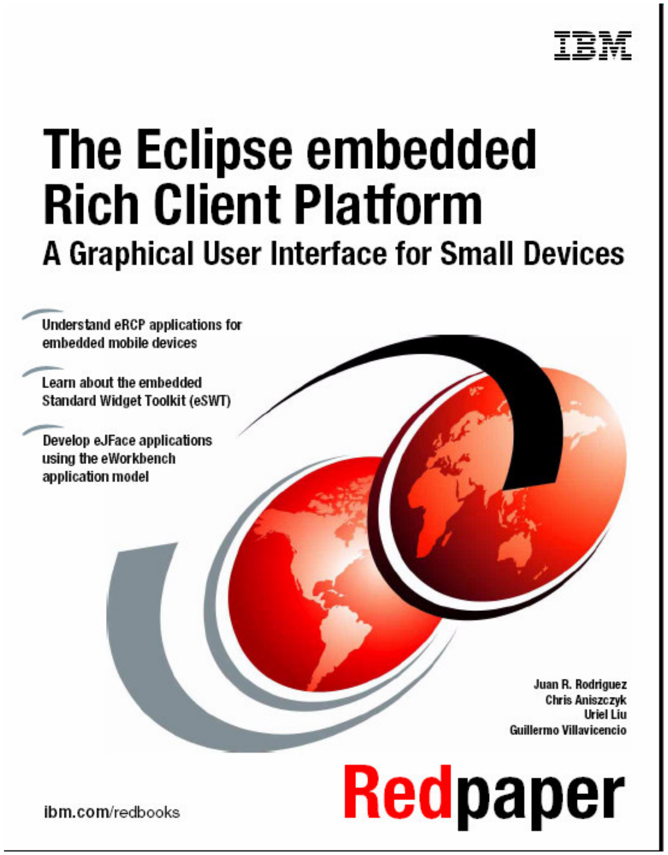
- **Packaging**
  - Feature Creation
  - Update-site Creation
- **Deployment**
  - Launch eWorkbench
  - Run the Application Manager
  - Install application from previous exercise
  - Restart eWorkbench
  - Run the newly installed application

Agenda

- **Demos!!!**
- **Background**
  - What is eRCP?
  - Community
- **Architecture**
- **Coding Exercises!**
- **Q&A**

Q&A

- **Questions?**
- **Thank you!**
- **Choose your own adventure!**
- **http://www.eclipse.org/ercp**
- **A future more in detail book is planned**



IBM

The Eclipse embedded Rich Client Platform

A Graphical User Interface for Small Devices

Understand eRCP applications for embedded mobile devices

Learn about the embedded Standard Widget Toolkit (eSWT)

Develop eJFace applications using the eWorkbench application model

Juan R. Rodriguez
Chris Aniszczyk
Uriel Liu
Guillermo Villavicencio

ibm.com/redbooks

Redpaper