Going Mobile: Mobile Application Design Principles for ADF Development

Peter Koletzke
Technical Director & Principal Instructor

Editor's Choice Award

Mobile Home or Mobile App?
I'm goin' home
And when I wanna go home
I'm goin' mobile
Well, I'm gonna find a home
And we'll see how it feels
Goin' mobile
Keep me movin'

—Pete Townshend (1945–),
Going Mobile (Who's Next album)

Agenda

• Brief history of Oracle Uls
• Design steps
• Further study

Slides and white paper are available on the UTOUG and Quovera websites.

In the Beginning Was...

• ...the “dumb” terminal
• PF and Fn keys!
• Training!

IAF to SQL*Forms 2.0 to Oracle Developer Forms 6
Then Oracle Created ...

- ...tools for client/server applications
- Graphics!
- Mouse!

SQL*Forms 4.0 to Oracle Developer Forms 6.0

Then, In The Year of the Internet...

- ...Java runs Forms within the web browser
- Familiar interface!
- Java install headaches replace Forms install headaches!

Oracle Forms 6i to Oracle Forms 11g

In the Next Year of the Internet...

- ... lightweight Web Apps (“Web 1.0”)
- Browser only!
- HTML rules!

Oracle UIX

In Another Year of the Internet...

- ...Web apps (“Web 2.0”)
- Interactive components!
- Single page!

Oracle ADF Faces 10g to 12c
Today

- Mobile
- Touch/gesture!
- “Intuitive”!

Oracle ADF 12c, Oracle MAF 2.1

Comparing UI Devices

<table>
<thead>
<tr>
<th>Display Paradigm</th>
<th>Graphics</th>
<th>Data Input</th>
<th>Navigation</th>
<th>User Training</th>
<th>Display Software</th>
</tr>
</thead>
<tbody>
<tr>
<td>Character mode</td>
<td>N</td>
<td>Keybd</td>
<td>Keybd (Tab, Fn keys)</td>
<td>Lots</td>
<td>Terminal software and/or host O/S</td>
</tr>
<tr>
<td>“dumb” terminal</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Client/server</td>
<td>Y</td>
<td>Keybd, mouse</td>
<td>Mouse</td>
<td>Some</td>
<td>Special runtime and app files</td>
</tr>
<tr>
<td>Java applet in a browser</td>
<td>Y</td>
<td>Keybd, mouse</td>
<td>Mouse</td>
<td>Some</td>
<td>Web browser, Java</td>
</tr>
<tr>
<td>Light client (not J2EE)</td>
<td>Y</td>
<td>Keybd, mouse</td>
<td>Mouse</td>
<td>Some</td>
<td>Web browser</td>
</tr>
<tr>
<td>Light client (Java EE) with AJAX</td>
<td>Y</td>
<td>Keybd, mouse</td>
<td>Mouse</td>
<td>Less</td>
<td>Web browser</td>
</tr>
<tr>
<td>Mobile (“smart”phone, tablet, e-reader)</td>
<td>Y</td>
<td>Keybd, tap</td>
<td>Tap, swipe</td>
<td>None(?)</td>
<td>Mobile O/S (Android, iOS)</td>
</tr>
</tbody>
</table>

My Lessons Learned

- Mobile is hot
- Mobile is different
- When you go into the woods, you are not alone
- Main challenges: screen size, interface methods
- Mobile will **not** replace desktop
- Two separate mobile “platforms”
- Design for mobile first
- Two-minute task
- Visual not textual

So...

How do I design for a mobile platform?

- The objective of this session
- Example: ADF applications
- Users have **Great Expectations**
  - Less (or no) tolerance for training
  - Less forgiving of non-intuitiveness
  - Short attention span
**Agenda**

• Brief history of Oracle UIs
• Design steps
• Further study

---

**Mobile is Hot**

• Mobile devices of all types
  – Not just PC or Mac
  – Smartphone, tablet, e-reader
• The number of mobile devices has exceeded the world’s population
• Apple’s 2014 Q4 iPhone revenue > Microsoft and Google revenue

**Mobile has Different Challenges**

• Interface methods
  – Touch and gestures (taps, swipes), voice
  – Keyboard not primary method
    • No full-size keyboard
    • No mouse
• Services
  – Camera
  – GPS
  – Contacts
  – Compass
  – Accelerometer
  – Telephone
  – Text messaging
Screen Size Variances

From User interface design for the mobile web, by James Lentz, WebSphere experience architect (July 2011)

- Severe size variance
  - The old display resolution issue
- Also, orientation can shift
  - Horizontal or vertical
Preview: Main Solutions

- **Display**
  - Show important content only
  - Allow drilldowns

- **Interface method**
  - Use gestures instead of mouse
  - Select or talk rather than type data
  - Big icons to tap for navigation and drilldown

1. Understand the problem
2. Know your users
3. Choose the platform
4. Remember three principles
5. Research existing patterns
6. Leverage new components
Know Your Users

- Where will they be when they run your app?
  - In transit: short attention span, distractions
  - Workplace: attention span can be longer
- Do they always work in a mobile mode?
  - Or, are they always in a workplace and just prefer a mobile device?
  - Workplace == desktop only?
  - Workplace == both?

a.k.a., Location, location, location

Why Know Your Users?

- Two mobile platforms
  - Small mobile: cell phone
  - Tablet/desktop: also e-reader
- Location determines screen real estate
  - Desktop and tablet have similar size capabilities
  - Smartphone is limited

Why Know Your Users?

Does Mobile Make Sense?

- Depends on your users
- Mobile will not replace desktop
  - Still a need for workplace computers
    - Heavy data input, reliable database connection
    - Design tasks, multiple monitors needed
    - Software development, etc.
- Platform determines the toolset
  - Small mobile
  - Desktop/tablet
Oracle Tools for Mobile

• Oracle ADF Faces components
  – Serves desktop and tablet – 12.1.3
• Oracle Mobile Application Framework
  – Serves small devices and tablets – 2.1
• One set of design guidelines to rule them all
  – Oracle Alta UI

“One Ring to rule them all, One Ring to find them.
One Ring to bring them all and in the darkness bind them.”

Mobile Application Framework (MAF)

• ADF development methods
  – Business services
  – Pages (features) with components
  – Task flows
• Supports iOS and Android
• Use local services like camera & contacts
• MAF components (HTML5/Java)
• AMX (application module XML) components
  – Subset of MAF; can be used with MAF

#1: Design for Mobile First

• Mobile-designed app can run on desktop
  – Desktop-designed app may not run well in a mobile device browse
• Use the new components
  – Gesture- and orientation-aware
• Design for the narrowest width device
• Use Alta skins, icons, colors

1. Understand the problem
2. Know your users
3. Choose the platform
4. Remember three principles
5. Research existing patterns
6. Leverage new components
#2: Visual, not Textual

- Really important for success
- Simple design elements – Alta skin
  - Quick to render
  - Flat – few lines, little “chrome,” no gradients or shadows
  - Easy to read
  - Big icons
  - Color to highlight important data
- Emphasize the most important content
  - It must appear regardless of orientation or device
- Offer analytics – visual not textual
  - For quick decisions, but don’t overdo

#3: Simple Navigation and Display

- Easy to return to opening page
  - Content, not chrome, provides navigation
- Two-minute tasks
  - Work done in short spurts
  - Solves the short attention span and distractions issues
- Summarize essential data
  - Driltdown to details
    - Information hierarchy

You are Not Alone

- When you go “into the woods” of mobile design
  - Oracle user experience (UX) research
    - For their Cloud (Fusion) Applications
  - Other vendors, too
- Use results of this research
  - Guidelines - for any toolset
  - ADF components - if you’re doing ADF
  - Learn lessons from those

The woods are just trees; the trees are just wood...
No need to be afraid there.
—Stephen Sondheim, Into the Woods
General Guidelines

• Search must be easy
  – Select criteria from lists
  – Aim for single field (google-like)
  – Voice-enabled, if possible
• Add social features
  – Collaboration, instant messaging, etc.
• Use mobile device services
  – Camera, GPS, contacts

More General Guidelines

• Scroll vertically only
• Use design patterns like “infotile”
  – Tabs containing data (related or not)
  – Drill into/down for details
• Alternative views for same data
  – Card, map
• Data entry using lists if possible

About Oracle Alta UI

• Released at OOW 2014
  – Developed for Oracle Cloud (& Fusion) applications
• Home page
  – bit.ly/oraclealta OR

Note: Future references to [ALTA_BASE] = www.oracle.com/webfolder/ux/
Engaging Visual Content
Data visualizations and info-graphics increase insight

To see more Oracle Alta UI samples check out the ORACLE ALTA UI GALLERY.

CUT THROUGH INFORMATION SMOG AND MAXIMIZE INSIGHT

A key design principle of Oracle Alta UI is to engage users with more visual content. Given the proliferation of mobile devices, social media, internet, and a 24/7 news cycle ours is a fast-paced, always-on, information society. Most people scan web pages rather than fully reading them. This means we need to approach modern application design differently. Modern UIs, like Oracle Alta UI, are much more graphical in nature, moving away from a traditionally text-based UI. A picture is worth a thousand words.

STACITIC AND TIGHTLY PACKED PRESENTATION OF INFORMATION IN VISUAL FORM.

New Development With Oracle Alta UI
Opt-in to the Alta UI skin today!

USING THE ORACLE ALTA UI SKIN FOR NEW DEVELOPMENT

The Oracle Alta UI skin is meant for new development and is an opt-in. To use Oracle Alta UI skin instead of Skyros, there are a few configuration steps. You configure your application to use the new skin by specifying values for the <skin-family> and <skin-version> elements in your application's trinidad-config.xml file, as shown below:

Introduction

Design Motivation

Cloud and Mobile – The Oracle Alta UI is both a visual design, as well as a layout approach (i.e., Lock and Flex). The motivation behind the Oracle Alta UI is to enable applications deployed to the Cloud and used on Mobile clients to present a compelling and comfortable UI.

Performance – For both Cloud and Mobile UIs, performance is key. The Oracle Alta UI advocates simpler page constructions to gain better performance.

Adaptive/Responsive Design – Responsive Design allows the same semantic markup to layout appropriate UI designs at different browser display widths. Therefore, the same web page can support multiple devices. Oracle Alta UIs are conducive to responsive design. That is, a product team may opt to apply responsive design techniques to enhance the Oracle UI without change to the Look and Feel.

This motivation has led us to make a number of design choices, such as:

- Less dense, light layouts
- Reduced noise in the UI chrome – UI designers can make page content stand out because there is less UI chrome with which to visually compete. This is achieved by:
  - A lighter, flatter visual design
  - Buttons, menus, and other UI components that are larger to support a less dense look and offer a larger target for touch input
Oracle Alta Mobile UI

• Subset/wing of Oracle Alta UI
• Oracle Alta Mobile Design Guidelines
  – Style
    • Custom color palette, icons, fonts, skin
  – Components
    • MAF
  – Patterns
  – Examples

For desktop/tablet apps, look for "Browser UI" or "Web UI" examples.
What About Examples?

- Oracle Alta UI website
  - Work Better demo app

[jdevadf.oracle.com/workbetter/faces/index.jsf]
ADF Faces 12c – 12.1.3

- Application Development Framework
- Offers new components that support multiple interface paradigms
  - Finger tap on a mobile device == mouse click on a desktop device
  - Rendering accommodates varying sizes and orientations
- Backported to 11g
  - 11.1.1.7 used for Fusion Apps

“New” ADF Faces Mobile Components

- Panel Springboard
- Panel Drawer
- List View
- Others
  - Card Deck (af:deck)
  - Transition (af:transition)
    - Child of a component that supports transitions
  - Diagram (dvt:diagram)
  - Led gauge (dvt:ledGauge)
  - Dial gauge, Status Meter Gauge

Panel Springboard - Navigation

- Not a pulldown or a list
- Graphical
  - Iconic
- Uses
  - Apps store
  - App functions
On a Tablet or Desktop

Panel Drawer - Alternative to Tabs
- Animated content display

List View - Data Display
- Abbreviated list (not table of all values)
- List expands to show more data

Panel Springboard
- ADF Faces: af:panelSpringboard
- MAF: adfmf:springboard
- Container for af:showDetailItem
- `displayMode`: set to grid or strip
**Panel Drawer**
- ADF Faces: af:panelDrawer
- MAF: not available
- Container for af:showDetailItem
- *position*: right, left, etc.

```xml
<af:panelDrawer id="pd1" position="right">
  <af:showDetailItem text="Jobs" id="sdi5"
    icon="/images/toolsOff.png">
    <af:region value="#{bindings.jobflow1.regionModel}" id="r4"/>
  </af:showDetailItem>
  <af:showDetailItem text="Locations" id="sdi6"
    icon="/images/globeOff.png">
    ...
  </af:showDetailItem>
</af:panelDrawer>
```

**List View**
- ADF: af:listView
- MAF: adfmf:navigationBar
- Container for af:listItem (e.g., Locations)
- Optional groupHeaderStamp facet
  - Content for heading of the group
- Model project objects needed to supply data

**List View Code Example**

```xml
<af:listView
  value="#{bindings.AllCountries.treeModel}" var="item"
  fetchSize="#{bindings.AllCountries.rangeSize}" id="lv1">
  <af:listItem id="li1">
    <af:panelGroupLayout id="pgl1" layout="vertical">
      <af:outputFormatted
        value="#{item.bindings.StreetAddress.inputValue}" id="of1"/>
      ...
    </af:panelGroupLayout>
  </af:listItem>
  <f:facet name="groupHeaderStamp">
    <af:listItem id="li2">
      <af:panelGroupLayout id="pgl2">
        <af:outputFormatted
          value="#{item.bindings.CountryName.inputValue}" id="of5"/>
      </af:panelGroupLayout>
    </af:listItem>
  </f:facet>
</af:listView>
```

**List View Page Definition**

```xml
<bindings>
  <tree IterBinding="AllCountriesIterator" Id="AllCountries">
    <nodeDefinition
      DefName="adfnutshell.model.views.CountriesView"
      Name="AllCountries0">
      <AttrNames>
        <Item Value="CountryName"/>
      </AttrNames>
      <Accessors>
        <Item Value="LocationsView"/>
      </Accessors>
    </nodeDefinition>
    <nodeDefinition
      DefName="adfnutshell.model.views.LocationsView"
      Name="AllCountries1">
      <AttrNames>
        <Item Value="StreetAddress"/>
        <Item Value="City"/>
      </AttrNames>
      <Accessors>
        <Item Value="LocationsView"/>
      </Accessors>
    </nodeDefinition>
  </tree>
</bindings>
```

The heading for zero to many locations

Allows listView to access Locations attributes
List View for Navigation

Agenda

• Brief history of Oracle UIs
• Design steps
• Further study

Oracle Website

• Oracle Technology Network
  – MAF home page
  – ADF home page
  – OTN demo videos
  – ADF Academy videos
• Chris Muir - design principals on YouTube
  – UI Development in Oracle MAF series
  – Part 12: Mobile UI Design Strategies
  – bit.ly/1Bch45B
• UX in Oracle Apps

From Shay Schmeltzer, Oracle PM Extraordinaire

• Introduction
  – https://blogs.oracle.com/shay/entry/a_guide_to_developing_your
  – https://blogs.oracle.com/jdeveloperpm/entry/get_the_oracle_alta_ui
• Building with Oracle Alta UI
  • Oracle Alta UI Development with Oracle ADF Faces
  • Related blog is mentioned in the Youtube notes
  • www.youtube.com/watch?v=oUNdxY7L-3w
Non-Oracle Perspectives

- User Interface Design for the Mobile Web, James L. Lentz, July 2011
- 20 Beautiful Mobile User Interface For Your Inspiration, Nancy Young
  - www.hongkiat.com/blog/mobile-app-ui/
- Smashing Magazine
- Code My Views
  - https://codemyviews.com/blog/mobilefirst

A MAF Must Have

- Luc Bors
- Oracle Press
- All you need to know to start working with MAF once design is complete

Quote

To be happy in this world, first you need a cell phone and then you need an airplane. Then you’re truly wireless.

— Ted Turner (1938-)

Summary - Lessons Learned

- Mobile is hot
- Mobile is different
- When you go into the woods, you are not alone
- Main challenges: screen size, interface methods
- Mobile will not replace desktop
- Two separate mobile “platforms”
- Design for mobile first
- Two-minute tasks
- Visual not textual
• Founded in 1995 as Millennia Vision Corp.
• Profitable without outside funding
• Consultants each have 10+ years industry experience
• Strong High-Tech industry background
• 200+ clients/300+ projects
• JDeveloper Partner
• More technical white papers and presentations on the web site

Please fill out the evals
Books co-authored with Dr. Paul Dorsey, Avrom Roy-Faderman, & Duncan Mills