User Interface Adaptations:
Indispensable for Single Authoring

Srinivas Mandyam
smandyam@covigo.com
Overview

1. Introduction
2. Single Authoring with UI Adaptations
3. Examples
4. Summary
Introduction

Building Device Independent Applications

Goal: Build one user interface that can be highly optimized for each terminal, without compromising interface quality.

High Precision User Interfaces

Desktop PC
Pocket PC
WAP
Introduction

Some of the Common Problems

- Disparate content sets across terminals
- CSS support not available for all devices
- Varying media support
- Content needs to be paginated
- No uniform support for structural elements (tables, lists)
- Need to use device specific markup
- Different Navigation on different devices
Design Methodology

Designers want full control to create High Precision UIs

- Design for device clusters, not individual devices
  - Device clusters represented using RDQL

- Design fragments that may be reused across multiple devices
  - Use existing authoring tools

- Specify adaptation techniques to apply on fragments (sets).
  - Pagination, Structural Transformation, etc.

- Preview and Optimize
  - Lot of time is usually spent here
Introduction

Design for Terminals, Not Individual Devices

Terminal = device cluster
- Class of devices that share a common trait
- “All portrait devices with color depth of 4096 or more”
- May be one device or many devices

Terminals are defined using RDQL (RDF Data Query Language)
- Device profiles (CC/PP, UAProf) are defined in RDF

Design using terminals
- Designing for one terminal works for many devices
- System supports RDQL aliases for ease of maintenance & reuse
Introduction

Example: Terminals based on screen size

Screen sizes categorized by frequency.
Based on a study of over 2 million hits from 600 devices
Introduction

Common Terminals

- **Screen size**
  - Small
    (Up to 135 pixels wide)
  - Medium
    (From 136 – 400 pixels wide)

- **Markup language**
  - WML
  - HTML
  - cHTML

- **Media support**
  - Color capable
  - Bit depth (bits/pixel)

- **Manufacturer**
  - Sanyo
  - Sony Ericsson
  - Nokia

SELECT ?os
WHERE (?a, <prf:OSVendor>, ?os),
  (?b, <prf:Vendor>, ?vendor)
AND (?os EQ "Nokia") ||
  (?vendor EQ "Nokia")
USING prf for
  <http://www.wapforum.org/profiles/UAPROF/ccppschem#include-20010330#>
Introduction

Overall Design process

Plan user interfaces for device clusters

Develop Adaptive User Interface fragments using current authoring tools

Apply Adaptations, Assemble and Preview application on all target devices
Introduction

Request-Response Cycle for Device Independent Applications

1. **Request**
   - Identify device using heuristics or CC/PP and UAProf

2. **Device Detection**
   - Device Profile

3. **Interface Adaptation**
   - Apply interface adaptation

4. **Response**
Introduction

Example Sequence of Adaptations

- Interface Adaptation is usually a sequence of actions
User Interface Adaptations
User Interface Adaptations

1. Content Inclusion & Exclusion

Show and hide content based on connecting device class

Common uses:
- Assemble modularized fragments of application
  - Portals
- Hide extraneous content from a larger page
  - Banner ads
  - Extra navigation
  - Voluminous content
  - Sidebars
- Show device- or markup-specific navigation
  - Softkeys
2. Pagination

Break large bodies of content into sections that can be displayed on multiple pages at runtime

- Based on device capabilities
  - Deck size
  - Number of records
- Repeatable header and footer on each page
- Cached result set

Common uses:
- Paginate long bodies of text
  - News stories
  - Essays
- Paginate large number of repeating records
  - Search results
  - Catalogs
User Interface Adaptations

3. Structural Element Transformations

Display the same content using different layout (structural) elements

- Main layout structures:
  - Tables
  - Lists

- Include/Exclude Content based on device capability

List types
- **Inline** (Item | Item | Item)
- **Br** (Item<br/>)
- **OL & UL** (<ol>, <ul>)
- **Select** (<select>)
- **Abstract** (<div type="list"><span listitem="true"/> </div>)
- **P** (<p>Item</p>)

Common uses:
- Reduce large tables of tabular data for smaller devices
  - Display selected columns or rows
- Reuse navigation
  - Convert navigation (list of links, etc.) to different form to take advantage of interaction behavior in different devices or browsers
- Exclude large areas of page layout
  - Tables often used to control layout structure
User Interface Adaptations

4. Pass Through

- “Pass through” device-specific content & code
- Used when Single Authoring is unsatisfactory for a specific device

**Common uses:**

- Take advantage of markup language-specific capabilities
  - Nested activities
  - Variables
  - Marquees (scrolling text)
- Use scripts
  - JavaScript
  - WMLScript
- Add/change markup- or device-specific attributes
  - Format masks
User Interface Adaptations

5. Page Systems for Navigation

Specify navigational relationships for different information architectures
- Content of each page determined during design time (not runtime, like pagination)

Common uses:
- Break up long list
  - Categorize list items
  - Add hierarchy to access information
- Divide contents of portal home
  - Main page becomes index
  - Each portlet becomes its own page or series of pages
Example
WebMail & Calendaring

One single-authored template file is transformed to work appropriately for each terminal type.
Example

How Did We Do It?

Four examples of how AUI adaptation mechanisms allow for high-quality single-authoring.

PC View

PocketPC View

WML View

1. Includes and Show/Hide
Example

How Did We Do It?

Four examples of how AUI adaptation mechanisms allow for high-quality single-authoring.

1. Includes and Show/Hide
2. Morphing: Tables vs Lists
Example
How Did We Do It?
Four examples of how AUI adaptation mechanisms allow for high-quality single-authoring.

- 1. Includes and Show/Hide
- 2. Morphing: Tables vs Lists
- 3. Pagination
Example

How Did We Do It?

Four examples of how AUI adaptation mechanisms allow for high-quality single-authoring.

- **PC View**
- **PocketPC View**
- **WML View**

1. Includes and Show/Hide
2. Morphing: Tables vs Lists
3. Pagination
4. Morphing: Transpose Tables
Example

Widget Morphing

The same tabular information is selectively adapted to the target device form factor and capabilities.
Example

Adaptive Media

- Show the appropriate version of an image, audio, video etc for each terminal
- Media Equivalency Group defines equivalent media files
- Server selects correct media file based on situation:
  - terminal characteristics, bandwidth, server load, personalization preferences
Summary

Conclusions

1. Using current standards such as XHTML and CSS with a few minor additions is a feasible approach for Single authoring

2. Developers want a high degree of control while building multi-channel user interfaces

3. A scalable single authoring solution needs UI Adaptation techniques

4. Designers need better tools for iterative UI design