Using Mobile Device Detection Approaches to Augment the Accuracy of Web Delivery Content

XATA’11

Ricardo Queirós\textsuperscript{1}  Mário Pinto\textsuperscript{2}

\textsuperscript{1}ricardo.queiros@eu.ipp.pt  UCTI-ESEIG/IPP & KMILT

\textsuperscript{2}mariopinto@eu.ipp.pt  UCTI-ESEIG/IPP & KMILT

1 de junho de 2011
Outline

1. Introduction
2. Device Detection techniques
3. ESEIG Mobile
   1. Architecture
   2. Device Capabilities Repository
   3. The WURFL REST Web Service
4. Validation
5. Conclusions
6. Current status and future work
Based on a survey at ESEIG

- large number of our students use mobile devices
- mobile devices with different platforms and characteristics
- they are eager to use their devices in several scenarios:
  - access ESEIG’s web site to consult news and events
  - access to the LMS for course content, assignments and grades

Lack a automatic mechanism to deliver uniform institutional and learning Web content to our students!
Device Detection (1)

- mobile content quality requires a full and demanding awareness of the constraints and benefits of mobile devices (Parsons & Hyu, 2007)
  - resolution
  - color and audio capabilities
  - screen size
  - keyboard functionality
  - media content type supported
  - encoding
  - …

- must be adapted to suit the mobile device characteristics
- adaptation means a process of selection, generation or modification of content (text, images, audio and video) to suit to the user’s computing environment and usage context (Parupalli, 2009).
Device Detection (2)

- Content adaptation requires firstly an accurate detection
- Device detection techniques:
  - Lazy mode - rely on the visualization browser features
  - LCD mode - W3C Mobile Web BPWG defined Default Delivery Context (DDC) as the Lowest Common Denominator (LCD)
  - User-Agent mode - use client (JavaScript) or server (PHP, JSP, ASP) scripting
Device Detection Techniques

PHP code for device detection based on the UA

```php
$ua = $_SERVER['HTTP_USER_AGENT'];
if (stristr($ua, "Windows CE") or
    stristr($ua, "AvantGo") or
    stristr($ua, "Mazingo") or
    stristr($ua, "Mobile") or
    stristr($ua, "T68")) {
    $DEVICE_TYPE="MOBILE";
}

if (isset($DEVICE_TYPE) and $DEVICE_TYPE=="MOBILE") {
    $location='mobile/index.php';
    header ('Location: '.$location);
    exit;
}
```
Device Detection Techniques

Device Detection (4)

- User-Agent (UA) header
  - RFC 2616 (HTTP/1.1) requires the UA format to be a string:
    User-Agent = "User-Agent:" 1*( product | comment )
  - some examples:
    - FireFox on my desktop PC:
      Mozilla/5.0 (Windows NT 6.0; rv:2.0.1) Gecko/20100101
      Firefox/4.0.1
    - Safari on the iPad:
      Mozilla/5.0 (iPad; U; CPU OS 3_2_1 like Mac OS X; en-us)
      AppleWebKit/531.21.10 (KHTML, like Gecko) Mobile/7B405

UA includes non-standard data about the device and the browser!
Device Detection Techniques

**Device Detection (5)**

- **User-Agent (UA) header**
  - Non-standardization of the UA string
    - affects the data interpretation and extension
    - triggered the appearance of the *device profiling* concept
    - covered by several standards: CC/PP, UAProf, WURFL, DeviceAtlas, ...
Device Detection (6)
1. W3C CC/PP (Composite Capability/Preference Profiles)
   - CC/PP is a RDF vocabulary extension
   - defines how client devices express their capabilities and preferences (the user agent profile) to the server that originates content (the origin server)
Device Detection (7)

2. UAProf (User Agent Profile)

- standard created by the Open Mobile Alliance
- represent a concrete CC/PP vocabulary for mobile phones
- conformant mobile phones provide CC/PP descriptions of their capabilities to servers that use this information to optimize the content
- a mobile device sends a header within an HTTP request containing the URL to its UAProf. The HTTP header is usually "x-wap-profile" (or "19-profile", "wap-profile", ...)
- drawbacks: Not all devices have UAProfs, others present dangling links, others are inaccurate
Device Detection Techniques

Device Detection (8)

3. WURFL (Wireless Universal Resource FiLe)
   - repository with a list of device capabilities and features
   - created by Luca Passani
   - repository can be used locally or accessed remotely using a REST WS approach
   - fed by a huge community of developers
Device Detection Techniques

Device Detection (9)

4. DeviceAtlas

- commercial database for device detection created by dotMobi
- includes many device databases such as WURFL and UAProf
- retrieve accurate JSON results
- DeviceAtlas Personal - a SOA aware version:
  1. user visits a Web site on his mobile device
  2. server forwards the UA header to the DA Personal service
  3. server receives a response with info about the user’s device
Device Detection Techniques

Device Detection (10)
Mobile device concurrency test

- data set includes 1,572 unique user agents
- DeviceAtlas presents smaller processing times, but more inaccurate results than WURFL

<table>
<thead>
<tr>
<th>Method</th>
<th>Time (seconds)</th>
<th>Mobile</th>
<th>Non-Mobile</th>
</tr>
</thead>
<tbody>
<tr>
<td>WURFL API</td>
<td>20.8</td>
<td>1090</td>
<td>482</td>
</tr>
<tr>
<td>DeviceAtlas API</td>
<td>1.2</td>
<td>527</td>
<td>1045</td>
</tr>
</tbody>
</table>
ESEIG Mobile

ESEIG Mobile project
- aims to standardize the delivery of learning and institutional content produced at ESEIG to the diversity of the students’ mobile devices
- a new layer on the top of the existent ESEIG infrastructure
Devices’ Capabilities Repository

- contains a file (wurfl.xml) with a large list of device capabilities
- formally described with a DTD file
- the latest file (2011-04-24) occupies 16MB
- can be downloaded for free at http://wurfl.sourceforge.net/
- can be installed locally or consumed remotely through WS
- it provides APIs (JAVA and PHP) to abstract and uniformize the access to the DB
Devices’ Capabilities Repository

- the WURFL is based on the concept of family of devices
- all devices are descendant of a generic device, but they may also descend of more specialized families
- a device element contains capabilities
- a device capability is a specific feature of a given device organized in groups
- groups are used to improve the readability of the WURFL XML database by humans (ex: sound_format group)
Devices’ Capabilities Repository

How it works?

- receives a UA
- uses a algorithm RIS (Reduction to Initial String) to match the UA
- uses a fallback mechanism to gather the request capability

```xml
<device fall_back="nokia_generic" id="N7110_v1"
    user_agent="NOKIA7710/1.0">
    <group id="sound_format">  
        <capability name="wav" value="true" />
        ...
    </group>
    ...
</device>
```
Devices’ Capabilities Repository
User Agent **Nokia7110/1.0 Mozilla... supports WAV format?**

```xml
<device fall_back="root" id="generic" user_agent="">
  <group id="sound_format">
    <capability name="wav" value="true"/>
  </group>
</device>

<device user_agent="Nokia" fall_back="generic" id="nokia_generic">
  <group id="sound_format">
    <capability name="mp3" value="false"/>
  </group>
</device>

<device user_agent="Nokia7110/1.0" fall_back="nokia_generic" id="nokia_7110_ver1">
  <group id="sound_format">
    ...
  </group>
</device>
```
REST Web Service

- **Detector** component
  - extracts the UA from the request
  - uses it to query the WURFL repository with the WURFL WS
- This WS provides a RESTful interface to the WURFL DB
- The use of this approach rather than a local implementation of WURFL avoids the maintenance of a local storage liable to the typical synchronization issues

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>ua</td>
<td>User-Agent</td>
<td><a href="http://api.wurflws.com/wurflws?ua=%5BUA">http://api.wurflws.com/wurflws?ua=[UA</a>]</td>
</tr>
<tr>
<td>search</td>
<td>Capabilities filter</td>
<td><a href="http://api.wurflws.com/wurflws?search=%5BF_1,F_2,...,F_n">http://api.wurflws.com/wurflws?search=[F_1,F_2,...,F_n</a>]</td>
</tr>
</tbody>
</table>
REST Web Service

- calling the following URL will return two capabilities of the Nokia 6630: resolution width and height

http://api.wurflws.com/wurflws?
    ua=Nokia6630/1.0(2.3.129)%
    20SymbianOS/8.0%
    20Series60/2.6%
    20Profile/MIDP-2.0%
    20Configuration/CLDC-1.1&
    search=resolution_width|resolution_height

- response is a JSON string that will be parsed by Adapter:

  {"resolution_height": "208", "resolution_width": "176"}
ESEIG Mobile access

- validate the usage of the ESEIG-Mobile web interface
  characterizing the access and rejection levels
- based on Google Analytics service data:
  - hit counters
  - rejected requests
  - new visitors
  - traffic
  - mobile operating systems used
- data was collected from November 2010 to February 2011
ESEIG Mobile access

1. ESEIG-Mobile usage: hits and new visitors
Validation

ESEIG Mobile access

2. ESEIG-Mobile users activity on the site
3. ESEIG-Mobile access by platform
Conclusions

ESEIG Mobile

- present and compare approaches for mobile device detection
- present the design of ESEIG-Mobile as an open system for the delivery of suitable content to the students’ devices
  - a devices’ capabilities repository to store a meaningful number of characteristics of mobile devices
  - a REST Web Service to obtain these characteristics based on the client’s HTTP header request
- validate our approach by presenting the usage statistics of ESEIG Mobile
- The analysis of this data helps to confirm the heterogeneity of the students’ mobile devices and their usage habits and preferences
ESEIG Mobile

- **Current status**
  - early development
  - only detecting if the HTTP request is made from a mobile device and query some device capabilities from the WURFL device repository

- **Future work**
  - prototype implementation process regarding, for instance, the transformation of the Web resources in the WNG format [13]. For this task we are considering using Extensible Stylesheet Language for Transformation (XSLT) to formally describe the transformations
  - increasing the device repository performance migrating from the WURFL XML database to a relational database (e.g. MySQL) using the Tera-WURFL project
Questions

Thanks!!

Questions?