Outline

• Introduction and Basic Concepts
• System Architecture
• Architecture Implementation
• Example Application
• Conclusions and Future Directions
Introduction / Motivation

Service provider

ADSL...

WIFI/GPRS

GPRS/UMTS

WIFI/GPRS

Others…

Goal of our work

A software development model where the presentation of an application (user-interaction) is able to adapt to the client execution context

This has been done:

• by separating the application logic from its presentation
• by using an XML-based language to specify the user interaction with the application in an independent manner with respect to the specific client execution context
• by delegating to a “renderer” the task of generating the actual user interface, adapting it to the used terminal characteristics
System Architecture

XML Vocabulary

Our vocabulary has been created by integrating the base vocabulary XForms with new tags in order to manage:

- Data Input
- Data Output
- Choice and multiple selections
- Information transfer
- Triggers
- Content displaying

```
<!ELEMENT UI (group+)>  
<!ATTLIST group caption CDATA #IMPLIED>
```
Example: choice and select Tag

```xml
<select ref="Sex">
  <label>Sex Field</label>
  <help/>
  <hint/>
  <choices>
    <item>
      <label>Male</label>
      <value>male</value>
    </item>
    <item>
      <label>Female</label>
      <value>female</value>
    </item>
  </choices>
</select>
```

Rendering in J2SE

Rendering in J2ME

Rendering and Profile Management

The choice of the interface renderer is made in a dynamic fashion, according to the client profile.

Based on client profile it will be able to:

- send to the client device the **renderer** and the **XML** description,
- send to the client device the **XML** description,
- send to the client the **rendered interface**.

UAProf (User Agent Profile) specifications for the client profile’s parameters:

- screen size,
- number of supported colors,
- supported image type
- etc.
Content Adaptation

Server based adaptation:

Client based adaptation:

Proxy Based adaptation:

Architecture Implementation

The approaches used in our framework are:

- Proxy based for the content adaption:
- Client based for the rendering process:

We have implemented:
- A renderer for the MIDP2.0 profile
- A Content Adapter
- A Profile Manager
- A Service Provider
The J2ME Renderer

1. XML file parsing
2. Rendering procedure
3. Events detecting

Profile Manager and Content Adapter
Example Application (1/2)

The application allows users to get information about a movie:

– The title
– The playbill
– The trailer

These information will be presented and adapted according to the device characteristics used by the user to access the service.
Example Application (2/2)

**UI Description**

```xml
<?xml version="1.0" encoding="UTF-8"?>
<UI>
  <group caption="MainForm" execution="Local">
    <output ref="Stringa">Title: Matrix Reloaded</output>
    <image meta-inf="Optional" ref="matrixreloaded.jpg" mime-type="image/jpg" align="center" maintain_aspect="true" scale="2" width="40" height="300"/>
    <output ref="Trama">Neo e gli altri ribelli hanno 72 ore di tempo prima che un esercito di …</output>
  </group>
  <output ref="Stringa">Trailer:</output>
  <video meta-inf="Optional" ref="http://localhost:8080/MyServlet/matrixj2me.mpg" mime-type="video/mpeg" start="onInit"/>
</UI>
```

Conclusions

In this work we have developed:

- A model in order to separate the presentation layer of the application from its business logic.
- A way to promote the automatic generation of the actual user interface starting from an abstract specification of the user interaction.
- An XML based language, giving the application developer the means to represent the user interaction with the application in an abstract way.
- A J2ME renderer prototype showing the viability of the approach.
Future Directions

• To refine the XML-based language in order to overcome some encountered limits (e.g. layout and size constraints)
• To develop “renderers” for PDAs (J2ME Personal Profile) and full-fledged desktop (J2SE)
• To integrate in our model a task model in order to specify more complex application behaviors
• To further exploit profile information (e.g. by also considering network information)
• To develop a more complex example application on different devices

Thank you for your attention!