MyMobileWeb

An open source platform for developing Mobile Web applications and portals

30/05/2007

FIT-350401-2006-2

Table of Contents

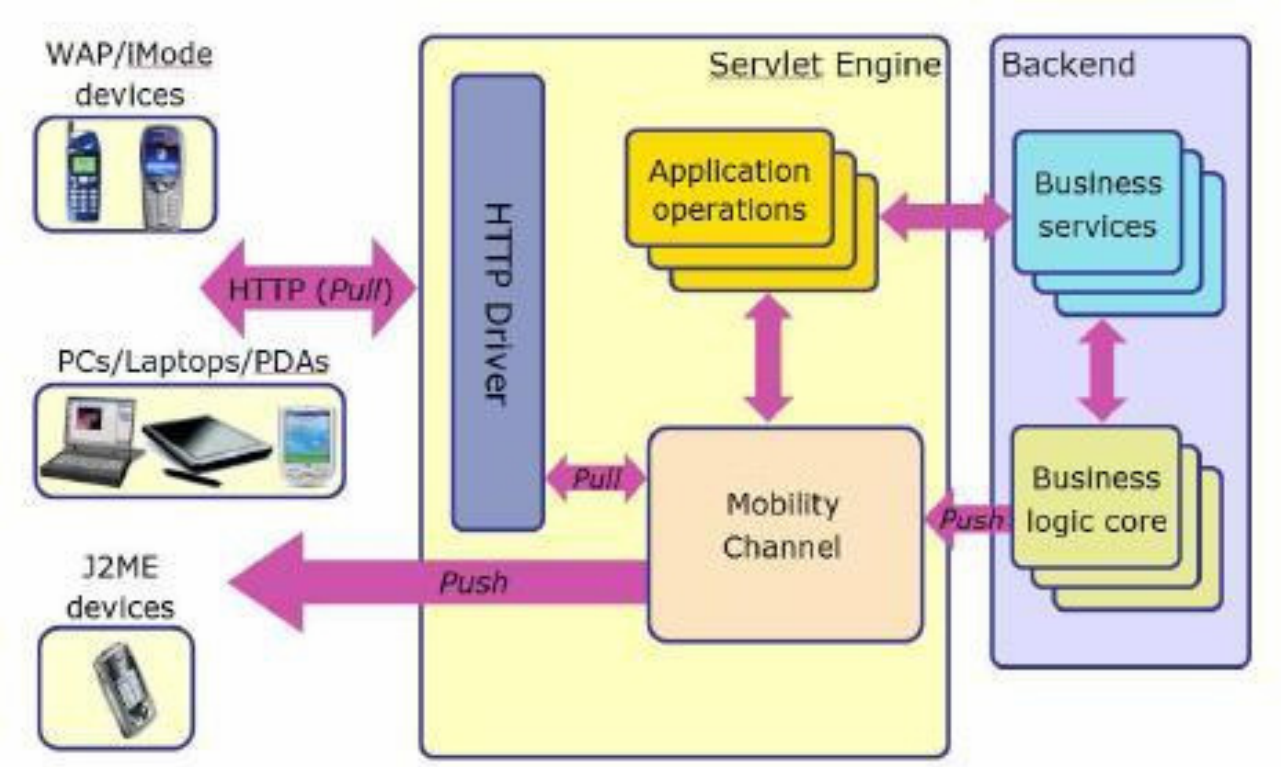
Introduction

- MyMobileWeb is an open source platform that simplifies the development of mobile web applications and portals
  - A low-cost platform (there are no license fees). GPL License moving towards Apache License
  - A modular product, open-standards-based
    - Other open-source technologies are used (WURFL, Batik, Xerces, Xalan, …)
  - All-Java product that only requires a minimal Servlet / JSP container (Tomcat, for example)
- Provides different modules which cover all the basic requirements that must meet a complete and integrated mobile solution, hiding all the complexity related to dealing with multiple delivery contexts
- Includes experimental modules related to the exploitation of semantics in a mobile environment
- Applicability
  - dotMobi applications and portals
  - Mobile solutions intended to work in an uncontrolled environment (multiple devices)
  - Creation of mobile content channels based on JSR-170-based CMS or RSS

Vision

- Mobile Web development should be driven by a ‘Channel Model’ based on Service Oriented Architectures (SOA)
  - Applications publish business services that are invokable from different channels: traditional Web Channel and Mobility Channel
  - Services are independent of the channel and need not to be duplicated
  - Mobile Channel is different than Web Channel (in general)
    - Markup transcodification is an anti-pattern
    - It has different views (presentations) and navigation schema
    - Both channels can be integrated in the same server and CMS but they are essentially different
    - Aligned with the dotMobi vision
  - Typically a mobility channel only addresses a small percentage of functionality
  - Mobile Channel is composed by
    - Information that users want to access while they are on the move
    - Information that people want all the time
- Rapid Application Development (RAD) of MultiChannel and MultiDevice Services
  - Reduction of time and budget
  - Common development skills (Web, Java, XML ...)

Target Architecture

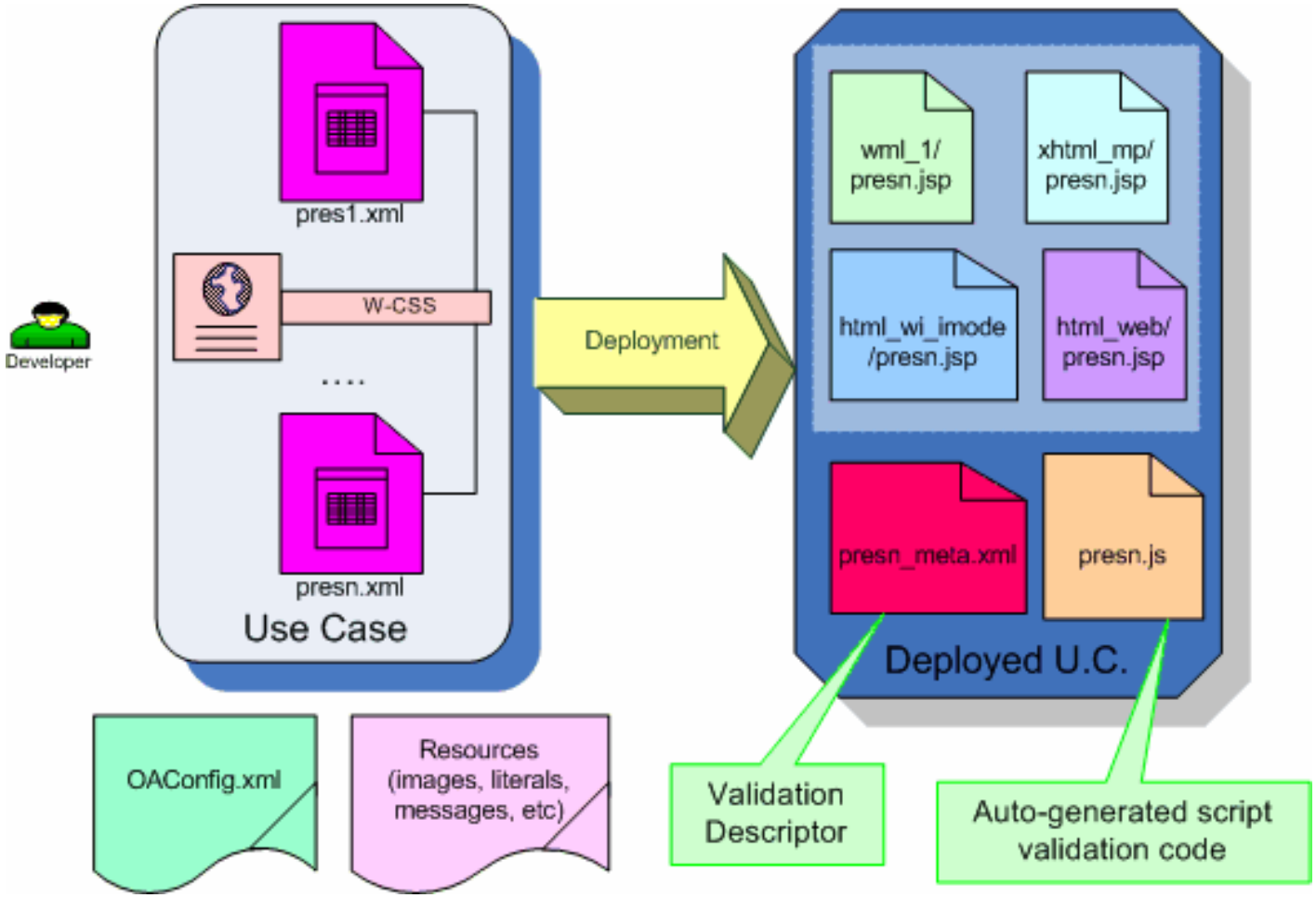


Main Features

- MyMobileWeb enables device independent development
  - It is followed a ‘flexible authoring’ approach
  - Flexible authoring is an authoring style that consists of designing once for all kind of delivery contexts, in addition to, when needed, a combination of adaptation policies and / or customized variants of few resources for specific delivery contexts.
- Differential aspects against competitors
  - High performance architecture. During each client request there is no markup transcodification
  - Integrated with WURFL (“de facto standard) for Device Description
  - It works with any Java Servlet/JSP Web/Application Server
  - Automatic code generation for local (Javascript-based) and server-side validations
  - Smart Literal Management (literals can be redefined for specific devices or clusters of devices)
  - Based on the concept of mobile visual controls and declarative user interface language
  - Intelligent management of paging, for each visual control, each container ...
  - Off-the-self (component-based) mobile adaptation of RSS data sources
  - Integrated with JSR-170-based CMS

Table of Contents

Development cycle (I)



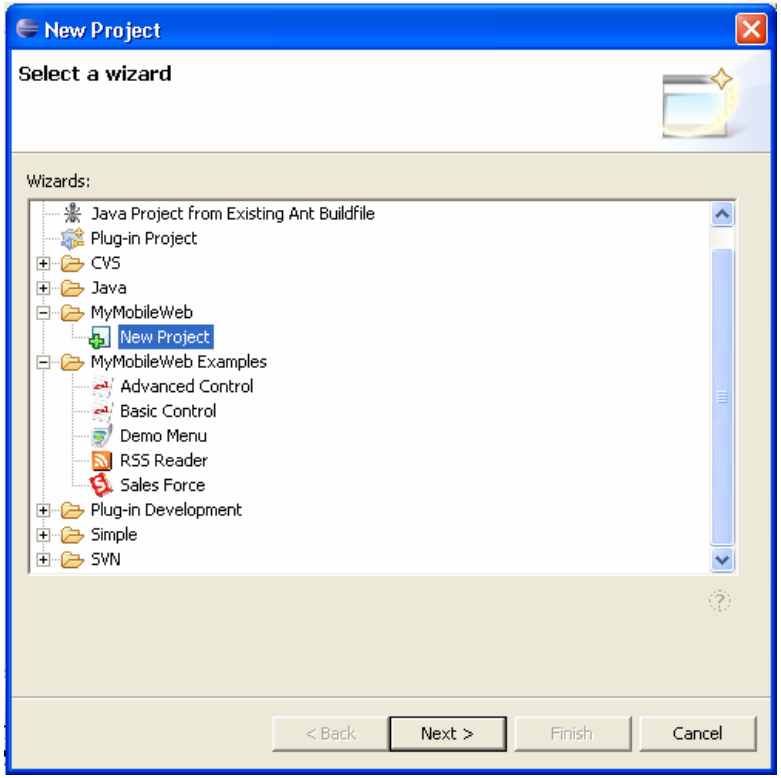
Development cycle (II)

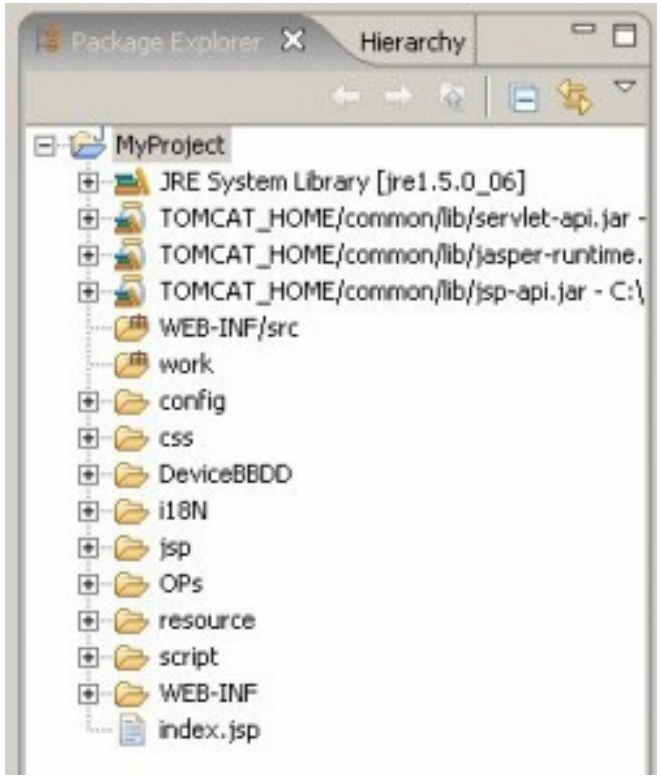
- MyMobileWeb applications are conventional Web J2EE applications that use additional libraries (in WEB-INF/lib)
- Presentation layer need to be defined declaratively using XML + CSS.
- A code generation tool need to be run to create
  - JSP pages, that will render the user interface specified in XML
  - Validation descriptors which will be used to perform automatic server-side validations
  - A set of Javascript functions in charge of performing local validations for devices with client-side scripting capabilities
- At runtime, these JSP pages, in collaboration with a set of libraries and components, will be in charge of rendering the presentation according to different delivery contexts.

Development cycle (III)

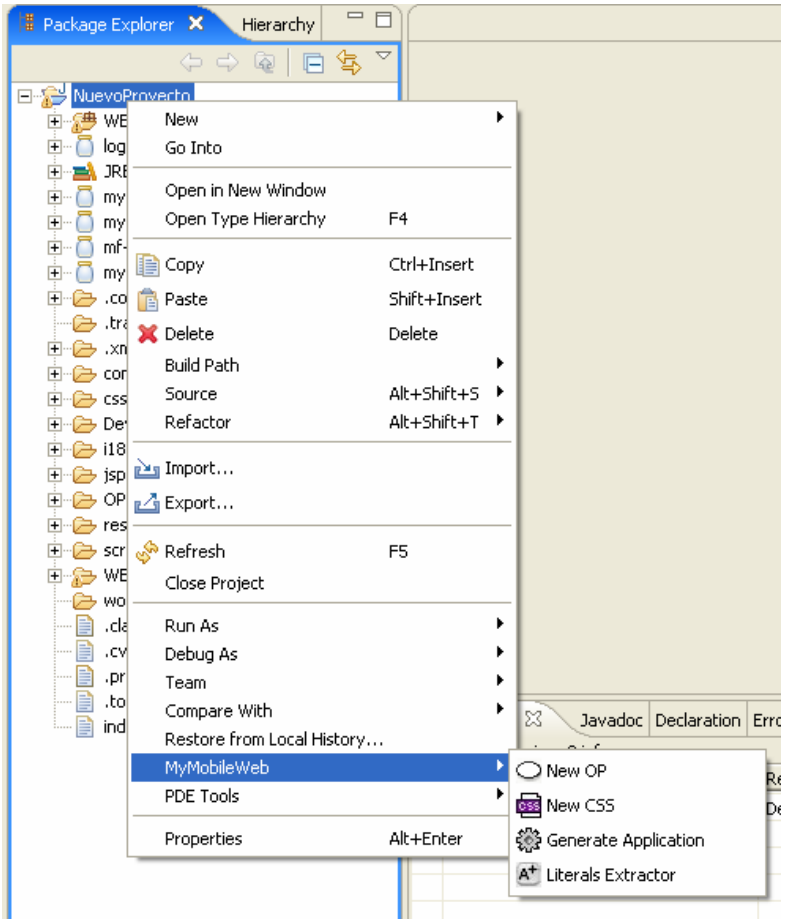
- This architecture, based on the pre-generation of pages, provides the best performance, as there is no execution of server-side XSLT code.
- A JSP page is not generated for each device, only are generated as much JSP's as needed, typically one for each markup technology.
  - If there are redefinitions of presentations or styles, then more JSP pages are generated to satisfy the requirements.
- Besides, developers need to specify:
  - Event handlers (Java classes) for the treatment of events triggered by each visual control.
  - Application Operations (OAs) (Java classes) that will be in charge, when needed, of getting new data or performing business logic operations (in collaboration with services).
  - Application resources, typically tiny images that will decorate the user interface. Developers can provide multiple variants of a resource and MyMobileWeb will select the best instance for each delivery context.
    - For more complex scenarios, MyMobileWeb provides an image transcodification component capable of adapting resources dynamically.

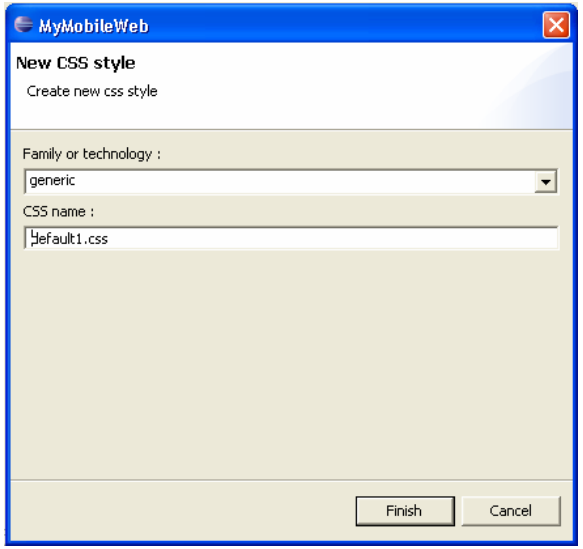
Development tools - Eclipse Plugin (I)

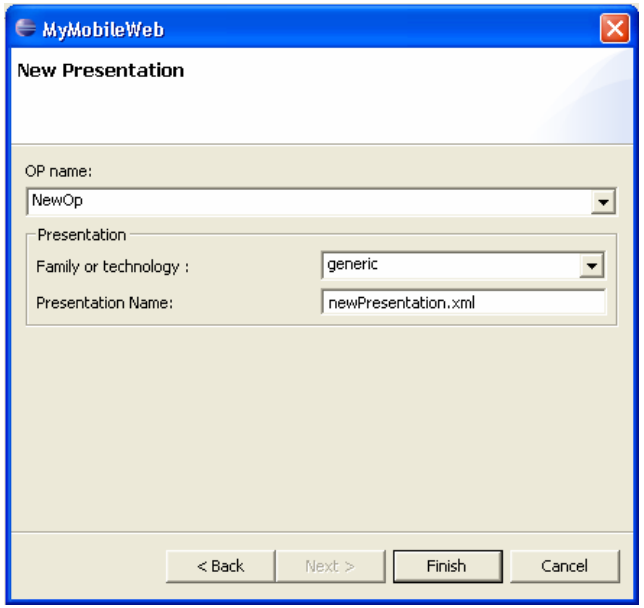




Development tools – Eclipse Plugin (II)







Example (I) - MyMobileWeb XML authored unit

```
<?xml version="1.0" encoding="UTF-8" ?>
<?xml-stylesheet href="example1.css" type="text/css"?>

<mymw:document xmlns:mymw="http://morfeo-project.org/mymobileweb" id="menu">
  <mymw:head>
    <mymw:title style="include">SalesForce</mymw:title>
  </mymw:head>
  <mymw:body>
    <mymw:p id="p1" style="nowrap expand" layout="vertical">
        <mymw:menu id="m1" style="mymenu paginate" bind="${menuOption}">
            <mymw:link resourceid="findps" id="cps" longtitle="The Find P/S option is implemented"
                       style="consult">Find P/S</mymw:link>
            <mymw:link resourceid="searchclients" id="cc" longtitle="Search Clients in our database"
                       style="clients">Search Clients</mymw:link>
            <mymw:link resourceid="contract" id="contr" longtitle="Do you want to contract one product?"
                       style="contract">Contract</mymw:link>
            <mymw:link resourceid="opinion" id="opin" longtitle="To know our clients"
                       style="opinion">Opinion poll</mymw:link>
            <mymw:link resourceid="exit" id="exit" longtitle="See you soon!" style="exit">
                       Exit</mymw:link>
        </mymw:menu>
    </mymw:p>
  </mymw:body>
</mymw:document>
```

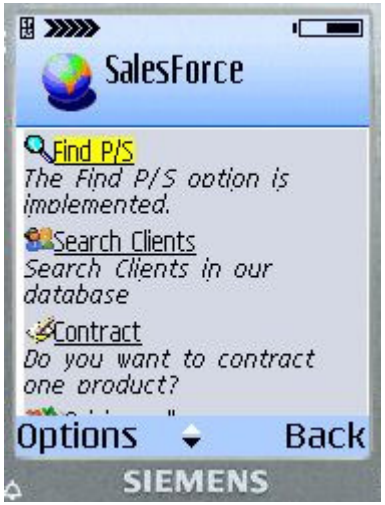
Example (II) - MyMobileWeb CSS style sheet

```
p.nowrap {
    white-space:nowrap;
}
p.expand {
    expand:true;
}
menu.mymenu {
    layout:vertical;
    align:left;
    white-space:nowrap;
    width:90%;
}
menu.paginate {
    paginate:false;
}
```

```
link {
    img-display:both;
}
link.longtitle {
    include:true;
}
link.contract {
    localsrc:149;
}
link.consult {
    localsrc:510;
}
link.clients {
    localsrc:163;
}
```

```
link.exit {
    localsrc:roundarrow2;
}
link.opinion {
    localsrc:chart;
}
menu.title {
    color:white;
    background-color:#336699;
    align:center;
    show-separator:false;
    include:true;
}
```

Example (III) - Rendering in different browsers

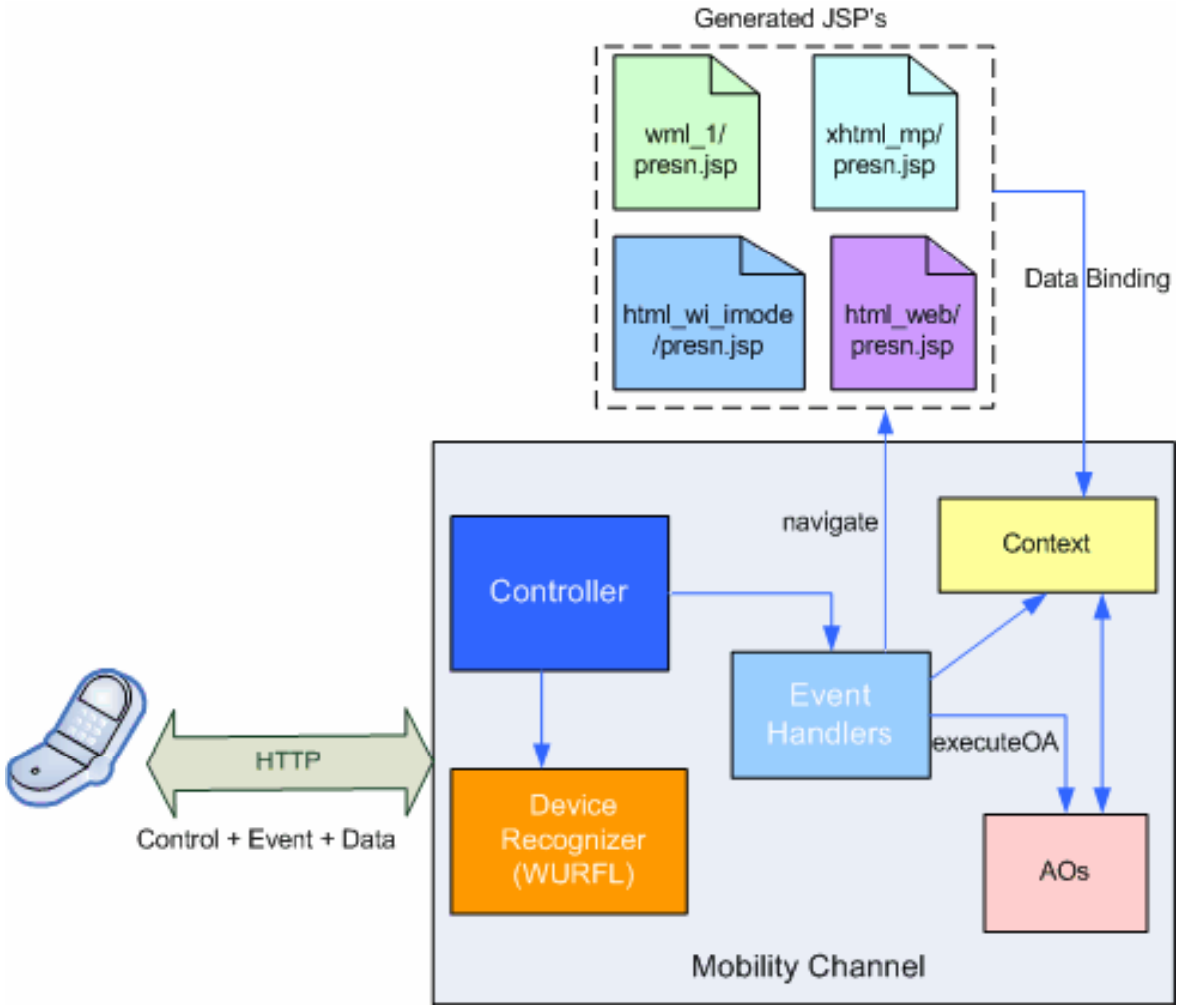


Example (IV) - Explanation

- An XML authored unit defines a user interface view
  - In the example it is defined a tiny screen with a set of menu options to be presented to the user. The menu is inside a container with vertical layout
  - Each menu option has an identifier, short and long descriptions
- The final aspect of the menu is controlled by means of a CSS style sheet declared at the top of the authored unit
  - CSS styles can be redefined for different devices or clusters of devices
- The data binding is specified by means of the attribute bind which defines the model variable that will be automatically updated with the option selected by the user (link identifier).
  - When the user selects an option, it will be raised an event that will be treated at the server side by means of handlers provided by the developer.
  - The event handler will find the option selected by the user in the bind variable menuOption.

Table of Contents

High Level Execution Architecture (I)



High Level Execution Architecture - Explanation (II)

- At the device side, when a user interaction causes an event to be treated at the server side, it is sent an HTTP request which contains:
  - the identifier of the visual control that raised the event.
  - the event identifier.
  - data that might have been entered by the user.
- At the server side, the request is initially processed by a Controller component, which recognizes the device (if it has not been done previously).
- Then, data is validated, and if validations are ok, data items are stored in the context. The context is a container that holds all the model data items, and it is structured hierarchically in different scopes (application, session, use case, view, etc.).

High Level Execution Architecture - Explanation (III)

- There are two kind of events sent from the device:
  - Application-specific events.
    - They have to do with the functionality of the application and are treated by specific handlers (methods of well-known Java classes) provided by programmers.
  - MyMobileWeb-specific events.
    - Handled automatically by MyMobileWeb, so application developers need not to worry about them.
      - For example, a next page event is raised when the user is paginating over the contents of a table.
- Application-specific event handlers decide on how to process an incoming request.
  - Typically they will call application operations (OAs) to get more data to be put into the model and, finally, a redirection to the next view (identified by a logical name) will be made.
    - At this point, MyMobileWeb will be responsible of locating the appropriate JSP page according to the delivery context.
      - This JSP page will render the presentation and will resolve all the data and content bindings with the help of runtime libraries.
- All event-handling related operations are accessible through an API provided by the MyMobileWeb MVC framework.

Mobile visual controls (I)

- MyMobileWeb provides a suite of mobile visual controls
  - A visual control is a user interface element of interaction that provides a well-defined functionality.
  - A visual control is finally rendered via a set of markup tags.
  - A visual control raises events, some of them managed by the application, some of them managed automatically by MyMobileWeb.
  - Any visual control might have a `display` attribute.
    - If the value of `display` is `true` then the control is rendered.
    - If it the value is `false` then the control it is not rendered.
    - `display` attribute can be an expression evaluated over the context.

Mobile Visual Controls (II)

- **hr** It puts an horizontal rule as a separator.
- **br** It is a line break in a presentation frequently used for better legilibility.
- **label**. It is used for labelling anything in an authored unit. It can contain any text. Using CSS it can be changed the aspect such as `color`, `font`, `underline`, `bold`, etc.
- **image** A control that displays a tiny picture in the screen. In MyMobileWeb images typically are specified by means of an abstract identifier, called `'resourceid'`. The programmer can provide different versions of the same resource and MyMobileWeb is in charge of selecting the best for a delivery context. Also MyMobileWeb provides an image transcoding module that can be used to dynamically transformate images to different formats and sizes as needed. Programmers or designers decide the schema to be used: content selection or transcoding.
- **object** It allows to put any multimedia object in a view. It has similar semantics than the `'object'` tag in (X)HTML.
- **entryfield** This is an input field in which the user can enter any text. It can be disabled or readonly. Developers can specify validations (`required`, `datatype`, etc) to be checked.
- **datefield** It is an input field that accepts a date as input. Depending on the delivery context it can be rendered as a calendar, as a set of input fields or as a wizard

Mobile Visual Controls (III)

- **timefield**. It represents an input field that accepts time as input (hours, minutes, seconds). It is parameterized by means of a mask.
- **textarea**. It is an area of free text (possibly with multiple lines). A textarea can also be `readonly` or `disabled`. If a textarea contains so much text for a given delivery context it will be automatically paginated.
- **link**. This is a hyperlink that can be optionally decorated with an image. Using style properties and depending on the delivery context it can be controlled if the image, the text or both are displayed.
- **select**. It allows the selection of one ore more values between a list of them. A select can be rendered as a `combobox`, `radio` buttons, `checkbox`, `menu`, etc. It depends on the style properties and the delivery context. When the user selects something no event is triggered.
- **menu**. It allows the selection of one between several items and, when done, automatically an event is raised. Menus can be rendered in different ways such as a list of links, a combobox sensitive to changes, a combobox plus a button, etc. Menus can have different layouts (`horizontal, vertical, grid`) and menu items could be decorated with an image or even be numbered with direct access through access keys. MyMobileWeb takes care of the pagination of the menu (if needed).
- **submit**. This control triggers an update of the model. When the user activates the control all the data in the current form on screen is posted to the server, starting the process of model (the context) updating, checking validations.

Mobile Visual Controls (IV)

- **action**. is a control to ask for something from the UI but no context updating occurs. Depending on the delivery context and the device it can be rendered as a button, as a link or even as an option mapped to a softkey (similar to the 'do' tag in WML).
- **list**. Represents data in list mode. Each list item can be decorated using an image or an index number. Lists are paginated automatically according to the delivery context.
- **table**. Represents data in tabular mode. Each column can be clickcable or not depending on the style settings. Tables are automatically paginated if needed. An interesting feature is that programmers can specify the visualization of more or less columns depending on the target delivery context.
- **chainedmenu**. It is a set of depending menus, i.e. the items of one menu depends on the value(s) selected in previous menus. A chained menu is rendered as a wizard in mobile phones and as a set of combos (popup lists) in PDAs.
- **rsspanel**. It is a control that provides the user interface for browsing and reading RSS feeds. The control is capable of dealing with different versions of ATOM and RSS.
- **telephonecaller**. A visual component to trigger a phone call from a user interface page. This control is an abstraction for developers that have not to worry about the different URL formats for specifying a call.
- **phonebookadder**. A visual component to add telephone number to my phone book. This control is an abstraction for developers that have not to worry about the different URL formats for specifying a phone book adder.

Styling

- MyMobileWeb has adopted CSS style sheets as the mechanism for specifying presentation aspects (look and feel, etc.) with respect to visual controls.
  - Each XML presentation file might be linked to one or more style sheets which must follow the CSS syntax.
  - The properties and selectors that can be used are those specified in W-CSS plus some specific extensions defined by MyMobileWeb.
- MyMobileWeb fully supports the cascading pattern for applying style sheets.
  - For instance, developers can specify a default style sheet that will apply to all the XML authored units.
  - Moreover, MyMobileWeb provides default CSS files that are applied when no style is specified by programmers.
- From a device independence perspective, the most important point is the style overriding feature.
  - Using this feature a developer can change specific style properties of an authored unit depending on the delivery context.
  - For example, if a developer realizes that a background color is not readable on a device, she can easily change it for one another more suitable, by means of style overriding and without needing to change anything in the XML authored unit.

Containers

- Visual controls need to be grouped into containers.
- MyMobileWeb provides two types of containers:
  - **p** It is a simple container that can only include visual controls (container nesting is not allowed). The layout can be `horizontal, grid` or `vertical`
    - A paragrah is paginated automatically, if it has too many (for a given delivery context) child controls inside it.
    - In that case the perceivable units by the user will be a set of screens chained by means of an interaction wizard.
  - **panel** It is a container that allows to switch between different subscreens, similar to 'tabs' in desktop applications.
    - There are different rendering options for a panel container: multiple cards in WML, anchors in XHTML-MP, link to each subscreen, etc. A panel can also include nested 'p' containers.
- MyMobileWeb also provides mechanisms for reusing fragments of user interface definition markup between different presentations.
  - This is accomplished by means of the include tag.
  - The inclusion mechanism is typically used for navigation bars ('back' and 'home' links) common to all the screens that compose an application.

Layout management

- For achieving device independence, the layout of containers need to be specified using CSS properties.
  - Using this method and the style overriding techniques provided by MyMobileWeb, a developer can alternate multiple layouts depending on different families of devices, tiny mobile phones, smartphones, PDAs, etc.
- For example, in a PDA a designer would like to see a grid layout, because she has plenty of screen space wherereas in a tiny mobile phone she would like to see a vertical layout.
  - Using MyMobileWeb, this can be achieved easily without duplicating any code and without additional effort by the developer.

Table of Contents

Example (I) - Menu control

```
<?xml version="1.0" encoding="UTF-8" ?>
<?xml-stylesheet href="menu.css" type="text/css"?>

<mymw:document xmlns:mymw="http://morfeo-project.org/mymobileweb" id="menu">
  <mymw:head>
    <mymw:title style="include">Example</mymw:title>
  </mymw:head>
  <mymw:body>
    <mymw:p id="p1" style="expand" layout="vertical">
        <mymw:menu id="ctr" title="Countries" style="countries" bind="${country}">
            <mymw:link resourceid="austria" id="au" longtitle="It's a landlocked country in Central Europe"
                       style="vertical">Austria</mymw:link>
            <mymw:link resourceid="usa" id="ee" longtitle="It's a country in North America"
                       style="vertical">USA</mymw:link>
            <mymw:link resourceid="france" id="fr" longtitle="His metropolitan territory is located in Western Europe"
                       style="vertical">France</mymw:link>
            <mymw:link resourceid="japan" id="ja" longtitle="It's an island country in East Asia"
                       style="vertical">Japan</mymw:link>
            <mymw:link resourceid="mexico" id="me" longtitle="It's located in Southern North America"
                       style="vertical">Mexico</mymw:link>
            <mymw:link resourceid="spain" id="sp" longtitle="It´s a country located in Southern Europe"
                       style="vertical">Spain</mymw:link>
            <mymw:link resourceid="sweden" id="ja" longtitle="It's a Nordic country in Scandinavia"
                       style="vertical">Sweden</mymw:link>
            <mymw:link resourceid="taiwan" id="me" longtitle="It's an island in East Asia"
                       style="vertical">Taiwan</mymw:link>
            <mymw:link resourceid="thailand" id="sp" longtitle="It lies in Southeast Asia"
                       style="vertical">Thailand</mymw:link>
            <mymw:link resourceid="turkey" id="ja" longtitle="It's a Eurasian country"
                       style="vertical">Turkey</mymw:link>
            <mymw:link resourceid="wales" id="me" longtitle="It's one of the four constituent nations of the UK"
                       style="vertical">Wales</mymw:link>
            <mymw:link resourceid="yemen" id="sp" longtitle="It's on the Arabian Peninsula in Southwest Asia"
                       style="vertical">Yemen</mymw:link>
        </mymw:menu>
    </mymw:p>
  </mymw:body>
</mymw:document>
```

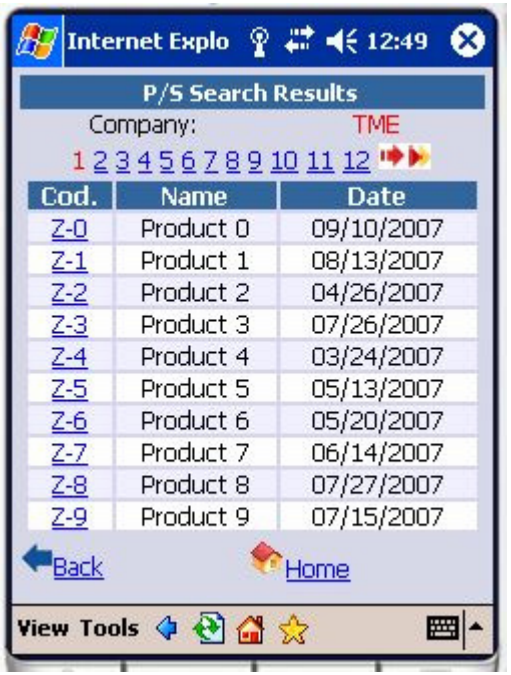
Example (I) - ... (rendering)

Example (II) - Table Control with binding and pagination

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet href="table.css" type="text/css"?>

<mymw:document xmlns:mymw="http://morfeo-project.org/mymobileweb" id="example">
  <mymw:head>
    <mymw:title style="include">P/S Search Res</mymw:title>
  </mymw:head>
  <mymw:body>
    <mymw:p id="p1" layout="vertical" align="center">
        <mymw:table id="myTable" bind="${selectedPS}" optionsbind="${searchPSResult}"
                                  keymember="code" paginate="true" style="body selcol0">
            <mymw:th style="headerfont headercolor">
                <mymw:td>Code</mymw:td>
                <mymw:td>Name</mymw:td>
                <mymw:td display="${_MYMW_DEV_BELONGS.PdaDevice}">Date</mymw:td>
            </mymw:th>
            <mymw:tr>
                <mymw:td member="code" />
                <mymw:td member="name" />
                <mymw:td member="date" />
            </mymw:tr>
        </mymw:table>
    </mymw:p>
    <mymw:include content="PSDetail/generic/product/p2"/>
  </mymw:body>
</mymw:document>
```

Example (II) - ... (rendering)



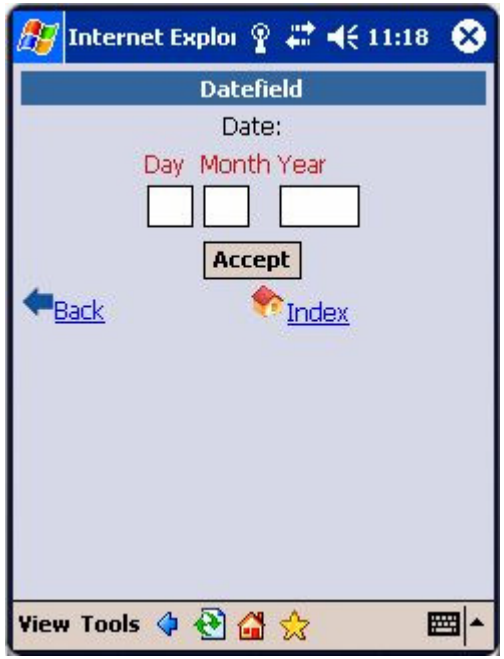
Example (III) - Datefield

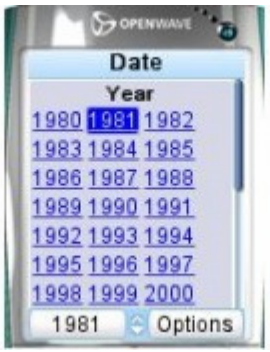
```
<?xml version="1.0" encoding="UTF-8" ?>
<?xml-stylesheet href="datefield.css" type="text/css"?>

<mymw:document xmlns:mymw="http://morfeo-project.org/mymobileweb" id="datefield">
  <mymw:head>
    <mymw:title>Datefield</mymw:title>
  </mymw:head>

  <mymw:body newcontext="true">
    <mymw:p id="p1" layout="vertical" align="center" style="nowrap" >
      <mymw:label id="date">Date:</mymw:label>
      <mymw:datefield style="mydate" labelid="date" id="date" bind="${ date }" />
    </mymw:p>
    <mymw:p id="p2" align="center">
      <mymw:submit id="submit" value="Accept" principal="true"/>
    </mymw:p>
    <mymw:include content="Entryfield/generic/example1/foot"/>
  </mymw:body>
</mymw:document>
```

Example (III) - ... (rendering)





Example (IV) – RSSPanel

```
<?xml version="1.0" encoding="UTF-8" ?>
<?xml-stylesheet href="feed.css" type="text/css"?>

<mymw:document xmlns:mymw="http://morfeo-project.org/mymobileweb"
               xmlns:user="http://morfeo-project.org/user" id="feed">
  <mymw:head>
    <mymw:title style="include">White House</mymw:title>
  </mymw:head>
  <mymw:body>
    <mymw:p id="p1" style="expand" layout="vertical">
        <user:rsspanel id="rss" src="http://www.whitehouse.gov/rss/news.xml"
                       paginate="true" />
    </mymw:p>
    <mymw:p id="foot" align="left">
        <mymw:link resourceid="back" id="feeds" style="both">Back</mymw:link>
        <mymw:link resourceid="index" id="index" style="both">Index</mymw:link>
    </mymw:p>
  </mymw:body>
</mymw:document>
```

Example (IV) - ... (rendering)

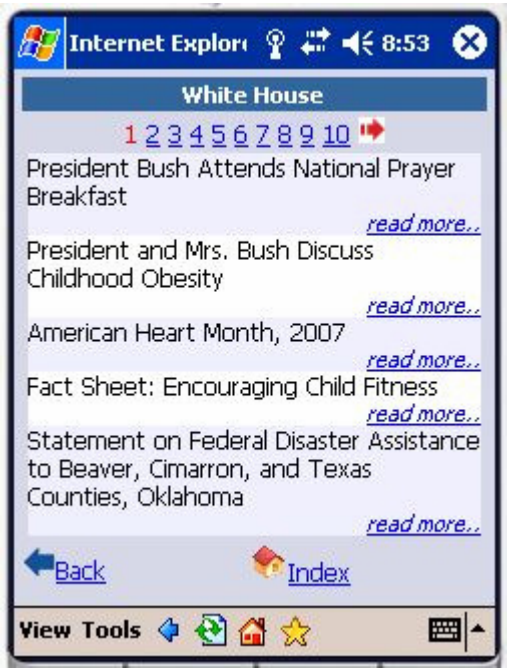


Table of Contents

Additional features

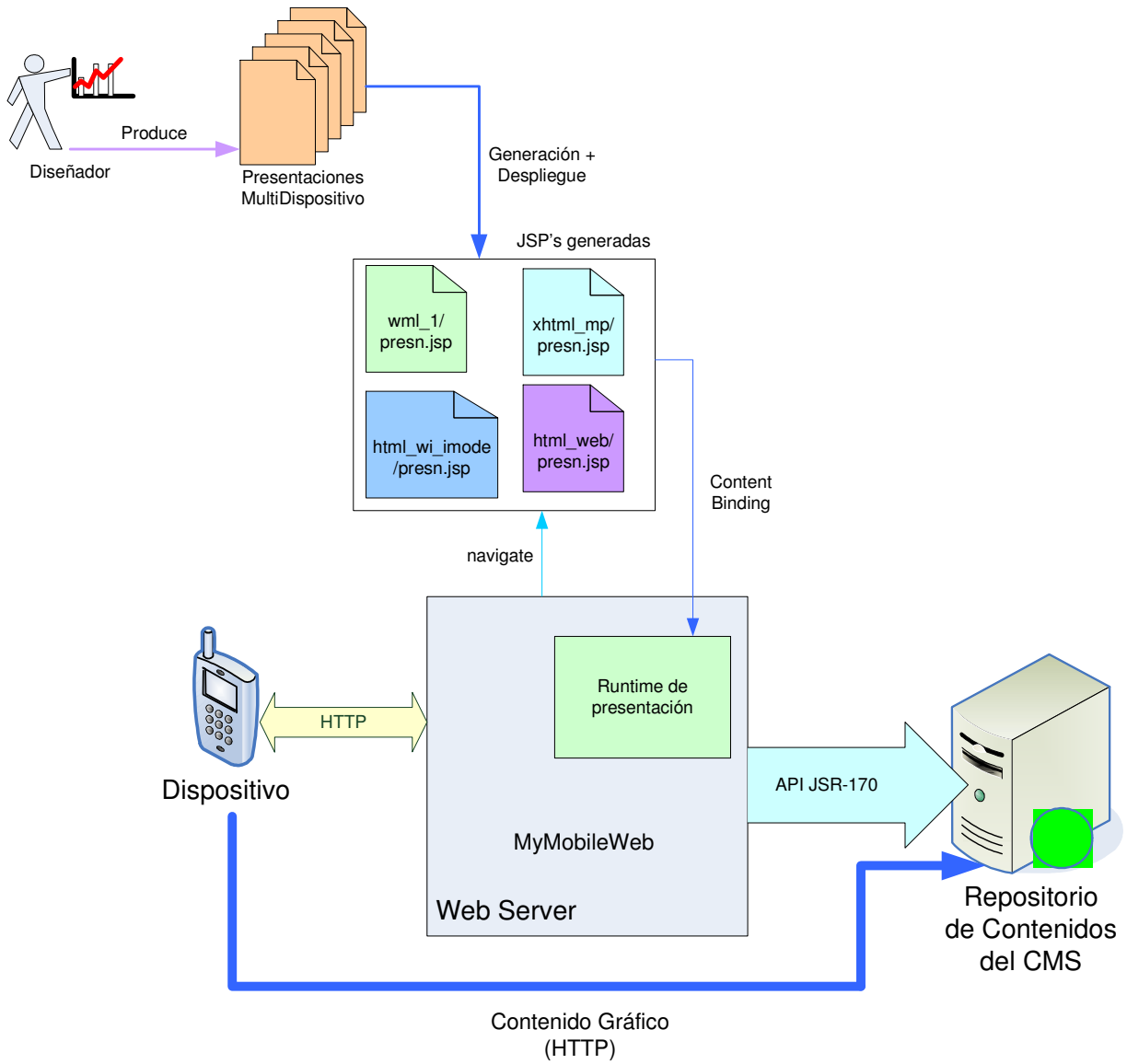
- Image transcoder module (dynamic adaptation of images)
  - Supports SVG also
- CMS integration (JSR-170)
- Semantic annotation of the user interface
  - Automatic form completion library
- Partial DCCI API implementation (Dynamic properties)
- Javascript modules for controlling the device
  - Screen orientation
  - Battery, Agenda, Phone Calling, GPS

Table of Contents

CMS and JSR-170

- Content Repositories and CMS Applications
  - *Content Repository*
    - A software component capable of performing store and retrieval functions over contents of different nature (documents, images, sound, video, text) and employing different media (FS, RDBMS, OODB)
  - CMS Applications
    - Provide a graphical front-end gráfico for the edición, publication and access to contents
    - Use the services provided by the repository
- JSR-170 is a standard Java API for accessing content repositories
  - implementation schema is similar in concept to JDBC
- There is a reference implementation of a JSR-170-compliant CR
  - Apache JackRabbit (open-source) http://incubator.apache.org/jackrabbit/

CMS integration architecture (JSR-170)



Content binding technology

- Content is associated to visual controls in a declarative manner
  - Avoiding developers to know all the details about JSR-170
- Contents are addressed following an URI-based schema
- Path based addressing
  - cmspath:repo:workspc/Demo/Images/telefonica
  - cmspath:repo/Demo/Images/telefonica
  - cmspath:/Demo/Images/telefonica
- UUID-based addressing
  - cmsuuid:/861735cf-068a-4981-9ba9-377301149557

Example : Image control

- Example 1
  - `<mymw:image src="cmspath:/Demo/Images/myimage" />`
  - The image will be the content of the specified node
  - Image transcoding can also be applied
- Example 2
  - `<mymw:image resourceid="cmsuuid: 861735cf-068a-4981-9ba9-377301149557" />`
  - The final content selected will depend on
    - the number of children of the specified node
    - the delivery context
  - The node should have one child for each version of the image (gif, wmbp, jpeg, ...)

Table of Contents

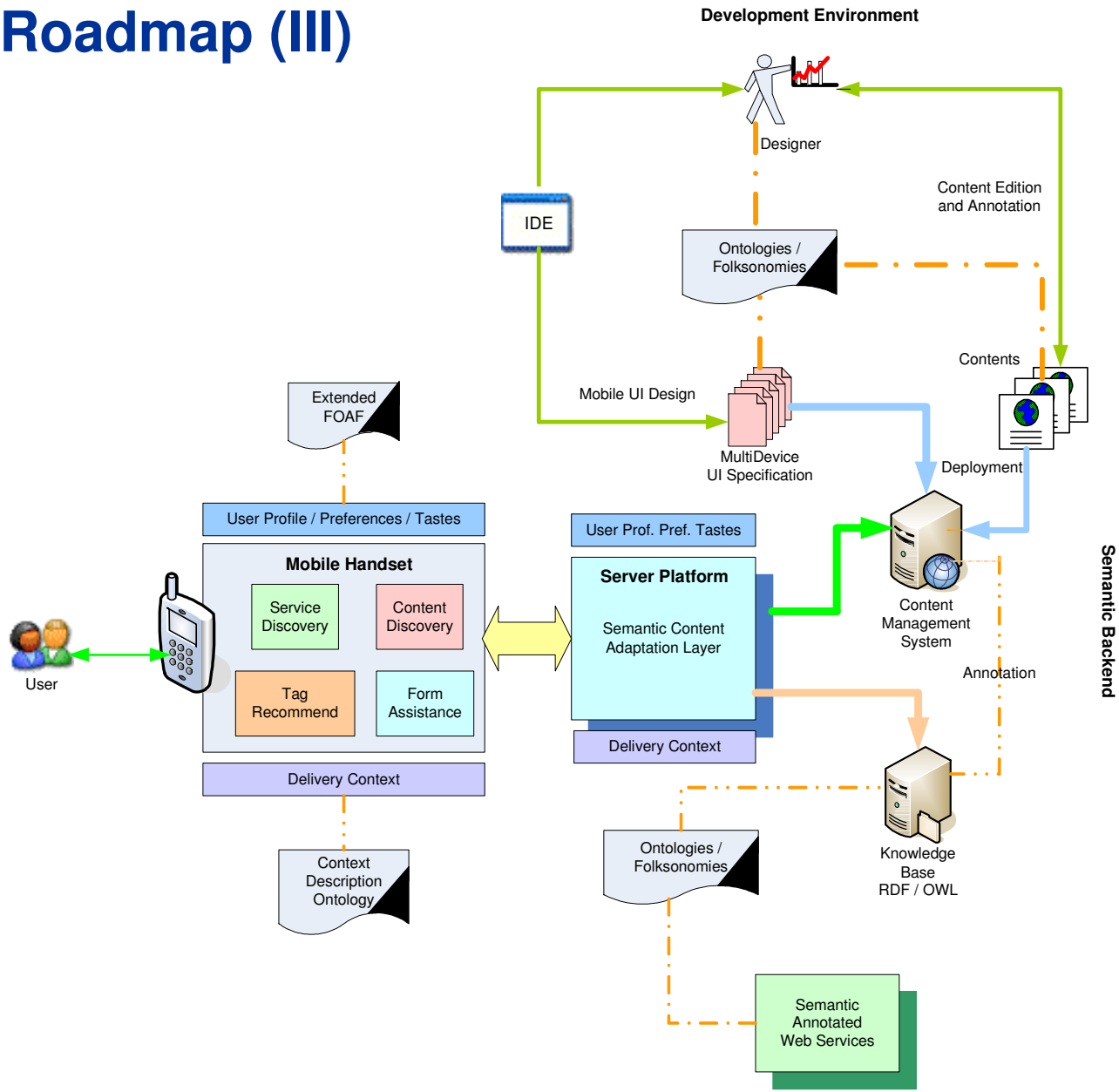
Roadmap (I)

- The project is being funded by the Spanish Government
  - "Proyecto tractor"
  - 9 partners (companies, SMEs and research institutions)
- Towards a "Eureka CELTIC" project (2007)
  - 15 partners
  - 4 countries (Spain, Italy, Germany, France)
- MyMobileWeb evolution is centered around
  - Tools
    - Graphical tool suite for designing the screens and flow
    - Plugins that assist the developer in creation and deployment of projects
  - Innovative standards development and adoption
    - Device Description Repository API
    - Declarative Format for Applications and User Interfaces
    - Device Indepedence Techniques and languages (DISelect, DIAL, ...)
    - Alliances with other communities (Apache)
  - Mobile Web 2.0 and 3.0

Roadmap (II)

- Development of the “Mobile Web 2.0 and 3.0” concept
  - Semantic Mobile Web that fully exploits ontologies and folksonomies
  - Ontologies / folksonomies for the description of user and delivery context
  - Semantic-based content adaptation and repurposing maintaining the thematic consistency
  - Semantic-based context-awareness
    - Exploiting semantic annotation of the user interface, contents and services
    - Content and service discovery depending on the user and delivery context (broadband, location, time, moment, objects nearby, ...)
  - Enrichment of the declarative language
    - Incorporate Mobile AJAX technology transparent to the developer
    - Incorporate rich multimedia contents using CDF standards
    - Incorporate declarative Multimodal Interaction (MMI) mechanisms
  - MyMobileSearch
    - Contextual search: Advanced algorithms that exploit page semantics, user and delivery context for producing optimized search results in the mobile web
    - Funded by Telefónica S.A.

Roadmap (III)


Development Environment
Designer
IDE
Content Edition
and Annotation
Ontologies /
Folksonomies
Contents
Mobile UI Design
MultiDevice
UI Specification
Deployment
Extended
FOAF
User Profile / Preferences / Tastes
Mobile Handset
Service
Discovery
Content
Discovery
Tag
Recommend
Form
Assistance
Delivery Context
User
User Prof. Pref. Tastes
Server Platform
Semantic Content
Adaptation Layer
Delivery Context
Content
Management
System
Annotation
Semantic Backend
Knowledge
Base
RDF / OWL
Ontologies /
Folksonomies
Context
Description
Ontology
Semantic
Annotated
Web Services

References

- MyMobileWeb is production-ready
  - Telefónica de España
    - Workforce Management System for Operation and Maintenance of fixed telephony network
    - 5000 users with different terminals (PDA, WAP, XHTML-MP ...)
  - BP
    - Petrol station auditory application
    - Light client (based on Pocket IE) and PDA MicroWebServer
    - User fills auditory forms off-line
    - Later, forms (stored as XML) are synchronized to backend
- There are ongoing projects that are using MyMobileWeb
  - andalucia.mobi
  - educamadrid.mobi
  - murcia.mobi (GoogleMaps and Google Earth integration)
- dotMobi Developers Portal
  - http://dev.mobi/node/208
  - http://dev.mobi/node/83

Downloads, documentation and support

- Downloads
  - Stable versions
    - http://forge.morfeo-project.org/frs/?group_id=28
  - Nightly Build (It might be unstable)
    - http://mymobileweb.morfeo-project.org/nightly.php
- Documentation
  - http://mymobileweb.morfeo-project.org/doc.php
  - Wiki: http://forge.morfeo-project.org/wiki/index.php/Plataforma_MyMobileWeb
- MyMobileWeb evolution maling list
  - mymobileweb-develop@lists.morfeo-project.org
- Help and support mailing list
  - mymobileweb-support@lists.morfeo-project.org

Who can participate in the MORFEO community?

Anyone who shares our dream and spirit …

"I am no longer captive to history.
Whatever I can imagine, I can accomplish"
Gary Hamel, "Leading the Revolution"

Visit our website !

http://www.morfeo-project.org

Telefonica
Telefónica Investigación y Desarrollo

germinus

YACO SISTEMAS

FUNDACIÓN CTIC
Centro Tecnológico de la Información y la Comunicación

ANSWARE TECHNOLOGIES

UNIVERSIDAD POLITECNICA MADRID
TECHNICA IMPENDI NATIONI

iWS

movistar