MyMobileWeb project's position

Workshop on Declarative Models of Distributed Web Applications

Dublin, 5-6 June 2007

José M. Cantera .- Telefónica I+D  jmcf@tid.es
Ignacio Marín .- Fundación CTIC  ignacio.marin@fundacionctic.org
Introduction
Developing applications for the Ubiquitous Web is hard. Main reason:

- (X)HTML is a general purpose language designed to create hypertext documents in the web, **but not for describing user interfaces**.

Scripting and server-side technologies “have filled the gap”, but:

Developers have always been demanding **more powerful abstraction mechanisms**. As a result, the market has responded with solutions:

- Ajax Toolkits
  - Dojo, Yahoo, GWT, ...
- Proprietary, tag-based, higher-level abstraction layers
  - JSF, XAML, XUL, Laszlo, MSXML

What about open standards? Alternatives (**all of them insufficient**):

- XHTML + XFORMS + Javascript and/or DIAL
- HTML 5 + Web Forms 2.0

**New standardization efforts are needed**

[http://morfeo-project.org](http://morfeo-project.org)
Why existing standards are insufficient?
Absence of a complete set of UI components

- grids, trees, menus, toolbars, progress bars, ...

No rich set of containers and layout abstractions.

- Developers end up using tables for layout → Not mobile nor accessible

No expression language notation for addressing objects

- Server-side scripts and Javascript → Lots of code to maintain
- XFORMS only works with XML data models and XPath.

They don't separate bindings, relevancy, formatting, validations.

No standard APIs for

- XFORMS elements, model mutation, creation of extensions

No sufficient mechanisms for specifying metadata or hints needed for adaptation to multiple delivery contexts
“A profile of XHTML 2” (DIAL) + DISelect:

- Goal: Content adaptation at server-side.
- It inherits all the problems coming from (X)HTML and XForms.

Some UWA use cases where DIAL + DISelect fails:

- Different layouts for different delivery contexts
- Date or time input component, rendered as a calendar or clock.
- Cool menus that degrade gracefully
- Select or menu component rendered as a popup list, or as list of links, or as a clickable map depending on the delivery context (device input mechanisms and browser capabilities).
- Big table or menu with dynamic contents that need to be paginated
- A big form which has to be paginated and divided in two or more chained subforms.
Web Applications specification developed by the WHATWG and candidate to be adopted by W3C

**Partial enhancements on**
- Validations
- Repetition model
- Extended elements (table, range, etc)

**Problems. All inherited from (X)HTML and more**
- Tag-soup reinvented → Not ready for enterprise development
- Backwards compatibility toll and browser vendor biases
- A rich component set is still missing
- Imperative against declarative : Scripting is encouraged
- 400 members to agree on something :(

http://morfeo-project.org
What is the trouble with existing, proprietary solutions?
Usage of AJAX toolkits is not transparent to the developer.

- They encourage imperative programming to the detriment of declarative formalisms
  - Example: declarative styling of components (CSS) no longer used.

Tons of Javascript code leads to

- bad performance and maintenance.
- applications not accessible nor friendly for mobile adaptation

They do not provide advanced standard UI mechanisms and formalisms (data binding, validations, formatting).

Extreme dependency on the selected AJAX toolkit

- Knowledge reuse and standardization are severely compromised.
- Writing new user interface components or extending existing ones is a difficult and tricky task.
A myriad of technologies, JSF, XUL, XAML, Laszlo, MXML ... all suffering from the same problems:

- Platform dependency: Java, .NET, Flash
- Openness: One implementation by one organization
- Desktop-orientation: Device independency was not a design goal
- Interoperability
- Reuse of user interface components between the different languages.

The open source community has started to understand the necessity of an open standard.

- Apache XAP project, leaded by NexaWeb Technologies
  - It defines the XAL language to reduce JavaScript in AJAX applications.
  - It can work with different AJAX Toolkits (Component Bridge Pattern)
MyMobileWeb: A successful implementation of a declarative language for developing applications and user interfaces in a multidevice environment
Low-cost, modular, open-standards-based, open source software platform intended to:

- **Simplify the development of top-quality mobile applications and portals**, providing an advanced adaptation environment.
- Aims at implementing the “Semantic Mobile Web” concept

**Declarative language (XML-based) for specifying the user interface in a device independent manner.** Features:

- UI components rendered in different ways depending on the delivery context
  - Grids, menus, etc and other specific to mobile: telephone call launcher ...
- Look & feel and adaptation policies are specified by means of CSS
- UI components grouped in containers with different layouts (CSS-specified) that can vary depending on the delivery context.
- JSP 2.0 expression language to resolve dynamic aspects
- Validation rules
- Data binding technology is widely used enabling advanced adaptation patterns such as pagination

[http://morfeo-project.org](http://morfeo-project.org)
Interface DEscription
Authoring Language
MyMobileWeb's language is a demonstration of the viability of using a declarative high level language for creating ubiquitous web applications.

**IDEAL** is a proposal for a new standard for Ubiquitous Web Applications.

- **Highly modular**, each module providing a well-defined functionality:
  - Core component set: Device Independent UI Components
  - Extended component sets: Desktop, mobile, set top boxes, etc.
  - Styling module (based on CSS syntax and properties, adding new needed properties)
  - Containers and layout managers (goal: to change UI layout without markup duplication)
  - Validation module (validation of user input; standard mechanisms for custom validations)
  - Binding module: simple and multiple binding for UI and binding policies: one way, two way, one time ...
  - Formatting module (for formatting information presented to the user)
  - Events module (based on XML Events and DOM Level 3 events, additions envisaged)
  - Model API (manipulation of data model and evaluation of expressions)
  - Expression language (interoperability with several expression languages - one being normative-)
  - Well-defined DOM-style API (to address use cases where declarative format is not sufficient)

- **Extensible** with well defined APIs for doing that

- **Interoperable** with XHTML, SVG → CDF

http://morfeo-project.org
Interface DEscription
Authoring Language

IDEAL Core

IDEAL Extensions (Standard)
IDEAL will be a highly modular language
IDEAL will be aligned with CDF specifications for example XHTML + IDEAL

IDEAL Extensions (User-defined, XBL or other)

Mobile Extensions
Desktop extensions

IDEAL Core

Device Independent UI components
Containers & LayoutManagers (spec. In a DI manner)
Events (DOM Level 3 + XML Events + Extensions)

Styling (CSS + Extensions)
Validation & Databinding & Formatters
Model API (IDL)

DOM API (IDL)

Each rounded box represents an IDEAL module
Modules can be extended DOM API can be extended

http://morfeo-project.org
How to develop IDEAL?
During 2006, Telefónica I+D and NexaWeb Technologies were working to come up with a **Declarative Format for Applications and User Interfaces (DFAUI)**

- Web Application Formats WG (WAF)
- **XAL and MyMobileWeb's language** were identified as starting points
- There is a requirements and use cases draft (not yet published)

However **we had to faced with**

- Hostile environment
  - Browser vendors saw the DFAUI as a competitor of WHATWG specs
- Absence of major players in this area IBM, Laszlo, Sun, Macromedia
  - Two companies is not enough for getting this work done
- Group members convinced that the gap should be covered with incremental add-ons to existing specs, such as HTML 5 for HTML

The work has stopped but **we want to resume it!!**

[http://morfeo-project.org](http://morfeo-project.org)
Alternatives to develop IDEAL (brainstorming)

**Within W3C**

- **Mechanisms**
  - A new language
  - An XHTML specific profile intended to develop UIs

- **Instruments**
  - “User Interface Incubator WG” → towards Recommendation
  - UWA WG
  - Rich Internet Application Backplane Taskforce
  - Joint task force: UWA + XFORMS + XHTML 2 + CSS WGs

**Outside W3C**

- Other SDOs interested in getting the work done
- Industry alliances
  - Open AJAX Alliance or other to be defined
- An open source project towards a “de facto” standard

[http://morfeo-project.org](http://morfeo-project.org)
Conclusions
Conclusions

- There is a gap wrt open, standards-based declarative models for UWA and, in particular, in the user interface area.

- Existing open standards are insufficient.

- AJAX and proprietary tag-based abstractions are more and more popular but create and extreme dependency on specific toolkits.

- MyMobileWeb: complete yet growing open source platform addressing some of the challenges found in declarative models for UWA.
  - UI declarative language independent of target device or interaction modes.
  - Language based on abstract UI components and containers, rendered in different ways (depending on delivery context).
  - Platform is working and ongoing projects use it. Correct approach.

- W3C to think of standardization of a new declarative language: IDEAL
  - modular, extensible and interoperable.

- Effort to be shared with the open-source community and industry alliances.

http://morfeo-project.org
Who can participate in the MyMobileWeb Project?

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/mymobileweb
Thank you for your attention!

MyMobileWeb project's position

Workshop on Declarative Models of Distributed Web Applications