XForms – the Future of XML Input

Mikko Honkala
Helsinki University Of Technology, Finland

Mikko.Honkala@hut.fi
http://www.x-smiles.org/
Outline of the Presentation

• Introduction to XForms
• Application areas - how can XForms help application development
• Implementation approaches & differences in them
• XForms implementation in X-Smiles
• Demos
Introduction to XForms
Introduction to XForms

- World Wide Web Consortium (W3C) creates Web recommendations.
- XForms
  - W3C Candidate Recommendation
  - Future Web forms technology. First requirements from september 1999.
  - Ideas from proprietary form languages (FML, Formsheets, XFA, XFDL).
  - Builds upon pre-existing XML technologies
What’s wrong with HTML forms?

• HTML Forms have not changed in ~8 years
• Forms defined as name=value pairs
  – Dynamic repeating constructs impossible
  – No multi-page or wizard type of interfaces
  – No datatype, validation or calculation support
• must be done at the server
  – more round-trips & reduced usability
• or programmed with scripts
  – reduced accessibility & maintenance
• Tied to single language: HTML
W3C XForms language

- Sends and receives structured data
  - XML instance data
- Form controls are bound to instance data
- Dynamic calculations and validation in the client
- Uses host language for the document layout
  - Can be almost any XML document language
- Advanced UI features
XForms document layers

• **Instance** – content (XML instance)
• **Model** – validation, constraints, calculations
• **User Interface** – the form controls embedded in a host language
• In addition:
  - **Binding** – binding between the instance, model and UI
• Can be seen as a MVC
## Purchase Order

<table>
<thead>
<tr>
<th>Units</th>
<th>Item</th>
<th>Price/unit</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>X-Smiles desktop</td>
<td>50 mk</td>
<td>150 mk</td>
</tr>
<tr>
<td>1</td>
<td>X-Smiles PDA</td>
<td>500 mk</td>
<td>500 mk</td>
</tr>
<tr>
<td>1</td>
<td>Java debugger</td>
<td>1500 mk</td>
<td>1500 mk</td>
</tr>
</tbody>
</table>

**Subtotal**: 2150  
**Taxes**: 473  
**Total**: 2623
XForms Example: Instance Data

```xml
<purchaseOrder>
  <items>
    <item>
      <name>X-Smiles desktop</name>
      <units>2</units>
      <price>50</price>
      <total>0</total>
    </item>
    <item>
      <name>X-Smiles PDA</name>
      <units>2</units>
      <price>100</price>
      <total>0</total>
    </item>
    <item>
      <name>Java debugger</name>
      <units>4</units>
      <price>75</price>
      <total>0</total>
    </item>
  </items>
  <totals>
    <subtotal>0</subtotal>
    <tax>0</tax>
    <total>0</total>
  </totals>
  <info>
    <tax>0.22</tax>
  </info>
</purchaseOrder>
```
<head>
  <xfm:model schema="purchaseOrder.xsd">
    <xfm:instance href="purchaseOrderData.xml" />
    <xfm:bind nodeset="purchaseOrder/totals">
      <xfm:bind nodeset="subtotal" calculate="sum(../../items/item/total)"/>
      <xfm:bind nodeset="tax" calculate="./subtotal * ../info/tax"/>
      <xfm:bind nodeset="/purchaseOrder/items/item/total" calculate="./units * ./price"/>
    </xfm:bind>
  </xfm:model>
</head>
XForms Example: The user interface

<body>
<table>
  <th>Units</th>
  <th>Item name</th>
  <th>Price</th>
  <th>Total</th>
</table>

<xfm:repeat nodeset="/purchaseOrder/items/item">
  <tr>
    <td>
      <xfm:range ref="units">
        <xfm:hint>Enter the quantity of this item.</xfm:hint>
      </xfm:range>
    </td>
    <td>
      <xfm:output ref="name"/>
    </td>
    <td>
      <xfm:output ref="price"/>
    </td>
    <td>
      <xfm:output ref="total"/>
    </td>
  </tr>
</xfm:repeat>

<xfm:output ref="/purchaseOrder/totals/total">
  <xfm:caption>Total price</xfm:caption>
</xfm:output>
</body>
Binding & Constraints using XPath

- XPath is a W3C recommendation
  - Developed mainly for XSLT but very general
- is used to:
  - select a single node from an XML document
    \[/purchaseOrder/items/item[1]\]
  - select multiple nodes (nodeset) from an XML document
    \[/purchaseOrder/items/item\]
  - perform calculations with the data in XML
    \[\text{sum}(/purchaseOrder/items/item/total)\]
Calculated Properties

- Properties apply to instance data in the model
  - **calculate** – parts of data calculated from other parts
  - **relevant** – is the item shown to the user
  - **readonly** – can the user edit the item
  - **constraint** – data is valid when this constraint is met
  - **required** – is the data required for submission
- Calculated properties can refer to other parts of the form data
- Uses XPath syntax

```xml
<bind nodeset="items/item/total"
  calculate="../units * ../price"
  relevant="../units>0" />
```
Validation Using XML Schema

- XML Schema is a W3C recommendation from 2001
  - Defines the structure of an XML document as well as datatypes
    - 'xsd:date' (1999-05-31)
    - 'xsd:time' (13:20:00.000)
    - 'xsd:decimal' (-123.4)
  - Datatypes can be created by the user with restrictions and unions
    - e.g. An integer smaller than 1000
  - Datatypes are more important to XForms
    - It is also possible to use a simpler 'schema for instance' syntax
XForms User Interface

- Very general level of user interface controls
  - input
  - select
- Label is associated with the control
- Can be used also in non-graphical environments (e.g. VoiceXML)
- Additional presentation hints with CSS stylesheets and presentation parameters
Datatype-aware form controls

- The form controls adapt to the datatype of the bound instance item.
- E.g. `<input>` bound to `xsd:date`, will be shown as a calendar control with the current locale, and `xsd:boolean` as a checkbox.
Advanced UI: Repeat

- Repeating user interface constructs
- Repeats the contained markup for each of the referenced nodes

```xml
<xforms:repeat nodeset="section">
  <xforms:input ref="name"/>
</xforms:repeat>

- Dynamic
  - Keeps UI and the instance in sync
  - Inserting and deleting items
- Similar to `<xslt:for-each>` but dynamic
Advanced UI: Switch

- Allows switching between different parts of the UI
  
  `<switch>`
  
  `<case id="case1">...`  
  `<case id="case2">...`

- Only one case active at a time

- `<toggle>` action used to toggle between cases

- Usages:
  - multi-page forms
  - tabbed user interfaces
Submission

- Multiple submission types
  - Posting XML document using HTTP POST / PUT
  - Legacy support: HTTP Post / Get
  - PUT XML files into filesystem
- Allows submission of only part of the instance data
- The reply does not necessarily replace the whole document as in HTML
Application areas - how can XForms help application development
Desktop applications in the web

- Create “desktop-like” applications in the Web
  - Document editing
  - Spreadsheet-type
  - Wizards / Dialogs

- Increased
  - **Usability** (e.g. dynamic calculations and validations)
  - **Maintenance** (no scripting)
  - **Accessibility** (no scripting)
Client (XForms processor)

- XForm
  - XML Schemas
  - XML Instances
- UI

Server

- XML Schemas
- XML Instances
- UI definitions
- XML DB

Application logic

HTTP
New platforms (Digi-TV, mobile devices)

- XForms UI defined in abstract level
- Single form can be show in
  - Different devices
    - Desktop
    - Digi-TV
    - Mobile device
  - Different modalities
    - Screen
    - Speech
As a tool in desktop application development

- Using a XForms implementation as a embedded component in a normal desktop application
  - Easier UI implementation effort
  - Easier migration to WWW
  - Easier maintenance
    - Document based development
SOAP front-end

• SOAP is a upcoming technology for creating machine usable services in the Web
• XForms could send receive simple SOAP requests
• Service discovery & configuration (WDSL) could be integrated with XForms
  – As an extension
  – In a later version of XForms
Implementation approaches & differences in them
Implementation approaches

- Plugin in a normal browser (No implementations currently)
- Language implementation in a normal browser (Formsplayer)
- Dedicated browser (X-Smiles)
- XForms as an application development component (Handwise)
- Server-side XForms (Chiba open source)
Current implementations

- X-Smiles – [www.xsmiles.org](http://www.xsmiles.org)
- Novell XForms preview - [www.silverstream.com](http://www.silverstream.com)
- Formsplayer – [www.formsplayer.com](http://www.formsplayer.com)
- Handwise XFUI - [www.handwise.com](http://www.handwise.com)
- Mozquito WebAccess- [www.mozquito.com](http://www.mozquito.com)
- E-XML Media XFE – [www.e-xmlmedia.com](http://www.e-xmlmedia.com)
- And more...
X-Smiles implementation
X-Smiles Browser

• Open Source
• XML Browser
  – XSL FO, SVG, SMIL, Xforms.
  – XSLT Transformations
  – ECMAScript
• Java-based
  – portability, available components, JMF
• Virtual prototype
  – Desktop, digi-TV, PDA, mobile phone
XForms Implementation in X-Smiles

• The first browser implementation

• Supports most of the XForms features:
  – validation
  – calculations
  – form controls
    • datatype adaption
  – events
  – dynamic binding

• We are co-specifying XForms
Embedding in different Markup languages

- XForms is designed to be used with another XML document layout format.
- Host languages in X-Smiles
  - Scalable Vector Graphics (SVG)
  - Synchronized Multimedia Integration Language (SMIL)
  - XML reformulation of HTML (XHTML)
  - XSL Formatting objects (XSL FO)
**X-Smiles extension: XML Signature**

- XML Signature & XForms integration
  - Custom action, that allows the user to sign the form using XML Signature
  - Signs both UI and the instance data
    - What-you-see-is-what-you-sign

Submission

- Instance data
- Signature
  - Reference to instance (inline)
  - Reference to UI (URI)
- UI Document
Level of support / Future work

- 80% of CR spec implemented
- Rest will be implemented in few following months
- Platforms
  - Runs well on desktop
  - Performance / size issues in smaller devices, since needs Xerces & Xalan for schema & xpath support
  - Digi-TV components will be integrated in the future
Thank you!