XForms for Web Services
Data In, Forms Out!

Kevin E. Kelly
kevin.kelly@us.ibm.com
Agenda

- Introduction
- XForms
- Auto Generation of XForms
- Demo
- Web Services invoked from XForms
- Demos
- Q&A
Introduction

- Work
  - Co-op'ed at IBM
  - US Air Force Officer
  - Automotive Embedded Startup
  - Rational Software
  - Acquired by IBM

- Standards
  - Chair Compound Document Format Working Group, was also in the XForms Working Group
  - Hypertext Coordination Working Group member
  - HL7 Advisory Council Rep to the W3C
What Are the Goals of XForms?

- Rich, XML-based, forms to meet the needs of business and consumer web applications
- Support for desktop browsers and mobile devices
- Decoupled data, logic, and presentation
- Reduce the need for script
- Support for structured form data in all XML
- Advanced forms logic
  - Bind, process, submit
- Seamless integration with other XML tag sets
  - Compound Documents
What Are XForms?

- XForms leverage XML in modeling, collecting, and serializing user input
- Enable forms designer to use a standard set of components and off-the-shelf tools for developing web applications
- Produces a user interface appropriate for the connecting device
- Provides interactive feedback via client-side validation
- Standards-based, and capable of embracing many devices
  - No proprietary vendor lock-in
Auto Generation of XForms

- Data in, Forms out!
- Since an XForm can use a separate XML instance as its data model
  - An XForm can be generated from a data model or prototypical XML instance
- The backing referenced schema(s) can be used to check for types (int, boolean, enumerations) and multiplicity (maxOccurs) to generate the appropriate form input control.
XML Forms Generator

XML Forms Generator plugin for Eclipse

XML Instance

Backin Schemas

XForm
Simple Input Field

Schema

```xml
<xsd:element name="OrganizationName" type="xsd:string"/>
```

XML Instance

```xml
<OrganizationName>Hewitt Associates LLC</OrganizationName>
```

XForm Input Markup

```xml
<xforms:input ref="#xml:OrganizationName" model="model_Enrollment">
    <xforms:label>Organization Name</xforms:label>
    <xforms:hint>Enter a value for OrganizationName of type string</xforms:hint>
</xforms:input>
```

XForm Rendered

```
| Organization Name | Hewitt Associates LLC |
```

Kevin E. Kelly – XForms for Web Services
Enumeration Generation

```xml
<xsd:element name="EntityRole">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="Plan Sponsor"/>
      <xsd:enumeration value="Insurer"/>
      <xsd:enumeration value="Broker or Sales Office"/>
      <xsd:enumeration value="First Party Administrator"/>
      <xsd:enumeration value="Second Party Administrator"/>
      <xsd:enumeration value="Third Party Administrator"/>
      <xsd:enumeration value="Other Affiliate"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>

<EntityRole>Third Party Administrator</EntityRole>
```
<xforms:select1 ref="hrxml:EntityRole" model="model_Enrollment">
  <xforms:label>Entity Role</xforms:label>
  <xforms:item>
    <xforms:label>Plan Sponsor</xforms:label>
    <xforms:value>Plan Sponsor</xforms:value>
  </xforms:item>
  <xforms:item>
    <xforms:label>Insurer</xforms:label>
    <xforms:value>Insurer</xforms:value>
  </xforms:item>
  <xforms:item>
    <xforms:label>Broker or Sales Office</xforms:label>
    <xforms:value>Broker or Sales Office</xforms:value>
  </xforms:item>
  <xforms:item>
    <xforms:label>First Party Administrator</xforms:label>
    <xforms:value>First Party Administrator</xforms:value>
  </xforms:item>
  <xforms:item>
    <xforms:label>Second Party Administrator</xforms:label>
    <xforms:value>Second Party Administrator</xforms:value>
  </xforms:item>
  <xforms:item>
    <xforms:label>Third Party Administrator</xforms:label>
    <xforms:value>Third Party Administrator</xforms:value>
  </xforms:item>
  <xforms:item>
    <xforms:label>Other Affiliate</xforms:label>
    <xforms:value>Other Affiliate</xforms:value>
  </xforms:item>
  <xforms:hint>Enter a value for EntityRole of type string</xforms:hint>
</xforms:select1>
**Enumeration Generation (Continued)**

XForm Rendered

<table>
<thead>
<tr>
<th>Entity Role</th>
</tr>
</thead>
<tbody>
<tr>
<td>Third Party Administrator</td>
</tr>
<tr>
<td>Plan Sponsor</td>
</tr>
<tr>
<td>Insurer</td>
</tr>
<tr>
<td>Broker or Sales Office</td>
</tr>
<tr>
<td>First Party Administrator</td>
</tr>
<tr>
<td>Second Party Administrator</td>
</tr>
<tr>
<td>Third Party Administrator</td>
</tr>
<tr>
<td>Other Affiliate</td>
</tr>
</tbody>
</table>

© Copyright 2006, IBM Corporation
Repeat Group Generation

```xml
<xsd:element name="IdValue" maxOccurs="unbounded">
    <xsd:complexType>
        <xsd:simpleContent>
            <xsd:extension base="xsd:string">
                <xsd:attribute name="name" type="xsd:string" use="optional"/>
            </xsd:extension>
        </xsd:simpleContent>
    </xsd:complexType>
</xsd:element>

<IdValue name="Mutually Defined">382235791</IdValue>
<IdValue name="Exclusively Defined">382235785</IdValue>
```
Repeat Group Generation (Continued)

```
<xforms:repeat model="model_Enrollment" id="repeat_IdValue_model_Enrollment"
  nodeset="hrxml:IdentificationCode/hrxml:IdValue">
  <xforms:input ref="." model="model_Enrollment">
    <xforms:label>Id Value</xforms:label>
    <xforms:hint>Enter a value for IdValue of type string</xforms:hint>
  </xforms:input>
  <xforms:input ref="./@name" model="model_Enrollment">
    <xforms:label>Name</xforms:label>
    <xforms:hint>Enter a value for name of type string</xforms:hint>
  </xforms:input>
</xforms:repeat>

<xforms:group>
  <xforms:trigger>
    <xforms:label>Add Id Value</xforms:label>
    <xforms:insert ev:event="DOMActivate" at="index('repeat_IdValue_model_Enrollment')"
      position="after" nodeset="hrxml:IdentificationCode/hrxml:IdValue"/>
  </xforms:trigger>
  <xforms:trigger>
    <xforms:label>Delete Id Value</xforms:label>
    <xforms:delete ev:event="DOMActivate" at="index('repeat_IdValue_model_Enrollment')"
      nodeset="hrxml:IdentificationCode/hrxml:IdValue"/>
  </xforms:trigger>
</xforms:group>
```

<table>
<thead>
<tr>
<th>Id Value</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>382235791</td>
<td></td>
</tr>
</tbody>
</table>

**Id Value**

**Name**

Mutually Defined

<table>
<thead>
<tr>
<th>Id Value</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>382235785</td>
<td></td>
</tr>
</tbody>
</table>

**Id Value**

**Name**

Exclusively Defined
XForms Bind

- **Types**

  - `<xforms:bind nodeset="/formBeginDate" type="xsd:date"/>

  - FormBeginDate is type xsd:date

- **Constraints**

  - `<xforms:bind constraint="string-length(.) &lt; 31"
    nodeset="/firstNameAndInitial" required="true()"/>

  - The firstNameAndInitial must be < 31 characters in length and cannot be blank

- **Calculations**

  - `<xforms:bind calculate="/wagesSalariesTipsTotal + /taxableInterest + 
    /unemploymentCompensation" nodeset="/adjustedGrossIncome"/>

  - AdjustedGrossIncome = wagesSalariesTipsTotal + taxableInterest + unemploymentCompensation
XForms Bind

- Relevant

```xml
<xforms:bind nodeset="/otherDiscuss" type="xsd:boolean"/>
<xforms:bind id="designee" nodeset="/designeesName" relevant="otherDiscuss='true'"/>
<xforms:bind nodeset="/areaCode" relevant="/otherDiscuss='true'"/>
<xforms:bind nodeset="/phonePrefix" relevant="/otherDiscuss='true'"/>
<xforms:bind nodeset="/phoneSuffix" relevant="/otherDiscuss='true'"/>
<xforms:bind nodeset="/designeesPIN" relevant="/otherDiscuss='true'"/>
```

- OtherDiscuss is type boolean
  If otherDiscuss is true then make relevant designeesName, areaCode, phonePrefix, phoneSuffix, and designeesPIN otherwise they are all not relevant

- (Most renderers hide them if they are not relevant, but they could be grayed out or something else too)
XForms Submit

- Submission
  
  - `<xforms:submission id="submit_form" action="http://xformstest.org/cgi-bin/showinstance.sh" method="post"/>
  
  - Define a submission action

    Submit

  - `<xforms:submit submission="submit_form">
    <xforms:label>Submit</xforms:label></xforms:submit>

  - Associate a submit button with the submission "submit_form"
When to Use Script?

- Long “select-one” choices
  - Usually implemented as a drop-list box
  - Script use to narrow choices for 3..n levels
  - Even in disconnected use mode, vocabularies can be local
  - Hierarchical enumerated lists

- “Special” Processing
Auto Generation of XForms

- HR-XML Enrollment Form with X-Smiles
- Firefox Samples, Tax Form with Firefox
XForms for Web Services

- Since WSDL is XML and it describes the input and expected output for a web service
  - It can serve as the data model for constructing a form that calls the web service on submission
Options for XForm Generation

- 3 approaches to discuss and demo (others are possible) for XForms-based form generation for web services:
  - Direct XHTML/XForm
  - Response JSP and Router Servlet
  - Request and Response XHTML/XForm
Direct XHTML/XForm

- Interrogate the WSDL
- Generate an XML data model instance derived from the WSDL
  - Simple SOAP envelope wrapping input parameters
- Generate the XForm from the XML data model instance (using same process as last demo)
  - But, not generating input fields for SOAP elements
- The submit is a call to the web service itself
- Direct XHTML/XForm

- Google Search with Firefox and XForms extension
Response JSP and Router Servlet

- Raw XML returned from web service is not the best user experience.
- The XForm can instead target the submit to a generic servlet that calls the web service and passes the returned information to a JSP generated from the same WSDL that was used to generate the XForm data model.
Response JSP and Router Servlet

XML Forms Generator

- WSDL
- XML
- XHTML / XForm

XForm Generation from Web Service

Input Form

Submit

Browser/Renderer

Server

Router Servlet

JSP

SOAP
XML

Publish to server

SOAP
XML

Output Page

Browser/Renderer

XHTML

SOAP
XML

SOAP
XML
- Response JSP and Router Servlet

Demo

Google Search with Firefox and XForms extension
Request Response XHTML/XForm

- JSP and Router Servlet require server-side logic and is not purely declarative
- XForms supports a switch-case construct that allows for rendering of different pages, which can be attached to the same data model which may be updated from the response of a web service call
- Retains an all declarative approach
- Request/Response XHTML/XForm

Google Search with Firefox and XForms extension
Summary

- XForms is useful for auto generation of forms for web services
  - Low footprint web enablement for data input
- 3 approaches for using XForms for web service invocation when manual input is required:
  - Direct XHTML/XForm
  - Response JSP and Router Servlet
  - Request and Response XHTML/XForm
Summary

- These approaches, as presented, will not work against all web service engines from an XForms 1.0 renderer due to the inability to set the SOAPAction HTTP header which many web services engines require for routing the service invocation.

  - This drawback is fixed in XForms 1.1 where SOAP is supported as a submit protocol.
Resources

- Eclipse 3.2 and Web Tools Project 1.5
  - www.eclipse.org .. click on Downloads tab
  - www.eclipse.org/webtools .. click on Download Now!

- XML Forms Generator

- Firefox 1.5.0.6 and XForms 0.7 extension
  - http://www.mozilla.com/
  - https://addons.mozilla.org/firefox/824/

- X-Smiles 1.0 alpha 1
  - http://www.xsmiles.org/

- Apache Tomcat 5.0
  - http://tomcat.apache.org/
Acknowledgments

- XML Forms Generator
  - Jan Kratky, Keith Wells, Steve Speicher

- Firefox XForms extension
  - Aaron Reed, Steve Speicher, Allan Beaufour, Olli Pettay, Doron Rosenberg, Merle Sterling
Questions
Thank You