6 XML

6.7 XForms

Motivation

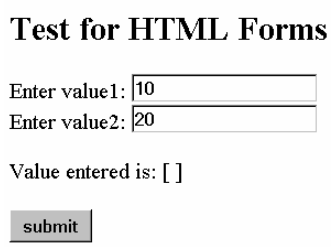
- HTML forms deliver the interactive WWW
- Forms are what make electronic transactions possible
- HTML forms are long overdue for an overhaul

1993:	**HTML forms**
1994–2001:	**Nothing happened**
2002:	**XForms**

HTML Form Example

```
<html>
  <head>
        <script language="JavaScript">
        </script>
  </head>
  <body>
        <h2>Test for HTML Forms</h2>
        <form name="firstForm" action="">
              Enter value1: <input id="val1" value="10"/><br/>
              Enter value2: <input id="val2" value="20"/><br/>
              <div id="htmlOut"><p>Value entered is: [ ]</p></div>
              <input type="submit" value="submit"/>
        </form>
  </body>
</html>
```

HTML Form Example



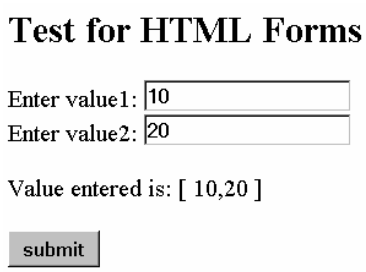
HTML Form with Script

```
<html>
<head>
<script language="JavaScript">
  function check() {
     if ((firstForm.val2.value - 0) > (firstForm.val1.value -0)) {
       htmlOut.innerHTML = "<p>Submitted values: “+firstForm.val1.value+","+firstForm.val2.value+"</p>";
       firstForm.submit();
     }
     else {
       htmlOut.innerHTML = "<p>Error: value 1 must be less than value 2 ( “
                                    +firstForm.val1.value+","+firstForm.val2.value+" )</p>";
      }
      alert("click to continue");
   }
  function display() {
     htmlOut.innerHTML = "<p>Value entered is: [ "+firstForm.val1.value+","+firstForm.val2.value+" ]</p>";
  }
  </script>
  </head>
```

HTML Form with Script (cont.)

```
<body onload="display()">
        <h2>Test for HTML Forms</h2>
        <form name="firstForm" action="" onsubmit="check()">
                Enter value1: <input id="val1" value="10" onchange="display()"/><br/>
                Enter value2: <input id="val2" value="20" onchange="display()"/><br/>
                <div id="htmlOut"><p>Value entered is: [ ]</p></div>
                <input type="submit" value="submit"/>
        </form>
  </body>
</html>
```

HTML Form with Script



Problems with HTML Forms

- Some Assembly Required
- Primitive Data Representation
- Data & presentation intertwined
- Need Script to do anything…
  - Validations
  - Calculations
  - Dynamic Forms
- Medium specific
- High cost of application development and support

How does XForms solve these problems?

- Content is separated from the presentation
- Presentation is defined elsewhere in the document, it is only bound to the content
- Validation is done in the client using XML schema and inter-data constraints
- Constraints and calculations are defined declaratively in the markup. Let the XForms processor implement them rather than program them in JavaScript
- XForms capable client receives and sends XML directly

XForms Design Goals

- Be a good XML citizen
  - Submit well-formed XML
  - Build on other XML vocabularies
- Anywhere, anyone, any time, any device
  - Support for desktop browsers, handheld, phones, ATMs, iTV, etc…
- Minimize need for Scripting
- From simple client/server to n-tier
  - Decoupled data, logic and presentation

XForms Design Goals – cont'd

- Accessible
- Not a standalone document type
  - Re-usable module
- Hosted by other document types
  - XHTML 1.1

XForms are the next generation of Web forms

- XForms is a W3C Recommendation
  - XForms 1.0 became a W3C Recommendation in October 2003
  - http://www.w3.org/TR/2003/REC-xforms-20031014/
- XForms provides a richer, more secure, more reliable, and presentation independent way of handling interactive Web transactions
- Future e-commerce solutions are expected to demand the use of XForms-enabled browsers
  - all major browsers will support XForms in the near future

XForms Conformance Profiles

- Two conformance profiles to support wide range of devices
- Basic: small devices, TV, phones, etc.
- Full: desktop browsers, servers

XForms Architecture

- Model Definition
  - Default data
  - XML Schema references
  - Business rules
- View Definition
  - UI characteristics (HTML/CSS)
- Controller Definition
  - View to Model bindings
  - Submission declaration

XForms Form Example <head>

```
<xforms:model id="MyDocument" schemas="local.xsd"
                                        xmlns="http://verifone.com/isd/NAXML/XFormsTest">
        <xforms:instance>
                <MyDoc>
                        <Val1>10</Val1>
                        <Val2>20</Val2>
                        <Res/>
                </MyDoc>
        </xforms:instance>
        <xforms:submission ref="MyDocument" id="s00"/>
        <xforms:bind type="xs:integer" nodeset="/MyDoc/*"/>
        <xforms:bind ref="/MyDoc/Val2" constraint="/MyDoc/Val2 &gt; /MyDoc/Val1" />
        <xforms:bind nodeset="/MyDoc/Res" calculate="/MyDoc/Val1 * /MyDoc/Val2"/>
        <xforms:action ev:event="naxml-setToOne">
                <xforms:setvalue ref="/MyDoc/Val1">1</xforms:setvalue>
                <xforms:setvalue ref="/MyDoc/Val2">1</xforms:setvalue>
        </xforms:action>
</xforms:model>
```

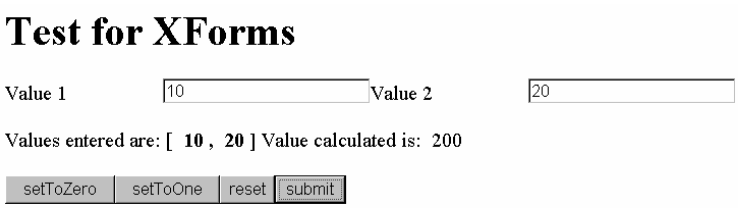
XForms Form Example <body>

```
<p>
        <xforms:input ref="/MyDoc/Val1">
                <xforms:label style="width:150px;">Value 1</xforms:label>
                <xforms:hint>Enter the first value, which must be less than the second</xforms:hint>
        </xforms:input>
        <xforms:input ref="/MyDoc/Val2">
                <xforms:label style="width:150px;">Value 2</xforms:label>
                <xforms:hint>Enter the second value, which must be greater than the first</xforms:hint>
        </xforms:input>
</p>
<p>
        Values entered are: [ <b><xforms:output ref="/MyDoc/Val1" />&amp;nbsp;,&amp;nbsp;<xforms:output
        ref="/MyDoc/Val2" /> ]</b>
        Value calculated is: <xforms:output ref="/MyDoc/Res"/>
</p>
```

XForms Form Example <body>

```
<xforms:trigger>
        <xforms:label>setToZero</xforms:label>
        <xforms:action ev:event="xforms-activate">
                <xforms:setvalue ref="/MyDoc/Val1">0</xforms:setvalue>
                <xforms:setvalue ref="/MyDoc/Val2">0</xforms:setvalue>
        </xforms:action>
</xforms:trigger>
<xforms:trigger>
        <xforms:label>setToOne</xforms:label>
        <xforms:action ev:event="xforms-activate">
                <xforms:dispatch target="MyDocument" name="naxml-setToOne"/>
        </xforms:action>
</xforms:trigger>
...
<xforms:submit submission="s00" class="submit">
        <xforms:label>submit</xforms:label>
</xforms:submit>
```

XForms Form Example

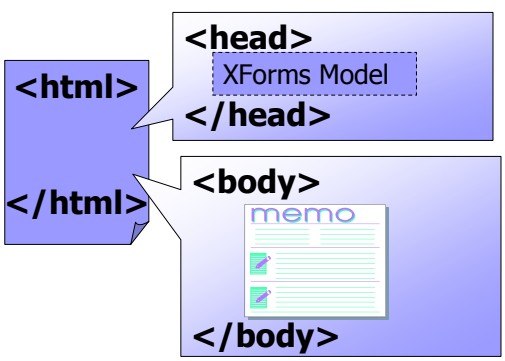


Terminology

- XForm or XForms?
- XForms Processor
- XForms Model
- Instance data
- Containing document
- Form control

How It Works

An Xforms form has two distinct parts:



Purpose
(non-visible)

Presentation
Form controls

XForms Model

- Defines the 'Purpose' of the form
  - i.e. presentation independent
- Includes:
  - Instance Data
  - XML Schema that constrains instance data
  - XForms (dynamic) constraints
  - Submit Information
  - Privacy information (P3P)

XForms Instance

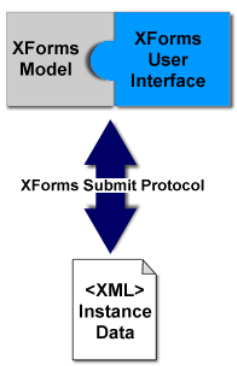
- Provides a template for data
  - default or partially submitted data
  - Can be inline or referenced externally
  - Used to create an "instance DOM"
- "instance DOM" continually updated
- A subset of the "instance DOM" is serialised and submitted

XForms Without Scripting

- XML Schema defines static constraints
  - Datatypes
  - minimum/maximum occurrences
- XForms extends these with ‘computed expressions’ evaluated at runtime
  - Is something relevant or required?
  - Calculated fields
  - XPath expressions

XForms Submission

- <submitInfo> specifies
  - What – a portion of the “instance DO
  - Where – target URI
  - How – serialisation format & transmi
  - Response – life after submit
- Default is ‘post’ encoded in XML



XForms Model - illustration

```
<model id="p1">
  <schema src=". . ." />
  <instance xmlns="">
  <my:age/>
  </instance>
  <bind ref="age" type="xsd:integer" . . . />
  <submission action=". . ." />
  <privacy src="policyref . . ." />
</model>
```

XForms UI

- Create a user interface
- "connect" the user interface elements to the appropriate data in the model
- Create interactive "views" of a model

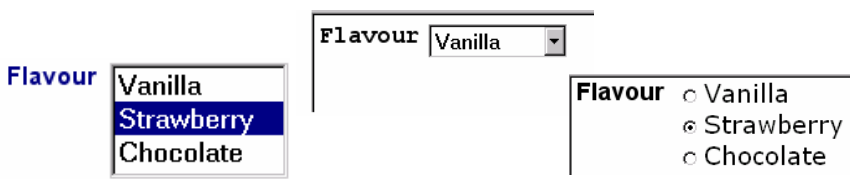
XForms Form Controls

- Bind to the model
- Metadata for the user
- Shortcuts & navigation hints
- Presentation hints
- CSS styling

<input>	<output>	<secret>
<textarea>	<range>	<upload>
<select>	<select1>	<submit>
	<button>	

XForms UI Controls

```
<select1 ref="my:icecream/my:flavor">
  <caption>Flavour</caption>
  <item><caption>Vanilla</caption><value>v</value></item>
  <item><caption>Strawberry</caption><value>s</value></item>
  <item><caption>Chocolate</caption><value>c</value></item>
</select1>
```



XForms UI

- Form controls bind to atomic data types
- Aggregations create sophisticated user interfaces
  - Obviating common uses of scripting

Construct	Purpose
<group>	Group related controls
<switch>	Conditional for dynamic UI
<itemset>	Dynamic selections
<repeat>	Repeating structures

XForms repeat - illustration

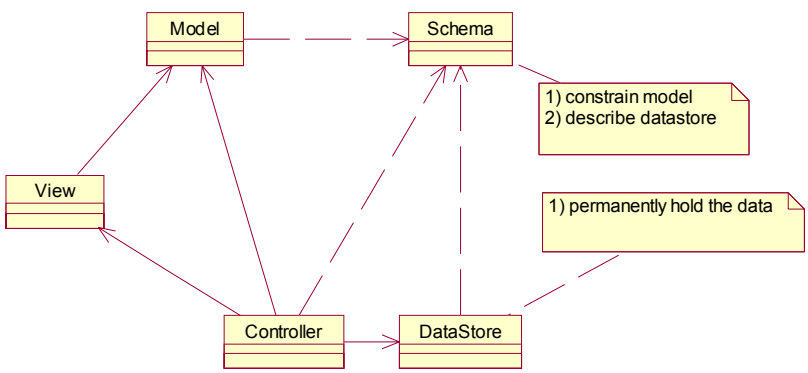
```
<repeat nodeset="/cart/item">
  <h:tr>
    <h:td>
      <input ref="product" />
    </h:td>
    <h:td>
      <input ref="description" />
    </h:td>
  </h:tr>
</repeat>
```

XForms Actions

- Declarative markup for event handlers
  - Uses XML Events
  - Reduces need for scripting

dispatch	refresh	recalculate	revalidate
setfocus	load	setvalue	send
reset	setindex	insert	delete
toggle	script	message	action

Classical Model View Controller (MVC)



XForms Deployment

- XForms can be implemented
  - In a client
  - In a server
    - Can deliver legacy markup to clients which lack native support
- Enables a front end to Web Services
- An XForms processor can be at multiple points in the network

References

- W3School XForms Tutorial
  - http://www.w3schools.com/xforms/default.asp
- W3C XForms page
  - http://www.w3.org/MarkUp/Forms/
- Several online presentations
- formsPlayer
  - http://www.formsplayer.com

Reading List

- W3School XPath Tutorial
  - http://www.w3schools.com/xpath/default.asp