How I Learned to Stop Worrying and Love the Data Model

Doug Tidwell
IBM Corporation
dtidwell@us.ibm.com
Objectives

- Introduce XForms, an open standard for business forms
- Discuss the advantages of declarative programming
- Illustrate the differences between XForms and more traditional AJAX approaches
- Examine ways to deploy XForms
The paradigm

- If you need to do CRUD operations with an XML document type, use XForms.
  - An XForms document is a custom application built around your XML document.
XForms

The next generation of forms on the Web
XForms

- A W3C effort to “bring HTML forms into the 20th century”
- Forms should be integrated with data models
- One form should be renderable on many types of devices
- Use a declarative model as much as possible
HTML forms - weaknesses

- Based solely on appearance
- Non-Western character support is spotty
- Cross-device forms are very difficult to write
- *Reliance on scripting 😞*
- No real data model
XForms at the W3C

- XForms Version 1.0 became an official W3C Recommendation in October 2003.
  - The latest revision to V1.0 was in March 2006.
- Version 1.1 is currently in Last Call at the W3C.
  - The last draft was published in February 2007.
Advantages of XForms

- XForms has an underlying data model that can be generated from your existing business objects.
- XForms has an underlying data model that can be generated from your existing business objects.
- XForms has an underlying data model that can be generated from your existing business objects.
Why is XForms so important?

- We’re creating more new XML data than new relational data
  - Growing at 2X rate of total database market [IDC]
- All major relational database vendors have some form of XML support
- XML is now pervasive in many organizations
- Almost every sector has XML based standards
Structure of an XForms file

- XForms tags are just XML tags imbedded in a standard XHTML file with a different namespace
- Most HTML form tags are exactly the same
Separation of concerns

- Model is in the header (non-visual section)
- Visual components are in the body (presentation)
XForms is model-driven

- XForms enables the developer to reuse business rules encapsulated in XML Schemas (XSD) and XML Transformations (XSLT)
- XForms reduces duplication and ensures that a change in the underlying business logic does not require rewriting in another language
Model-View-Controller

- XForms uses a variation of the model-view-controller (MVC) design pattern
- The model has no user interface concepts
- The control layer moves data to and from the model
Strong typing

- Submitted data can be strongly typed if necessary and can be checked on the client
- Strong typing enables automatic client-side validation
  - When serving the same XForms document to an non-compliant browser, these constraints can be used to generate client-side JavaScript automatically
Beyond XML Schema

- XForms authors can go beyond the basic set of XML Schema constraints and add complex validation rules.
- XForms provides additional constraints as part of the Model.
- This enhances the overall manageability of the resulting Web applications.
Direct XML submission

- XML Forms can send XML data directly from the web client to the server
- Data can be validated directly in the client web browser
- Complex multi-part forms can be broken down into tabs but do not need re-fetching from the server
Direct XML submission

- There is no need for custom server-side logic to transform the submitted data to the business application if it already uses XML
  - XML submission is like AJAX++ - it has a lot of configurability. You can submit a fragment, prune elements, you can turn validation, etc. on or off in 1.1.
  - You can work with XML on the client side, but you can still submit to non-XML systems on the server side.
Device independence

- Abstract user interface controls lead to **intent-based authoring** of the user interface
- An XForms application can target many different devices
  - XForms can be deployed to a range of accessing devices
  - Look at PicoForms, Satec
A brief tour of XForms

- We’ll spend a few minutes looking at some of the capabilities of XForms.
  - The Tax Form and other samples are at mozilla.org/projects/xforms/samples.html
  - To see the Orbeon samples, install Orbeon, start your servlet container and go to http://localhost:8080/ops.
Declarative programming
Declarative programming

- Declarative programming says what has to be done; it doesn’t say how to do it.
- Works in the context of a particular problem
  - We want to get the information about a customer living in the United States, for example.
- The requirements of that problem can be encoded in a specialized language
  - Define that language as XML; from there, you can generate an XML Schema, the XForms user interface, test data, etc.
Problem context

- If we’re issuing an insurance policy to a person living in the United States:
  - We know the format of the person’s name
  - We know the format of the person’s address
- Maybe we need other details about that person
  - What organization they work for (we have discounts for employees of certain companies)
  - Are US citizens?
- We use datatypes to define the information
  - A person’s name is a string, their date of birth is a date, a yes or no question is a boolean, etc.
Defining business objects

- Given our knowledge of the context, we can define business objects. Here’s a person in the U.S.:
  - Title (optional): Dr., Mr., Mrs., Ms., Rev.
  - First name
  - Middle name (optional)
  - Last name
  - Suffix (optional): Jr., Sr., II, III, IV

- We can define all of these with XML Schema.
Generating user interfaces

- Given our business object, it’s easy to generate a user interface with XForms.
- Everything in the form is mapped directly to something in the XML schema.
XForms document structure
An XML document... is a tree

- Both the model and the views are trees of data elements.
The XHTML... is a tree

- Both the model and the views are trees of data elements.
The model... is a tree

- HTML
  - head
    - title
    - meta
    - style
  - xf:model
    - Person
      - Name
        - FirstName
        - LastName
    - Organization
      - Address
        - Street
        - City
        - State
        - Zip
Linking the model and view

Diagram:

- HTML
  - head
  - xf:model
  - body
  - form
    - fieldset
    - label
    - input
    - <bind>
  - <bind>

Model:

Person
  - Name
    - FirstName
    - LastName
Just do the right thing

```
<html xmlns:xlink="http://www.w3.org/1999/xlink">
<head>
  <meta charset="utf-8"/>
  <title>Person</title>
</head>
<body>
  <form>
    <fieldset>
      <label for="BirthDate">BirthDate</label>
      <input type="xs:date" id="BirthDate" value=""/>
      <label for="USCitizen">USCitizen</label>
      <input type="xs:boolean" id="USCitizen" value="false"/>
    </fieldset>
  </form>
</body>
</html>
```
XForms compared to AJAX
XForms compared to AJAX

- AJAX doesn’t have a built-in data model
- AJAX uses JavaScript for everything
- AJAX doesn’t automatically generate XML data
XForms has a data model

HTML

head

title

meta

style

xf:model

Person

Organization

Address

FirstName

LastName

Street

City

State

Zip
AJAX doesn’t
XForms doesn’t need JS

```html
<html>
  <head>
    <xf:model>
      <Person>
        <BirthDate type="xs:date"/>
        <USCitizen type="xs:boolean"/>
      </Person>
    </xf:model>
  </head>
  <body>
    <form>
      <fieldset>
        <label>
          BirthDate type="xs:date"
        </label>
        <input />
        <bind>
        </bind>
        <label>
          USCitizen type="xs:boolean"
        </label>
        <input />
        <bind>
        </bind>
      </fieldset>
    </form>
  </body>
</html>
```
AJAX has to use JS

```
FirstName = document.person.first;
LastName = document.person.last;
CurrentOnTaxes = document.person.cot.checked;
DateOfBirth = new Date(document.person...)
```
XForms generates XML

```xml
<Person>
  <Name>...</Name>
  <Organization>...</Organization>
  <USCitizen>true</USCitizen>
  <BirthDate>1995-04-21</BirthDate>
</Person>
```

BirthDate type="xs:date"

USCitizen type="xs:boolean"
AJAX has to use JS

// Build XML structure
//   by hand; use JSON? DOM?

// Create element "Person"
// Create element "FirstName"
// Create text element "Bob"
// Add text element as child
// of "FirstName" . . .
Building and deploying XForms
Building and deploying

- One of the challenges of deploying XForms is viewing them.
- We’ll look at three approaches:
  - Requiring a browser plug-in
  - Generating JavaScript from the XForms document
  - Using a standalone viewer
Forms Players

XForms Extension
FormPlayer
Presentation Server
OpenOffice.org
Mobile Client
AjaXForms
Oracle
Netfront Mobile
Orbeon
XForms Players

Rendered Form
See Wikipedia “XForms”
Using a browser plug-in

- Typically useful in an intranet
  - For Internet Explorer, there are some options:
    - FormsPlayer by Mark Birbeck
    - MozzIE, written by Peter Nunn [sourceforge.net/projects/mozzie](http://sourceforge.net/projects/mozzie)
Using Ajax controls

- The Chiba project provides an open-source XForms engine.
  - It uses XSLT to transform the XForms document into Ajax-enabled HTML pages and send them to the client.
- The Orbeon Presentation Server (OPS) is based on Chiba and provides similar functions.
- These are server-centric. They can generate some client-side validation.
Using a standalone viewer

A third option is to use a standalone viewer that renders the XForms and manages the data stored in the model.

- FormsPlayer is a good example of this approach.
- You can embed the FormsPlayer inside a .Net application.
A Web 2.0 UI
Data models

- Lost in all the excitement of Web 2.0 is the fact that our spiffy interfaces have to do something useful.

- You can build Web pages with JavaScript libraries to put a happy face on a business process
  
  Are these maintainable? Probably not.  
  
  We want these pages to be tied to our data model.
Our data models
The application

Round 1, Matchup 1

- [1] Donuts
- [16] Food
The application

How I learned to stop worrying and love the data model

Competitors

Round 1

Round 2

Round 3

Round 4

[1] Donuts

[16] Food

[8] Colorado Avalanche (hockey team)

[9] Colorado avalanche (natural disaster)

[9] Colorado avalanche (natural disaster)
The application

- I wrote an article about using XSLT to generate the bracket from the XML document:
  

- An article about building the XForms application will be published on dW soon.
The new paradigm

An XForms document is a custom editor for your XML document type.
Resources
Resources

- The XForms 1.1 spec:
  w3.org/TR/xforms11/ [Last call draft]

- XForms 1.0:
  w3.org/TR/xforms/

- Steven Pemberton’s XForms presentation on the W3C site:
  w3.org/2007/Talks/05-15-steven-xforms11/
Resources

- XForms.org
- XForms Institute: xformsinstitute.com/
- O’Reilly’s XML.org
- IBM developerWorks: ibm.com/developerworks/xml/xforms
Resources

- **MozzIE:**
  - [sourceforge.net/projects/mozzie](sourceforge.net/projects/mozzie)
  - Embeds the Mozilla Gecko rendering engine into IE

- **Firefox XForms add-on:**
  - Select Tools→Add-ons, then click Get Extensions, search for “XForms”
Resources

- FormFaces: formfaces.com
- Chiba: chiba.sourceforge.net
- Orbeon Presentation Server:
  ➢ Available at orbeon.com, open-sourced under the GPL
Resources

- FormsPlayer: 
  
  formsplayer.com

- Also see Mark Birbeck’s *Introduction to XForms* at: 
  
  formsplayer.com/introduction-to-xforms
Resources

- Visual XForms Designer:
  `ibm.com/alphaworks/tech/vxd`

- XML Forms Generator:
  `ibm.com/alphaworks/tech/xfg`
XForms Tutorial & Cookbook

- An excellent tutorial on XForms
- Includes dozens of working samples that illustrate the full power of XForms.
- [en.wikibooks.org/wiki/XForms](en.wikibooks.org/wiki/XForms)
Micah Dubinko was one of the editors of the XForms 1.0 spec.

You can get an online copy of his book at xformsinstitute.com/essentials/

The DocBook source is available as well.

A brief ethics test
Is it ethical... No!

- ...to mention the second edition of my book is on its way? (ISBN 0-596-51415-8)
- Lists for $49.99, but it’s a bargain at twice the price.
- Rough Cuts version available today!
Summary
Summary

- We’ve covered several important topics:
  - XForms, an open standard for forms
  - Declarative programming
  - Differences between XForms and AJAX
  - How to deploy XForms
Thanks!

Doug Tidwell
IBM Corporation

dtidwell@us.ibm.com