XForms

XML Powered Web Forms

T. V. Raman

IBM Research

Outline

- Motivation and goals.
- XForms model.
- XForms user interface.
- Creating and deploying XForms.

Motivation And Goals

Motivation

Bring benefits of XML to online forms.

- HTML forms power the interactive Web.
- Forms are key to online transactions.
- HTML forms in serious need of an overhaul.

HTML Forms Shortcomings

- No separation of data and presentation.
- Unstructured, losely typed user input.
- Custom code for data validation.
  - Client-side Javascript.
  - Server-side application logic.
- Business logic duplicated in multiple layers.

Increases cost of Web application deployment.

Goals

- Structured, strongly typed XML for user input.
- Bind *user interface* to data.
- Interaction using high-level UI vocabulary.
- Transmit and process user input as XML.

Goals

XForms Designed to embed and be embeddable.

- *Not* a stand-alone document type.
- Hosted by other document types:
  - XHTML 1.1
  - SVG
  - XHTML2 —Default forms module is XForms.

Re-usable forms module for XML vocabularies.

XForms Model

Model Components

⟨*model*⟩ *holds presentation independent components.*

- XML instance that holds user input.
- XML schema that constrains instance data.
- Dynamic runtime constraints.
- Submission metadata:
  - What, Where, How.

XForms Instance Data

⟨*instance*⟩ *declares instance data.*

- XML template with default values.
- Parsed to create instance DOM.
- Instance DOM continuously updated.
- Instance DOM Serialized during submission.

Instance data reflects state of user interaction.

Constraining The Instance

⟨*schema*⟩ *encodes static type constraints.*

- Atomic data types.
- Complex types for data structures.
- User defined types.

XForms re-uses XML Schema.

Dynamic Constraints

⟨*bind*⟩ *declares dynamic constraints.*

- Expressed as computed XPath expressions.
- Re-evaluated during user interaction.
- Add to and refine static schema constraints.
  - Relevant, Required, Calculate, Isvalid.

Submission

⟨*submission*⟩ *holds submission metadata*

- What —address portion of the instance tree.
- Where —specify target URI.
- How —serialization, transmission protocols.
- Response —Life after submit.

XForms Model

```
<model id="p1"
  schema="person.xsd">
  <instance>
    <person> ... </person>
  </instance>
  <bind .../>
  <submission .../>
</model>
```

XForms UI Controls

User Interface Controls

User interface vocabulary for intent-based authoring.

- Create user interface,
- Bind interaction to data,
- Create interactive *views* of the model.

Anatomy Of A UI Control

Declarative markup encodes salient features.

- Binding attributes that wire control to model,
- Metadata for giving feedback to the user,
- Keyboard shortcuts and navigation hints,
- Presentation hints,
- CSS-based styling.

XForms UI Controls

XForms defines the following UI controls.

input	secret	textarea
output	range	upload
select	select1	submit
trigger —Generic Widget		

Input Date

```
<input model="p1"
  ref="/person/birthdate"
  class="...">
  <label>...</label>
  <help>...</help>
  <hint>...</hint>
  <alert>...</alert>
  </input>
```

XForms User Interface

Higher Level Constructs

Complex user interaction created via aggregation.

- Basic controls bind to atomic data types.
- Aggregations create complex user interfaces.
- Obviate common uses of scripting.

Aggregation Constructs

Construct	Purpose
⟨group⟩	Group related controls
⟨switch⟩	Conditionals for dynamic UI
⟨itemset⟩	Dynamic selections
⟨repeat⟩	Repeating templates

Dynamic UI Using ⟨switch⟩

```
<switch id="sw">
 <case id="i" selected="true">
  <input ref="yourname">
   <toggle case="o" event="..."/>
  </input> </case>
 <case id="o" selected="false">
  <trigger id="edit">
   <toggle event="..." case="i"/>
  </trigger> </case>
</switch>
```

Dynamic Selection

Available choices are determined at runtime.

```
<select model="cart"
  ref="/order/item">
  <label>...</label>
  <itemset model="catalog"
    nodeset="/sf/asimov/book">
    <label model="catalog"
    ref="title"/>
    <value ref="@isbn"/>
  </itemset>
</select>
```

Repeating Constructs

⟨*repeat*⟩ *enables template based user interfaces.*

```
<repeat nodeset="/cart/item">
  <input ref="prod">...</input>
  <input ref="qty">...</input>
</repeat>
```

XForms Processing Model

DOM Event Model

XForms processing model is defined declaratively.

- XForms events define processing model.
- Event processing defined by DOM2.
- Event semantics exposed via XML events.

XForms Processing Model

- Model
  - Construct instance and bind constraints.
- User interface
  - Bind controls to model,
  - Generate initial presentation,
  - Listen for interaction events,
  - Invoke event handlers,
  - Recalculate, Revalidate, Redisplay.
- Submit —transmit and serialize instance.

XForms Actions

XForms defines following declarative event handlers.

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

Creating And Deploying XForms

Creating XForms Applications

Connecting business logic to the Web.

- Map back-end data models to XForms model.
- Bind XForms UI to the result.
- Deliver appropriate final form presentation.

Deploying XForms

XForms can live at multiple points on the network.

- XForms server can manage data model.
- Can deliver DHTML to legacy clients.
- Can map to multimodal presentations —X+V.

XForms Summary

XForms Model

Decrease cost of Web applications.

- Encapsulates data and constraints,
- Enables automatic server-side validation,
- Enables client-side code generation,
- Enables Web front-ends to Web Services.

XForms User Interface

Designed for cross-device, multimodal access.

- UI markup captures underlying intent.
- *Accessible* by design.
- Factors interaction logic from presentation.
- Controls can be optimized for target device.

XForms Submit

Delivers well-formed XML to the server.

- Submitting XML makes I18N easy.
- XML simplifies server-side processing.
- Response can transmit instance updates.