WebWork in Action

A hands-on look at the future of Struts

Who is Patrick?

- Founder of Autoriginate, Inc.
- Previously worked for Jive Software
- Founder of OpenQA - open source QA tools
- President of OpenSymphony Group, Inc.
- Author of WebWork in Action (free copies!)

Introduction

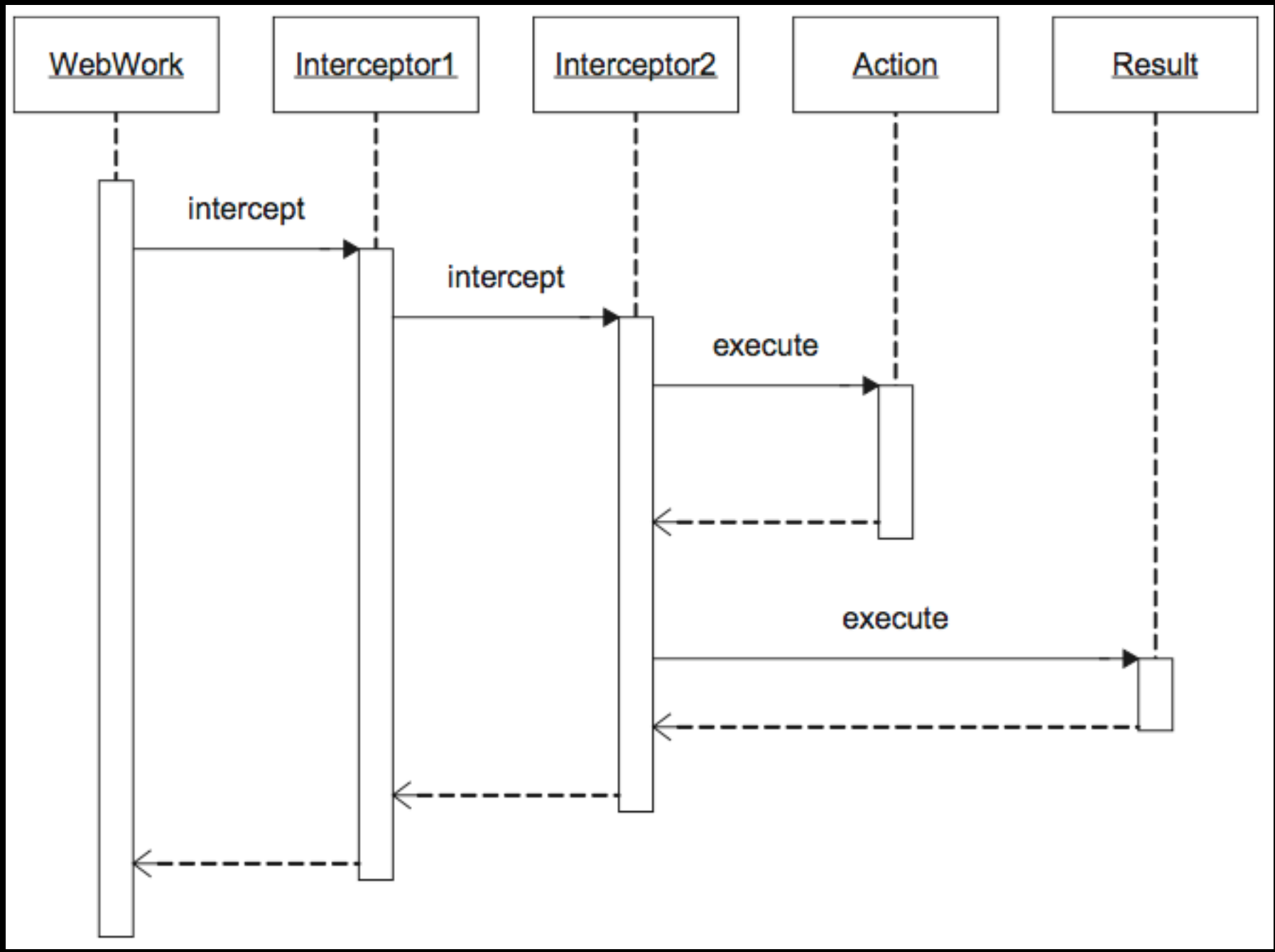
- Overview of WebWork
- Comparison to other frameworks
- About the Struts merger
- WebWork basics: validation, tags, and more
- Rapid development with WebWork
- AJAX Support

Overview: WebWork...

- Is built upon the Command Pattern
- Works directly with POJOs
- Uses OGNL for expression language and data binding
- Has an advanced validation framework
- Includes an extensible widget system
- Supports JSP, Velocity, FreeMarker, Jasper Reports, XSLT, and other view technologies

Core Concepts

- Three key pieces:
  - Actions (POJOs, ActionSupport)
  - Results
  - Interceptors
- No “form beans”: the action is the model
- Value stack allows loose coupling


WebWork
Interceptor1
Interceptor2
Action
Result
intercept
intercept
execute
execute

Getting Started

- Two options:
  - Standard Servlet (2.3) container
  - Built in QuickStart server (more later)
- Both methods are compatible
  - develop in QuickStart and deploy in a standard container

Setting up the Filter

```
<filter>
  <filter-name>webwork</filter-name>
  <filter-class>
    com...FilterDispatcher
  </filter-class>
</filter>
<filter-mapping>
  <filter-name>webwork</filter-name>
  <url-pattern>/*</url-pattern>
</filter-mapping>
```

Configuration

- Actions, Interceptors, and Results configured in xwork.xml
- Support for packages and package inheritance
  - Optional mapping to namespaces
- Additional files can be included using <include>

Example xwork.xml

```
<xwork>
  <include file="webwork-default.xml"/>
  <package name="default"
           extends="webwork-default">
    <action name="listPeople"
            class="ListPeople">
      <result>listPeople.jsp</result>
    </action>
  </package>
</xwork>
```

Example xwork.xml

```
<xwork>
  <include file="webwork-default.xml"/>
  <package name="default"
           extends="webwork-default">
    <action name="listPeople"
            class="ListPeople">
      <result>listPeople.jsp</result>
    </action>
  </package>
</xwork>
```

Example xwork.xml

```
<xwork>
  <include file="webwork-default.xml"/>
  <package name="default"
           extends="webwork-default">
    <action name="listPeople"
            class="com...ListPeople">
      <result>listPeople.jsp</result>
    </action>
  </package>
</xwork>
```

Example xwork.xml

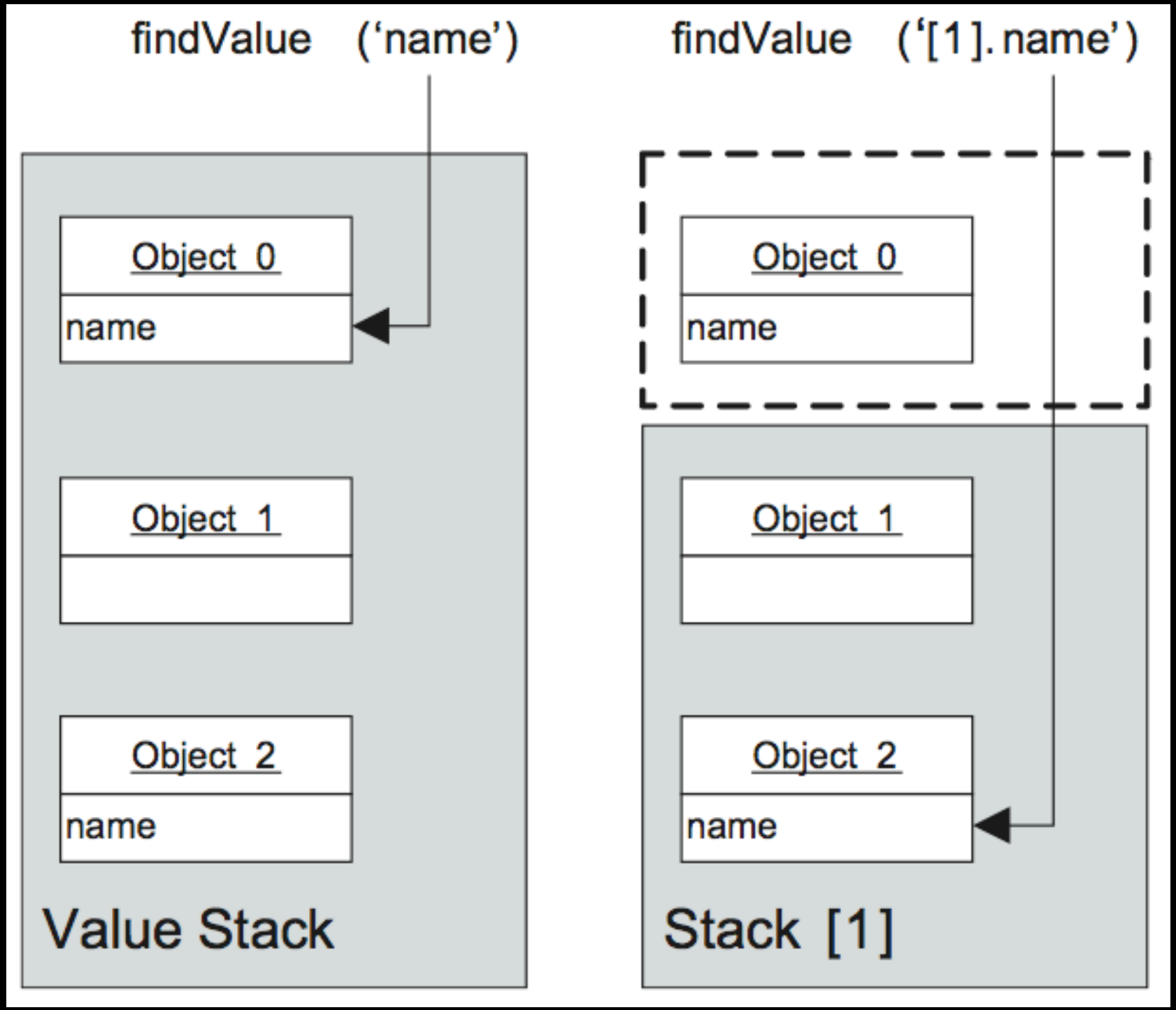
```
<xwork>
  <include file="webwork-default.xml"/>
  <package name="default"
           extends="webwork-default">
    <action name="listPeople"
            class="com...ListPeople">
      <result>listPeople.jsp</result>
    </action>
  </package>
</xwork>
```

Interceptors

- Provide the very core features for WebWork
  - Logging
  - Applying HTTP request parameters
  - Invoking the validation framework
- Can also provide advanced features
  - Automatic “Please wait...” pages for long-running requests
  - Prevent double click problems

Value Stack

- All expressions (OGNL) work against the value stack
- Actions are pushed on the stack before anything else happens
- Additional objects, such as those in an iterator, can be pushed on to the stack
- Allows for loose couple of web components


findValue ('name')
findValue ('[1].name')
Object 0
name
Object 1
Object 2
name
Value Stack
Object 0
name
Object 1
Object 2
name
Stack [1]

Comparisons

- WebWork vs Struts?
- WebWork vs Ruby on Rails?
- Action vs Component?
- WebWork vs JSF?
- Can’t we all just get along?

Java Web Frameworks

- Action frameworks
  - URL binding
  - WebWork, Struts Action, RIFE, Stripes, Spring MVC
- Component frameworks
  - Event binding
  - JSF, Tapestry, Shale, Seam

About the Merger

- Struts Action 2.0 == WebWork 2.2 + some Struts features
- WebWork will cease to be actively developed
- Code, developers, and community moves to Struts
- Future focus on development productivity

Struts: No Longer a Framework

- Struts is not a framework, it is a community
- Two frameworks:
  - Action: Action model
  - Shale: Component model
  - Action and Shale will share code as much as possible
- Various sub-projects, such as Tiles

WebWork Basics

- UI Tags
- Validation
- Data Binding
- Continuations

UI Tags

- Platform to create reusable UI widgets
- Form controls provided out of the box
- Groups of templates form “themes”
- The “xhtml” theme is a simple two-column layout
- Themes can extend each other
  - ajax -> xhtml -> simple

Example

```
<ww:form method="post">
    <ww:textfield label="Name"
                  name="name"/>
    <ww:textfield label="Age"
                  name="age"/>
    <ww:select label="Favorite color"
               name="color"
               list="%{
                  { 'Red', 'Blue',
                    'Black', 'Green' }
               }"/>
    <ww:submit action="quiz"/>
</ww:form>
```

First name is required!

First name:

Last name: Lightbody

Email: plightbo@yahoo.com

Gender: Male Female

Street Address: 123 ABC Street

Zip Code: 97209

City: Portland

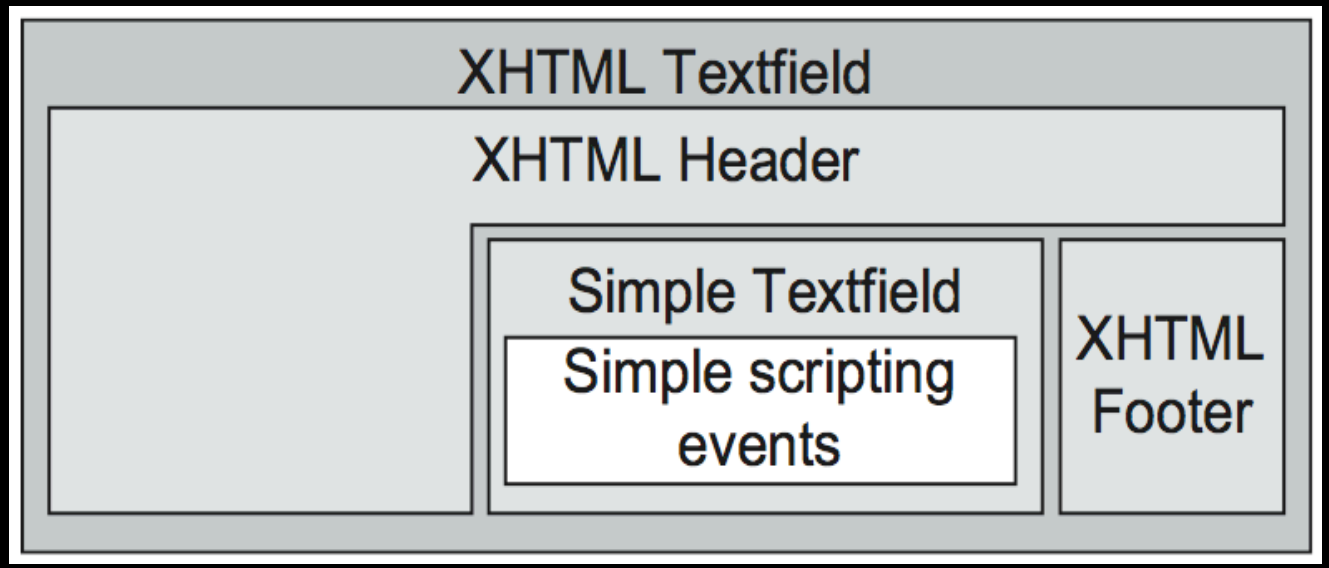
State: Oregon

Country: USA

P.O. Box

Update Profile

The xhtml theme extends the simple theme and provides a standard two-column layout…



Demo

Validation

- Abstracts validation rules from core code
- Common rules already created (required, regex, date range, etc)
- Same rules work with client side validation using AJAX
- Rules can defined in XML or with annotations

Example

```
<validators>
    <field name="age">
        <field-validator type="int">
            <param name="min">13</param>
            <param name="max">19</param>
            <message>
                Only people ages 13 to
                19 may take this quiz
            </message>
        </field-validator>
    </field>
</validators>
```

Data Binding

- HTTP is not aware of data types... but Java is!
- WebWork helps with this mismatch by letting you work with your raw POJOs rather than type-less strings
- Can support basic objects, lists, maps, sets, and more
- Binding rules are based on generics and annotations

Examples

- String -> int
  - <input name="id"/>
- String[] -> List<String>
  - <input name="name"/>
- Complex types
  - <input name="person.id"/>
  - <input name="people[0].id"/>
  - <input name="person.friends.name"/>

Continuations

- Is a native feature in some languages, but not Java
- Lets you define application flow as Java code
- State is stored as simply local method variables
- WebWork uses byte-code manipulation to emulate continuations in the Java language.

Example

```
int answer = ...;

while (answer != guess && tries > 0) {
    pause(SUCCESS);

    if (guess > answer) {
        addFieldError("guess",
                      "Too high!");
    } else if (guess < answer) {
        addFieldError("guess",
                      "Too low!");
    }

    tries--;
}
```

Demo

Rapid Development

- What makes development "rapid"?
- Why is Ruby on Rails so popular?
- J2EE productivity fallacies
- QuickStart: bringing scripting benefits to Java web apps

QuickStart

- Is the quickest way to get started
- Is inspired by AppFuse, Ruby on Rails
- Is powered by a built-in Jetty server
- Automatically compiles your source files
- Gets you started in three steps:
  - Unzip webwork-2.2.2.zip
  - cp -R webapps/starter webapps/showcase
  - java -jar webwork.jar quickstart:showcase

Class (re)Loading

- Java *can* support the same edit-refresh style development that scripting languages have
- Commons-JCI is used, which delegates to Janino, Eclipse compiler, and others
- To be done properly, libraries and frameworks need to get out of the edit-compile-package-deploy-wait-refresh mindset

Demo

AJAX Support

- WebWork provides basic building blocks for AJAX development:
  - Remote divs
  - Remote forms
  - Validation
- Builds on top of Dojo and DWR

Validation

```
<ww:form method="post" theme="ajax"
         validate="true">
    <ww:textfield label="Name"
                  name="name"/>
    <ww:textfield label="Age"
                  name="age"/>
    <ww:textfield label="Favorite color"
                  name="answer"/>
    <ww:submit/>
</ww:form>
```

Demo

Remote Forms

```
<ww:head/>
...
<ww:form id="myForm"
         cssStyle="border: 1px solid black;"
         action="myAction"
         method="post"
         theme="ajax">
    <ww:textfield name="..."/>
    ...
    <ww:submit resultDivId="myForm"/>
</ww:form>
```

Remote Forms

```
<button type="submit"
        dojoType="BindButton"
        formId="myForm"
        value="Submit"
        targetDiv="myForm">
    Submit
</button>
```

Questions?

Questions?

Questions?

Questions?

? ?

?

Questions?

? ?

Questions?

?

Questions?

Questions?

Out of stock, please
try Amazon.com :)