





Overview

- Beyond HTML: An overview of client-side technologies
- "Live" clients: the architecture of Java client-side components
- Client-side components in the real world: the AISJava Framework
- FDF/PDF: Interactive PDF for the Web
- Q&A



Why Use Client-Side Technologies?

- Haven't you ever wanted more?
- HTML/HTTP provides a limited model of interaction
- Standard HTML doesn't provide fine layout control



Why Not Roll Your Own?

- Circumventing or enhancing HTML requires a lot of work on the programmer's part
 - Communication with the server
 - Connectivity to existing data and applications



The WebObjects Solution

- Java Client-Side Components
- Interactive FDF/PDF support



Java Client-Side Components

- Provides more interactive user interface model
- Enables integration with existing Java applets
- Uses same back-end logic as for WebObjects HTML interfaces





Overview

Applets can cooperate with the server

- The role of applets
 - Display and manage values for a set of keys
 - Invoke actions in the Web Application
- WebObjects takes care of the hard stuff
 - Allows multiple applets to participate
 - Handles communication with the server
 - Synchronizes values with the server



Keys and Values

Values are property list objects

- Property lists may be any combination of:
 - String (NSString)
 - Vector (NSArray)
 - Hashtable (NSDictionary)
 - byte[] (NSData)



Flow of Control (part 1)

Client

Page arrives in browser Applets initialized Initial values set



User uses the applets User causes an action



Changed values are gathered Action invocation is sent

Server

WOF constructs page Initial keys & values in HTML



Flow of Control (part 2)

Client

Changed values are gathered Action invocation is sent

Server

WOF updates bound variables WOF invokes action



Action executes



Changed values are gathered Response is sent

Response is received Applets get updated values



Client Side Architecture

How does it work?

- Applet Group Controller
 - "Hidden" applet that runs the show
 - Knows about other applets on the page
 - Handles all synchronization of values
 - Handles all communication with server
- Associations
 - Know how to deal with specific applets
 - Provide uniform interface to controller



API

Writing your own associations or applets

- Association API
 - public Vector keys()
 - public Object valueForKey(String key)
 - public void takeValueForKey(Object value, String key)
- SimpleAssociation
 - Can be used if the applet is designed for it
 - Just passes on Association messages



Uses

What can I do with it?

- Use our applets instead of HTML forms
 - Can update values without reloading page
- Write your own applets
 - For controls that forms don't support
- Write Associations for existing applets
 - For integrating other people's applets





AIS Java Presentation Framework

Thomas Charubas

Director, Java Solutions Practice

AIS



The Problem

- We need to distribute real applications with sophisticated GUIs in a 3-tier fashion
- HTML is very limited
- Certain functions should be handled on the client:
 - Data validation, formatting, manipulation, presentation
 - Searching and sorting
 - Feedback



Java Presentation Framework

- Flexible and rich set of client side Java components for WebObjects based on IFC/JFC
- Inter-applet workflow handler
- Extensible validation and formatting
- Highly configurable from the middle tier
- Components talk objects
 - WebObjects, CORBA/HOP and Java Beans





FDF/PDF

- Allows precise layout control of pages using PDF for presentation
- Provides a form overlay to PDF documents
- Forms can trigger server actions
- Forms can receive data from server applications





ютт	3119	

First name

Last name

Phone _____

PD F



Form	2	1	1	0	
TOTAL	,	•	-		

First name _____

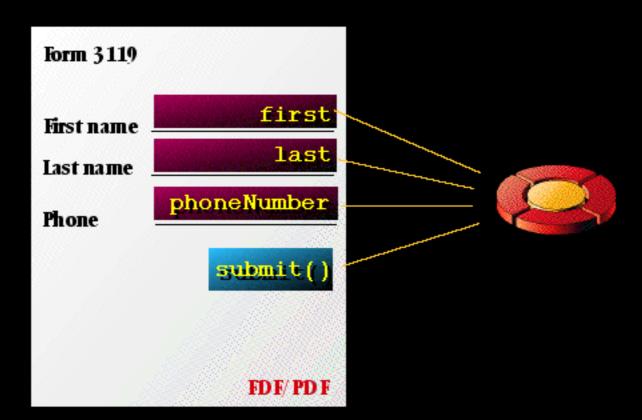
Last name

Phone

PDF



Form 3119 first First name last Last name phoneNumber Phone submit() FDF/





Subclass the FDF Component

```
public class FormThirtyOneNineteen extends
PDFComponent {
  // all the action happens here
}
```



Bind Variables to I/O Elements

```
public String first =
pdf.PDFTextField.defaultValueTag;
public String last =
pdf.PDFTextField.defaultValueTag;
public String phoneNumber =
pdf.PDFTextField.defaultValueTag;
```



Bind Actions To Form Controls

```
public Component submit() {
//do something cool
return this;
}
```



FDF/PDF Implementation Steps

- Start with PDF file
- Create overlay in Adobe Acrobat
- FDF file contains both layout and bindings data for components
 - Replaces HTML/WOD files in WebObjects
- Create implementation in script or Java



