Changing the rules of business™

Building rich web applications using ILOG JViews and evolved technologies such as JSF and AJAX

Mathias Mouly
Technical Account Manager
mmouly@ilog.fr

Overall Presentation Goal

Changing the rules of business™

Learn how to build advanced 2D graphics web interface using Java Server Faces, AJAX and ILOG Visualization Technology

Agenda

Changing the rules of business™

- Introducing JSF technology
- Building JSF advanced 2D graphics components
- Few words about ILOG JViews
- JSF JViews Chart component example with DHTML rendering
- JSF/AJAX to improve user experience
- Summary

Introduction - JSF

Changing the rules of business™

- JSR 127
- A component framework for building user interfaces for web applications
- A set of APIs for representing UI components, managing state, handling events inputs and defining page navigation
- A JavaServerTM Pages (JSP) core tag library
- A set of predefined server-side components and an HTML render kit implementation

JSR 127 : JavaServer Faces

Changing the rules of business™

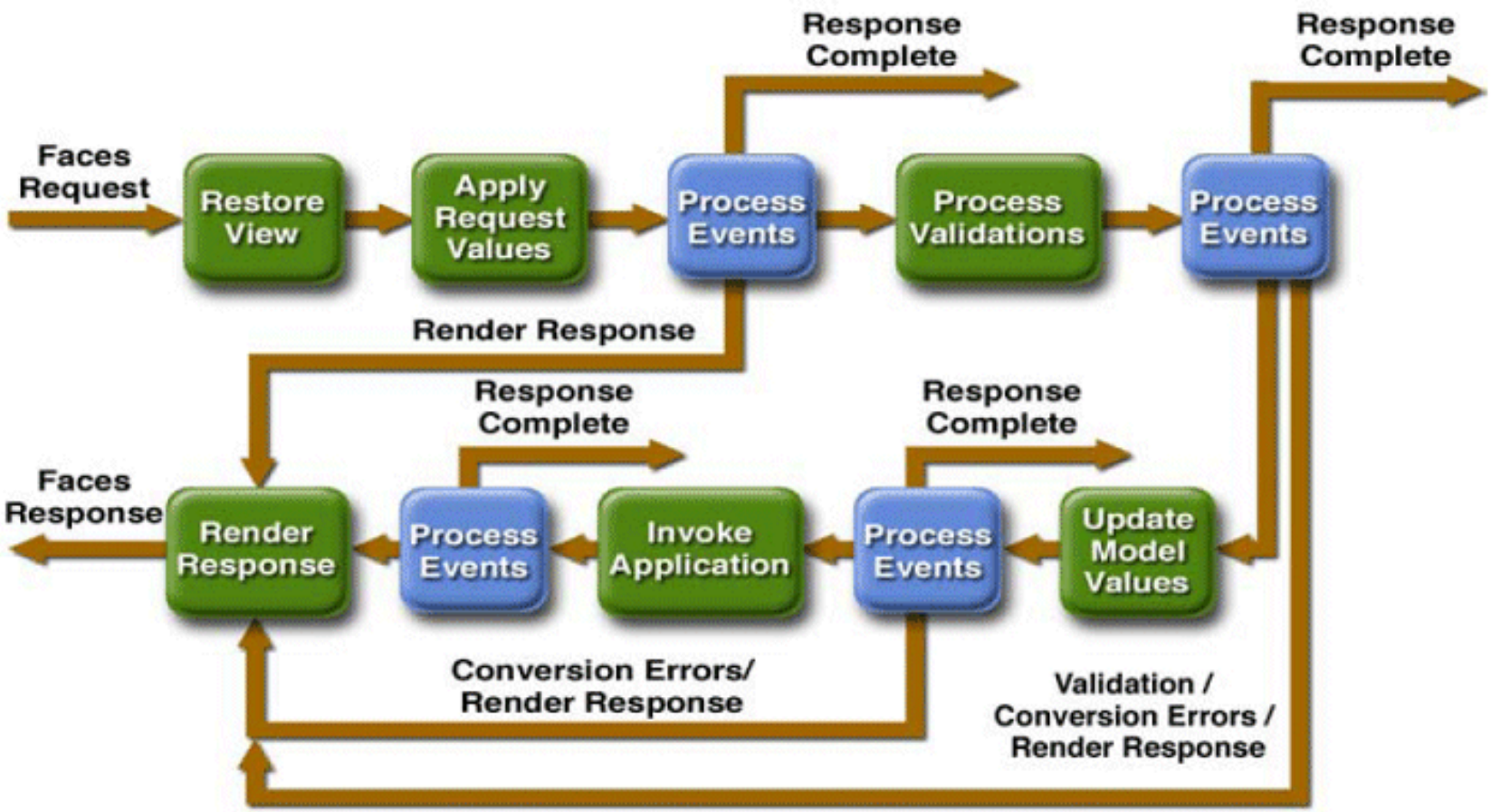
Features

- A well defined request processing lifecycle
- Managed UI Component state across requests
- Event model that allows the application to write server-side handlers
- Multiple render kits for one component (HTML, DHTML, SVG, Flash,...)
- Navigation between pages
- Managed beans
- Bindings or Value Expressions (1.2)
- Input value validation and conversion
- Multiple implementations of the specification

JSR 127 : JavaServer Faces

Changing the rules of business™

Request Processing Lifecycle



Introduction - JSF

Changing the rules of business™

- **Basic Use**
  - **Develop the model objects and register them**
  - **Create the JSP pages**
  - **Define page navigation**
- **Customizing**
  - **Customize converters and listeners**
  - **I18N**
- **Extending**
  - **Adding new render kits**
  - **Defining new components**

Introduction - JSF

Changing the rules of business™

```
package myPackage;

public class Customer {
   private String name;

   public void setName(String name) {
      this.name = name;
   }

   public String getName() {
      return this.name;
   }
}
```

Introduction - JSF

Changing the rules of business™

```
<%@ taglib uri="http://java.sun.com/jsf/html" prefix="h" %>
<%@ taglib uri="http://java.sun.com/jsf/core" prefix="f" %>

<html>
 <head>
  <title>Enter your name</title>
 </head>
 <body>
   <f:view>
     <h:form id="myForm">
      Enter your name:
      <h:inputText value="#{theCustomer.name}" />
      <h:commandButton action="send" value="Submit" />
     </h:form>
   </f:view>
 </body>
</html>
```

Changing the rules of business™

http://localhost:8080/demo1

DEMO

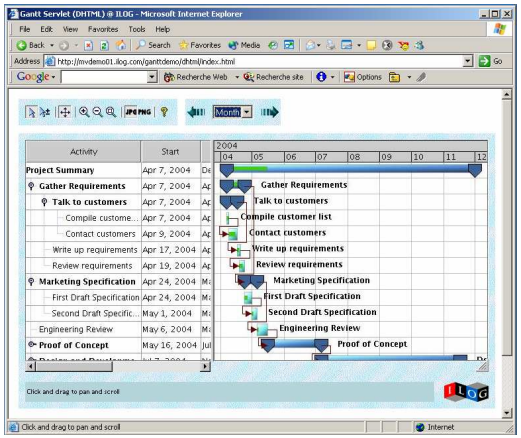
Advanced 2D Graphics Components

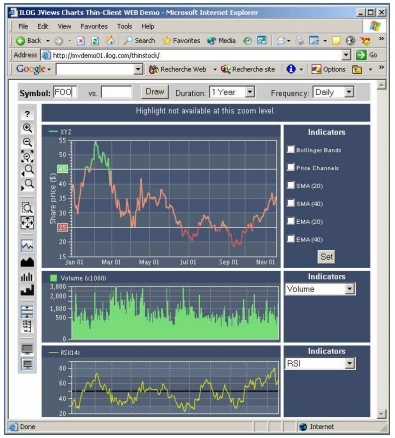
Changing the rules of business™

- An advanced 2D graphics component goes beyond simple UI components to provide enhanced feedback and interaction to the user
  - e.g. tooltips, zooming, data selection, data edition
- Examples:
  - Charts components
  - Map components
- Why should you care?
  - To display new types of data
  - To improve user experience on all kinds of data

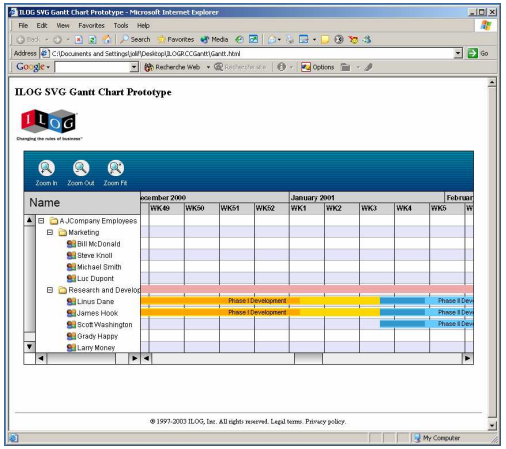
Advanced 2D Graphics Components

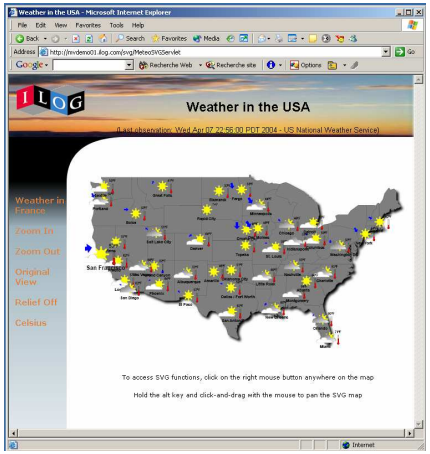
Changing the rules of business™





SVG + Servlet/JSP Examples





Advanced 2D Graphics Components

Changing the rules of business™

- Enable DHTML or SVG components for the client-side display and interaction
- JSF is the standardized server-side framework
  - To generate the client representation
  - To handle and validate user inputs
  - To manage the component state
  - To synchronize with the business data model
  - …
- IDE support

Advanced 2D Graphics Components

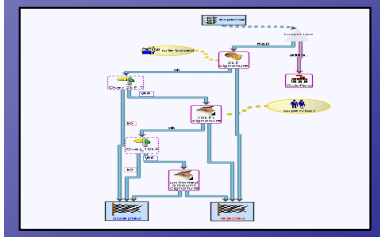
Changing the rules of business™

- Create a JSP tag handler and its descriptor
- Create the server-side component classes
- Create the renderer(s)
- Register the components and the renderers

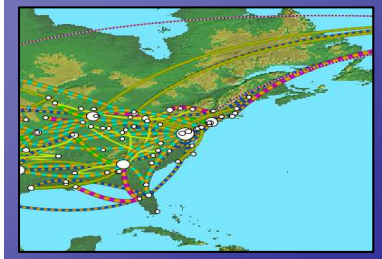
Few words about ILOG JViews

Changing the rules of business™

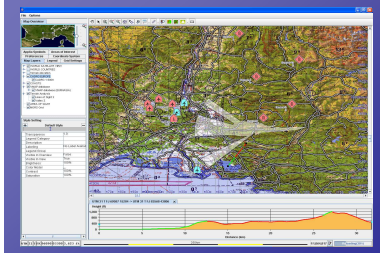
Diagrammer



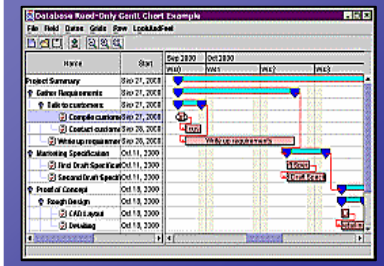
Maps



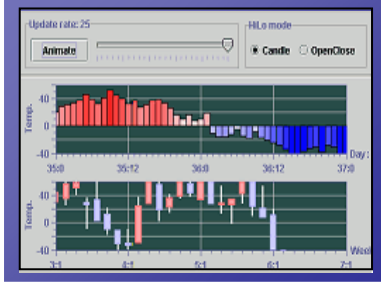
Maps for Defense



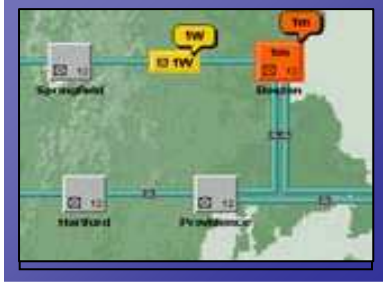
Gantt



Charts



JTGO



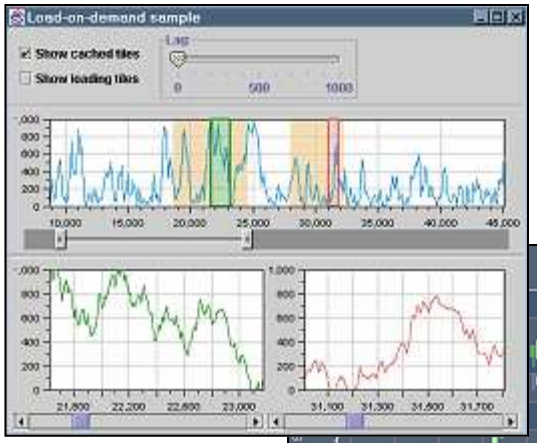
Changing the rules of business™

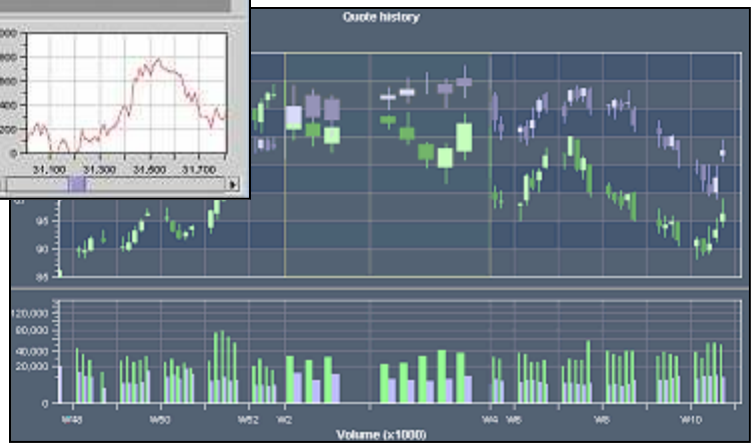
DEMO

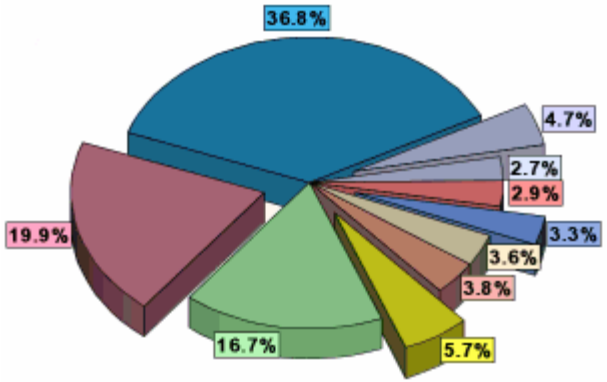
JViews Charts

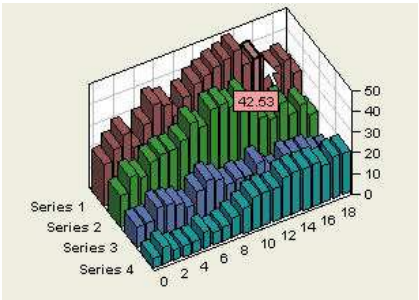
Changing the rules of business™

Charts







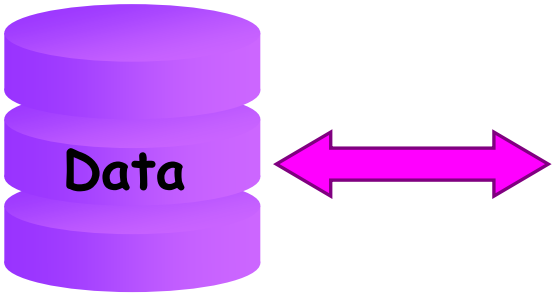


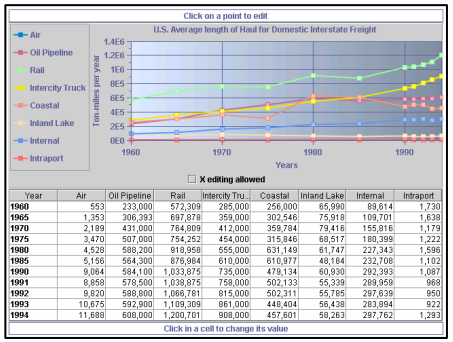
- **Interactive Charting**
  - **Point, line, bar, bubble, etc.**
  - **Radar, pie, polar**
- **Features:**
  - **2D and 3D rendering**
  - **Innovative interactions**
  - **Powerful MVC architecture**
  - **Real-time redraw capability**
  - **Connectors for XML, JDBC, and Swing table provided**

JViews Charts : Principles

Changing the rules of business™

- **MVC architecture**
  - **Data and graphical representations are separated**
  - **Data object / chart displayer object**
- **Data aware charts**
  - **Data changes are automatically reflected in the charts that display these data**
  - **Modifications made by interacting with a chart are automatically reflected on the data**



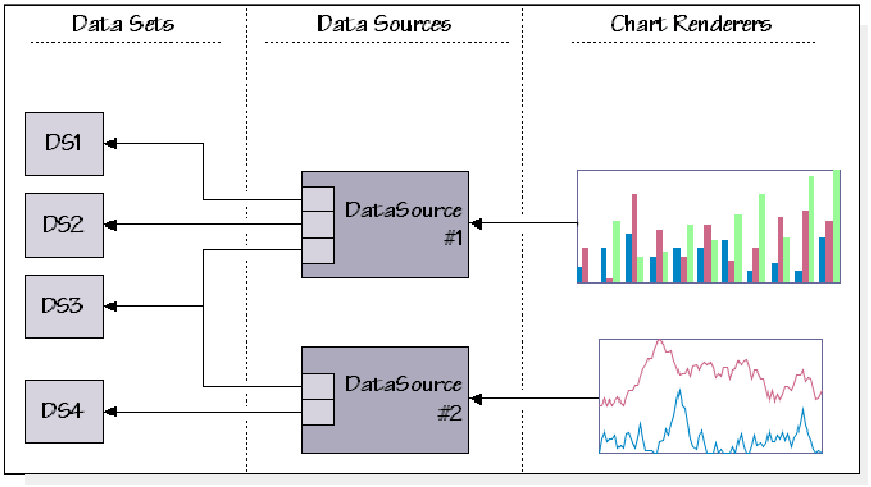


Year	Air	Oil Pipeline	Rail	Intercity Tru...	Coastal	Inland Lake	Internal	Intraport
1960	553	233,000	572,309	285,000	256,000	85,980	89,614	1,730
1965	1,353	306,393	697,879	359,000	302,546	75,918	109,701	1,639
1970	2,189	431,000	764,809	412,000	359,784	79,416	155,816	1,179
1975	3,470	507,000	754,252	454,000	315,846	68,517	180,399	1,222
1980	4,528	588,200	919,958	555,000	631,149	61,747	227,343	1,596
1985	5,156	564,300	876,984	610,000	610,977	48,184	232,708	1,102
1990	9,064	584,100	1,033,875	735,000	479,134	60,930	292,393	1,087
1991	8,859	578,500	1,038,875	758,000	502,133	55,339	289,959	969
1992	9,820	589,800	1,066,781	815,000	502,311	55,785	297,639	950
1993	10,675	592,900	1,109,309	861,000	448,404	56,438	283,894	922
1994	11,689	608,000	1,200,701	908,000	457,601	58,263	297,762	1,293

JViews Charts : Flexible Integration

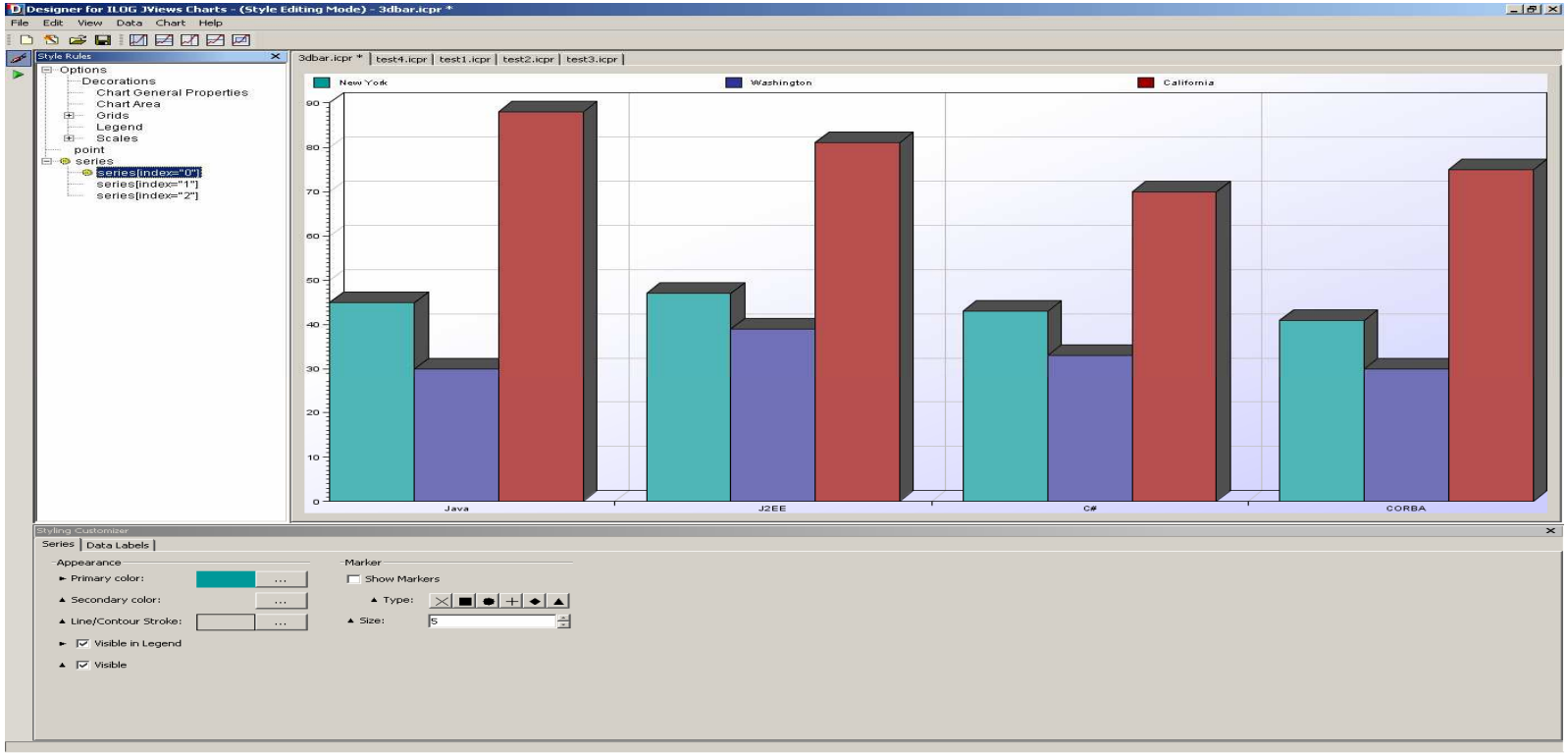
Changing the rules of business™

- **Data model designed for integration**
  - **Open API for adaptation to any backend**
  - **Data sets**
    - Array of points, cyclic array, function, combinations, load-on-demand, …
  - **Data sources**
    - Memory, input, XML, JDBC, CSV, Swing table, …



JViews Charts : The Designer

Changing the rules of business™



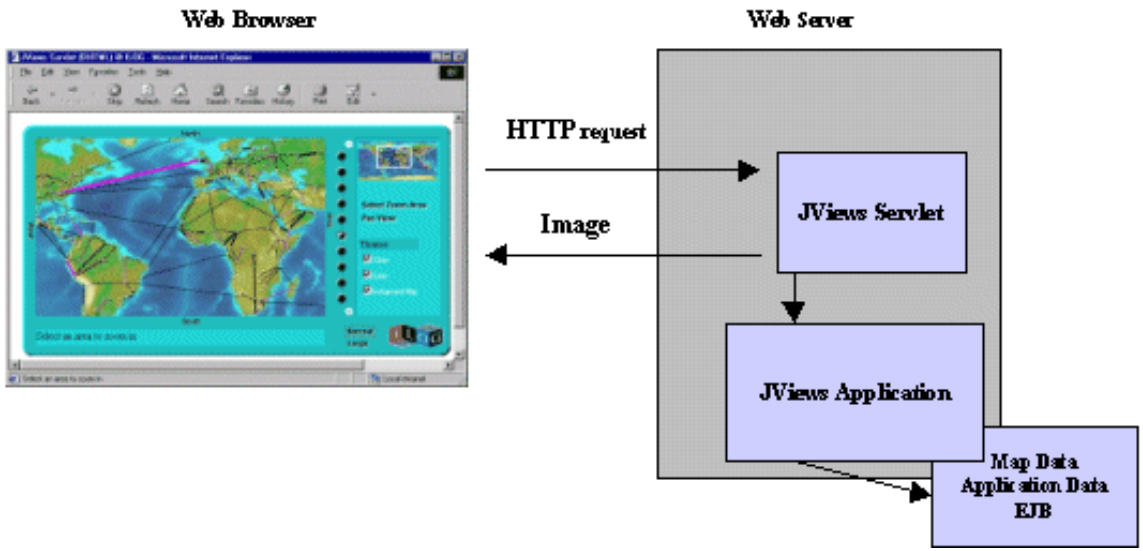
Changing the rules of business™

DEMO

JViews DHTML components

Changing the rules of business™

Architecture



- JavaScript API
- Image Servlet from Java Graphic component

JViews JSF DHTML components

Changing the rules of business™

Purpose

- Make developing Web Application with JViews easier
- Leverage the DHTML components with JSF
- JSF component view manages the graphic object
- DHTML renderers render HTML + JavaScript
- Heavily subclassed image servlet support
  - The servlet must be subclassed by the customer as a last resort

JViews JSF DHTML components

Changing the rules of business™

Architecture

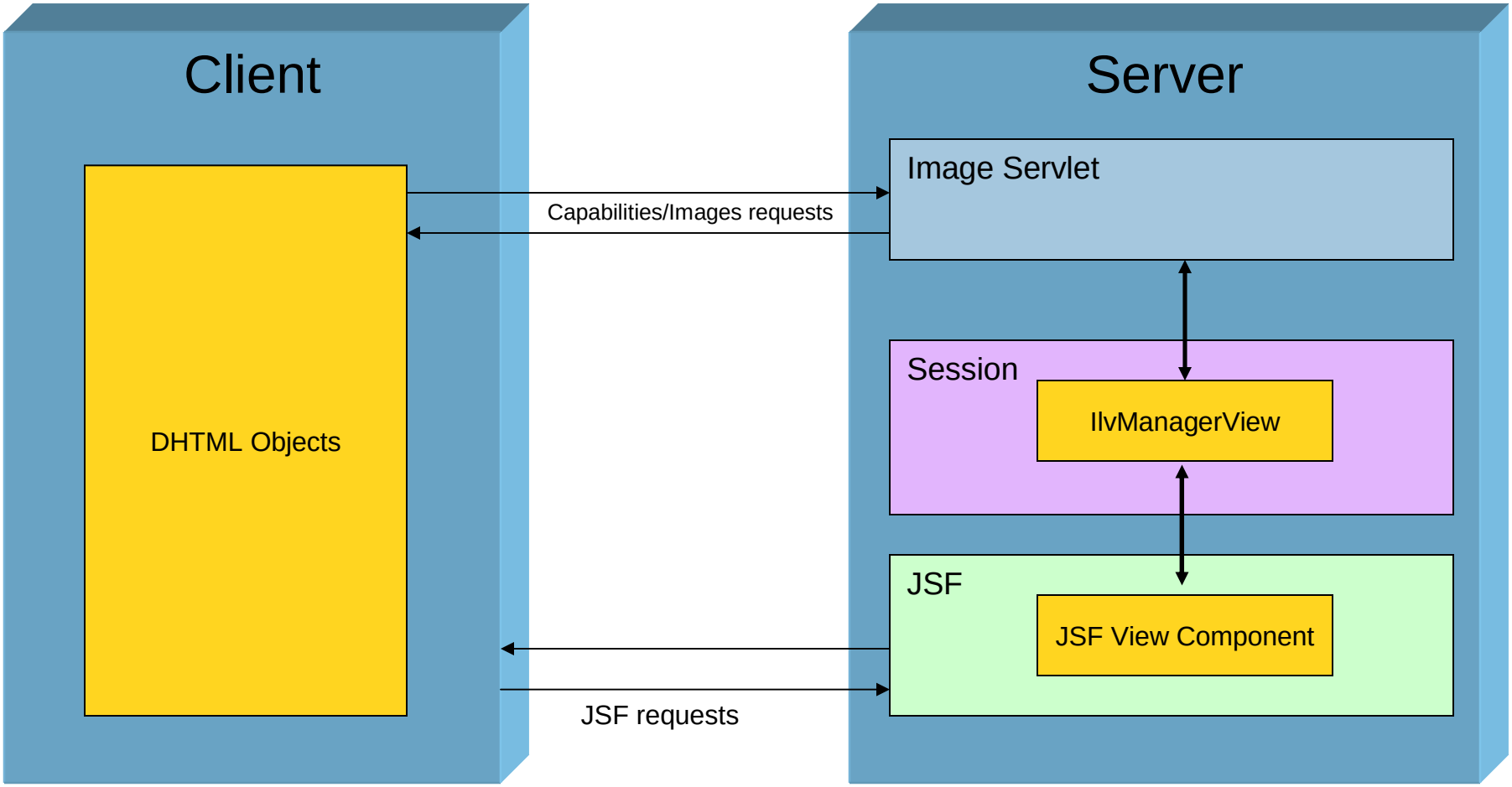


Chart JSF Component Example

Changing the rules of business™

```
public class ChartTag extends UIComponentTag {
  public String getComponentType() {
    return “ChartComponent”;
  }
  public String getRendererType() {
    return “ChartRenderer”;
  }
  protected void setProperties(UIComponent c) {
    super.setProperties(c);
    if (url != null)
      c.getAttributes().put(“url”, url);
    // ...
  }
  public void setUrl(String url) {
    this.url = url;
  }
  public void setDataSourceID(String dataSource) {
    this.dataSource = dataSource;
  }
  public void setType(String type) {
     this.type = type; // PIE, BAR...
  }
}
```

Chart JSF Component Example

Changing the rules of business™

```
public class ChartComponent extends UIGraphic {
  public void setDataSourceID(String dataSource) {
    this.dataSource = dataSource;
  }
  public Object saveState(FacesContext ctx) {
    Object values[] = new Object[3];
    values[0] = super.saveState(context);
    values[1] = dataSource;
    // ...
    return values;
  }
  public void restoreState(FacesContext ctx,
                           Object state) {
    Object[] values = (Object[])state;
    super.restoreState(context, values[0]);
    setDataSource((String)values[1]);
    // ...
  }
}
```

Chart JSF Component Example

Changing the rules of business™

```
public class ChartRenderer extends Renderer {
 public void decode(FacesContext ctx, UIComponent c) {
   super.decode(ctx, c);
   Map parameterMap =
    ctx.getExternalContext().getRequestParameterMap();
   String url = (String)map.get("url");
   if (url != null)
     ((ChartComponent)c).setUrl(url);
   // ...
 }
 public void encodeEnd(FacesContext ctx, UIComponent c)
   throws IOException {
   super.encodeEnd(ctx, c);
   ChartComponent chart = (ChartComponent)c;
   ResponseWriter writer = ctx.getResponseWriter();
   writer.startElement("img");
   writer.writeAttribute("src", chart.getUrl());
   writer.endElement("img");
 }
}
```

Chart JSF Component Example

Changing the rules of business™

- The image generator Servlet uses a Chart rendering engine that dumps drawings to a Graphics2D
- A Graphics2D is obtained from a BufferedImage and passed to the Chart engine
- The BufferedImage is encoded according to the required image type and sent back to the client

Chart JSF Component Example

Changing the rules of business™

```
<%@ taglib uri="http://java.sun.com/jsf/html" prefix="h" %>
<%@ taglib uri="http://java.sun.com/jsf/core" prefix="f" %>
<%@ taglib uri="http://www.ilog.com/jviews/tlds/jviews-chart-
   faces.tld"    prefix="c" %>

<html>
  <body>
    <f:view>
      <c:XMLDataSource id=“myDataSource" filename="data.xml" />
      <c:chartView id="chart" style="width:800;height:600;"
         dataSourceId=“myDataSource" />
    </f:view>
  </body>
</html>
```

Changing the rules of business™

http://localhost:8080/myJSF/faces/chart-user1.jsp

DEMO

Adding interaction

```
<c:XMLDataSource id="myDataSource" filename="data.xml" />

<c:chartView id="chart" style="width:800;height:600;"
  dataSourceId="myDataSource" interactorId="zoom" />

<c:chartZoomInteractor id="zoom" />
```

Changing the rules of business™

http://localhost:8080/myJSF/faces/chart-user9.jsp

DEMO

Adding CSS rendering flexibility

Changing the rules of business™

```
<jvcf:chartView id="chart8" project="data/test2.icpr"
  style="width:1000;height:800; "
   binding="#{chartBean.chartView}"/>

<h:commandButton value="Reset"
   actionListener="#{chartBean.reset}"/>
```

Changing the rules of business™

http://localhost:8080/myJSF/faces/chart-user11.jsp

DEMO

Changing the rules of business™

AJAX

- XmlHttpRequest + XML + JavaScript
- IFrames are working too !
- Security problems for the two solutions
  - ActiveX
  - Some browsers don’t like IFrames
- Allows to send requests and process response asynchronously
- Cool …. But no standards
- New issues
  - History & bookmarks
- Real solutions are proprietary

Changing the rules of business™

JSF/AJAX Strategies

- **External servlet**
  - Use a dedicated servlet out of the JSF lifecycle to provide the component with its update data
  - Ex: Popup menu
  - Ex: RWC data servlet
- **Servlet Filter**
  - Use the JSF lifecycle
  - Add a servlet filter that cuts the unecessary information
- **Client side parsing (IBM JavaOne)**
  - Use the JSF lifecycle
  - Parse on the client

Changing the rules of business™

JSF/AJAX Strategies

- **Phase listener (JViews RWC)**
  - **Register a phase listener that intercepts the AJAX request**
  - **Render the update data**
  - **Cut the JSF lifecycle**

- **Complete architecture (ex ADF)**
  - **Proprietary components that handles incremental refresh at the base**

Changing the rules of business™

DEMO

Summary

Changing the rules of business™

- Advanced graphics components need a server-side framework such as JSF
- JSF based components can be easily assembled in JSF-enabled IDEs
- Think to JViews for advanced 2D Graphics usage

For More Information

Changing the rules of business™

- **JavaServerTM Faces**
  - **http://java.sun.com/j2ee/javaserverfaces/**
- **Java and SVG**
  - **Batik Toolkit: http://xml.apache.org/batik**
  - **Bring SVG power to JavaTM applications: http://java.sun.com/developer/technicalArticles/GUI/svg**
  - **ILOG JViews: http://jviews.ilog.com**
- **SVG**
  - **http://www.w3.org/Graphics/SVG**
  - **Adobe SVG Viewer: http://www.adobe.com/svg**

Next Steps

Changing the rules of business™

- Visit the ILOG booth in the exhibition space for further information
- Evaluate ILOG JViews
  http://www.ilog.com/products/jviews/eval/index.cfm
- Try the online tutorials
  http://www.ilog.com/products/jviews/webSeminar.cfm
- Look at the demos
  http://www.ilog.com/products/jviews/demos/index.cfm

Changing the rules of business™

Q&A