

Changing the rules of business™



Christophe Jolif
Software Architect

Ajax use at ILOG



Agenda

- **Building Advanced Visualization Components**
- **Working with other Ajax Components**
- **What do we expect from OpenAjax?**

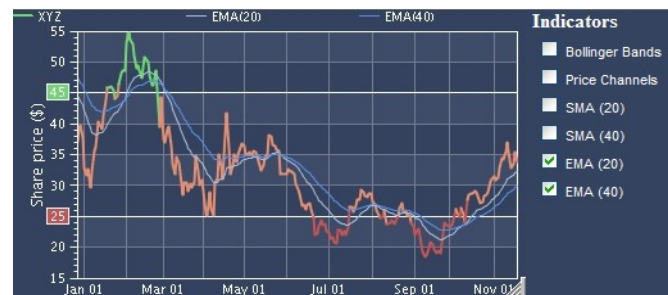
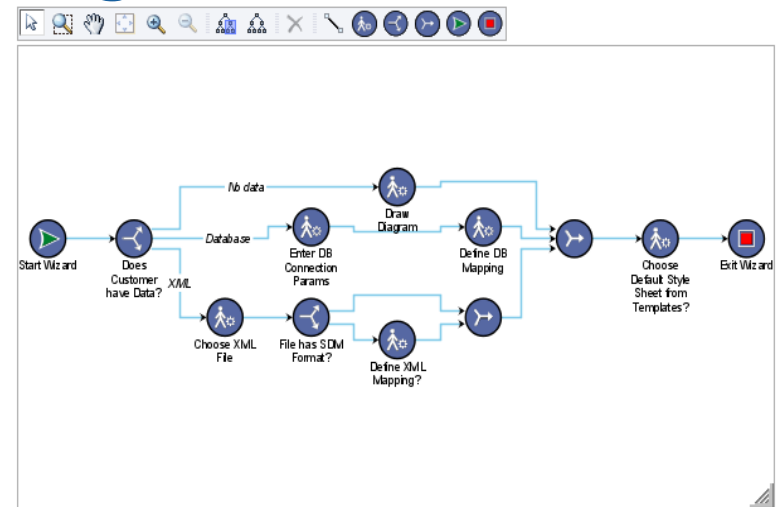
Building Advanced Visualization Components

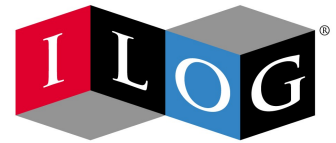


Changing the rules of business™

Examples

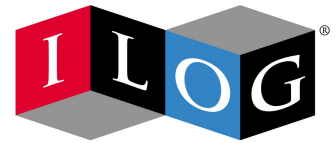
- Any Kind of Interactive Diagrams
- Editable Gantt Charts
- Maps Display
- Rules Editor
- Pie, Bar, Radar... Charts
- ...





History

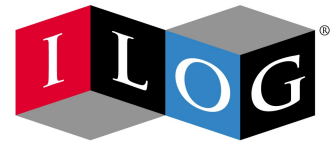
- **1999 – DHTML Components**
 - No component model, entirely Servlet-based on the Server
 - First version of ILOG JavaScript framework (“class-like” inheritance, tracing, browser workarounds...)
- **2001 - ILOG “WebComponents”**
 - Component model
 - Incremental refresh of components and “Ajax” features using iframes or XMLHttpRequest
 - But proprietary => No interoperability, no IDEs...



History

- **2003 - JavaServer Faces**
 - IDEs, interoperability with other 3rd party components
 - But we lost the “Ajax” features
- **2005 – JavaServer Faces and Ajax**
 - We introduce Ajax features in our JSF components like: dynamic contextual menus, objects editing with minimal round trip, incremental data refreshes, local caching...

Building Advanced Visualization Components



Changing the rules of business™

Technologies

- **Common Technologies:**
 - **JavaServer Faces as UI Component Framework**
 - **XML and JSON for data communication**
- **DHTML Components:**
 - **Use DHTML and Rasters generated on the server to render most of the display**
 - **Our proprietary legacy JavaScript framework**
 - **Sometimes use SVG/VML generated on the client to render “dynamic” part of the display (interaction feedback for example)**
 - **Use iframe as communication object**

Building Advanced Visualization Components



Changing the rules of business™

Technologies

- **Rich Web Components:**
 - **Use SVG generated on the client from the data sent from the server**
 - **Use JavaScript objects on the client to expose data model and rendering options to the client**
 - **Use XMLHttpRequest as communication object**

Working with other Ajax Components



Changing the rules of business™

What for?

- For demonstrating our components
- For our customers to build enterprise applications

Solving our remaining issues ;)

- **Standard ways for avoiding collision between Ajax libraries**
- **Definition of interfaces/hubs allowing Ajax components to talk to each others in a standard manner**
- **OpenAjax should define best practices on how Ajax component should work with server-side environments (JSF, portlets...)**