Using jMaki in a Visual Development Environment

Gregory Murray, Ludovic Champenois, Craig McClanahan

Sun Microsystems, Inc.
https://ajax.dev.java.net/

Session TS-9516

Goal of Our Talk

Understand how you can leverage visual development tools to rapidly build web applications with jMaki.

Agenda

Introduction to jMaki
jMaki and IDEs
jMaki and Visual Development
Summary and Q&A

Agenda

Introduction to jMaki

jMaki and IDEs

jMaki and Visual Development

Summary and Q&A

Origins of jMaki?

- ‘j’ is for JavaScript™ technology
- Maku == to wrap in Japanese
- Started as a way of wrapping JavaScript technology functionality
- Project jMaki has evolved to provide more

Why Use jMaki?

- Common interface to all JavaScript technology libraries
- Common models
- Multi-server support (program in what you know)
- JavaServer Pages™ (JSP™)/JavaServer™ Faces technology/facelets/portlets
- PHP/JavaScript technology (Phobos)
- It's easy to get started (convention over configuration
- Templates
- It promotes a clean separation of content/style/ JavaScript technology

jMaki

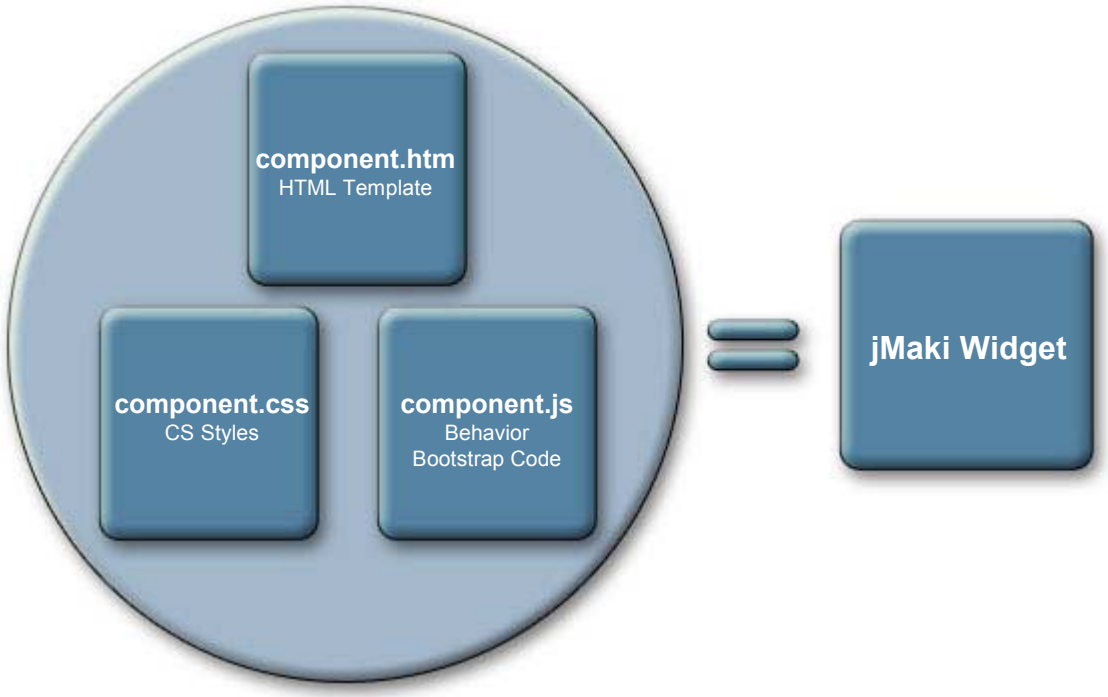
A Client-Server Framework for Providing JavaScript Technology-Centric User Interfaces

- A widget model for creating widgets or wrapping existing functionality
- A server runtime to provide JavaScript technology dependencies and client/server data bindings
- A set of client services for managing widgets
- Layouts and themes
- Glue for tying everything together

http://ajax.dev.java.net/about.html

Widget Model

Simple



https://ajax.dev.java.net/widget-model.html

Hello World Widget

component.htm

```
<div id="${uuid}"></div>
```

component.js

```
jmaki.hello = {};
jmaki.hello.Widget = function(wargs) {
  var mydiv = document.getElementById(wargs.uuid);
  mydiv.innerHTML = "Hello " + wargs.args.name;
}
```

index.jsp

```
<%@ taglib prefix="a"
                  uri="http://jmaki/v1.0/jsp" %>
<a:ajax name="hello" args="{name: 'greg'}" />
```

Hello World Widget

JSP—index.jsp

```
<%@ taglib prefix="a"
                  uri="http://jmaki/v1.0/jsp" %>
<a:ajax name="hello" args="{name: 'JS'}" />
```

PHP—index.php

```
<?php require_once 'Jmaki.php'; ?>
<?php addWidget("hello", null, "{name: 'PHP'}"); ?>
```

Phobos index.ejs

```
<% library.jmaki.insert({
   component:"hello",
   args: {name: 'Phobos'}
}); %>
```

Client Services API

- jmaki.attributes
- jmaki.publish/subscribe
- jmaki.Timer
- jmaki.doAjax
- jmaki.loadScript/loadStyle
- jmaki.namespace
- jmaki.extend
- jmaki.log

XHP XML Http Proxy

A Window to the Outside World

- Access RESTful web services
  - Yahoo GeoCoder
- Access to RSS feeds
  - Atom/RSS
- Can convert data to JSON
- Widgets are tuned to use it

jMaki Glue

- Based on publish/subscribe
- JavaScript technology-based and loaded application-wide or based on a page
- JavaScript technology handlers mapped to topic names
- Widgets configured to work by default

jMaki Layouts

- CSS-based
- Layouts use common naming conventions
- Widgets size to fit the layouts
- HTML templates provided for the layouts

https://ajax.dev.java.net/source/browse/*checkout*/ajax/ws/lib/css/index.html

jMaki Themes

- CSS-based and separate from the layouts
- Layouts use common naming conventions
- Widgets provide default themes which are overridden
- True CSS cascade
  - Library-level CSS
  - Widget CSS overrides library CSS
  - Theme overrides widget and library CSS

https://ajax.dev.java.net/source/browse/*checkout*/ajax/ws/lib/css/index.html

jMaki Recipe

- Choose a layout
- Drop widgets into a layout
- Configure widgets (if necessary)
- Provide glue if widgets interact
- Choose a theme

Agenda

Introduction to jMaki

jMaki and IDEs

jMaki and Visual Development

Summary and Q&A

jMaki Component Challenges

- Multiple IDEs support
  - Eclipse, NetBeans™ IDE, CLI
- Multiple languages support
  - JSP technology, JavaServer Faces technology, Phobos JS, Ruby, PHP
- Multiple Ajax Tookits support
  - Dojo, Yahoo, Google, Ext, Scriptaculous, Flickr
- Integration for your own components

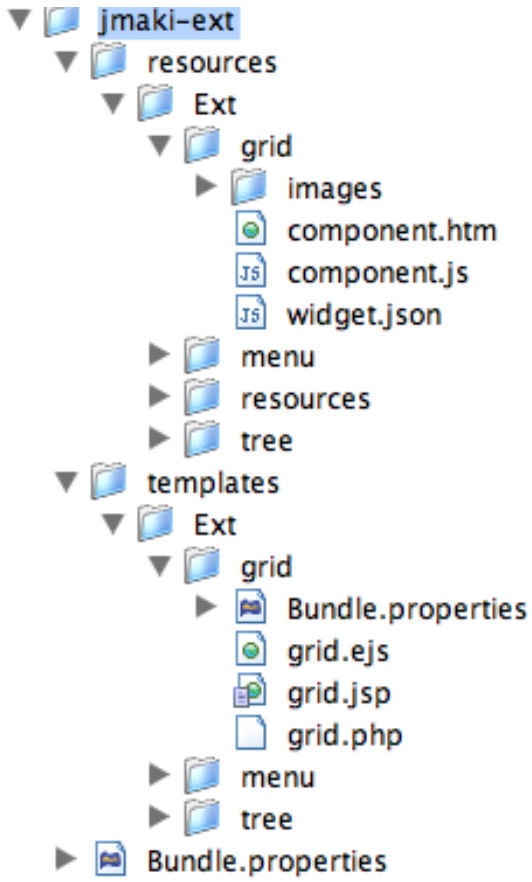
jMaki Toolability Requirements

- Register jMaki libraries to projects
  - Web project, Phobos project, Ruby projects…
- Expose jMaki components via a palette
- Install new jMaki components
- Upgrade components
- Customize components
  - Need for IDE agnostic component meta-data
- Expose jMaki layouts and themes

jMaki Library Structure

- Collection of named components
  - Name if directory path of the component
  - A component is component.html, component.js, component.css and widget.json
- The toolkit library content
  - For example, dojo or yahoo libs
- Snippets for JSP technology, EJS, PHP, etc.
- Palette entry names in a bundle file
- All this packaged as a zip file for installation

jMaki Library Structure



jMaki Widgets

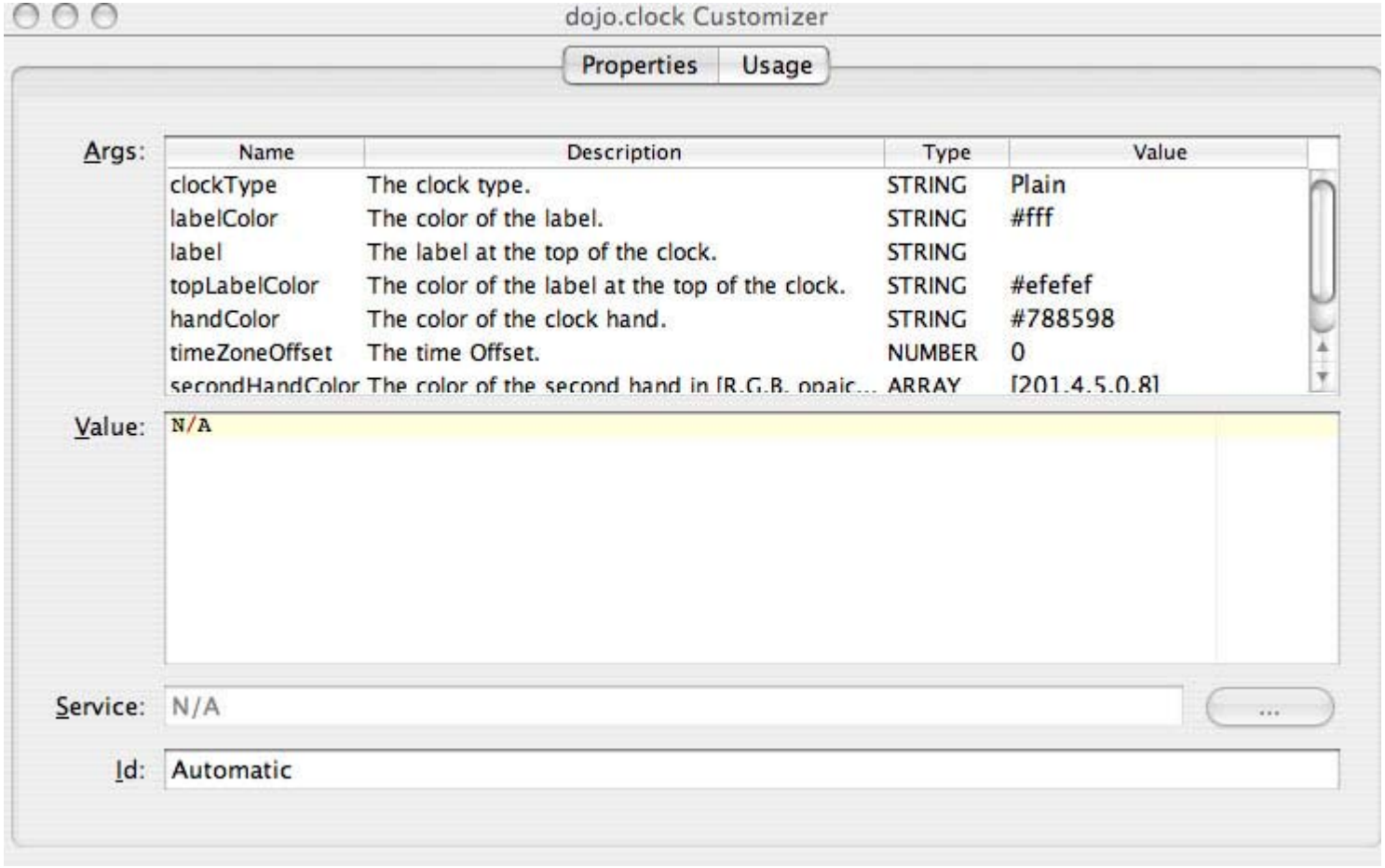
widget.json for Dojo Clock

```
{    "name" : "Clock",
     "type" : "dojo",
     'version' : '.8',
     'jmakiVersion' : '.8.2.3',
     "image" : "images/dojo-clock.jpg",
     "description" : "This widget is a clock.",
     "args": [
         {"clockType" : {
            "type":"STRING",
             "description" : "The clock type.",
             "defaultValue" : 'plain',
              'values': [
                 {'name' : 'Plain', 'value' : 'plain',
'description' : 'Clock with blue background.'},
                 {'name' : 'Black', 'value' : 'black',
'description' : 'Clock with black background.'},
```

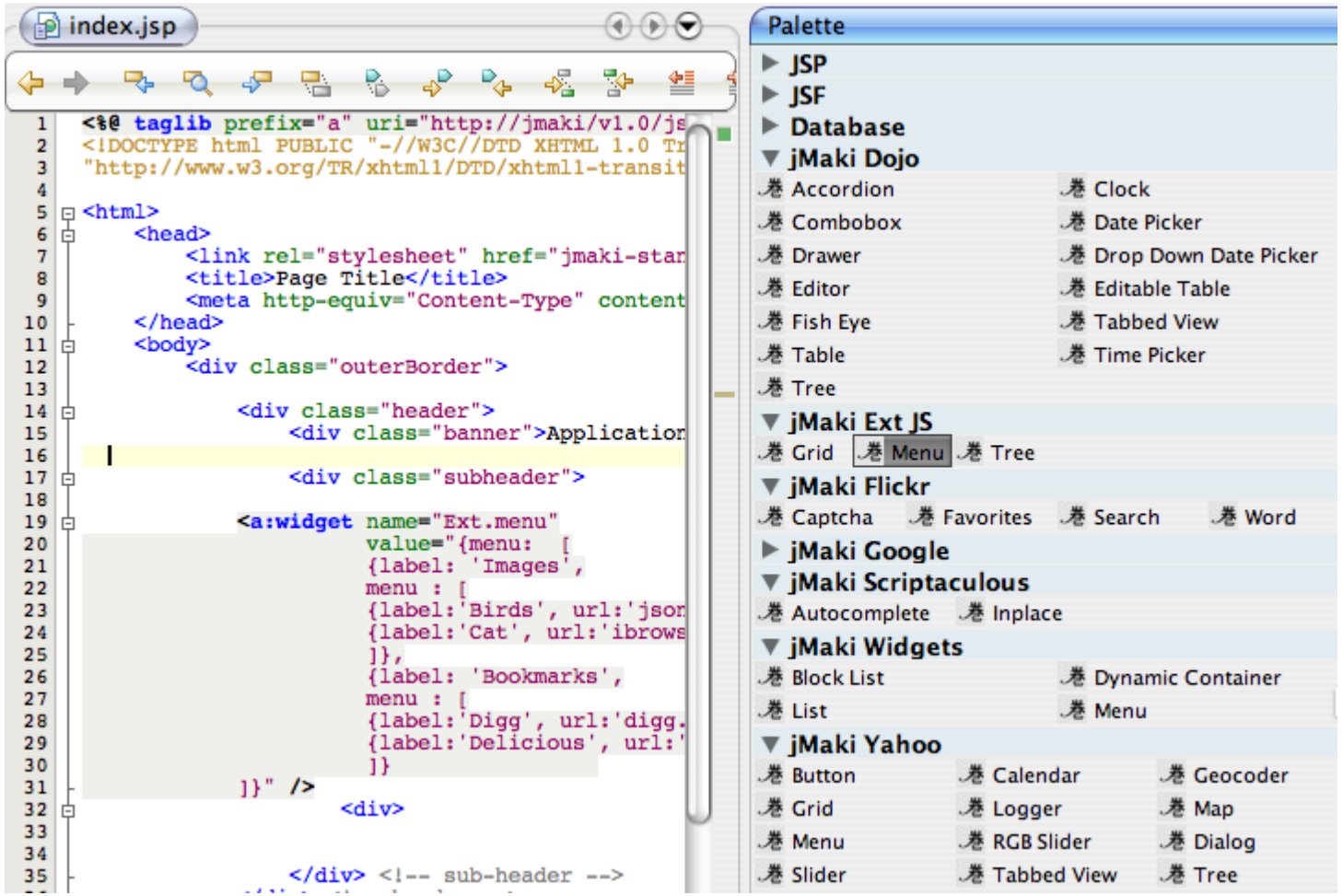
widget.json for Dojo Clock

```
...
    'config' : {
          'type' :
           { 'id' : 'dojo',
             'libs' : [
                  '../resources/libs/dojo/v.4.2/dojo.js'
               ],
               'preload' : 'if (typeof djConfig
==\'undefined\') djConfig = { parseWidgets: false,
searchIds: [] };',
               'resources' : [
                 '../resources/libs/dojo/v.4.2/src'
               ]
         }
       }
```

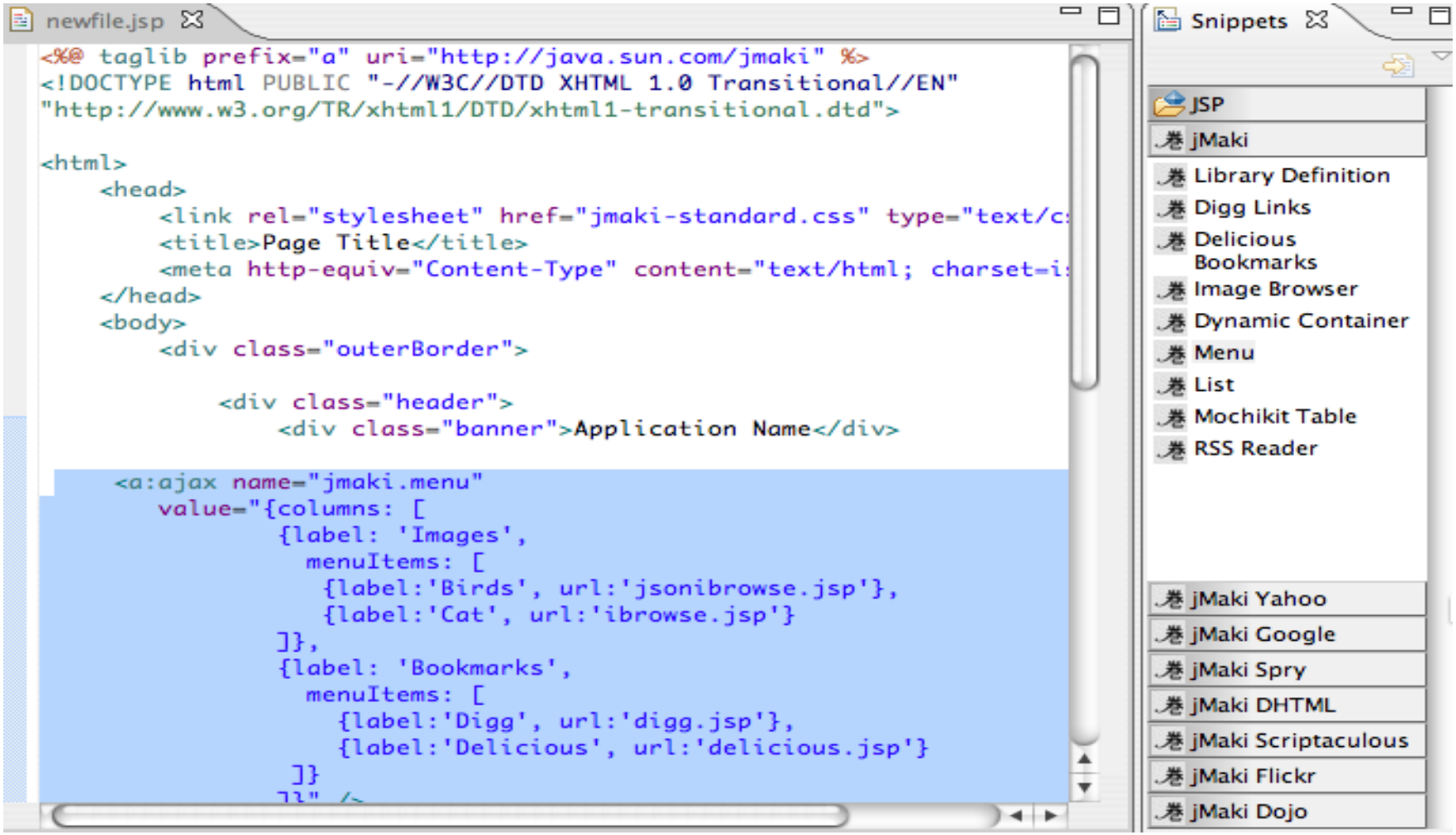
jMaki Component Customizer



jMaki Palette in NetBeans IDE



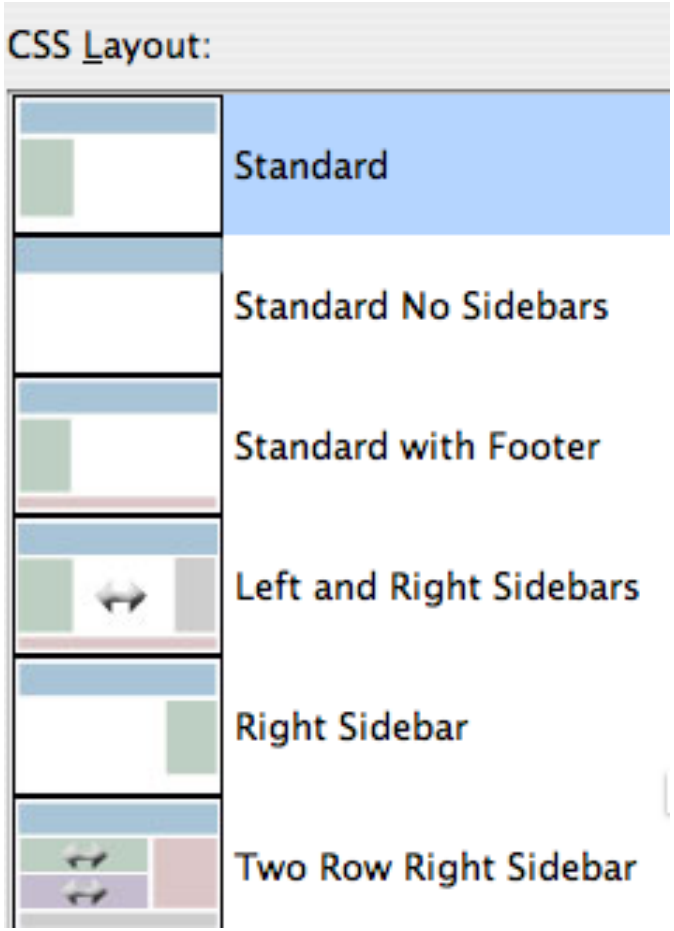
jMaki Palette in Eclipse



jMaki Layouts

- Define structure of the page
- Defined under/resources/css directory
- Naming convention
  - jmaki-2column-footer.css
  - jmaki-2column-footer.html
- The HTML file is used for JSP technology and Phobos
  - The IDE replaces correct content on the fly
- Image for IDE usage
- You can add your own or edit them

jMaki Layouts



jMaki Themes

- Work in conjunction with layouts
  - Not structural definitions
- Defined under: /resources/css/themes directory
- ThemeName/theme.css: contains the theme
- ThemeName/images: contains necessary images
- IDE exposes a theme chooser
- Global per app
  - Stored in /resources/config.json

DEMO

jMaki Integration in NetBeans IDE; jMaki Integration in Eclipse

Agenda

Introduction to jMaki

jMaki and IDEs

jMaki and Visual Development

Summary and Q&A

jMaki and Visual Development

Background and Motivation

- As we have seen, the standard jMaki approach:
  - Wraps widgets from a variety of libraries
  - Provides a consistent programming interface
  - Minimizes need for application developer to write low level JavaScript technology code
- However, the standard jMaki approach:
  - Still requires developers to understand JavaScript technology object literal notation for transferring values
  - Has no direct support component oriented composition
- JavaServer Faces technology provides such an environment
- Can we integrate jMaki and JavaServer Faces technology capabilities?

Yahoo Tree View—The Standard Way

```
<a:widget name="yahoo.tree"
  value="{root: {title: 'Tree View', expanded: true,
   children: [
     { title: 'Tree View Node 1',
       children: [
         { title: 'Tree View Node 1-1' },
         { title: 'Tree View Node 1-2' },
       ]
     },
     { title: 'Tree View Node 2', expanded: true,
       children: [
         { title: 'Tree View Node 2-1' },
         { title: 'Tree View Node 2-2' },
       ]
     },
   ]}" />
```

Yahoo Tree View—The JavaServer Faces Technology Way

```
<yahoo:treeView id="treeView1" expanded="true"
                value="Tree View">
    <yahoo:treeViewNode value="Tree View Node 1">
        <yahoo:treeViewNode value="Tree View Node 1-1"/>
        <yahoo:treeViewNode value="Tree View Node 1-2"/>
    </yahoo:treeViewNode>
    <yahoo:treeViewNode expanded="true"
                        value="Tree View Node 2">
        <yahoo:treeViewNode value="Tree View Node 2-1"/>
        <yahoo:treeViewNode value="Tree View Node 2-2"/>
    </yahoo:treeViewNode>
</yahoo:treeView>
```

Standard and JavaServer Faces Technology Approaches

Compare and Contrast

- Using the standard approach:
  - Single tag for all widgets (name attribute to select)
  - Complex data composed as JS object literal values
    - Developer must understand required attribute names/values
  - (Not shown) Many widgets support acquiring their content from external URLs as well as literal strings
- Using the JavaServer Faces technology approach:
  - Individual component(s) for every widget
  - Complex data composed from component hierarchy
    - Developer must understand component properties
  - (Not shown) JavaServer Faces technology value expressions can be used

Standard and JavaServer Faces Technology Approaches

Compare and Contrast

- Using the standard approach:
  - Developer interacts with JSP technology/PHP/etc. at source level
  - Widget customizer for basic property configuration
- Using the JavaServer Faces technology approach:
  - Developer can interact with JSP technology at source level, or
  - Developer can interact with visual design surface in a tool like the Visual Web Pack add-on to NetBeans IDE
  - Property sheet and individual property editors for property configuration

Visual Development Considerations

jMaki and Visual Tools

- jMaki uses client side JavaScript technology renderers
  - Visual design surface may not execute JavaScript technology
  - Solution—design time vs. runtime renderers
- jMaki widget wrapper components
  - Provide typesafe properties for configurable attributes
  - Delegate rendering to standard jMaki widgets
  - Can be used in standard web projects
  - Can be used in visual web projects
    - Including support for custom design time behavior
- See LAB-4460 for help creating such components

DEMO

Visual jMaki Development

Agenda

Introduction to jMaki

jMaki and IDEs

jMaki and Visual Development

Summary and Q&A

Summary

- jMaki provides comprehensive, sophisticated wrappers around client-side JavaScript technology libraries
- jMaki supports server side implementation for multiple languages and platforms
  - Java platform (JSP and JavaServer Faces technologies)
  - PHP
  - Serverside JavaScript technology
- jMaki is specifically designed to be supported by IDEs and other development tools

Resources

jMaki on the Web

- Main jMaki site
  - https://ajax.dev.java.net/
- jMaki Widgets site
  - https://widgets.dev.java.net/
- Sun web developer pack
  - http://developers.sun.com/web/swdp/
- NetBeans IDE (and visual web pack)
  - http://netbeans.org/

Resources

jMaki at JavaOne℠ Conference

- Using jMaki in a Visual Development Environment
  - TS-9516, Tuesday, 4:40pm–5:40pm
- Benchmarking and Profiling Web 2.0 Applications for Performance
  - LAB-4410, Tuesday, 5:40pm–7:40pm
- Dynamic Portals and Ajax in Portlets
  - BOF-4664, Tuesday, 10:00pm–10:50pm
- Testing Web 2.0 Features Using Real-World Applications
  - BOF-6825, Tuesday, 10:00pm–10:50pm

Resources

jMaki at JavaOne℠ Conference

- jMaki: Web 2.0 App Building Made Easy
  - TS-6375, Wednesday, 10:55am–11:55am
- Building Ajax-Enabled JavaServer Faces Components and Applications with jMaki, Dynamic Faces, and NetBeans IDE
  - LAB-4460, Wednesday, 6:35pm–8:35pm
- Assembling Ajax Applications With Power Tools
  - TS-9517, Thursday, 10:55am–11:55am
- Services Interoperability with JavaTechnology and .NET
  - TS-8840, Thursday, 5:30pm–6:30pm

Q&A

Gregory Murray, Ludovic Champenois,
Craig McClanahan

Using jMaki in a Visual Development Environment

Gregory Murray, Ludovic Champenois, Craig McClanahan

Sun Microsystems, Inc.
https://ajax.dev.java.net/

Session TS-9516