Jddddd

a state-of-the-art solution for control panel development

Elke Sombrowski
Content

- What is jddd?
- Thin client versus rich client concept
- Advanced features/widgets of jddd:
  - Dynamic components
  - Logic components
  - Buttons with special function
  - Wildcards, filters and JavaScript
- Conclusion & Outlook
What is jddd?

- JDDD = Java Doocs Data Display
- Developed at DESY (MCS4 group)
- Panel builder and runtime engine for control system displays
- Editor is similar to the editor of standard IDEs like Netbeans
  - Rich set of predefined components / widgets
  - No programming skills needed
- Stores panels in XML format
Jddd supports all control systems used at DESY.

- DOOCS
- TINE
- TANGO
- EPICS
jddd, a state-of-the-art solution for control panel development

The jddd editor

ICALEPCS 2011, E. Sombrowski
The Editor has standard functions like:

- cut / copy / paste
- align
- flip
- rotate
- undo / redo
- snap to grid
- zoom

These functions are available in the context menu and in the icon bar.
The jddd editor: Predefined component/widget types

Pane Components:
- LayeredPane
- TabbedPane
- IncludeComponent
- DynamicList
- DynamicIncludes
- Alive

Static Components:
- Label
- Line
- Oval
- Rectangle
- Triangle
- Icon

Dynamic Components:
- Button
- ToggleButton
- MouseOver
- Value
- Dial
- TextField
- ComboBox
- CheckBox
- ProgressBar
- Slider
- StatusRegister
- Audio
- ColouredIndicator
- LocationChooser
- CameraImage
- TextArea
- HtmlArea
- XmlViewer
- PropertiesTable
- DeviceTree

Logic Components:
- If
- Switch

Plot Component:
- PlotSpectrum
- PlotHist
- PlotLocation
- PlotXY
FLASH orbit feedback overview

Horizontal Orbit: line = diff, bar = actual, ref

Vertical Orbit: line = act. orbit, bar = set point, error

Server inactive - feedback is switched off
Thin clients do as little processing as possible on the client side and rely on accessing the server to get data.

Thick/rich clients do a significant amount of data processing on the client system, while relatively little is done on the server.

Jddd uses the thin client approach!
Advantages of thin clients

- The interface between client and server is clearly defined.
- Control panels are rapidly developed using a set of predefined components/widgets.
- No programming skills are needed. Engineers, technicians and operators are able to design their own panels.
- Panels are easily adapted to changing server properties and new hardware.
- Thin client software provides a standard look&feel and uniform functionality.
Advantages of rich clients

- The functionality is not restricted to generic possibilities of thin client software. Graphical components can be adapted to special use cases. The display design is more user-friendly.

- Sometimes it's difficult to put all functionality in a server. Mathematical operations or data analysis might be included on the client side.

→ Jddd has advanced widgets and features for reaching more flexibility in control panel design
A dynamic list is a scroll pane where multiple components can be placed. In run mode the specified data is read from the name server and the list is filled automatically.

### in edit mode:

<table>
<thead>
<tr>
<th>Name</th>
<th>Rotation Speed</th>
<th>current</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value</td>
<td>Value</td>
<td>no value</td>
<td></td>
</tr>
</tbody>
</table>

### in run mode:

![Dynamic List of Pump Stations](dynamicList.xml)

<table>
<thead>
<tr>
<th>Name</th>
<th>Rotation Speed</th>
<th>current</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>7ACC6.1.PS</td>
<td>734.496</td>
<td>0.70</td>
<td></td>
</tr>
<tr>
<td>PS.86</td>
<td>0.000</td>
<td>0.00</td>
<td></td>
</tr>
<tr>
<td>PS.85</td>
<td>0.000</td>
<td>0.00</td>
<td></td>
</tr>
<tr>
<td>PS.11.5</td>
<td>0.000</td>
<td>0.00</td>
<td></td>
</tr>
<tr>
<td>PS.11.1</td>
<td>0.000</td>
<td>0.77</td>
<td></td>
</tr>
<tr>
<td>PS.11.9</td>
<td>0.000</td>
<td>0.00</td>
<td></td>
</tr>
<tr>
<td>PS.11.7</td>
<td>0.000</td>
<td>0.00</td>
<td></td>
</tr>
<tr>
<td>PS.12.0</td>
<td>0.000</td>
<td>0.00</td>
<td></td>
</tr>
<tr>
<td>PS.45</td>
<td>565.470</td>
<td>0.00</td>
<td></td>
</tr>
<tr>
<td>PS.11.2</td>
<td>0.000</td>
<td>0.00</td>
<td></td>
</tr>
<tr>
<td>PS.11.3</td>
<td>0.000</td>
<td>1.08</td>
<td></td>
</tr>
<tr>
<td>PS.11.6</td>
<td>779.880</td>
<td>0.00</td>
<td></td>
</tr>
<tr>
<td>PS.43</td>
<td>0.000</td>
<td>0.00</td>
<td></td>
</tr>
<tr>
<td>PS.11.4</td>
<td>0.000</td>
<td>0.00</td>
<td></td>
</tr>
</tbody>
</table>
Example for using a dynamic list in a watchdog overview panel.

The dynamic list displays the server status for a selected hardware device.
The dynamic includes component is a transparent layered pane which displays automatically multiple IncludeComponents according to their Z_POS and X_POS.

FLASH orbit feedback display: Dynamic includes are used for displaying BPMs and steerer magnets.
The if component displays one of two different layers (cases). The switch component displays one of multiple layers. Any case may contain nested logic components.
Buttons with “Set Component Property” function
Wilcards and regular expression filters are used to specify a dedicated subset of data.

Example: In the location plot below the channel address is:
TTF2.DIAG/ORBITFEEDBACK/*_X/BPK_XYZ_RB
Mathematical operations on control system values can be performed using **JavaScript** syntax.

Example:

```
$address1 * 44 + $address2
```

At runtime “$address1” and “$address2” are replaced by the current channel values.
Conclusion

In general:

- Thin clients have a lot of benefits, especially because they are easy-to-use and control panels can rapidly be developed. They are a good choice for simple control panels.
- Thick clients offer more possibilities, but the development is time consuming. They should only be used for high-level control displays.

But jddd is more than an editor for simple thin clients:

- Jddds advanced widgets, filters and JavaScript support offer the possibility for nearly rich client development. Most control system displays with complex display logic can be implemented with this software.
Next steps in jddd development:

- We will focus on performance tuning:
  - Improve startup speed
  - Implement search index for SVN
- Improve jddd widgets according to wishes of the users.
- Improve EPICS interface.
Thank you for your attention

- Webpage:  http://jddd.desy.de
- Documentation:  http://jddd.desy.de -> Help
- Examples:  http://jddd.desy.de -> Start Panels
- Contact:  jddd@desy.de