

JDDD, A STATE-OF-THE-ART SOLUTION FOR CONTROL PANEL DEVELOPMENT

E. Sombrowski, A. Petrosyan, K. Rehlich, W. Schütte, DESY Hamburg, Germany

Abstract

Software for graphical user interfaces to control systems may be developed as a rich or thin client. The thin client approach has the advantage that anyone can create and modify control system panels without specific skills in software programming.

The Java DOOCS Data Display, jddd [1,2,3,4], is based on the thin client interaction model. It provides inclusion of panel components and channel address inheritance for the creation of generic displays. Wildcard operations and regular expression filters are used to customize the graphics content at runtime, e.g. in a dynamic list component the parameters have to be painted only once in edit mode and then are automatically displayed multiple times for all available instances in run mode. This paper will describe the benefits of using jddd for control panel design as an alternative to rich client development.

- Control system 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 control panels. This reduces the commissioning time because the hardware and software development is in one hand.
- Panels are easily adapted to changing server properties and new hardware. The design has high flexibility to later modifications.
- Thin client software provides a standard look&feel and uniform functionality for all control panels independent from the developer.

INTRODUCTION

DESY is currently building a new 3.4 km-long X-ray free electron laser XFEL. This facility will deliver ultra-short light pulses with a peak power up to 100 GW and a wavelength down to 0.1 nm. The commissioning is planned in 2014. The software development has already started and concepts are currently being evaluated and improved at the FLASH FEL facility.

Jddd is developed as the main tool for the creation of graphical displays for the XFEL accelerator. It uses the thin client concept. In the following paragraphs advantages and disadvantages of thin and rich clients will be discussed.

THIN CLIENT VERSUS RICH CLIENT APPROACH

When implementing a client/server architecture it has to be determined if it will be the client or the server that handles the bulk of workload.

In a rich client system, a significant amount of data processing is done on the client or desktop system, while relatively little is done on the server. In contrast, a thin client generally does as little processing as possible on the client side and relies on accessing the server each time input data needs to be processed or validated.

Each of these approaches has its own benefits:

Advantages of Thin Clients

- The interface between client and server is clearly defined.

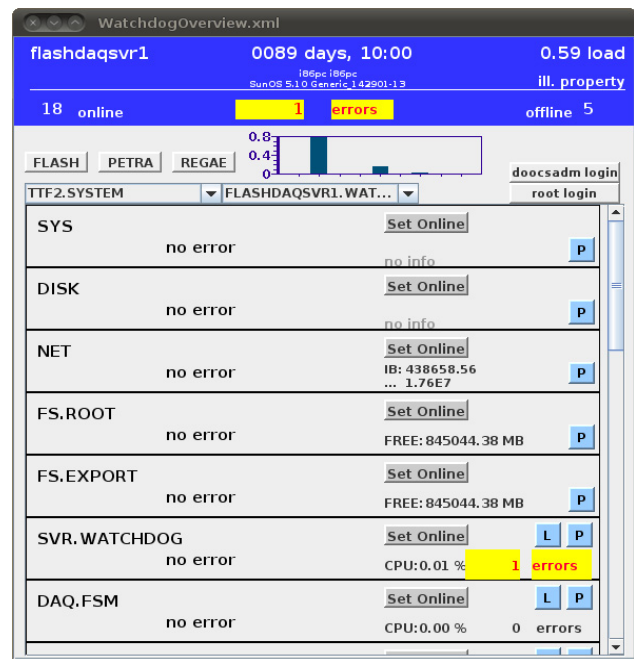


Figure 1: Screenshot of the watchdog overview panel in run mode. A dynamic list component in the lower part of the panel displays the status of all available servers on a selected CPU. A location chooser component, which is placed above the list, allows to change the facility and CPU address and therefore the list content. The buttons labelled FLASH, PETRA, REGAE switch between the different facilities.

Advantages of Rich Clients

- The functionality is not restricted to generic possibilities of thin client software. The display design is more user-friendly. Graphical components can be adapted to special use cases.
- Sometimes it's difficult to put all functionality in a server. Using rich clients, simple mathematical operations or even complex data analysis might be included on the client side.

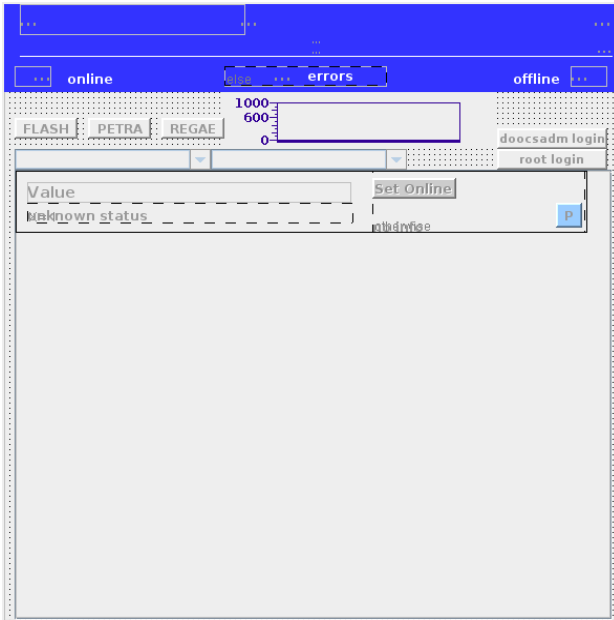


Figure 2: Screenshot of the watchdog overview panel in edit mode. The lower part of the panel contains a dynamic list component. This list is filled in run mode as displayed in Figure 1.

ADVANCED COMPONENTS AND FEATURES

To compensate the disadvantages of thin clients, to reach more flexibility in control panel design and to implement data manipulation, jddd uses different approaches:

- Dynamic components, which change their graphical content according to server properties.
- Logic components (if and switch component) switch the view between different layered panes (cases) depending on a user defined control system value.
- Buttons with “Set Component Property” function change e.g. include files or the panels base address at runtime.
- Wildcard operations, regular expression filters and JavaScript are used for mathematical data operations.

DYNAMIC COMPONENTS

Dynamic Lists

The dynamic list is a layered pane which can be filled with any jddd components in edit mode. It is assigned a base address with a wildcard for the facility, device, location, or property. At runtime the wildcard is replaced by all available addresses and the list is automatically filled with the required number of components. Figure 1 and 2 show an example for using the dynamic list in a watchdog overview panel.

Dynamic Includes

The dynamic includes component is a transparent layered pane which displays automatically multiple include components at dedicated positions according to their Z_POS and X_POS. In the FLASH orbit feedback example (Figure 4) the dynamic includes component shoes all available BPMs and steerer magnets.

LOGIC COMPONENTS

If and Switch Component

The if component contains two cases which consist of two different layered panes. The switch component contains an arbitrary number of cases. Depending on a user defined control system value one of these layered panes is displayed at run mode.

The condition for each case of a logic component is displayed in the jddd component inspector. Any case may contain nested logic components (see Figure 3).

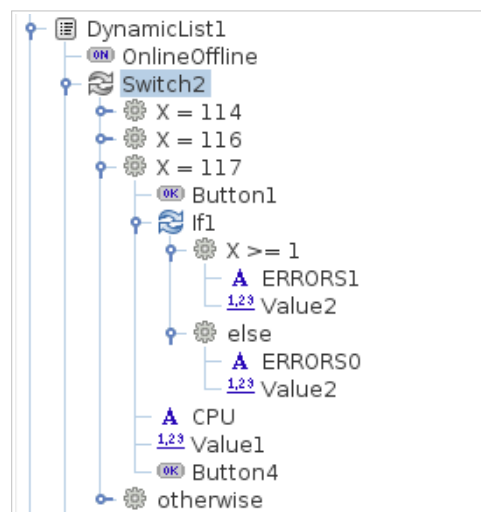


Figure 3: Example for nested switch and if components displayed in the jddd component inspector.

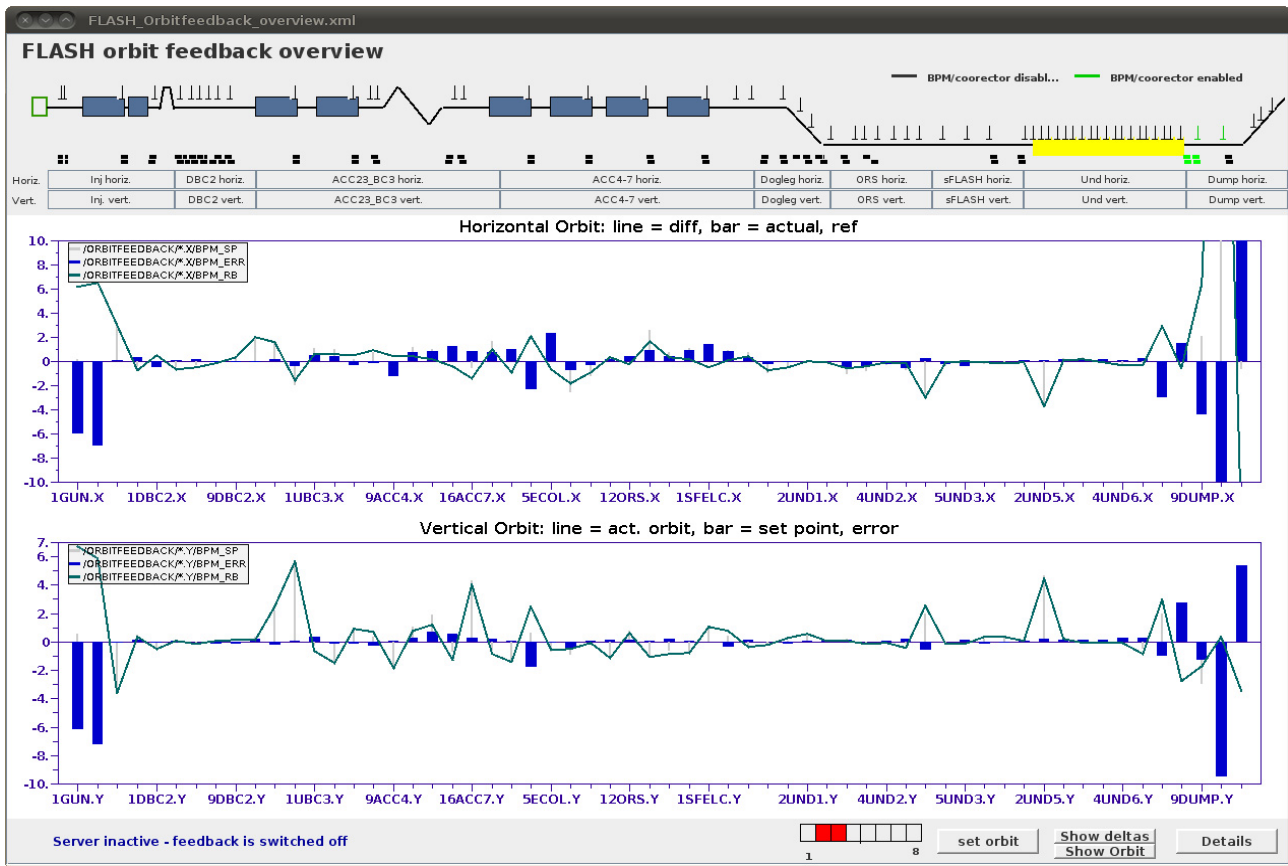


Figure 4: Screenshot of the FLASH orbit feedback panel. In the upper part of the panel dynamic include components are used for displaying BPMs and steerer magnets. The buttons “Show Deltas” and “Show Orbit” use the “Set Component Property” function to switch between orbit and delta plots.

SPECIAL BUTTON FUNCTIONALITY

A jddd button can have different functions. One of them is the “Set Component Property” function. Pressing this type of button one or multiple properties of any component in the panel are changed.

One possible use case is to modify the XML file of an include component. This changes the graphical content of the panel at runtime. In the FLASH orbit feedback (Figure 4) this button function is used to switch between delta and orbit view.

Another use case is to set the panels base address at runtime. In Figure 1 the FLASH, PETRA and REGAE buttons change the base address of the Watchdog panel and therefore switch the view between the different facilities.

WILDCARDS AND FILTERS

To allow the creation of dynamic control panels, which adapt the displayed information according to the selected server properties, jddd uses wildcard operations and regular expression filters.

Wildcards

To specify the content of dynamic lists, dynamic includes and location plots, wildcard operations in the address string are used. For example: If the address of a location plot is

TTF2.DIAG/ORBITFEEDBACK/*X/BPM_XYZ_RB

all available locations ending with “X” will be displayed in the plot (see upper location plot in Figure 4). For further filtering regular expression filters may be used.

Filters

For dynamic lists, dynamic includes, location chooser and the properties table a regular expression filter is available to filter the displayed data. It uses standard Java syntax for regular expressions, like `*(?!_SVR)` to remove all locations ending with the letters “_SVR”. This filtering mechanism allows to display a dedicated subset of data.

JAVA SCRIPT

The "dataOp" property is used to perform mathematical operations with control system values using JavaScript syntax. It is available for the jddd components button, value, progress bar, coloured indicator, status register and switch. An example how to use this property is e.g.:

```
dataOp = $address1*2+$address2
```

At runtime "\$address1" and "\$address2" are replaced by the current values read from the specified addresses.

CONCLUSION

Jddd is more than an editor for simple thin clients. With logic components, "Set Component Property" buttons, filters and JavaScript support it offers advanced possibilities for nearly rich client development. Most panels with complex display logic can be implemented with this software and only a few high-level displays have to be created as rich clients.

Dynamic components and the use of wildcards ensure that displays do not have to be reworked if new devices are added to the control system.

Using jddd with its easy-to-use editor the development time for graphical displays is minimized. Engineers, technicians and operators are able to design control panels themselves, which reduces the workload of the control group. All panels will have a common look and feel and functionality independent from the developer.

REFERENCES

- [1] <http://jddd.desy.de>
- [2] E. Sombrowski, P. Gessler, J. Meyer, A. Petrosyan, K. Rehlich, "jddd in action", ICALEPCS'09, Kobe, Japan, October 2009.
- [3] E. Sombrowski, K. Rehlich, "First Experiences with jddd for Petra Vacuum Controls", PCAPAC'08, Ljubljana, Slovenia, October 2008.
- [4] E. Sombrowski, A. Petrosyan, K. Rehlich, P. Tege, "jddd: A Java Doocs Data Display for the XFEL", ICALEPCS'07, Knoxville, Tennessee, October 2007.