

# FIRST EXPERIENCES WITH JDDD FOR PETRA VACUUM CONTROLS

E. Sombrowski, K. Rehlich, DESY Hamburg, Germany

## Abstract

The commissioning of the rebuilt PETRA and the to be built XFEL accelerators at DESY, requires the creation of numerous new control panels. For a quick and easy creation of these panels the Java Doocs Data Display jddd [1] has been developed. jddd is a Java editor for designing and running control panels. It provides a rich set of ready-made components/widgets, e.g. status indicators, logic components to animate graphics and powerful plot components including online data analysis functionality. The panels are saved in an XML file format and kept in a centrally hosted subversion repository to allow proper bookkeeping and history tracking. Currently the PETRA vacuum controls are used as a real live test for jddd. This way the innovative concepts are being evaluated and the data access to multiple control systems, currently DOOCS, TINE and TANGO, can be optimized.

## INTRODUCTION

Since 2007 the PETRA accelerator at the Helmholtz research center DESY is converted into the most brilliant storage-ring-based X-ray source worldwide. It will offer excellent conditions for top-class research with extremely intense and sharply focused X-ray light. The user operation will start in 2009.

The commissioning of PETRA has a tight time schedule, which means much work for the engineers as well as for the control groups. The first controls needed are vacuum controls. These displays are currently designed with jddd. The high complex vacuum displays are an ideal test for jddd and the underlying jdoocs API.

## THE JDDD EDITOR

jddd has a graphical editor with a set of different ready-made components/widgets for control panel design. With the assistance of this editor synoptical displays are designed in a short time without any programming effort or knowledge. All displays may be used as generic library components and may be added to other jddd displays. The created panels are saved in an xml file format including a Base64 encoded thumbnail image of the panel. These xml files are parsed and executed in jddd run-time mode.

The main goals for the editor design were stability, extendability, high performance, a standard, ergonomic and intuitive machine interface and a modern look & feel. Java was chosen as programming language for its operating systems independency. jddd is started using the

Java Web Start technology. The Web Start links for the editor and runtime version are available on the jddd homepage [2].

In the following paragraphs a couple of useful jddd components will be introduced and some special jddd features like the subversion browser and the plugin interface will be described.

## SPECIAL COMPONENTS/WIDGETS

### Plot Components

The plot components are designed for high performance data presentation. Four different plot types are currently available in jddd: Time domain, history, location and XY plots.

In the plots multiple channels can be displayed and new curves can be added by drag and drop. With the mouse a data region may be selected to analyse the data in this selection, e.g. using fourier transformations, linear fits, displaying RMS, min/max values or showing histograms or making data correlations (see screenshot figure 1).

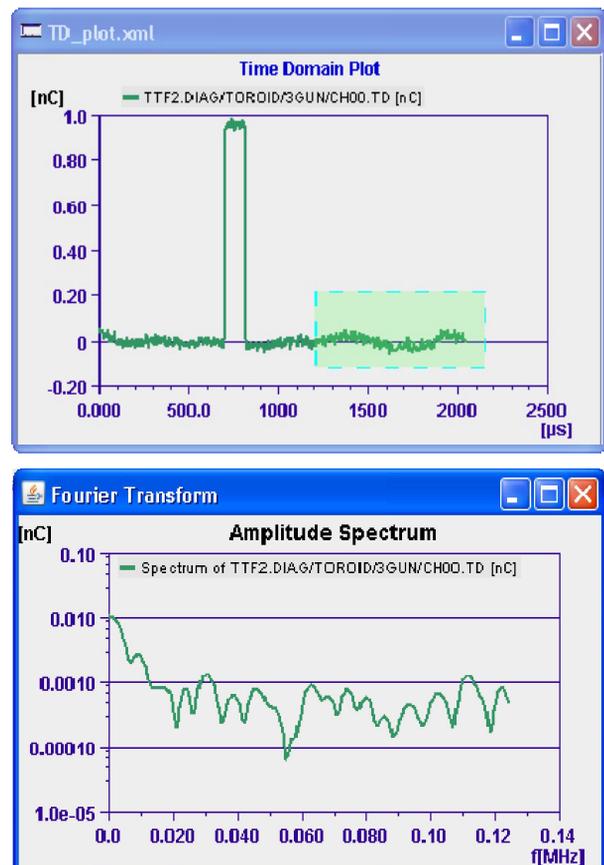


Figure 1: Time domain plot with FFT of selected range.

## Dynamic Lists

The dynamic list is (as the name implies) a component used for dynamic panel creation. It consists of a panel where different kinds of jddd components, displaying control system values can be placed. On editing only the components for one location have to be added. At runtime jddd automatically fills the lists with components for all available locations. In figure 3 a dynamic list in run mode is located in the upper right corner of the panel. In this list all available pump stations with corresponding pressure values are displayed.

A special feature of the dynamic list is the location filter. If the location of the list is set to “SOL\*” only locations starting with the letters “SOL” will be displayed.

## Buttons

A jddd button cannot only be used to set one or multiple control system values, it may also be assigned various other functionalities like:

- Open new window, replace window, override window
- Set component property (to set one or multiple property values)
- Execute shell command
- Print panel
- Help (opens a predefined web page in a browser)

The most interesting functionality for creating interactive displays with jddd is the “Set Component Property” functionality, which can be used, e.g. to change the content of a Dynamic List or the reference to an xml file of a library component in run mode.

This button feature has been well approved by the users of the vacuum displays.

## A SUBVERSION REPOSITORY FOR DATA STORAGE

A subversion browser was developed using the svnkit library [3]. With this browser, files are committed to, loaded from, updated, moved or removed from a user specified subversion repository. For file browsing two different views are available:

In the default “Info” view all information about the most current file version are displayed. For jddd files these information include the version number, the date, the file size, the author name, a description text and a thumbnail preview (see screenshot figure 2).

In the “Change Log” view older file versions can be viewed and loaded.

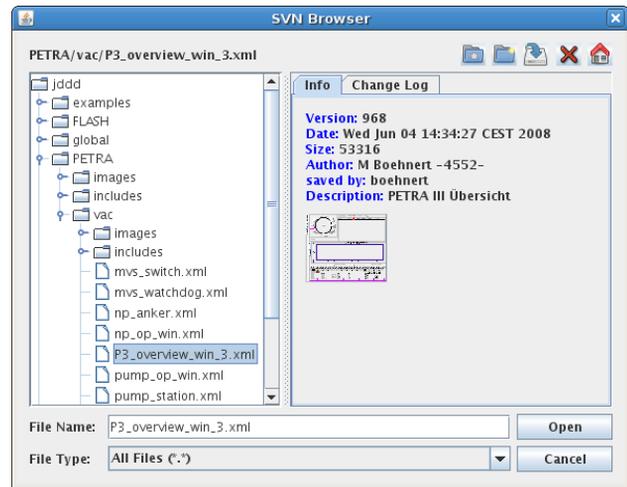


Figure 2: The subversion browser.

## THE PLUGIN INTERFACE

Many people have spent a lot of time to develop high level graphical Java beans with very special functionality. These beans may be imported to jddd using a plugin interface. Only a plugin.jar is needed containing the wrapper classes for each java bean. If the plugin.jar path is set in the jddd preferences, the beans which are included in the jar will automatically be visible in the component palette and may be added to a panel using drag and drop. All bean properties which are specified in the wrapper classes are editable in the jddd component properties table using the standard property editors.

The advantage of this plugin technology is, that the plugin beans keep their direct connection to the local control system. A name server is not required. The plugin beans can be connected to any control system and not only to the restricted number of control systems which are supported by the jdoocs API.

## FIRST EXPERIENCES USING JDDD FOR PETRA VACUUM CONTROLS

jddd has proved to be an excellent tool for quick and easy control panels design. With a rich set of powerful components/widgets (like logic components and button with the “Set Component Property” functionality) it is possible to create not only simple test panels, but also high level displays with an interactive layout (see vacuum display, figure 3). jddd is fast enough to operate highly complex control system panels with more than 1000 components. Using library components which can be exchanged on a button click at run time, the loading time could successfully be reduced to a minimum.

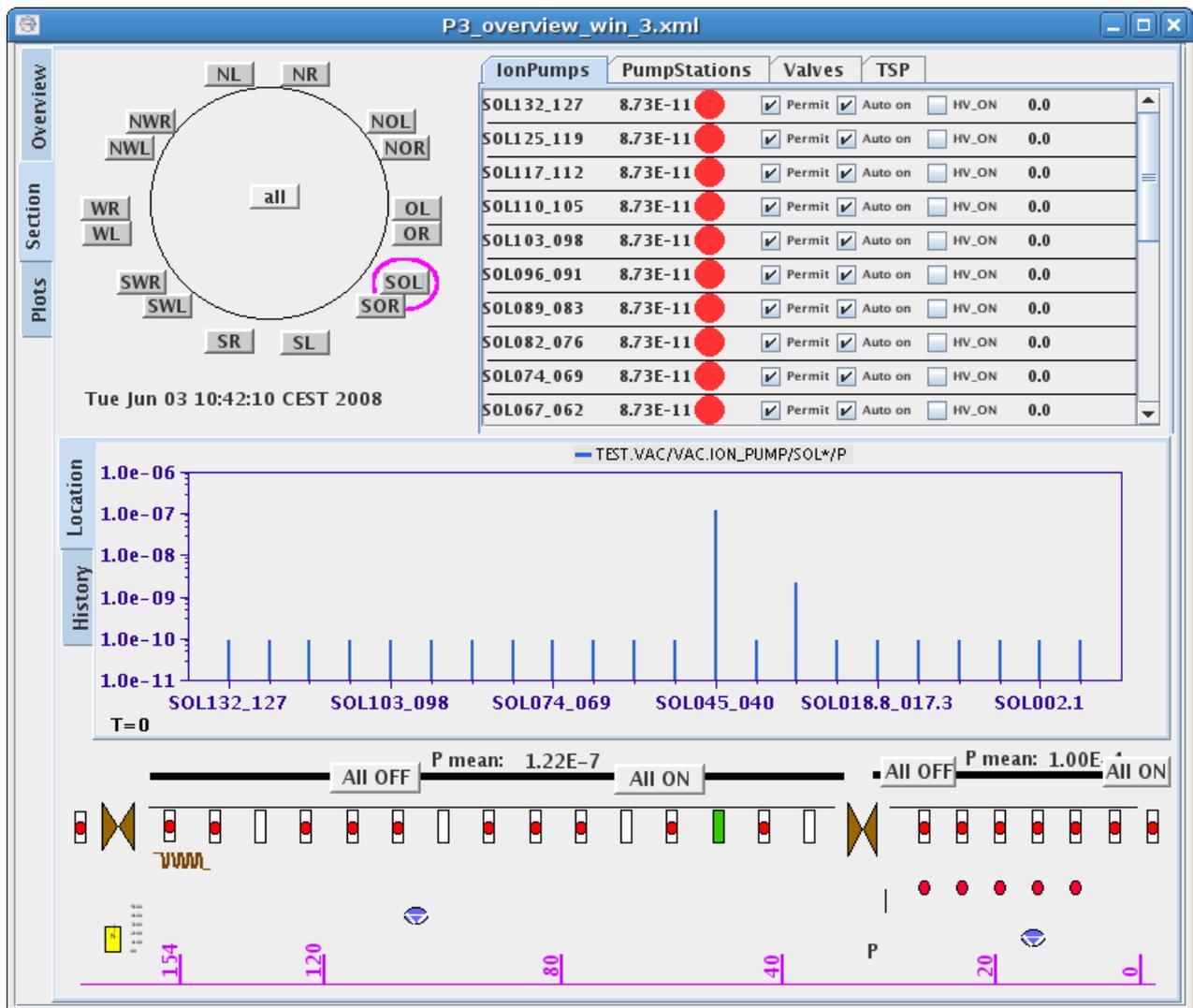


Figure 3: Running jddd panel for PETRA vacuum controls.

The subversion browser allows to store the control panels in a central place and keeps the history of the panels. Many developers are able to work on the same displays.

With the plugin interface external Java beans like ATK(Tango) and ACOP(Tine) beans are easily integrated into jddd. This enables other institutes to keep their specialized components/widgets. For the vacuum displays this feature has not been used, but it may be useful in the future.

## OUTLOOK

The next steps in jddd development will be an improved interface to the TANGO and a new interface to the EPICS control system. This way all control systems

used at DESY will be supported by jddd. Furthermore it is planned to implement asynchronous DOOCS calls and to read history data from the data acquisition system (DAQ) and to display these data in jddd panels.

Another idea is the development of a jddd middle layer server. Because searching files in the subversion repository could be accelerated, this server may be used for caching the subversion file system. It could also be used to keep a list of the xml file dependencies, to provide for runtime statistics and to update panels in running applications

## REFERENCES

- [1] E. Sombrowski, A. Petrosyan, K. Rehlich, P. Tege, "jddd: A Java Doocs Data Display for the XFEL", ICALEPCS'07, Knoxville, Tennessee, October 2007.
- [2] <http://jddd.desy.de>
- [3] <http://svnkit.com>