An Object-Oriented Data Display for the TESLA Test Facility

K.Rehlich
DESY, Notkestrasse 85, Hamburg, Germany

Abstract

This paper describes an object oriented editor for designing and running graphical user interfaces for control screens of the Distributed Object Oriented Control System (DOOCS). The whole DOOCS system [1] is based on the object-oriented methodology, i.e. devices are handled as objects, data objects are transferred between client and server programs and the data is visualized by display objects. The control system has to display and change the parameters of many devices of the same type. With the drawing editor the user creates a single instance as a synoptical representation of such a device and reuse it in all locations of the display by copy and paste commands. The device display is animated by actual device data and may contain sensitive fields to activate commands to the devices or to start new sub-windows. The required information is passed to the sub-window, therefore, a single prototype of a sub-window has to be created once for all devices of the same type. Historical trends or digital scope plots are also integral parts of the DOOCS data display (ddd) program.

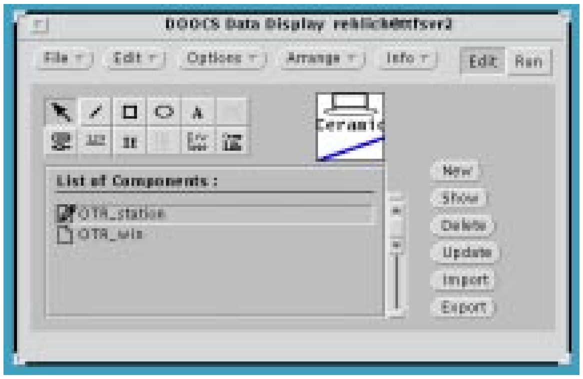


Figure. 1. Main window of the ddd editor. The ‘OTR station’ is selected and shown in the small preview window.

1 Introduction

The DOOCS Data Display is a program to draw components for a control system with an editor and to run the created drawing with real data. Two versions of the program are available. One is to draw and run or test the drawings. The second one is a run-time version only which does not allow change of the drawing.

A control system has a lot of similar devices. All this equipment has the same types of properties to display and to control. Just the device addresses are different. The main improvement of ddd compared to other tools is the possibility to create reusable components. A once-created component may be used in several places in a synoptic diagram or displayed in separate windows. The functionality of the components are the same. They only differ in the addresses to access the data from the control system. The ddd program uses a hierarchy of addresses for passing to the reusable components.

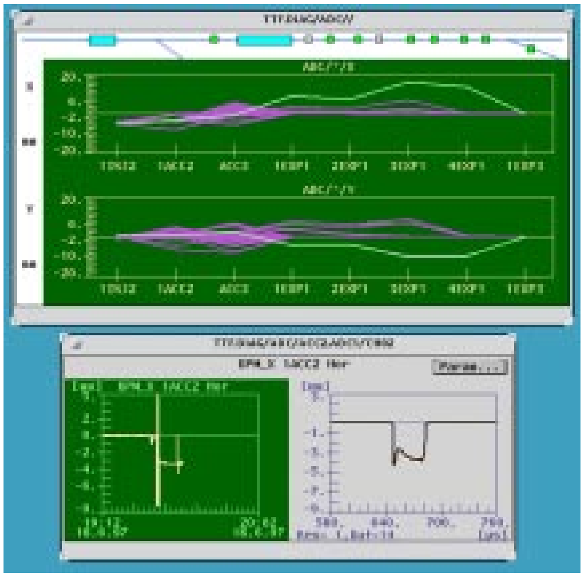
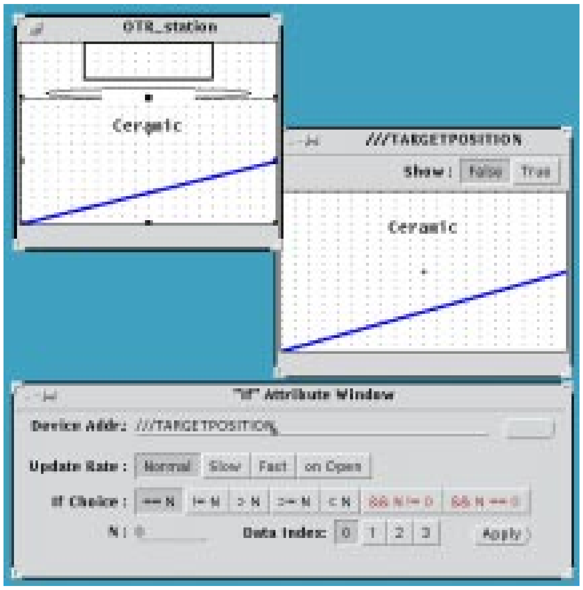


Figure. 2. Plot of the horizontal and vertical beam displacements of the TTF linac as read from the beam position monitors and a single monitor signal including the historical trend.

2 The editor

All synoptical pictures are created with the assistance of the ddd editor. The editor permits the drawing of simple shapes like lines or text and complex elements which are used to display and control data in the control system. These elements are used to display the device status as floating point or integer values, as text strings, status register or different types of plots. Parts of the drawing may depend on the data of a device. Any group of shapes can be drawn to depend on on-line data. An example is shown in figure 3. Furthermore, components or whole windows are stored in a library and can be reused in other drawings. These library components may consist of device dependent data also. Such a component could be a view screen/OTR station for instance that shows which of the targets is inserted into the beam pipe. This library component is later used by drawing an outline in a paint window. The component is scaled to the drawn outline and the address of the device has to be specified in a panel. Further clones of this view screen station are generated by copy and paste commands. Each clone then gets its individual relative device address. The addresses are propagated to the subcomponents in a hierarchical manner. Components can also be created by including other library components that contain other library parts.

This view screen example shows the basic information of the device only. A library element can contain sensitive fields that allows activations of a further window with detailed information during run-time. These windows are drawn once for all devices of the same type. A click on the sensitive field passes a base address to the activated window in order to display the device specific data of the selected device only. Several devices of the same type, but with different addresses in the control system can be displayed at the same time.



Show : False

Device Addr: ///TARGETPOSITION

Update Rate : Normal

If Choice : == N

N : 0

Data Index: 0

Figure. 3. This example shows the creation of an ‘OTR station’ component. The five different target positions are animated by the ‘TARGETPOSITION’ property of the device and are specified in ‘if’ conditions.

3 Running the application

The synoptical displays for the control system are organized in a hierarchical manner, i.e. by mouse clicks the user selects more and more detailed information on certain subsystems or devices. All information including archived data and other types of plots can be accessed just by mouse clicks from the graphical status displays.

4 Implementation

The whole editor and run-time environment is written in C++ as class libraries. Output can be generated on any X-display and as PostScript in a file or on a printer. The drawings are stored in ASCII format.

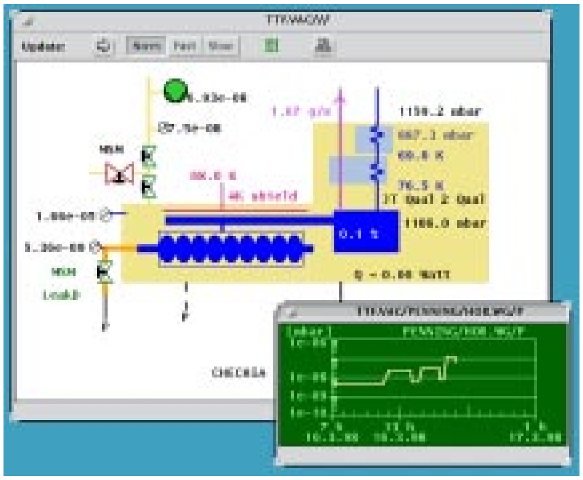


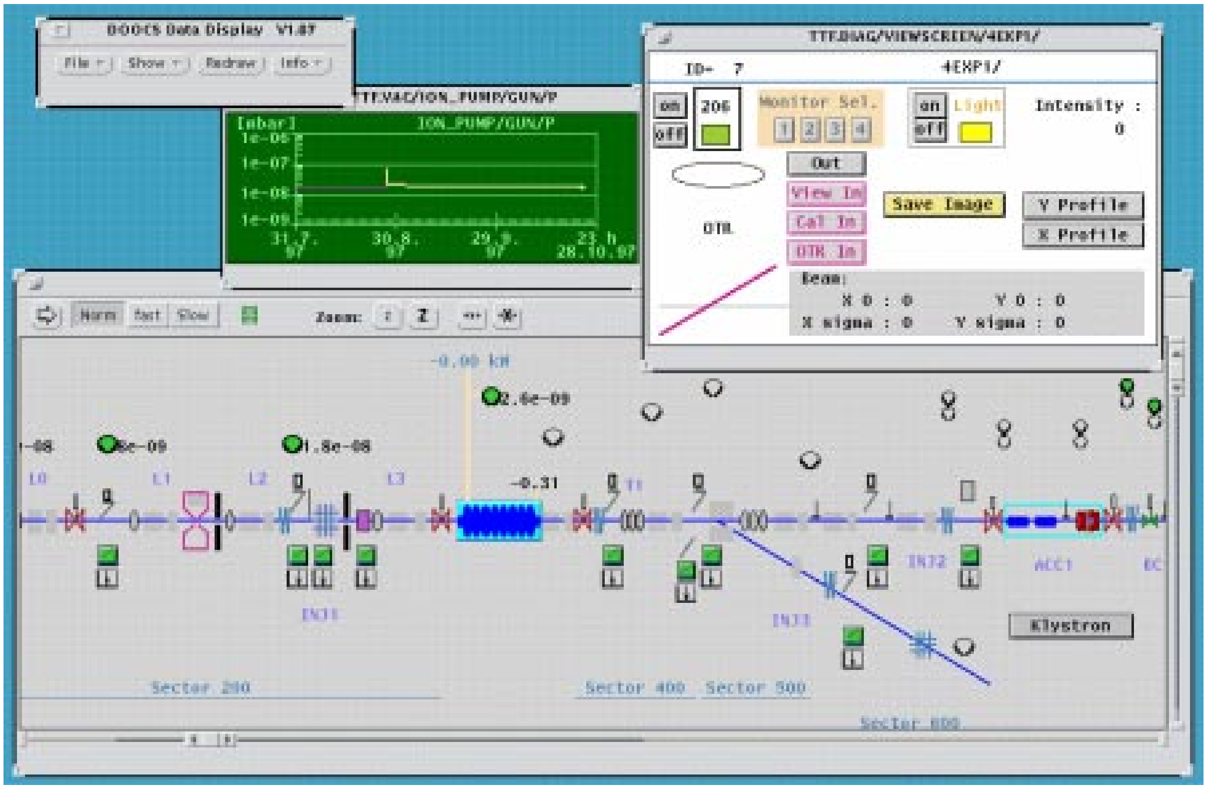
Figure. 4. Synoptical display of a cavity test stand. The status of the devices is indicated by different colors and different forms.

5 Conclusions

With the assistance of the *ddd* editor synoptical displays of a control system are created in an efficient way since all parts can be reused. For TTF we created several hundred of displays in a short time.

References

[1] S. Goloborodko, G. Grygiel, O. Hensler, V. Kocharyan, K.Rehlich, P.Shevtsov ”DOOCS: an Object-Oriented Control System as the Integrating Part for the TTF Linac”, ICALEPCS 97, Beijing.



ID= 7

4EXP1/

Intensity : 0

Beam:

X 0 : 0 Y 0 : 0

X sigma : 0 Y sigma : 0

Figure. 5. Zoomed display of the TTF linac. The visible part can be shifted by the sliders. A click on an object activates a window with detailed information (e.g. a viewscreen station) or a plot (e.g. history of the pressure of an ion pump).