XCON - A Motif Based Toolkit for Graphical User Interface Design for Control Applications

L. Kopylov, M. Mikheev
Institute for High Energy Physics
Protvino, Russia.
B. Hendricks
Fermilab
Batavia, IL, USA

The XCON toolkit has been created and used for the development of a number of applications for the Fermilab and IHEP accelerator control systems. It provides an easy way to create a sophisticated graphical user interface for control applications, in particular ones with complex synoptic diagrams. The toolkit is based on Motif and contains a wide set of control widgets in addition to a powerful graphical editor and has the capability of including GIF images as a screen background. Equipment access is implemented as a library layer which allows this package to be adapted to other control systems. A description of the XCON toolkit and a few examples of its use are presented in this report.

1. Introduction.

The Graphical User Interface (GUI) is an essential part of accelerator control applications. The large quantity of equipment and the complexity of physical and technological processes place serious demands on the operator's and engineer's professional knowledge to control and maintain the facility. An approach should be considered in application development which facilitates operation of the accelerator on the basis of a general knowledge of physics and technology. The solution is to develop an application with a powerful and comprehensive GUI that contains enough descriptive information about the facility to be controlled. The most advanced type in this direction is a synoptic-based GUI. It presents a schematic drawing of the facility or schematics of technological processes together with the set of the parameters to be controlled.

A number of commercial systems provide such solutions, for example Vsystem [1] or EPICS [2]. As a rule, those systems are a general complex solution of a facility control task including, together with the GUI, the internal database, equipment access and network communication. For a control system developed from scratch this is an attractive solution, while for an existing mature control system it is not the case because a complete set of services is already implemented which has its own local peculiarities. In such a case, the implementation of a complex package could be an expensive solution or not acceptable at all.

While developing the package called XCON we had a goal to provide a graphical tool to build a GUI for control applications that would allow powerful and comprehensive synoptic diagram development with minimal programming and which could be easily adopted into an existing software infrastructure. Despite a large number of commercial packages for Motif GUI development (eg. DecVUIT, FaceMaker, and X-Designer), such activity still requires special software support and remains very time and effort consuming. XCON liberates the developers from routine coding and allows them to concentrate on the technological and engineering aspects of application design.

2. Package contents.

An interactive graphical interface builder utility (XCON) and a runtime kernel for application execution (XEXE) are combined in the package together with a number of routines for file format conversion. XCON is a complete set of tools to create the GUI. It includes a graphical editor and control widget set for data representation and setting. The result of development in the XCON environment is a GUI description file that is executed later under the XEXE kernel. Such functional separation allows for the simplification of the execution kernel and an increase in the efficiency of its operation. Dedicated to GUI development, the XCON utility has an extensive user interface based on the Motif tool kit. A package FIG [3] has been used as a prototype of a graphical editor. An example of a GUI design session in the XCON environment is shown in figure 1. There are three main windows: command, mode selection and canvas window. In the last one, the developer is actually drawing the GUI components.

Conversion utilities allow the user to compile an input for XCON from different descriptive files. In these files, a position and an active element type should be specified. On the basis of this data, XCON builds a frame that can be combined with a desired background later on. This approach is suitable for quick GUI generation for large numbers of similar equipment described in tabular form. The same method was used to port an existing application to the XCON environment for Antiproton Accumulator control at Fermilab.

3. Interface development.

An application GUI consists of a passive background picture and a number of active graphical elements (AE) representing the equipment behavior. The background picture provides additional information to describe the operation of the facility. It could be either a synoptic diagram or a functional schema of the installation. XCON provides a wide set of graphical primitives to draw the background picture including lines, polygons, curves, circles, ellipses, and text. By grouping a few primitives, a complex graphical element can be produced. Once developed, an element could be saved in a library and reused later.

A set of functions is implemented to edit the elements once drawn. It includes Move, Copy, Size, Mirror, Rotate, Delete, and Group/Ungroup functions. The curves are editable by moving individual nodes. Files in GIF format can be included as a background. Multiple space layers in the element description provide the correct element superposition on the control screen. A grid with variable spacing and two coordinate side rulers simplifies the positioning of objects. The position of the object on the canvas can be snapped to the grid nodes. Drawing one selected primitive or performing a desired edit function is referred to as an individual mode of XCON. The mode of operation can be selected either by clicking on the corresponding button on the Mode Panel or by means of a cascading menu in the Command Window. For each mode, the corresponding attributes are indicated on the Command Window (line type, color, filling style, text font, etc.) where they can also be modified.

Active elements (AE) of the GUI present on the screen the state of equipment while the application is running. They show the values of measured parameters and device operational status and allow the setting of control parameters to desired values. In active (running) mode, all AEs are tightly coupled to their corresponding accelerator devices through equipment access library functions. A number of AE parameters should be specified during GUI design. Three groups of parameters can be distinguished: geometric, graphic and parameters belonging to the corresponding equipment. Geometric parameters determine the position and size of active elements on the control screen. The XCON graphical editor is used to specify this group of parameters (so called WYSIWYG style).

Equipment description parameters include the control device name and control device type. The first one defines the addressing of equipment through the control network, while the second one defines the method of device access which includes the data type and the corresponding equipment access routine name. In some control systems, a method is implemented to retrieve the device descriptive information from the database based upon the device name (like dio_lib at Fermilab [4] or EqLib at IHEP [5]). In this case, only the device name should be specified. The rest of the device specific information will be gathered from the database.

Active elements are implemented by the use of widgets. The graphical parameters of active elements define how the information will be presented on the screen (ie. the widget type and its resources). A set of widgets are included in the package for managing analog and digital data including text field, scroll bar, scale, plot, LCD style indicator, and wheel switch selector. The standard Motif widgets were used as well as specialized control widgets [6..8]. The resources of AE are specified through descriptive forms. An example of such a form for a plot widget is shown in figure 1 at the right. The resources that could be modified by means of XCON are the subset of the corresponding widget resources set. The rest are set to their optimal value for this application.

To show the status of a multistate device, a special active element was developed which is called a token. It changes the image according to device status. In the design phase, a set of graphical planes are prepared and are assigned to the corresponding device state. The device type defines how the conversion should be done from device state to the proper token plane. The token might be used for setting data. To do this, an action or a list of actions could be specified. It will appear on the screen as a menu of actions when the token is activated by clicking on it with the mouse. A set of standard actions allows the user to set the desired value to the device, open a new or close the existing window, open a Control Panel, or initiate the execution of the predefined action. The Control Panel (CP) is a combination of AE dedicated for the operation of one complex device. In the design phase, a dummy device name is specified for all AE on the CP. It will be replaced with a real device name during application execution. Figure 2 shows (on the right) the CP for the power supply of a magnet element.

There are two modes of the canvas window, development and simulation. In the first phase, the developer draws and specifies the GUI components using the XCON graphics editor and resource specification forms. Active elements are drawn as a named rectangle in this phase. In the simulation phase, the canvas presents a real view of the designed interface screen. All widgets pop up in their final appearance according to the resources specified even though they are disconnected from the equipment.

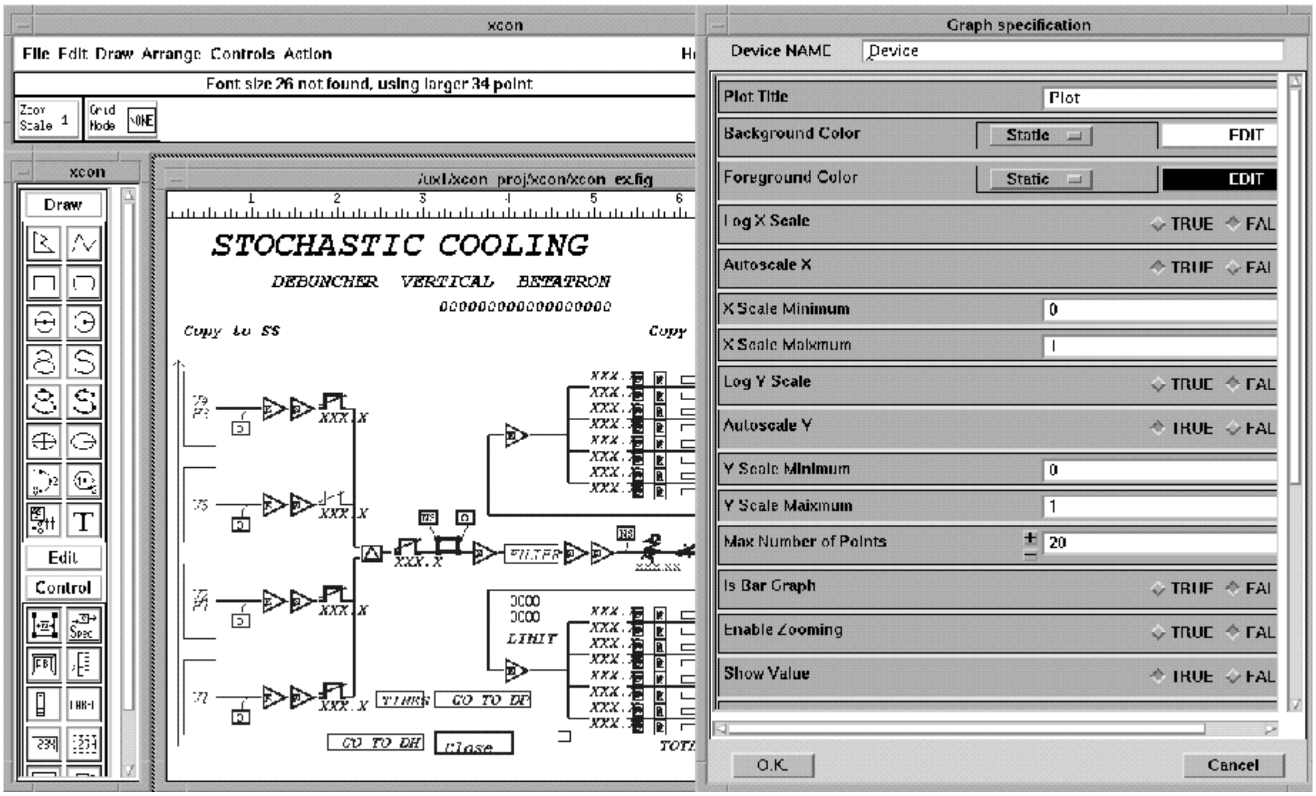

Fig.1 Example of XCON interface design.

4. GUI execution.

For execution, descriptive files are supplied to the XEXE kernel. The kernel creates a window on the control screen with a background and active widgets connected to the accelerator equipment. The widgets reflect the data corresponding to the device states.

The communication is implemented through an existing equipment access library (dio_lib in the case of Fermilab and EqLib in the IHEP version). The executive kernel attaches to the library through the layer of Customization Functions (CF) which convert the specific kernel calls to the standard library functions. The number of specific functions corresponds to the number of different device types. Here is the place where the developer can put his own code creating specific or virtual device descriptions. For most "standard" devices, the communication is covered completely by equipment access functions, and CF serves just an interface purpose.

A refresh of the control screen is implemented by means of periodic polling with a rate specified by the operator. The kernel consequently passes the list of devices presented on the control screen. The polling list is updated each time when a new window appears or an old one is closed. Two specific aspects exist in the implementation with dio_lib at Fermilab. The first one is the extended set of routines in dio_lib that return values in engineering units. This significantly simplifies the CF layer. The second aspect concerns the possibility of requesting a list of devices in one call instead of with consecutive individual calls. Due to that, the device access time was drastically decreased. For such calls, a list of devices should be prepared. XEXE generates a few of these lists combining devices with a similar return data type.

5. Example of application.

On the basis of XCON, a few applications have been implemented at IHEP (Protvino) and Fermilab. Figure 2 shows the prototype of the application for control of the fixed target beam switchyard. This application in now under preparation for the next fixed target run of the Fermilab accelerator complex. The main window on figure 2 contains the synoptic diagram for the beam transfer lines with number of active elements (magnets and loss monitors). On the right, a control panel for a magnet power supply is shown.

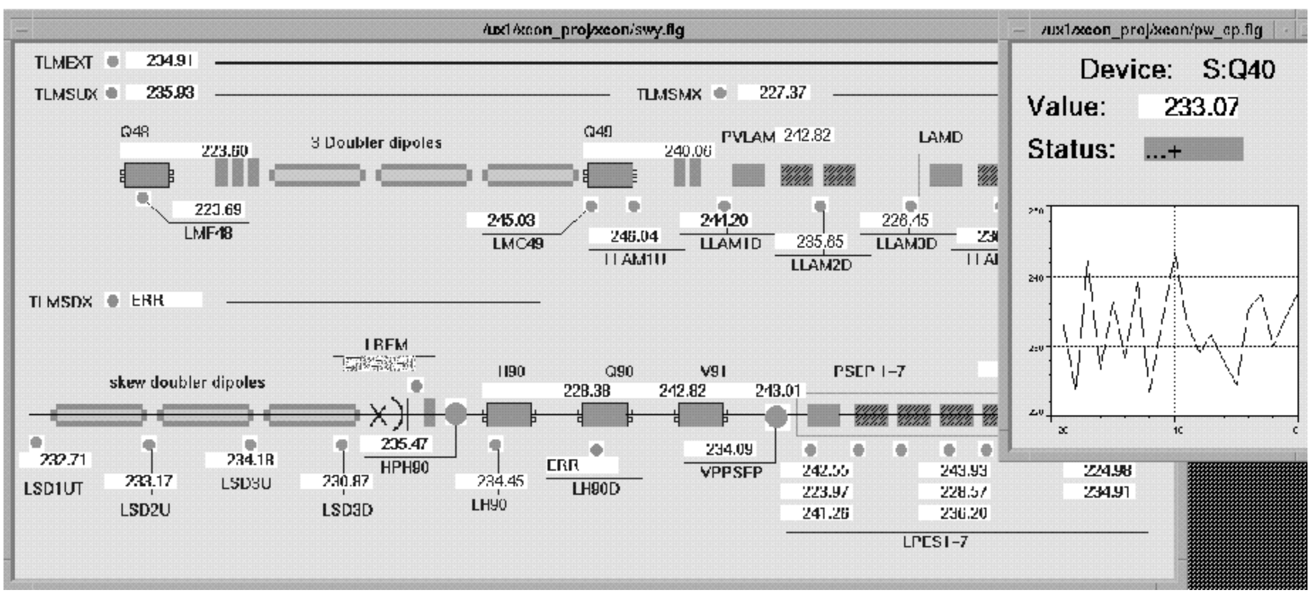

Fig.2 Example of control screen for the operation of Beam Switch Yard

References

1. P.Clout The status of Vsystem. Nucl. Instr. and Meth A352 (1994) 442-446
2. R.L.Dalesio et al., Nucl. Instr. and Meth A352 (1994) 179-184
3. S.Sutanthavibul Facility for Interactive Generation of Figure (FIG), Copyright 1985.
4. B.Hendricks, et al. Overview of the Next Generation of Fermilab Collider Software, ICALEPCS91, Tsukuba, KEK.
5. N.Trofimov et al. The equipment access software for a distributed UNIX-based accelerator control system. A352 (1994) 274-276
6. P.D.Johnson. Control panel widget Set (Xc) v1.3, Copyright 1992.
7. D.F.Flanagan et al., MIT Project Athena widget set (At) v.5, Copyright 1992.
8. P.Antonsanti et al., Workstation as console for the CERN PS. ICALEPCS91, Tsukuba, KEK Proceedings