CAQTDM, AN EPICS DISPLAY MANAGER BASED ON QT

A.C. Mezger, H. Brands, Paul Scherrer Institut, Villigen, Switzerland

Abstract

At the Paul Scherrer Institut (PSI) the display manager MEDM was used until recently for the synoptic displays at all our facilities, not only for EPICS but also for another, in-house built control system ACS. However MEDM is based on MOTIF and Xt/X11, systems/libraries that are starting to age. Moreover MEDM is difficult to extend with new entities. Therefore a new tool has been developed based on Qt. This reproduces the functionality of MEDM and is now in use at several facilities. As Qt is supported on several platforms this tool will also run on Microsoft Windows. The MEDM data files (.adl) were used as an initial set for the new system and were converted into the new xml format using the parser tool adl2ui. These were then edited further with the Qt-Designer and displayed with the new Qt-Manager caQtDM. The integration of new entities into the Qt designer and therefore into the Qt based applications is very easy, so that the system can easily be enhanced with new widgets. New features needed for our facility were implemented. The caQtDM application uses a C++ class to perform the data acquisition and display; this class can also be integrated into other applications.

INTRODUCTION

In the EPICS environment of most accelerator laboratories graphical synoptic display tools like MEDM, EDM as well as many others were developed. These tools have been used for many years with success, but many of them are currently reaching their end of life due to the evolution of the graphic platforms. Moreover these in-house solutions are nearly impossible to maintain and to extend, because either a developer has to be found for the older underlying graphic systems or because the systems are likely to disappear in the near future [1] [2].

A new tool should therefore be independent of the underlying graphical system and its components have to be also as much as possible widely supported. The actual solution for these requirements lies in the world of Qt, which provides modern graphical support on many platforms. Qt comes also with its own graphical editor and a developer can easily add new widgets to it, so that an open source (LGPL v2.1) solution that can also be used under commercial terms is available. Most of the work for rendering the data on the synoptic display is then represented by the handling of the data acquisition system.
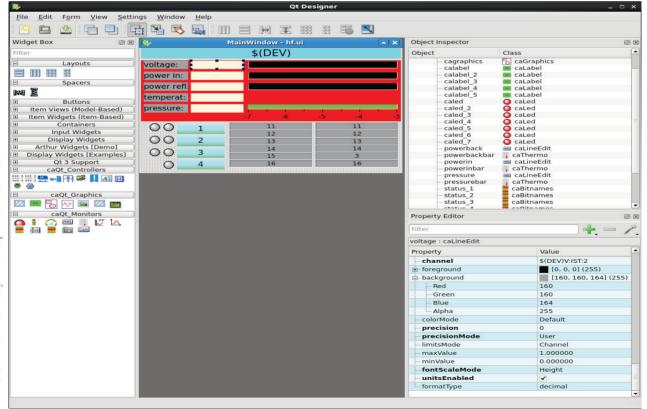

Figure 1: Qt-Designer editor where the graphical objects are situated on the left and can be dragged on the middle window for composing a display. The properties of the objects are display in the right corner window.

QT, A GUI FRAMEWORK

Qt Framework

Qt represents a cross-platform application framework that is widely used for developing applications with a graphical user interface. It runs on the major desktop platforms on top of their native display system, so that a Qt-based application will run on these platforms with the same look and feel. A major advantage of this system lies in the fact that it is used by a huge community and that very detailed documentation is available.

Qt comes with many libraries, an integrated development environment (IDE) **Qt-Creator** and a user interface designer **Qt-Designer**. The Qt libraries are already included in many Linux distributions, albeit not always with a version greater than 4.8. Where necessary newer versions can easily be downloaded and installed. In addition to the libraries, the Qt-Designer and Qt development package should also be installed in order to build graphical user interfaces and Qt-applications.

For Linux all relevant Qt-Packages have to be downloaded and installed. Afterwards the downloaded caQtDM package [3] must be build. For Windows, the windows distribution of caQtDM including EPICS is all that is needed.

Some third party libraries based on **Qt** like **qwt** for 2D graphics and **qwtplot3d** for 3D graphics can be integrated and the caQtDM package uses qwt for drawing strip charts and x/y Cartesian plots. At PSI **qwtplot3d** is used in dedicated applications to display x/y plots as a function of time and can be integrated in caQtDM if the need for it arises.

The current version of Qt used at PSI is **Qt 4.8.2**

Qt-Designer, the Graphical User Interface Builder of Qt

The Qt-Designer (Fig. 1) allows a developer to plug-in his own graphical objects with their methods and properties, so that integrating new graphical control objects is quite easy and will give the user a rich environment for developing user interfaces. Many graphical objects have been integrated, not only the standard MEDM objects like strip plot, cartesian plot, wheel switch, text monitor, bar monitor, etc. but also some new objects for camera display, LEDS, toggles, internal channels and many more.

The include object, called composite widget in MEDM, can now also be represented in Qt-Designer. This was not the case in the first versions of the plug-ins as was presented in earlier communications [4].

The Qt-Designer gives the control display "architect" a performant interface to draw his displays. It is straightforward to drag an object into the editor window. As soon as the object has been placed and resized, the properties of the object can be edited. The most important one is of course the process variable (called channel in EPICS 3). This property may present a variable as in

MEDM that will be replaced by defining a macro at run time of caQtDM.

Actually 30 control widgets are integrated, these widgets are not directly CS-aware, but provide the properties and methods to be used with applications. The application caQtDM will then add the control information coming from the control system (usually EPICS).

CAQTDM, THE DISPLAY TOOL

Requirements

caQtDM has been developed with Qt version 4.8.2 qwt version 6.0.1 and EPICS version 3.14.12 and has been extensively tested with these versions. Since version 3.2.0 of caQtDM the package can be built also with Qt version 5.1.0 and qwt version 6.1.0. However this version has not yet been thoroughly tested and a bug in Qt 5.1.0 must be patched (QTBUG-33130) in order to treat some properties correctly.

Components

For caQtDM, many C++ classes had to be developed. First of all, the necessary graphical objects were created. These objects are pure graphical objects and could be used for other purposes. The objects are plugged into the Qt-Designer. With this tool a display description is written that will be read by the caQtDM_Lib class that will combine the display definitions with the control system data. This architecture separates display definitions and control system data acquisition; we think that this allows to support multiple control systems in a clean way. At our site we support simultaneously 2 control systems, EPICS and an older in house made system ACS. caQtDM uses this class and represents the stand-alone application for the control synoptic displays. The architecture of the different components is shown in Fig. 2.
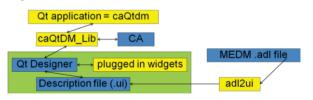

Figure 2: Architecture of the Qt synoptic display package.

In case of GUI applications where more logic and data treatment is required, the above class can be integrated. The user does not need to bother about the data acquisition while provided by the library class. Access to the display items and data can then be used to enhance the application with the necessary logic.

At PSI caQtDM supports two control systems, EPICS and our in-house made control system ACS. Additionally a skeleton for EPICS4 can be activated by selecting **the appropriate** option when building the package. Other control systems could be easily integrated by filling the internal data container.

```
                              XHIPA:TIME          09-09-2013 18:48:44
Ion Source
IVVH2OF:SOLA:2            59.30 %  IVGF:IST:2            5.95 cc/min
IVMWFP:SOL:2                419 W  IVMWFP:IST:2           413 W
IVGD:IST:2             162.38 bar  IVMWRP:IST:1          0.00 W
IVMWFPA:SOL:2                0 W   IVMWFPA:IST:2         0.00 W
                                   IVMWRPA:IST:2        -0.183 W
IVWC:SOL:1            -50243 Units  IVWC:IST:2          -2.299 A
Ion Source ECR autotuner
IVMWTX:IST:1             1.035 V   IVMWTY:IST:1          0.927 V
Extraction
EGRID:IST:2             55.42 kV   EVEX:STR:2           24.70 mA
EVEX:IST:2              61.38 kV   EVF:STR:2             0.38 mA
EVF:IST:2               -1.90 kV   EVFI:SOL:2            1.00 mA
Extraction
GV2:IST:2          1.322e-05 mbar  GV3:IST:2        2.959e-06 mbar
```

```
!grid 2
comment comment -span 1 -bg "#ffffff" " "
XHIPA:TIME                              formRead
comment comment -span 2 -fg "#ff00ff" Ion Source
IVVH2OF:SOLA:2                          formRead %5.2f
IVGF:IST:2                              formRead 5.2

IVMWFP:SOL:2                            formRead %5.0f
IVMWFP:IST:2                            formRead %5.0f
```

Figure 3: Example of display generated with the scripting language shown below.

The adl2ui tool is included in the installation package to convert MEDM display files to the caQtDM xml files. This tool presents an easy and fast path to switch from MEDM to the new synoptic display tool. However at PSI we also modify the resulting screens in order to have a consistent look and behavior. A style guide for developers has therefore been written for this purpose [5].

For fast prototyping of control displays caQtDM also supports a restricted set of instructions of our in-house developed scripting language [6]. This allows us to display rapidly data in a tabular way (see Fig. 3) or print several representations in a batch manner. This feature is constantly extended.

DEVELOPING / RENDERING A DISPLAY

It has to be pointed out that the package does not consist of a single application capable to edit and run the displays as in other tools like MEDM, EDM,

The toolkit consists of the Qt-Designer as display editor and caQtDM as display run time tool, the Qt-Designer being delivered with Qt and supported by Qt.

When developing a control display from scratch, the Qt-Designer tool is used. The various control objects are located in the left panel and can be dropped on the user display situated normally in the center panel.

The properties for the selected object can then be changed in the property panel. In order to have some default styles for the control panel a default style sheet should be loaded for the "mainwindow", the parent object of the control display. This style sheet is provided by the package and its contents can be pasted into the style sheet definition. In the style sheet you will be able to change some styles, but one of them represents the background of

your control display that can be changed by specifying the color with Qwidget#centralWidget {background: rgba(187, 187, 187, 255);}

The most important property of a control object is of ourse the process variable (pv) name. In case of some objects like the strip tool several pv's can be given separated by a semi-column. The display rate by default is 5 Hz and by specifying a JSON string of the following type after the pv this rate can be modified: {"monitor":{"maxdisplayrate":20}}. This is useful when displaying camera images or when one really wants to limit the rate for performance reasons.

Qt-Designer has also a concept of layouts that will not be described here. Using this feature gives the possibility to control the resizing of a display. However this feature is normally not needed, because caQtDM will resize the control displays hopefully correctly in all cases and will also respect the user layout behavior.

After or during the development of the control display the display may be viewed with its control values by the application caQtDM. This application provides also a function to reload the already displayed control panel, so that at any time the panel can be updated.

CONCLUSIONS

Qt-Designer extended by our plug-ins and caQtDM represents a good control system environment for developing and rendering control system data displays and a powerful replacement for the well known MEDM tool used in the EPICS community. Moreover the tools are not restricted to EPICS and other control system could be easily supported.

Also the fact that these tools run on many platforms, including Microsoft Windows and are not bound any more to X-Windows, allows the package to be ready for next generation graphical platforms like Wayland [1] and Mir [2]. This presents a wide range of platforms and gives thus the possibility of a rapid deployment of the tools.

ACKNOWLEDGMENT

REFERENCES

[1] Information about Wayland can be found at http:// wayland.freedesktop.org

[2] Information about Mir can be found at http://unity.ubuntu.com/mir

[3] caQtDM package can be found at http://epics.web.psi.ch/software/caqtdm

[4] Epics collaboration meeting at SLAC, spring 2012.

[5] A. Luedeke, "Application Development Style Guide for the SwissFEL", internal PSI communication, August 2009.

[6] W. Portmann, "Pep Scripting Language Manual", internal PSI communication.