caQtDM: a Qt-based EPICS display manager, status report

Helge Brands, **Anton Mezger**, Paul Scherrer Institut

Wir schaffen Wissen – heute für morgen

At the Paul Scherrer Institut the new Qt-based EPICS display manager has successfully replaced the previous used display manager MEDM.
Qt and therefore Qt-based applications present many advantages like portability and usability over several platforms, independent of the underlying graphical layer, integration of new display objects by the Qt plugin mechanism and has its own extendable graphical design editor.

We will give a short explanation of the Qt-based systems, show the possibilities of our EPICS display manager and how the display manager can be used on different platforms.

Moreover we will give some information about the EPICS-QT initiative and how several products can integrate into a single one.

- ❑ Qt: a new system for writing graphic user interfaces.

- ❑ The Qt-Collaboration.

- ❑ caQtDM: our new display manager.

- ❑ caQtDM: features.

- ❑ caQtDM on IOS, problems to solve,
    startup of App,
    human interaction.

- ❑ Writing applications using caQtDM core libraries.

- ❑ adl2ui, the translator for converting MEDM files.

- ❑ Conclusion.

- ❑ Offline Demo on IPAD.

What is Qt:

- **Qt** is an application framework used for developing application software with a graphical user interface.

- **Qt** is open source, but enterprise licenses can be obtained for support.

- **Qt** is a cross platform framework, thus available for Desktop, Mobile and Embedded devices.

- **Qt** is widely used and programming techniques can be easily found (OOOH, Google, smurfs!).

- **Qt** provider is very reactive to new developments as well as to problems.

Qt
Code less.
Create more.
Deploy everywhere.

A new collaboration has been setup last year to develop new EpicsQt tools by:

- The Paul Scherrer Institute

  **caQtDM display manager and graphical widgets
  the .adl to .ui conversion tool**

- The Australian light source

  Australian display manager and epics aware widgets

- Lyncean technologies

  Science applications (Python)

Both display managers can integrate the functionality of the other widget sets.

caQtDM is an EPICS display manager written in C++ and using Qt as graphical user interface:

- caQtDM is PSI's actual successor of MEDM

- caQtDM can be extended with other control systems (at PSI 3 CS were used simultaneously)

- caQtDM follows mainly the MEDM philosophy, Composite=Include files

- MEDM .adl files can be converted to .ui files with the adl2ui application

- caQtDM can be easily extended with other plugins (Australian lightsource) and with other widget sets.

- caQtDM relies on the Qt-Designer for developing GUI's
    - ⟹ modern open source as well as commercial tool
    - ⟹ powerful, reliable, plugin support for widgets.

- Resizing of displays ⟹ fast and reproducible (fonts resize too !)

- Any custom widget can be integrated.

- Widget set is growing.

- Performance optimizations and use of multiple cores when available
    (waveforms with several 100 kpoints, images, waterfall plots)

- Libraries of caQtDM can be used to develop new Qt applications

Qt is widely used for many platforms, i.e. caQtDM will run on these platforms too when EPICS is available:

as of today: **Linux's, Windows, MacOS and IOS**

The PSI site for caQtDM allows to download the sources, as well as installation packages for Windows and MacOS.
http://epics.web.psi.ch/software/caqtdm/

Why IOS:
When constructing a new accelerator, it is important to be able to test the apparatus on the spot !!

For IOS an App can be made available too and is already running on our Ipad's (However can only be distributed when the application follows Apple's rules and requirements. This has not yet been verified).

Philosophy of Ipad/Iphone is different from a classical pc:

1. It is a touch device only (no mouse)
2. It has only access to its application sandbox
3. For distribution, Apple licensing is necessary

⟹ the application needs another approach for loading files and human interaction

caQtDM has some extensions to account for these:

- All these extensions are actually developed with Qt-code and the latest version of Qt5.4 is necessary on IOS 10.

However on Mac OSX and IOS some problems with EPICS still exist that have to be addressed by the core developers:

- disconnection of channels when the ipad is closed. (port closing?).
- Spurious disconnection and reconnection of channels.

At startup a configuration file is loaded with information about http file server and configuration file, that can be changed by the user:

```
<?xml version="1.0" encoding="UTF-8"?>
<configuration>
    <url value="http://epics.web.psi.ch/software/caqtdm/qtDir"/>
    <url value="http://epics.web.psi.ch/software/caqtdm/qtDir"/>
    <config value="ProScan_MA85.config"/>
    <config value="HIPA_MA85.config"/>
</configuration>
```

The configuration file contains environment definitions, url display path, caQtDM display path and launcher file to be loaded.

```
EPICS_CA_MAX_ARRAY_BYTES 150000000
EPICS_CA_AUTO_ADDR_LIST NO
EPICS_CA_ADDR_LIST some_gateway
EPICS_CA_SERVER_PORT 5062
CAQTDM_LAUNCHFILE launcher.ui
CAQTDM_URL_DISPLAY_PATH http://epics.web.psi.ch/software/caqtdm/qtDir/Office
CAQTDM_DISPLAY_PATH ../Documents
```

All other ui files will be loaded from the server when needed !

Gestures have to be implemented for human interaction:

1. Fingerswipe Gestures are implemented for closing windows and cycling through displays.

2. TapAndHoldGesture is implemented for getting the context menu

3. Other gestures for moving slider and for wheelswitch are default interactions given by Qt.

Activ objects like sliders and wheelswitches have to be made big enough in order to be manipulated !!

Otherwise your usual files are displayed without problems.

The class libraries can be used to write your own application with GUI support.

Initialisation and display of ui file:

```
MutexKnobData *data = new MutexKnobData();
MutexKnobDataWrapperInit(data);
caQtDM_Lib *mainwindow = new caQtDM_Lib(this, uifile, macro, data);
Mainwindow->show();
```

Adding own monitors:

```
monitorIndex = mainwindow->addmonitor(mainwindow, &kdata, monitorName, this, specData, map, &pv);
```
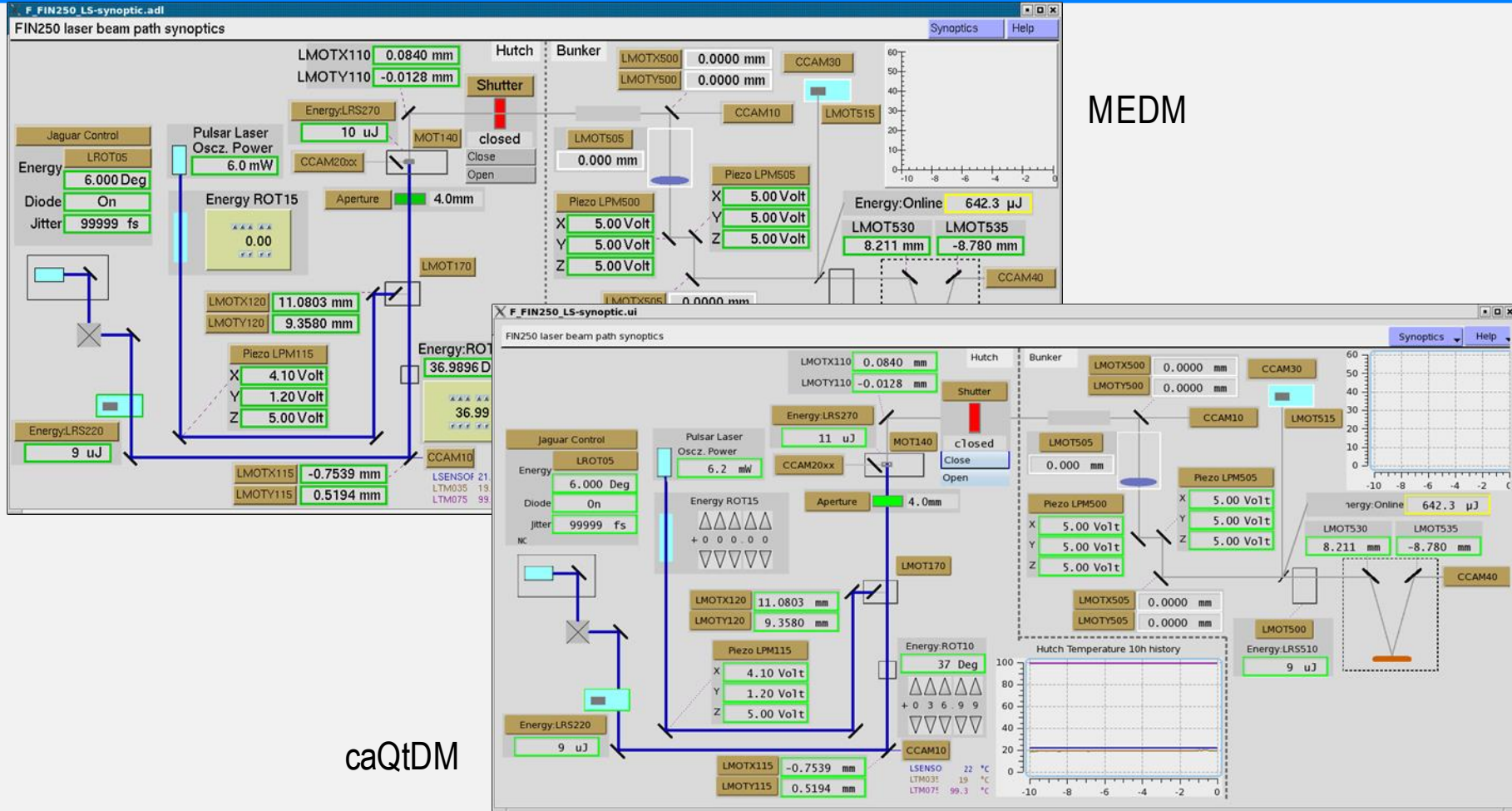
Controlsystem data:

```
knobData *ptr = data->GetMutexDataPtr(monitorIndex);
qDebug() << ptr->edata.rvalue << ptr->edata.units;
```

A complete example can be found on our web site.
http://epics.web.psi.ch/software/caqtdm/

adl2ui, the translator for converting MEDM files.

MEDM

caQtDM

1. Easy conversion of adl files to the new ui files, but files have to be checked!

2. The widget ordering after conversion has some problems: Active widgets can be occulted by other objects. This can be changed in the designer.

3. In Qt, the width of a text widget has to fit the text, but in MEDM the width is not always adjusted.

Conclusion

- caQtDM replaced successfully MEDM, but not everything has yet been replaced; adl2ui gives good results.

- caQtDM has been extended with new features like images, waterfall plot, Led, table and wavetable

- caQtDM has been optimized for performance, so that cartesian plots with over 100'000 points can be used.

- caQtDM runs on many platforms including IOS, where still some problems have to be solved.

- caQtDM is continuously responding to the users requests and bugs finding !!!

- The Qt designer presents a good editor tool.

Acknowledgments:

The authors of MEDM for their code.
Elettra people for providing some widgets.

❏ Thank you for your attention

and have a look at:
http://epics.web.psi.ch/software/caqtdm/