ISBN: 978-3-95450-238-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2023-THPDP001

NEW GENERATION QT CONTROL COMPONENTS FOR HIGH LEVEL SOFTWARE

G. Strangolino, G. Gaio, R. Passuello, Elettra Sincrotrone Trieste, Trieste, Italy

Abstract

A new generation of Qt graphical components, namely cumbia-qtcontrols-ng is under development at ELETTRA. A common engine allows each component to be rendered on traditional Qt Widgets and scalable Qt Graphics Items alike. The latter technology makes it possible to integrate live controls with static SVG in order to realize any kind of synoptic with touch and scaling capabilities. A pluggable zoomer can be installed on any widget or graphics item. Apply numeric controls, Cartesian and Circular (Radar) plots are the first components realized.

MOTIVATION

The ELETTRA synchrotron light source in Trieste, Italy, will be subjected to a major upgrade in the forthcoming years. The rationale is a substantial reduction of the emittance of the stored electron beam, targeting its levels so as to provide a diffraction limited X-ray source also in the horizontal plane. On that account, the new machine, ELETTRA 2.0, will provide intense nano-beams in the range of VUV to X-rays for the analytical study of matter with very high spatial resolution. New magnet design, innovative vacuum technology and revolutionary beam monitoring and orbit feedback systems provide at present a solid basis for the realization of the new machine. New lattice design shall be adapted to the existing ELETTRA storage ring tunnel, building and infrastructure, including the injector and the beam-lines. This implies the substitution of the twelve arcs of the existing storage ring with as many new ones of almost identical length. The new magnetic lattice reduces the present horizontal emittance of 7 nm-rad down to 0.147 or 0.212 nm-rad at 2.4 GeV, largely increasing the brilliance and coherence of the X-ray beam, whilst preserving the injection system. The implementation of the storage ring upgrade will enable ELETTRA to maintain its leadership in the context of synchrotron light sources operating within the same energy range.

ELETTRA 2.0 and the continuous challenges posed by the FERMI@ELETTRA free electron laser encouraged the development of a new generation of Qt graphical components, based on the *cumbia* [1] libraries and the Qt SVG [2] and Graphics Scene [3] technologies. The acquisition of a 55 inches 4K touch screen for the control room inspired the design of the new graphical objects with touch interaction in mind. They are offered by the new *cumbia-qtcontrols-ng* library, which flanks the *cumbia-svg* module.

[1] a minimum impact on the object's "paint" code is an objective, presently not completely accomplished.

TECHNOLOGY

cumbia-qtcontrols-ng is a graphical library providing special twofold components capable of drawing themselves and accept interaction both within traditional Qt widgets and as scalable Qt items within a Qt Graphics Scene. The integration with *cumbia-svg* allows drawing composite *synoptic* user interfaces where static SVG items stand beside live items such as plots, gauges, numeric input objects, and so on.

Qt Graphics View Framework

Qt Graphics View provides a surface (scene) for managing and interacting with a large number of custom-made 2D graphical items, and a view widget for their visualization, with support for zooming and rotation. Graphics View provides an item-based approach to model-view programming. Several views can observe a single scene, and the scene contains items of varying geometric shapes.

Qt SVG

Scalable Vector Graphics (SVG) is an XML-based language for describing two-dimensional vector graphics. Qt provides classes for rendering and displaying SVG drawings in widgets and on other paint devices.

cumbia libraries

The *cumbia* libraries are a multi threaded set of components written in C++ aimed at interfacing the lower level control system servers to the graphical user interfaces for the control room. The separate layers and plugins that make up the framework are designed to integrate with Tango [4], EPICS [5] and potentially any other distributed control system and offer an engine independent interface to the clients.

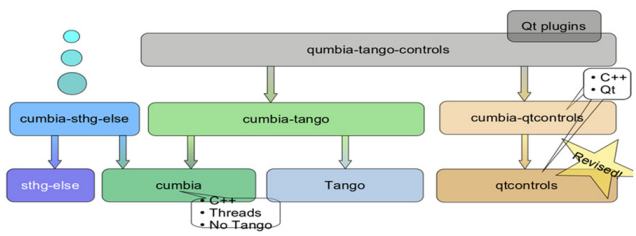

Figure 1: Main components of the *cumbia* library.

An *http* module, relying on the PUMA [6] services, allows for Qt applications running everywhere natively. This strategy has proved to be effective for developers as well as users working remotely.

ISBN: 978-3-95450-238-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2023-THPDP001

CUMBIA SVG

The Qt SVG C++ module provides functionality for handling SVG elements. The cumbia engines and infrastructures can be used to change any SVG element. SVG nodes in the XML document are mapped into *QuGraphicsSvgItem* instances on a Qt Graphics Scene. *QuGraphicsSvgItem* is an extension of the Qt Graphics Svg Item. An external SVG editor, e.g. Inkscape [7] is used to draw. SVG elements in the drawing can be connected to values obtained from the available *cumbia* engines and their properties changed accordingly. In several cases, the connections defined and the type of attributes in the SVG elements bring automatic changes in the representation of the object within the drawing. In more complex ones, the programmer shall map values from the engines to values of the attributes in the SVG DOM document. The library supports the SVG *layer* nodes. The class *QuSvgLayerHelper* can be used to test and change layer visibility. The class also notifies when *sources* (readers) are hidden or shown again, so that the readings linked to hidden elements can be suspended. Fig. 2 shows a snippet of a synoptic application employing a *cumbia svg* view.

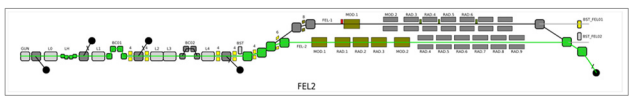

Figure 2: An application with an SVG view.

CUMBIA QTCONTROLS NG

The new generation of cumbia controls focuses on the design of high level software, offering components for custom-tailored data visualization as well as for synoptic applications. The module offers a set of graphical components that are designed to be used both in the context of traditional Qt widgets and as items in a Graphics View. A common painter draws on either paint device, thus presenting the same appearance regardless the technology. In the context of a Graphics View, the objects shall be scalable and integrated into SVG. The library is at an early stage and hiterto offers a small assortment of objects. The diagram in Fig. 3 represents the grounds of the component design.

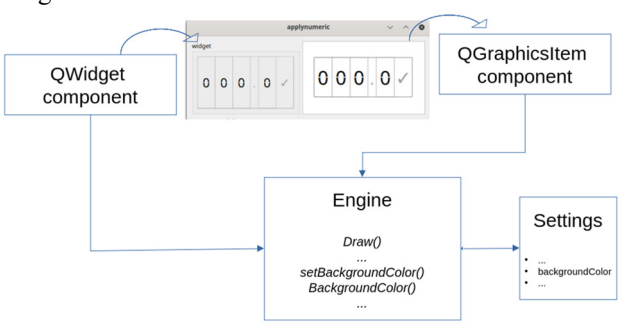

Figure 3: Class diagram of a *cumbia-qtcontrols-ng* object.

Cartesian Plot

The Graphics Plot Item is a Qt Graphics Item that draws a chart. It is highly customizable, supports multiple curves with either scalar or spectrum data (Figs. 4 and 5).

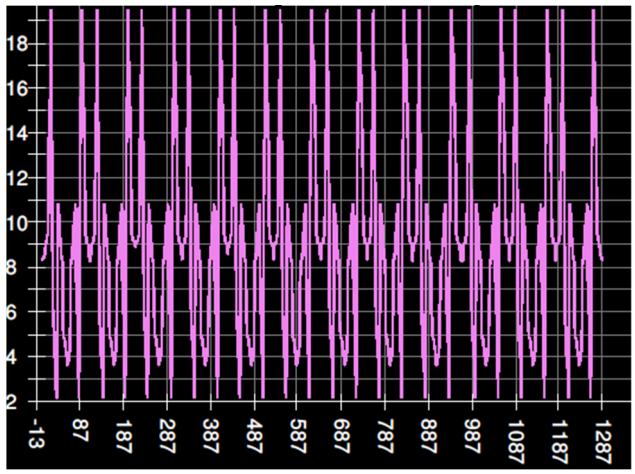

Figure 4: Cartesian Plot. Spectrum data.

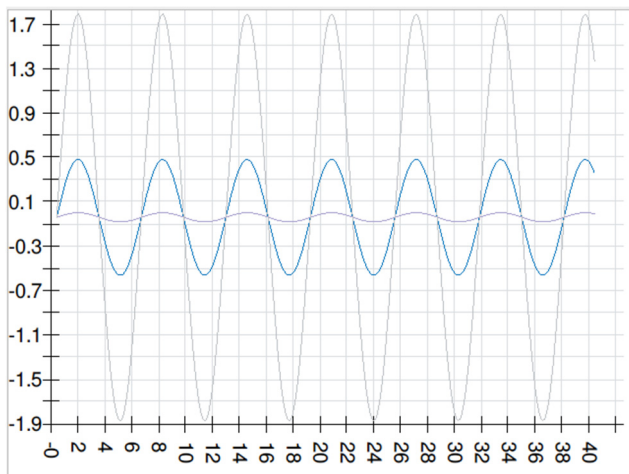

Figure 5: Cartesian Plot. Different style, scalar curves

Numeric Item

The numeric item is a scalar number input widget. Intended to replace the legacy Apply Numeric from the classic *cumbia-qtcontrols* distribution, accepts touch events as well as traditional mouse clicks. A preview of the element can be seen in Fig. 3.

Circular (radar) Plot

Rethinking human computer interaction under the perspective of the ELETTRA 2.0 upgrade, a component expressly tailored to represent values along the storage ring has been conceived to replace traditional histogram charts where spectrum data with umpteen points shall be drawn at extreme refresh rates. The design pursues instinctive engagement of the user in the context of circular machines, notably in the representation of orbit and optics figures. The canvas and the aspect of the curves can be fully customised, as well as the circumference divided into sections mapping those of the facility. Selecting a point on the plot shows the corresponding value at the center of the

ISBN: 978-3-95450-238-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2023-THPDP001

area (0.00) and may optionally start an edit operation. The latter allows a user to drag a point in the curve and change a value on the system accordingly. Figure 6 shows a circular plot in action. The "x axis" can be seen in correspondence of the "0" Y value, concentric to the lower and upper bounds.

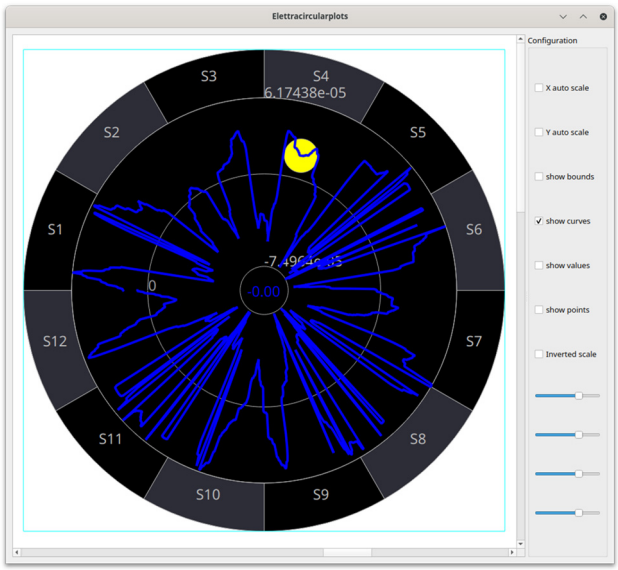

Figure 6: Circular (radar) plot with storage ring sections, selection point and configuration options.

Figure 7 shows another radar plot with a different style. Besides, details of the edit procedure are represented. A click on a point pertaining to an editable curve activates the operation. Dragging the orange circle either towards or off the center modifies the value of the point. The yellow circle moves exactly radially, irrespective of the off radius finger (or mouse) motion.

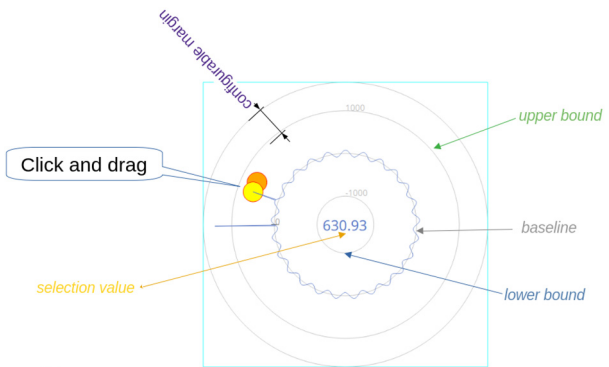

Figure 7: Radar Plot editing, baseline, bounds, selection.

Zooming

Graphics View is equipped with rotation and scaling capabilities on the items in the scene. Nevertheless, the user may need to zoom a given area of a chart without affecting the scale of the other items in the scene. A generic zoomer is included in the cumbia new generation controls library with the following characteristics:

- shall zoom everything (Graphic Items, Widgets) almost[1] automatically.
- Shall be activated by selecting a rectangular area on a widget or item.
- Shall be movable.
- Shall be stackable (supplying discrete unzoom).

Figure 8 displays a zoomed area on a radar plot.

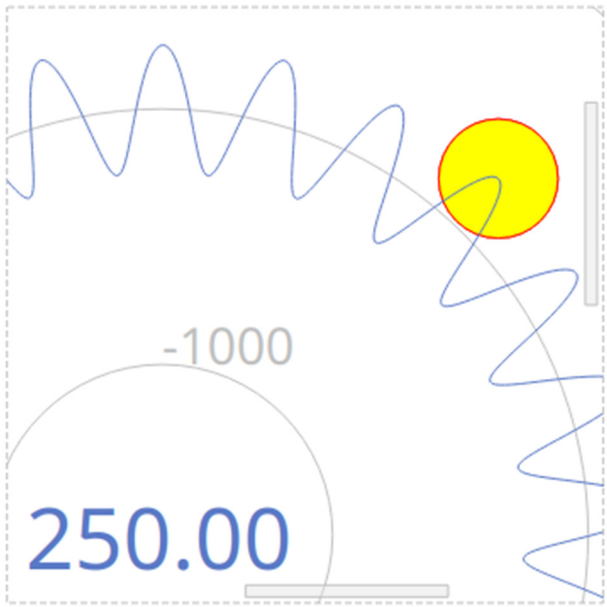

Figure 8: Radar plot zoomed. The area can be moved.

The object that can be installed on any traditional widget in order to enable close-ups. An image comes to mind and the situation is represented in Fig. 9.

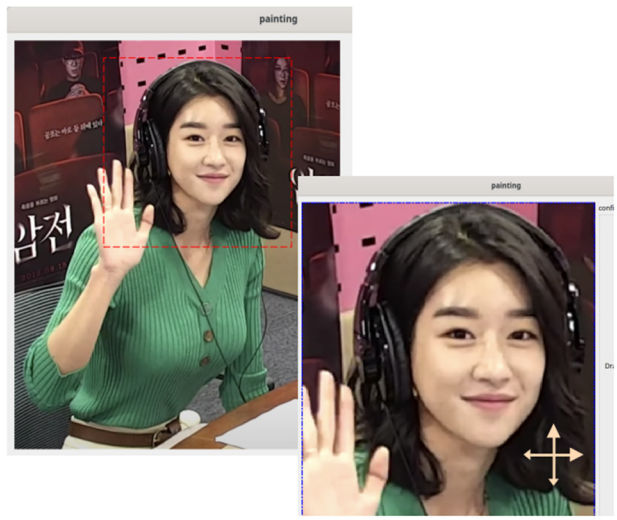

Figure 9: Zoomer on a traditional Qt Widget.

As aforementioned, the new generation of the cumbia controls library is at an early stage. The revision of existing components and the realisation of new ones shall be in step with the necessities and challenges posed by the ELETTRA 2.0 advancements and those continuously brought on by the dynamic nature of the FERMI Free Electron Laser.

ISBN: 978-3-95450-238-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2023-THPDP001

CONCLUSIONS

At ELETTRA the lack of a profitable solution for synoptic design, satisfactorily complete and flexible as well as performant, user-friendly and beautifully drawn has been a common perception over the years. The rising need for big amounts of data displayed at high rates through unconventional graphical representations that fit together the user expectations and the machine morphology led to the choice of the C++ programming language and the cumbia libraries, constantly crafted with great attention to performance and reliability. The Qt technology is the perfect companion when these are major constraints. Finally, the whole infrastructure is complemented by the cumbia HTTP engine, allowing the native execution of applications remotely.

Combined SVG and Graphics View technologies yield dynamic, user-friendly and composite applications where static items flank live and fully interactive custom drawn objects. Figure 10 shows yet another synoptic view with a plot drawn on top of an SVG view.

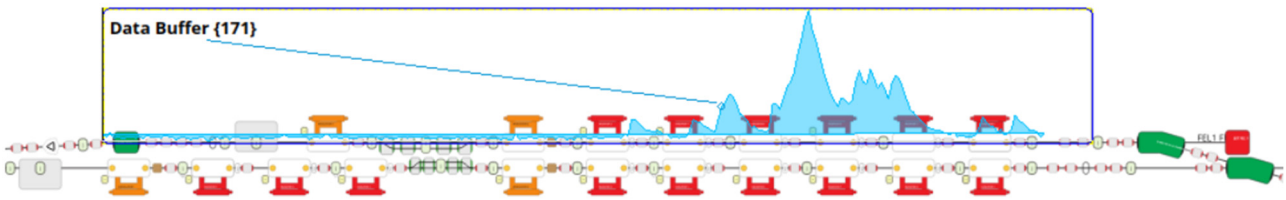

Figure 10: A combination of SVG items and a Plot Graphics Item.

REFERENCES

[1] G. Strangolino, "Cumbia: a new library for multi-threaded application design and implementation", in *Proc. ICALEPCS'17*, Barcelona, Spain, Oct. 2017, pp. 830-834. doi:10.18429/JACoW-ICALEPCS2017-TUPHA174

[2] Qt SVG, https://doc.qt.io/qt-6/qtsvg-index.html

[3] Qt Graphics View Framework, https://doc.qt.io/qt-6/graphicsview.html

[4] TANGO control system, https://www.tango-controls.org/

[5] EPICS control system, https://epics-controls.org, https://epics.anl.gov

[6] G. Strangolino and L. Zambon, "Canone 3: a new service and development framework for the web and platform independent applications", in *Proc. ICALEPCS'21*, Shanghai, China, Oct. 2021, pp. 1023-1028. doi:10.18429/JACoW-ICALEPCS2021-FRAR02

[7] Inkscape, https://inkscape.org/