

# A FLEXIBLE SYSTEM FOR END-USER DATA VISUALISATION, ANALYSIS PROTOTYPING AND EXPERIMENT LOGBOOK

Roberto Borghes\*, Valentina Chenda, Georgios Kourousias, Marco Lonza, Milan Prica, Martin Scarcia, Elettra-Sincrotrone Trieste S.C.p.A, Basovizza, Italy

## Abstract

Experimental facilities like synchrotrons and free electron lasers, often aim at well defined data workflows tightly integrated with their control systems. Still such facilities are also service providers to visiting scientists. The hosted researchers often have requirements different than those present in the established processes. The most evident needs are those for i) flexible experimental data visualisation, ii) rapid prototyping of analysis methods, and iii) electronic logbook services. This paper reports on the development of a software system, collectively referred to as DonkiTools, that aims at satisfying the aforementioned needs for the synchrotron Elettra and the free electron laser FERMI. The design strategy is outlined and includes topics regarding: dynamic data visualisation, Python scripting of analysis methods, integration with the TANGO distributed control system, electronic logbook with automated metadata reporting, usability, customization, and extensibility. Finally a use case presents a full deployment of the system, integrated with the FermiDAQ data collection system, in the free electron laser beamline EIS-TIMEX.

## INTRODUCTION

At the present moment there are 32 experimental stations operating at the synchrotron Elettra and the free electron laser (FEL) FERMI [1]. The data acquisition and control software of each end-station is almost unique, being highly focused on the specific technique applied. Nevertheless, for the user-oriented applications it is not unusual to meet the requirements that are common to a number of end-stations. For those, the most efficient solution would be a product with a high level of customization reusable in a multitude of situations. As a general idea, such an approach may be compared to a drill tool: it has been designed to perform a specific task but it needs to mount accessories for the alternative uses like drilling in metal, wood or concrete.

The Scientific Computing team had the chance to apply this strategy in different situations, building a collection of software tools called the DonkiTools. Any of the DonkiTools aims at becoming a standard in its field of application at Elettra and FERMI. The name of the suite recalls that of a donkey, a robust and versatile beast of burden, so highly appreciated in the Mediterranean region.

## DONKITOOLS OVERVIEW

Each application of the DonkiTools suite (Fig. 1) originated to solve a specific requirement, but has been designed from the start using a plug-in based architecture that makes it flexible and reusable. The plugin-based extensibility allows for constant evolution of the DonkiTools. It is possible to add a new functionality without the hazard of destabilizing others or the core behaviour.

As a programming language, Python was the natural choice as it is supported by TANGO and offers a high degree of dynamicity and a large set of scientific and visualization libraries.

PyQt4 [2] has been adopted as the GUI development framework. PyQt ensures that the DonkiTools will run on all platforms supported by the Qt, including Windows, MacOS/X and Linux. The plug-ins that can extend each application are basically PyQt Widgets included in the main panel at runtime.

In the next sections, we will present a short overview of three successful applications used by the staff and visiting researchers at Elettra and FERMI:

- DonkiLOG: an extensible electronic logbook for experiments with automated metadata reporting.
- DonkiCCD: a flexible visualizer of scientific images that may be customized and integrated to the end-station control system.
- DonkiPY: a tool for rapid prototyping of data collection and analysis scripts.



Figure 1: DonkiTools icons.

## DONKILOG

The first release of the DonkiLOG (Fig. 2) was developed for the needs of scientists working at the free electron laser FERMI. Each experimental shift at FERMI consists of several scan operations. Scans are often

\* roberto.borghes@elettra.eu

performed at the speed of 10Hz making the classical paper notebook completely unfit for the logging task. That created a pressing demand for a user-friendly logbook application capable of taking automated snapshots of the instrumentation status before and during each experimental scan. Initially we investigated the existing solutions but finally the decision was made to develop a brand new application that should be portable, customizable and re-usable.

Main features of the DonkiLOG include:

- WYSIWYG text editor
- HTML based, compatible with any web browser
- export to PDF
- screen-shot insertion
- integrated Data server for uploading images and text from remote machines
- automatic file saving
- plug-in based extensibility

Currently the logbook has been installed and integrated with control and acquisition systems on six different end-stations running both Linux and Windows. The plug-in based extensibility allows the customization of each installation with the addition of extra buttons to the user interface in order to grab data and images from beamline-specific external devices.

Another appreciated feature is the embedded data server that allows to push data from the remote clients: the data acquisition system can log directly into the user logbook or another scientists can upload screen-shots and annotations from another workstation through a remote text editor.

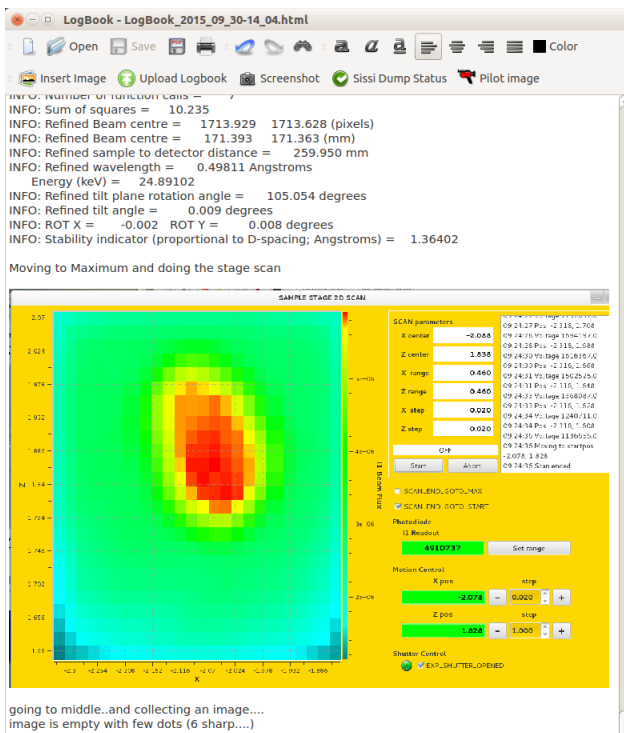


Figure 2: DonkiLOG electronic logbook.

## DONKICCD

There is a wide variety of image visualization tools installed at Elettra and FERMI endstations but during the data acquisition phase, the basic needs of all the labs are more or less the same. Based on this observation, we started the development of an application capable of visualizing most of the scientific image formats present at Elettra and FERMI creating a new standard visualizer, powerful and flexible enough to fit the needs of any new imaging end-station.

DonkiCCD (Fig. 3) has been developed using the guiqwt framework [3], a 2D data-plotting library based on PyQwt and on the scientific modules NumPy and SciPy. In order to support different file formats produced by 2D X-ray detectors, the FabIO [4] python library has been used.

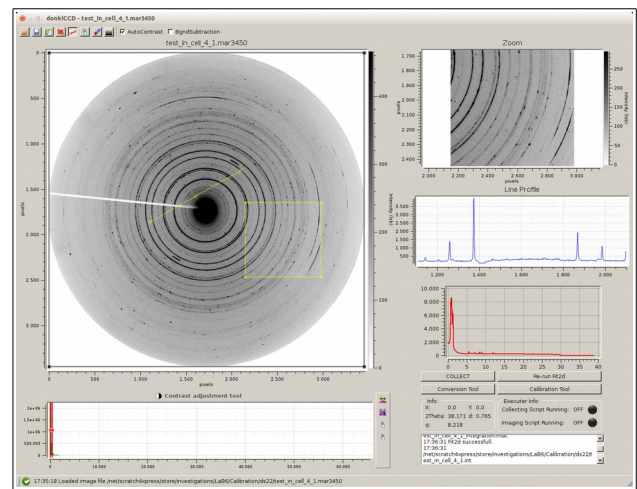


Figure 3: DonkiCCD scientific image viewer.

Main features of the DonkiCCD application include:

- support for common scientific image formats: CBF, TIFF, MAR345,...
- region of interest (ROI) zooming
- line profile tool
- automatic and manual contrast adjustment
- background subtraction
- plug-in based extensibility

Currently, the DonkiCCD is used on seven end-stations for the visualization and near real-time data analysis of images from different scientific instruments: Dectris Pilatus, mar345 imaging plate, Princeton Instruments CCD, Basler and Andor cameras. Through the plug-in architecture each installation can interact with the data acquisition system in order to automatically update the visualized images. Furthermore, upon the requests from the scientists, new data processing tools have been added:

averaging display, projections plotting, ROI intensity trend, etc...

### DONKIPIY

Experimental data acquisition and analysis systems are usually based on well defined data workflows, but sometimes it is necessary to investigate a new experimental technique or a data analysis pipeline. In such cases, a rapid prototyping tool is needed that permits scientists to develop new methods in a fast and simple manner.

Once again, Python turned out as the optimal choice for the scripting language. Its immediacy and readability allows any scientist or student even with basic programming skills to learn the essentials quickly and write simple scripts with a limited effort. This led to the design of a tool for rapid prototyping based on Python and a set of customizable libraries with a user friendly help system.

DonkiPY (Fig. 4) is an extension module of Spyder [5], the well known Python integrated development environment that features advanced editing, interactive testing, debugging and introspection tools. The DonkiPY extension allows the usage of custom libraries within the Spyder environment. Furthermore, the IDE is enriched with an interactive help system that visualizes documentation and examples relative to the supplementary libraries. These are standard Python scripts written with a specific header used to fill the help system.

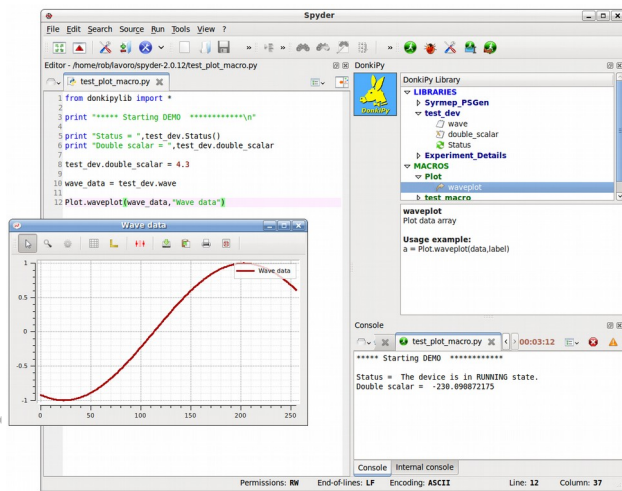


Figure 4: DonkiPY rapid prototyping of analysis methods.

This tool has an extremely wide application field. It has been developed in order to let scientists apply their ideas without a detailed knowledge of the control system or the mathematical methods involved. The DonkiPY custom libraries, also called *macros*, are developed and tested by the specialized staff and afterwards deployed and shared on the experimental end-stations.

Currently DonkiPY is being successfully used at Elettra and FERMI for two main purposes: fast prototyping of new data collection sequences or implementation of new data analysis pipelines. In the first case the added macros simplify the interaction with the instrumentation, in the latter case they are more focused on mathematical algorithms and plotting utilities.

### EIS-TIMEX USE CASE

EIS-TIMEX end-station [6] has been designed to exploit the high intensity, energy domain and time structure of the FERMI Free Electron Laser to probe fundamental properties of dense matter under extreme thermodynamic conditions. The experimental chamber and the sample environment configurations have been kept fairly flexible in order to accommodate various possible configurations for single-shot experiments, including simple EUV and soft x-ray absorption/reflection and pump-probe experiments where the probe can be either an external laser or a FEL pulse.

The control and data acquisition system, as on all the FERMI end-stations, is based on the TANGO framework [7]. At the end-station three workstations are used: two are dedicated to instrumentation control and data acquisition while the third one is used for data visualization and analysis (Fig. 5).

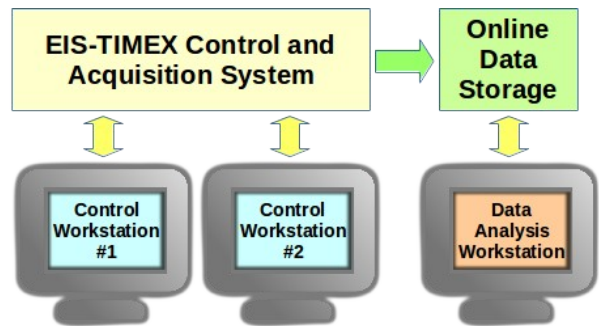


Figure 5: FERMI end-station EIS-TIMEX.

All the DonkiTools described in this paper are fundamental resources for the scientists during the experimental period at EIS-TIMEX. As an experimental shift may last an entire week, 24 hours a day, it is extremely important to ensure the support of easy to use applications, capable of maximizing the automation of the tasks.

### Working with Image Detectors

Different experimental techniques have been applied at EIS-TIMEX. Some of them require image detectors like Basler GigE cameras or Princeton Instruments CCD. For those, we have successfully applied DonkiCCD to visualize and perform fast data analysis on acquired images (Fig. 6). The basic functionalities of the

DonkiCCD have been extended according to the needs of scientists:

- multiple regions of interest support
- running average plotting over multiple images
- image intensity axis projections
- data thresholding
- integrated instrumentation control panel

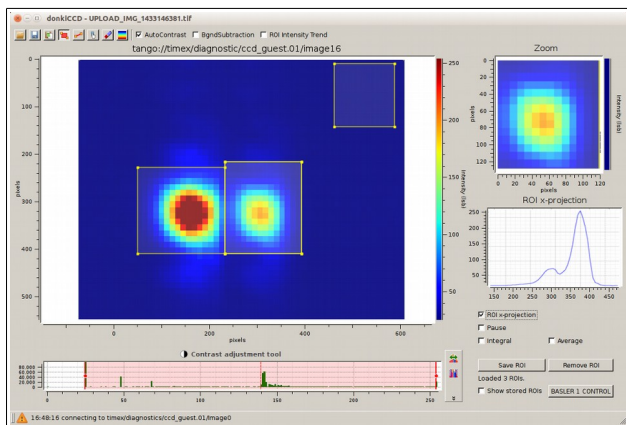


Figure 6: DonkiCCD is used to visualize Basler cameras and Princeton Instruments CCD.

### Experiment Log

Experimental activities are logged using the electronic logbook DonkiLOG integrated with the TANGO data acquisition system in order to automatically log additional metadata about FERMI status and the experimental scan in progress (Fig. 7). Furthermore, using a remote client application, it is possible to append data analysis results and screen-shots from other workstations. At the end of the experimental shift the logbook is exported as PDF file and saved with the scientific data on the online storage.

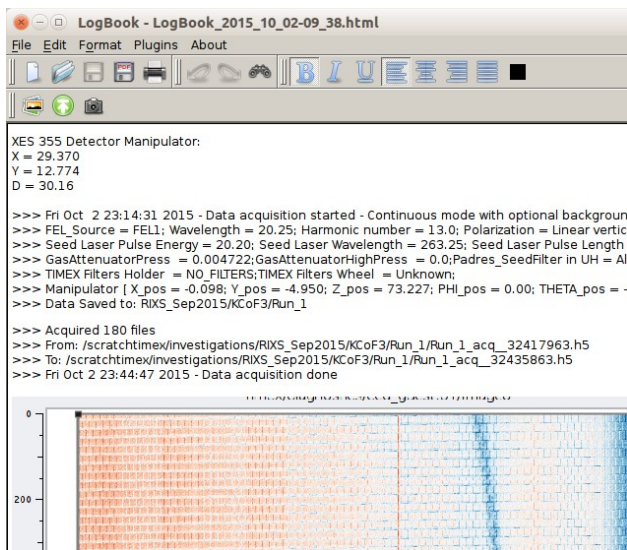


Figure 7: DonkiLOG accepts log messages from data acquisition system.

### Data Analysis

An experimental data collection may produce hundreds of HDF5 files that need to be analysed, correlated and plotted. Scientists can access and analyse data on the central storage using a dedicated Linux workstation with the DonkiPY tool installed that permits the rapid prototyping of analysis methods. A set of custom macros developed ad-hoc simplifies the procedures of correlating data coming from multiple HDF5 archives. Users may immediately add results and charts to the user logbook using the DonkiLOG remote client application.

### CONCLUSIONS

The DonkiTools approach has been repeatedly put to test at Elettra and FERMI end-stations and was proven to be efficient and successful. DonkiTools applications are flexible, portable over different platforms and user friendly. The strength of this suite resides in its design choices: a modern and powerful language like Python at the core, a highly portable graphics library like Qt for the GUI and a plug-in based architecture that permits a high degree of customization.

### ACKNOWLEDGEMENT

The authors, members of the Scientific Computing team of Elettra, thank the IT Group and the beamline Groups of Elettra Sincrotrone Trieste. Moreover they acknowledge the importance of advanced technologies and open source software like that of the distributed control system TANGO.

### REFERENCES

- [1] Elettra and FERMI Light Sources website: <http://www.elettra.eu/lightsources/index.html>
- [2] PyQt application framework website: <https://riverbankcomputing.com/software/pyqt/intro>
- [3] GuiQwt Python library website: <https://pythonhosted.org/guiqwt>
- [4] Fabio Python I/O library for images website: <https://pypi.python.org/pypi/fabio>
- [5] Spyder the Scientific PYTHON Development website: <https://pythonhosted.org/spyder/>
- [6] A. Di Cicco et al., “Probing phase transitions under extreme conditions by ultrafast techniques: Advances at the FERMI free-electron-laser facility”, Journal of Non-Crystalline Solids, Vol. 357 – 14, (2011)
- [7] R. Borghes et al., “A Common Software Framework for FEL Data Acquisition and Experiment Management at FERMI”, ICALEPCS2013, San Francisco, USA, (2013)