

CS-STUDIO TRAINING AT REA3*

T. Summers[†], D. B. Crisp, National Superconducting Cyclotron Laboratory, Michigan State University, 48824 East Lansing, USA

A. C. C. Villari, Facility for Rare Isotope Beams or Affiliation, Michigan State University, 48823 East Lansing, USA

Abstract

In the past year, Control System Studio (CS-Studio) [1] has become the predominant graphical user interface tool at ReA3, the 3 MeV/u rare isotope beam (RIB) Reaccelerator at Michigan State University's National Superconducting Cyclotron Laboratory (NSCL) [2].

CS-Studio is a set of control system interface tools that include operator interfaces, history plots, an alarm handler, save/restore, scanning, and more. Becoming an effective user of these tools takes considerable time and training.

This contribution will describe the challenges and strategies for training operators on the general use of the CS-Studio tools. It will describe the use of a simulated user interface environment for training operators at any time without affecting the operating facility.

BACKGROUND AND MOTIVATION

Commissioned in 2015, ReA3 re-accelerates RIBs produced by in-flight by projectile fragmentation. The RIBs are provided by the Coupled Cyclotron Facility (CCF) and will be provided by the Facility for Rare Isotope Beams (FRIB) in the future. ReA3 and FRIB use CS-Studio as the primary graphical user interface tool to the EPICS [3] control system. Details on initial implementation of CS-Studio at ReA3 can be found in [4].

Responsibility for operating ReA3 is transitioning from system experts to the CCF operators, who will eventually merge into the FRIB operations department. CCF operators are accustomed to control system tools such as MEDM, LabVIEW, and locally developed Qt and Tcl/Tk applications. Several years ago, they adopted CS-Studio's Data Browser as a plotting tool but have only recently started to develop and use CS-Studio interfaces. However, as the cyclotrons will be replaced by FRIB, there is no intention of replacing the bulk of its existing interfaces.

As the operators became more involved in the operation of ReA3 it was apparent that lack of familiarity with CS-Studio was a barrier to operating it effectively. Initial efforts at training were intermittent and often during times of machine setup or troubleshooting.

Both ReA3 and the CCF are currently operating facilities supporting a user community. This poses a challenge for training on control system tools since the equipment is generally either running for an experiment or shut down for maintenance. A solution to this lack of access is to create a set of simulated devices and tools which can be used at any time without affecting operating equipment.

* This material is based upon work supported by the National Science Foundation under Grant No. PHY-1565546.

[†] summers@nsl.msu.edu

TRAINING WORKSHOPS

Half-day training workshops were held during a recent maintenance period when the entire operator staff was available. Using simulated content, they had the opportunity to explore interfaces and gain experience with the plotting tools, alarm handler, save and restore program, and scan interface.

CS-Studio is a complex set of tools, and a few hours is not enough time for an operator to become an expert with them all; follow-up sessions focusing on specific aspects of the program will be beneficial. The simulated tools used, together with reference slides and a task walkthrough, will remain accessible for self-guided learning and exploration.

SIMULATION

The Accelerator Controls Simulation (*acsim*) project consists of an EPICS input-output controller (IOC) and related CS-Studio content. This project is used for CS-Studio training as well as developing and testing display styles, high-level summaries and commands, and scripts.

EPICS IOC

At the core of the simulation is an EPICS IOC (R3.15). Standard record types are used to create device process variable (PV) templates for a source, power supplies, RF cavities, vacuum gauges and gate valves, x-ray monitors and faraday cups. The records are configured to have realistic noise, ramping time, alarms, and interlocks.

Convenience PVs, essential to creating efficient CS-Studio interfaces, were developed using a combination of standard records, C++, and Python programs. Examples of these PVs include system summaries, waveforms of setpoints and readbacks with statistics, grouped commands, smart alarms, and the ability to revert setpoint changes - all of which were deployed on ReA3 after being prototyped and tested on the simulation.

Multiple copies of the IOC's record and application templates can be loaded to run independent instances of the simulation. The IOC is configured to provide 'IOCstats', auto-saving, channel access security, population of channel finder, and automated archiving of tagged records.

CS-Studio Content

To interact with the simulated PVs, a set of CS-Studio operator interfaces (OPIs) were created, following a hierarchical approach. At the top level is the launch page shown in Fig. 1, which is split into an Accelerator section for high-level or global interfaces, and a Subsystem section for device-focused interfaces. This page demonstrates the banner layout used on all pages with title, logo with link to

home page, timestamp, and help menu. The help menus contain the ‘Color and Symbol Legend’ shown in Fig. 2, as well as any help content specific to the page.

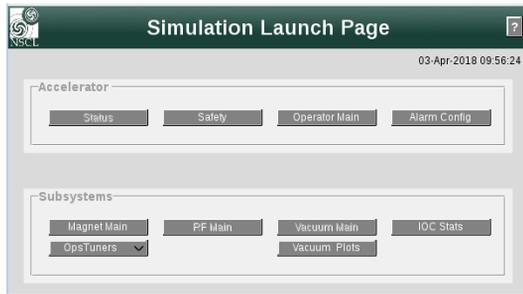


Figure 1: Simulation launch page.

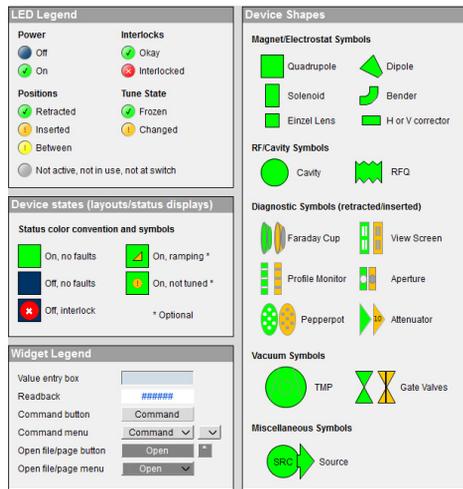


Figure 2: Color and symbol legend help page.

The main operator page, shown in Fig. 3, consolidates commonly used control and status indicators into one graphical interface. The main feature is a cartoon layout of the accelerator with magnet devices color-coded to indicate operating status. Labels can be shown by toggling ‘Show Labels’. Device interlocks are indicated by overlaying a red icon with an ‘x’ symbol. Vacuum readings with alarm-sensitive borders are displayed next to the gauge widgets. Vacuum valve controls are placed next to the status indicators. Faraday cups can be inserted or retracted and display current readings, which are also displayed in a history graph. The source controls are available as well as an ‘Ilk’ popup window which will indicate the status of all devices that can interlock it.

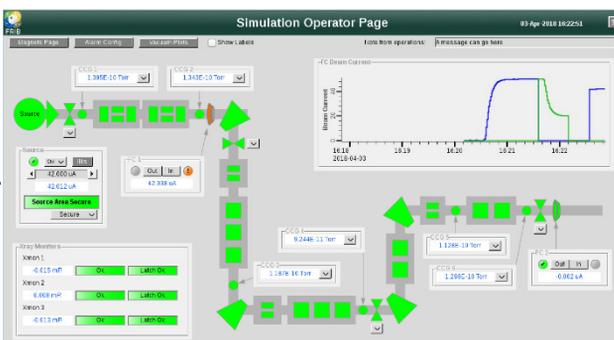


Figure 3: Main Operator page.

A main accelerator status page, not shown, has a layout visually similar to the main operator page but excludes device control, and is optimized for a large TV display. This page also provides a section of device summaries in text form for easy scanning.

There are two pages for safety status. The first is the detailed layout schematic seen in Fig. 4 showing device summaries and their physical location, intended as a status display to supplement the main status page. The second page, seen in Fig. 5, shows the devices in a row layout for easy scanning and providing details and resets for each device. This page also features an embedded plot showing three dose rate PVs, one of which is archived at full resolution and one archived with a deadband.



Figure 4: Safety layout page.



Figure 5: Safety device list page with history plot.

Power supplies are often the devices operators interact with the most, so there are several interfaces developed for them. First, the Magnet Main page shown in Fig. 6 has a section at the top with the summary status, control and in-use indicator for each family. The in-use indicator can be used for unused beamlines to mask alarms and grey-out the status. Below the summary section is a tabbed container with control rows for each power supply with power and setpoint control and readback, link to details, a hold/revert function for temporarily storing a setpoint, and the option to undo the last setpoint change.

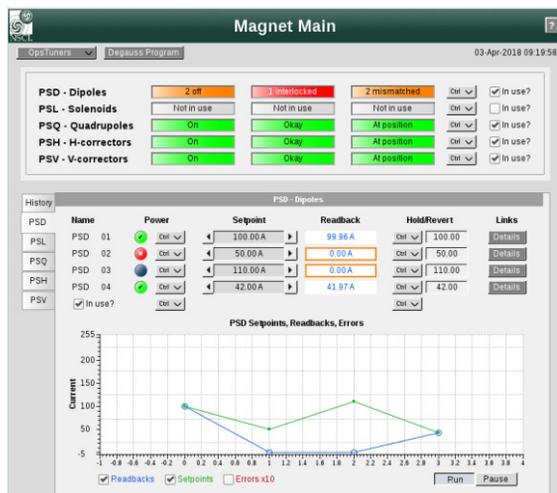


Figure 6: Magnet Main page.

The Details links open engineering pages with all available details, settings, and alarm configuration for the power supply, as well as a history plot

A new type of application ‘opstuner’ has been developed to mimic one of the previous control system tools. The opstuner, shown in Fig. 7, is easy to configure and provides a simple row layout for devices. Power and setpoint controls and readbacks are displayed in each row and grouped at the bottom. The application allows for grouped adjustments and keyboard shortcuts for adjusting the step sizes.



Figure 7: Magnet opstuner page.

The magnet Degaussing page, shown in Fig. 8, gives an example of interacting with a Python application through a CS-S interface. The Python application orchestrates the ramps of solenoid and corrector pairs.

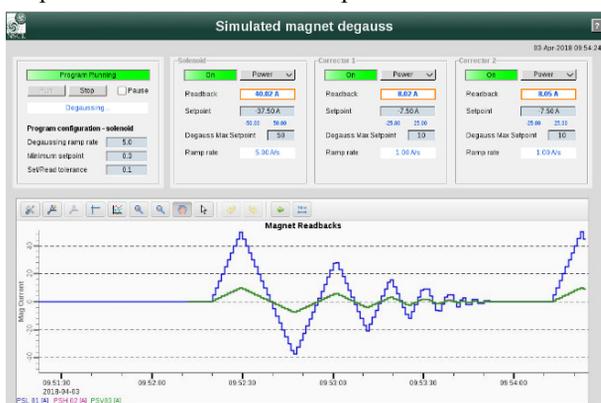


Figure 8: Interface for Python-based degaussing program.

Supporting Applications

An alarm configuration was created to include power supply interlocks and setpoint/readback mismatch errors. These were configured to not alarm if the power supply

section was marked ‘not in-use’. The interlock will only alarm if the power supply is commanded to be on, and the mismatch condition will only alarm if it is present for longer than 30 seconds. An email server is available to test automatic email notifications.

A save/restore configuration was created for power supply setpoints. It also includes non-restorable vacuum gauge readings. The interface is configured to use the version control software Git for file storage.

Finally, a simple example was created to demonstrate how to setup and submit a scan to the scan server, and to view and save the results from the interface.

FUTURE DIRECTION

Future development of the *acsim* project will include the addition of more device types and increasing the realism of these devices. The goal is to create a set of interfaces reasonably similar to the real accelerator controls. Not only does this simplify operator training, it provides a realistic environment to prototype new interfaces, perform usability tests, and make improvements to layout and workflow before deploying anything to the real machine.

A new project is under development to use the *acsim* tools to teach operators to develop CS-Studio content and applications. These will serve as a set of templates for making new pages, helping to ensure a consistent experience across operator interfaces.

A long-term goal of the *acsim* project is to explore the possibility of including physics parameters or incorporating a lattice code in order to use the program to simulate beam transport.

CONCLUSION

CS-Studio is a complex tool that includes operator interfaces, history plots, an alarm handler, save and restore, scanning and more. Becoming an effective user of all these tools can take considerable time and training, which can be challenging in an operating facility. An effective solution is to create a set of simulated EPICS devices and related CS-Studio content to use for training.

The additional benefit of developing these simulated tools is that they provide an environment to test new applications and display styles, as well as acting as a style guide and example repository for all content developers.

REFERENCES

- [1] CS-Studio, <http://controlsystemstudio.org>
- [2] A.C.C. Villari *et al.*, “Commissioning and First Accelerated Beams in the Reaccelerator (Rea3) of the National Superconducting Cyclotron Laboratory, MSU”, in *Proc. 7th Int. Particle Accelerator Conf. (IPAC'16)*, Busan, Korea, May 2016, paper TUPMR024, pp. 1287-1290.
- [3] EPICS, <http://epics.anl.gov>
- [4] T. Summers *et al.*, “Implementing CS-Studio at ReA3”, in *Proc. 16th Int. Conf. on Accelerator and Large Experimental Control Systems. (ICALPCS'17)*, Barcelona, Spain, Oct. 2017, pp. 1887-1889, doi:10.18429/JACoW-ICALPCS2017-THPHA190