NSLS II COMMISSIONING TOOLS*

G.M. Wang#, T. Shaftan, B. Bacha, A. Blednykh, E. Blum, W.X. Cheng, J. Choi, M. Davidsaver, J. De Long, L. Dalesio, R. Fliller, M. Fulkerson, G. George, W. Guo, K. Ha, H. Hseuh, Y. Hu, W. Louie, M. Maggipinto, J. Mead, D. Padrazo, G. Shen, K. Shroff, O. Singh, K. Vetter; Y. Tian, H. Xu, L. Yang, X. Yang, F. Willeke. Photon Sciences, BNL, Upton, NY, 11973, U.S.A

V. Smalyuk, S. Karnaev, S. Sinyatkin, S. Gurov, A. Erokhin, E. Simonov, A. Derbenev, P. Cheblakov, R. Kadyrov BINP, SB, RUS

Abstract

NSLS-II is a state-of-the-art third-generation light source under construction at BNL. As many facilities worldwide, NSLS II uses the EPICS control system to monitor and control all accelerator hardware. CSS is used for simple tasks such as monitoring, display, setting of PVs and browsing the historical data. For more complex accelerator physics applications, a collection of scripts are mainly written in Python. The controls group developed the services, such as channel finder, machine snapshot, data archiving, twiss server, unit conversion etc. This paper will present the tools that we have been using for commissioning.

INTRODUCTION

The NSLS-II [1] is a state of the art 3 GeV synchrotron light source under construction at Brookhaven National Laboratory. It consists of a 200 MeV linac, a booster ring accelerating beam from 200 MeV to 3 GeV, a 3 GeV Storage ring and transport lines in between them.

As many other facilities [2,3], NSLS II chose the Experimental Physics and Industrial Control System (EPICS) as its control system to monitor and control the accelerator hardware. It interfaces to the accelerator instruments and devices (such as Power supply, digitizer, and motor) with IOC (Input Output Controllers). Channel Access is used as the interface to the machine process variables (PVs).

The typical control applications include two types, simple monitor/control device and complex accelerator physics application for studies and machine optimization (measurements, data analyses and applying correction). Mainly, we use cothread to access the process variables (PVs) in Python script for complex accelerator physics applications and the CSS (BOY) [4] panel for simple tasks such as monitoring, displaying and setting of the machine Process Variables (PVs). Besides, MATLAB is also available for programming and EDM panels are used for Linac control system from the linac vendor Research Instruments, GmbH. All source codes including panel configurations are controlled and managed by mercurial version in the control system.

To keep the beam commissioning time efficient, various tools were developed for data displaying, collecting and processing. Besides machine status and physics applications, we also developed the daily used operation tools, such as save/compare/restore tool, data archive, alarm and warning monitor, elog, eticket, channel finder service and applications.

Prior the beam commissioning, we have two stages test with above tools, subsystem integration testing (IT: the hardware system level function verification) and machine extended integration test (EIT: mimic the beam commissioning and tune the machine). These verifications require time and look boring and redundant, but it is necessary as it may take weeks of commissioning time to fix different technical problem.

MACHINE STATUS APPLICATION

All of the monitor and control panels are created with software tools in Control System Studio (CSS), which is an Eclipse-based collection of tools to monitor and operate large scale control systems. The interface allows the operators to edit the desired ramp/soak profiles and provides the option to use predefined recipes.

The NSLS-II operation panels [5] include the user panels and the expert panels. They display the device status and parameters (setpoint and readback) and show the device performance. A user panel shows only information that user is required, such as the magnet current setpoint and readback. The expert panel shows all the information that expert can fully control the device or diagnose the device, such as power supply operation mode and magnet temperature.

HIGH LEVEL APPLICATION

Besides the devices status and control, we also develop high level applications to realize the accelerator physics application tools. It is to translate the raw/processed data from PV to the “physics” parameters, such as from the beam size reading from flags for beam emittance or beam energy measurement. The code was programmed with Python. Although we programmed the high level applications with Python, some of their GUIs (depending on operation needs) are implemented in the CSS, as part of operator panels, so that the operators have consistent live information and interface environment. The user can set parameters and monitor the progress in GUI as regular device PV set and monitor. This feature is realized through soft IOCs.

* This manuscript has been authored by Brookhaven Science Associates, LLC under Contract No. DE-AC02-98CH10886 with the U.S. Department of Energy.

#gwang@bnl.gov

A screenshot of beam energy and energy spread measurement with flag VF2 is shown in Figure 1. It is in continuous mode, which means whenever the beam image updates, the beam energy and energy spread is updated correspondingly. It shows the beam image, horizontal beam profile, beam energy, beam energy spread, and the charge ratio in every 0.5% energy deviation.

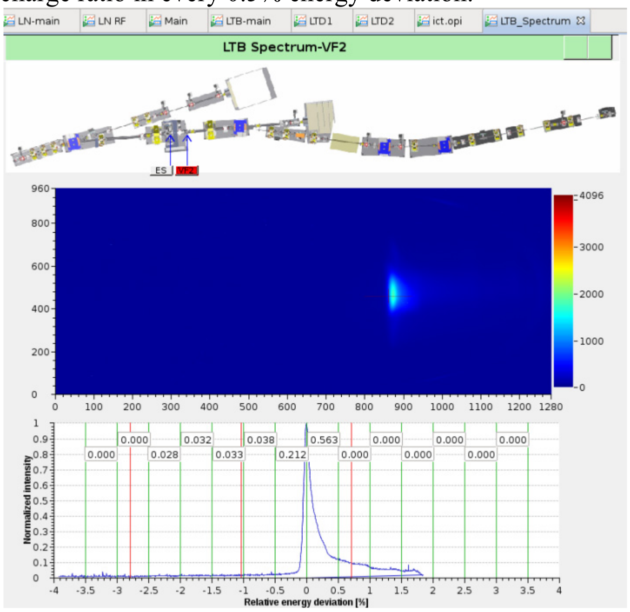

Figure 1: Screenshot of beam energy and energy spread measurement with flag VF2 in LTB transport line.

COMMISSIONING TOOLS

MASAR

The MASAR (MAchine Snapshot, Archiving, and Retrieve) [6] is an important tool used during the commissioning, which is developed by the NSLS-II Controls group for machine save/compare/restore function. MASAR is an EPICS V4 service, and it is a server-client based tool-set. The server and client are implemented in C++ and Python. The server takes machine snapshots, archives data in a relational database, and retrieves data from the database. The client provides a GUI that retrieves snapshot data, compares the data with the live machine, compare multiple snapshots, and restores the machine with given snapshot data.

The user creates a list of channels, which can be any record data type, such as scalar values, arrays or strings, that are important for the system in a configuration file and, then, MASAR saves the current configuration. To take a snapshot, the client specifies the name of the snapshot configuration and a name for this snapshot event. When the server receives a command to take a machine snapshot, the server retrieves the list of channel names in the configuration from the database. It then gets the current value of all the channels from IOC, and saves the data as a snapshot event into the database.

Channel Finder Service and Applications

The High level applications tend to prefer a hierarchical view of the control system where they can group channels by type (such as BPM, horizontal corrector) or location. These applications require retrieving a large set of channels' value or alarm state. Channel finder applications have simplified user interaction by requiring the user to providing only the channels' criteria that they are interested in, such as element type BPM or element Name ICT1, instead of the complete set of channels name.

User provides a list of channel names (PVs) with given properties and tags and channel finder service (CFS) [7] stores these information for different applications. The Properties are keyword-value pair data, where the keywords are predefined, such as element name with value P1, element type with value BPM, s Position with value P1 0.5 et. al, whereas the tags are plain strings, such as aphla.sys.LTB, aphla.sys.SR. The channel finder applications use CFS plugin, to query the service for a group of channels based on the set of criteria, which is based on one or any combination of channel name, properties and tags. The channel finder applications client are supported by CSS (such as channel viewer to show a group of PVs with their name, properties and tags, Channel line plot viewer to plot a group of PVs live data, waterfall plot to show time plot of all queried channels) or python (APHLA: accelerator physics high level application).

Channel Archiver

An EPICS Channel Archiver is installed to record the values of many interesting PVs over time (such as ICT charge, power supply readback, vacuum pressure, which are predefined by PVs' owner). This data is then available for online queries, for plotting with the CSS Data browser, and may be exported in several formats (such as spreadsheet, Matlab, h5) for analysis.

Figure 2 shows the historical view of the charge reading after linac from one ICT and two Faraday Cups with data browser.

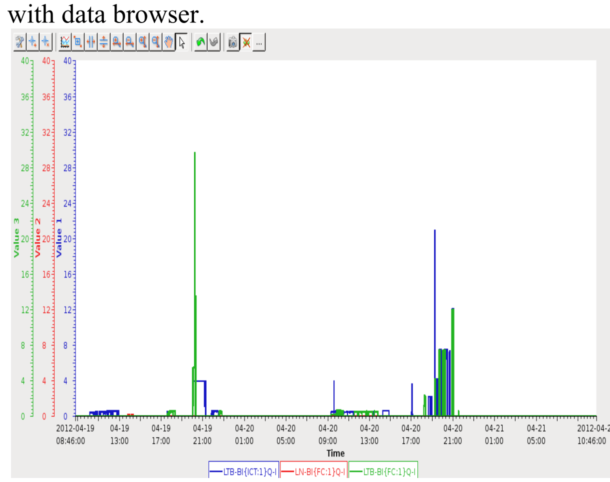

Figure 2: History view of charge reading.

ISBN 978-3-95450-138-0

SUB-SYSTEM INTEGRATION TEST

The purpose of the sub-system integration test [8] is to verify the device function along with utilities, timing system and control system in control room environment. Meanwhile, we do the diagnostics system calibration and develop more functions to improve the control system features for users. The subsystem testing include all diagnostics (such as ICT, FCT, FC, Flags, BPMs) devices, power supplies and RF system. We developed the test procedure, which includes the system interface block diagram, the system physics parameters and ranges, the control parameters and procedures on mechanical/controls testing, calibration and check points. During the process, we use external "known" signal to simulate the real beam in various situations, such as 1 Hz/2 Hz/ stacking mode with low/high bunch charge at single bunch mode/long pulse mode.

Most of the technical issues are solved during the integration test. But this test cannot fix some possible problems, such as magnet polarity error, which could only be revealed by beam.

For example, during the BPM system integration testing [9], a number of hardware issues has been found out and fixed (such as cable loose connection on a patch panel). Some BPMs four buttons' signal shows different behaviour and noisy frequency up to a few kHz were observed. This is solved by modifying the oscillator and installing a microwave absorber. The BPM position noise level is now improved down to 1μm. During the BR flags test, a parasitic background was observed on the screen images, which will also affect the beam image. To fix this problem, A set of the absorbers has been produced by BINP and installed into the flag tubes. During the ICT test, the beam signal source is 'replaced' by a pulse generator. We found LTB-ICT2 connection issues and four ICTs exhibit the readback difference, which is larger than 5% due to calibration. After repeating the calibration, the four ICTs readback difference came down to be less than 1% and the charge readback noise reduced below 10 pC.

EXTENDED INTEGRATION TEST

Prior to the Booster (BR) beam commissioning, we carried out the beam commissioning with simulated beam signals, called extended integrated testing (EIT) to prepare for the booster ring commissioning by operating the whole system as during the actual beam commissioning. The beam signals are simulated and generated by a computer program, ELEGANT. Then they are transported by the same data channel as the real beam signal would travel.

We developed shift schedule, daily shift plans and shift summaries, closely corresponding to the actual beam commissioning. The EIT activities consist of 4 hour shifts per day with shift team from BNL and BINP. The short and long term shift plans are based on real beam commissioning steps. The whole test lasts 10 weeks.

During this process, we tested and optimized all the operation screens, the high level applications and upgraded the operation tools. With readbacks from the power supply and RF systems, commissioning activities such as injection, first turn steering, achieving circulating beam, RF capture, closed orbit, acceleration ramp optimization and extraction can be set up and optimized. This proved itself to be an efficient way of debugging the controls- and the correction systems, as the applications bug can be quickly revealed with the known 'beam' signal. Many small issues and bugs, which would have us slowed down during the commissioning, have already been discovered and fixed during the test. For example, the large dipole BD1 IGBT fail caused the PS broken, which takes power supply group ~1 month to find and fix the problems. The feedforward function in quads is implemented as the error is larger than specification. The MASAR function is upgraded to accommodate the waveform channels. The ramp manager application is upgraded so that RF control with sequence and limitation is included and RF voltage waveform setting can be easily correlated with PS setpoint.

Moreover, the testing is carried out from the control room in a way which is almost identical to the commissioning with real beam. The operations procedures, the shift routines, the use of the logbook and much more were practiced and improved during EIT. For example, the online shift summary service was developed for easy search.

ACKNOWLEDGEMENTS

All of this work would not have been possible without constant support of all the NSLS II teams and the close collaboration with the BINP team.

REFERENCES

[1] F.J. Willeke, Status of the NSLS-II Project, TUOBS3, PAC2011.
[2] Emery, L., Commissioning Software Tools at the Advanced Photon Source, p 2238 - 2240, PAC1995.
[3] R. Bartolini et al., High level software for DIAMOND commissioning and operation, THPCH112, EPAC2006.
[4] http://cs-studio.sourceforge.net/
[5] G.M. Wang et al., Preparation for NSLS II linac to booster transport line commissioning, MOPPR094, IPAC12.
[6] G.B Shen et. al, "NSLS II Middlelayer Services", MOPPC155, ICALEPCS 13.
[7] K. Shroff, et al, Multi Channel Applications for Control System Studio(CSS), MOPMN015, ICALEPCS2011.
[8] G.M. Wang et al., NSLS II injector integration test, THPEA063, IPAC2013.
[9] V. Smaluk et al., Status of beam diagnostics for NSLS II Booster, MOPC02, IBIC2013.