

Hall C EPICS Controls and Monitoring System Featuring Control System Studio

Peter Bonneau, Mary Ann Antonioli, Pablo Campero, Brian Eng, Amanda Hoebel, George Jacobs, Mindy Leffel, Tyler Lemon, Marc McMullen, and Amrit Yegneswaran

Physics Division, Thomas Jefferson National Accelerator Facility, Newport News, VA 23606

February 12, 2019

This paper summarizes the development and implementation an EPICS controls and monitoring system for Hall C’s HMS and SHMS, featuring Control System Studio (CSS). Integration of CSS applications would standardize control systems across the Physics Division.

CSS is an open-source, integrated development platform of software applications to monitor and operate large-scale EPICS control systems. Earlier applications for EPICS control systems were developed independent from each other. Each application was different and it was often difficult to exchange data between them. To resolve these issues, CSS was developed by a collaboration of US national labs (Oak Ridge, Brookhaven, Lawrence Livermore), Michigan State University, and the Deutsches Elektronen-Synchrotron facility.

Table I summarizes the CSS programs to be implemented in the Hall C slow controls system. In Fig. 1, the slow controls architecture diagram, the blue boxes are the CSS software applications to be implemented.

Software	Function
CSS-BOY (Best OPI, Yet)	operator interface development and runtime environment
CSS-BEAST (Best Ever Alarm System Toolkit)	distributed alarm system with CSS interface
CSS-WebOPI	Read-only CSS screens viewed in web browser for remote monitoring

TABLE I. Proposed control system studio software for Hall C.

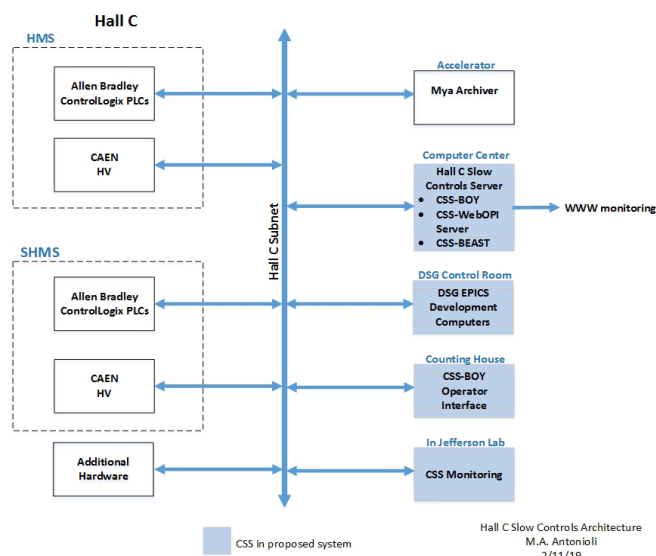


FIG. 1. Proposed Hall C CSS - EPICS slow controls architecture.

CSS-BOY is an operator interface (OPI), development, and runtime environment. This is the main application users run to display and control EPICS systems.

In the development mode, tools are available to create control screens. CSS-BOY allows easy design of control screens. With integrated scripting and rules, the designer can create dynamic screens with a minimum of effort.

To start executing the OPI, the operator switches the program to runtime mode. The program connects to the network and the EPICS process variables (PVs) are displayed or controlled on the OPI screen.

The runtime mode looks and functions like a web browser. If a PV is not found on the network, the program indicates a disconnected state by displaying a pink border surrounding the PV. Figure 2 shows an operational Hall C CSS-Boy OPI for the SHMS horizontal bender magnet.

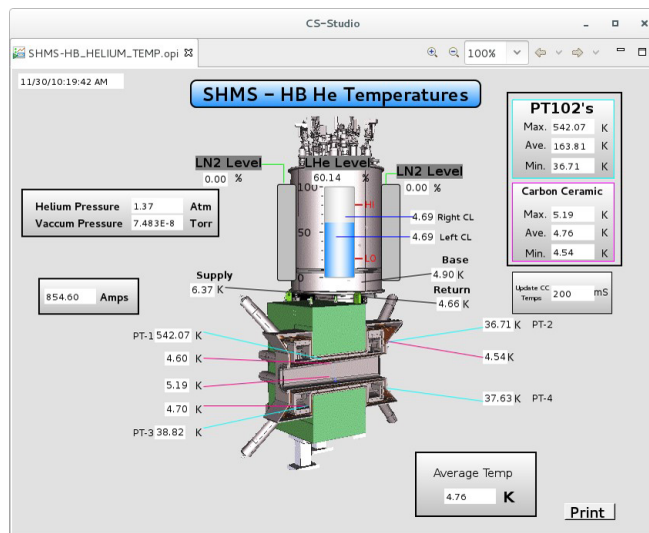


FIG. 2. Working CSS-BOY OPI for Hall C HB magnet.

CSS-BEAST is an integrated distributed alarm system for EPICS control systems. Table 2 highlights the major functions of BEAST.

The BEAST user interface, Fig. 3, is used to acknowledge alarms and edit the alarm configuration. The user interface can also provide guidance to operators on how to handle specific alarms, invoke links to related operator interfaces, or start other CSS tools triggered by alarm PVs.

CSS-WebOPI provides web access to OPIs created in CSS-BOY. With WebOPI, users can monitor their control

Beast feature	Function
alarm server	uses EPICS channel access to monitor and trigger alarms in control system
CSS user interface	views current alarms as table or hierarchical tree (see Fig. 3)
relational database	configures alarm server and logs alarms
web reports	analyzes number and frequency of alarms; searches alarm configurations

TABLE II. Beast alarm system features.

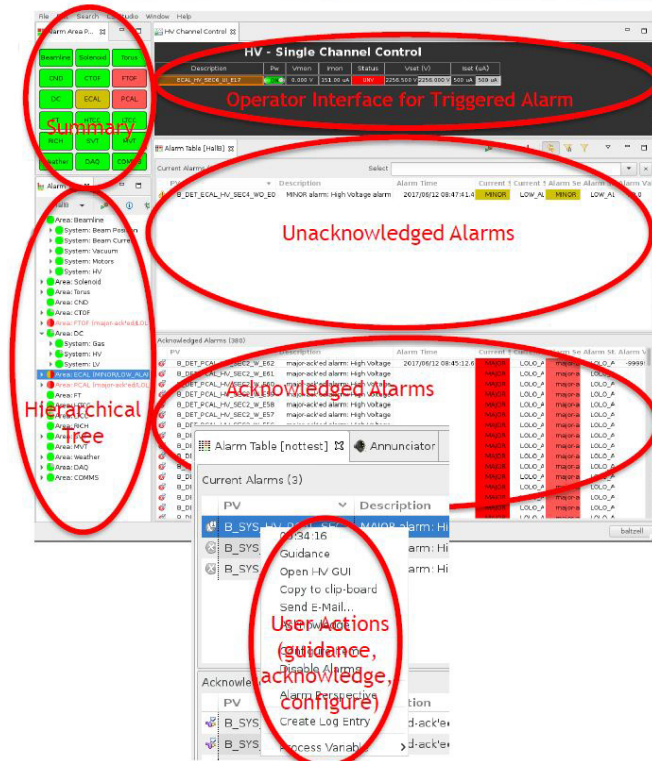


FIG. 3. Beast user interface.

system screens from anywhere in the world, at any time, via a web-browser-equipped device such as a PC, laptop, tablet, or smart phone.

The WebOPI server is used by the control system to translate CSS-BOY screens and serve them to the web. Without any additional work, the user can deploy screens by copying the file developed in CSS-BOY to a pre-configured directory on the WebOPI server.

On the client side, no additional software is needed other than a standard web browser. A Jefferson Lab CUE login is first required on the client to view the screens. Only signal monitoring is allowed via WebOPI. No control of signals will be permitted.

To sum up, an EPICS controls and monitoring system for Hall C featuring CSS is being developed and implemented. This development platform of software applications integrates applications to streamline and increase the reliability of EPICS control systems. The implementation of CSS will standardize control systems across the Physics Division.