ISBN: 978-3-95450-184-7 doi:10.18429/JACoW-IPAC2018-THPML078

WEB-BASED CONTROL ROOM APPLICATIONS AT TRIUMF

C. Barquest*,[1], M. Corwin[2], P.M. Jung[1,3], S. Kiy[1], K. Lucow[1], S. Marcano[2], T. Planche[1], S.D. Rädel[1], D. Sehayek[2], O. Shelbaya[1], B.E. Schultz[1], D. Tattan[4]
[1]TRIUMF, Vancouver, Canada, [2]University of Waterloo, Waterloo, Canada, [3]University of Victoria, Victoria, Canada, [4]Trinity College Dublin, Dublin, Ireland

Abstract

Control room applications are programs that interface with control systems and beam physics models. These tools range from real-time diagnostic visualizations to post-processing data analysis. At TRIUMF, the concept of web-based control room applications has been adopted to advance the capabilities of these applications and facilitate operations. This online model takes advantage of server-based continuous integration and a centralized middleware layer. Continuous integration of server-based applications allows for easy deployment and maintenance. A centralized middleware layer allows a single application to work for many different accelerator configurations. Some motivating examples of web-based applications currenly being developed are presented, demonstrating this online approach to be an effective method for deploying applications for use in the control room and beyond.

INTRODUCTION

Optimal efficiency is achieved when distinct notes of expertise harmonize as one. In the control room, this corresponds to the coordination of beam physics theory with the controls implementation and real-world operation of complex machines. Applications lie at the intersection of these spaces. We aim to take advantage of a tech-driven model, forging a path for efficient operation and application deployment by developing open-source applications which eliminate the dependence on licensed software. Lessons learned on code maintenance and continuous improvement based on user feedback have guided our decisions thus far. We will discuss this online model and its usefulness in our approach to control room applications.

ONLINE MODEL

There are many advantages to deploying control room applications on a web server. We will focus on two features specifically: server-based continuous integration and a centralized middleware layer.

Server-based Continuous Integration

Continuous Integration (CI) of server-based applications allows for easy deployment and maintenance. The basic principle behind CI is to automate the building and testing of applications as teams of developers commit to a shared repository. We currently use GitLab's Continuous Integration and Deployment [1] feature to automatically deploy applications to our production server, and three additional subdomains.

The `devel` subdomain corresponds to a project's master branch; this development mode allows for quick testing of new features. The `staging` mode allows for additional test deployment, and `beta` is reserved for beta-testing. Each mode corresponds to a branch in the git project structure, and are configured to deploy automatically to their respective locations on the server, as seen in Fig. 1. A parallel data structure is also maintained for security, as deployed applications have restricted write permissions.

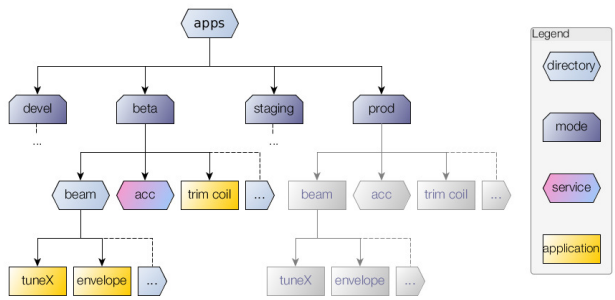


Figure 1: Deployment framework on application server.

A significant advantage to GitLab's archictecture is its built-in issues feature [2] which allows users to provide feedback. This greatly encourages communication and serves as an effective interface between developers and users.

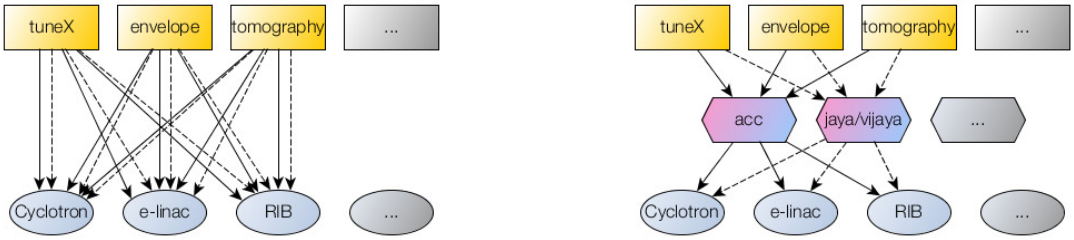
Centralized Middleware Layer

Beyond the deployment advantages, a further improvement of this web-based architecture is what we are calling a "centralized middleware layer". This is a layer of service applications that translate the many different accelerator configurations and raw data types into common descriptions, typically in an XML text-based format, that can then be parsed by each application. For example, the `acc` project is an XML-based description of TRIUMF accelerators and beamlines. Service applications for getting and setting EPICS control values, called `jaya` and `vijaya` respectively, are also deployed.

As seen in the difference between Fig. 2a and Fig. 2b, the number of connections to maintain decrease significantly when this middle service layer is employed. If a change needs to be made for the way we access certain values, for instance, only one edit is needed in order to update all applications to the new method. This leads to code that is easier to maintain, working for a broader range of facilities, and quicker to deploy as interfaces are clearly defined.

* cbarquest@triumf.ca

ISBN: 978-3-95450-184-7 doi:10.18429/JACoW-IPAC2018-THPML078



(a) Schematic without the XML service middle layer.

(b) Schematic with the XML service middle layer.

Figure 2: Making the connection between Applications and Accelerator Descriptions.

MOTIVATING EXAMPLES

A few foundational examples of applications developed and deployed at TRIUMF within this framework are presented here. Note that applications beginning with the word `beam` are closely coupled. These applications are written within a larger modular umbrella application, `beam`, which provides a framework that allows other applications to be quickly developed and deployed as blueprints [3]. This modular template approach allows new applications to quickly take full advantage of the centralized middleware layer as discussed previously, and integrates the applications into a cohesive union.

Beam TuneX

`TuneX` is our flagship web app, developed in response to a survey of user requirements for increasing tuning efficiency.

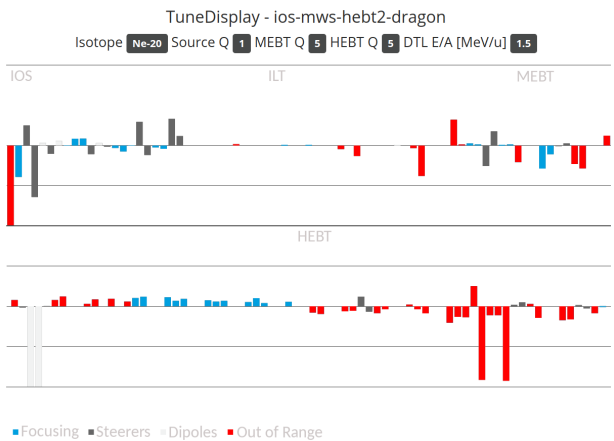


Figure 3: `TuneX` Screenshot example of display mode.

Reference tunes are now curated in a version-controlled XML database. Physical beamline characteristics including position, length, control systems process variables, and operational limits are stored in one collection of XML files.

`TuneX` is composed of three core functionalities – Loader, Display, and Scale. These are the main ways in which an operator will interact with a tune. The Loader function consolidates the many disparate sources of information which previously were needed for setting a tune. The GUI is kept as simple as possible, only ever representing information relevant to the present task-at-hand.

The Display function gives a live overview of differences between the control system setting and reference setting, allowing operators an at-a-glance view of a tune. This is vital for hand-offs between shifts so that any issues or abnormalities in the tune can be brought up and discussed before the outgoing shift leaves. The display platform also gives the opportunity to systematically study tune quality as a function of changing variables.

Last, the Scale function consolidates multiple scaling utilities into one central tool. Not yet fully implemented, this will allow operators to specify both the current and desired beam parameters and adjust the system settings accordingly based on the current state of the tune. This will be done in a nearly identical GUI to that used for the Loader portion of `TuneX`.

Beam Envelope

Beam `Envelope` is an interactive display of the `TRANSOPTR` [4] envelope calculations for given beam paths in our accelerator database. This application allows users to easily take advantage of the optimization and fit routines of `TRANSOPTR` [5] in order to develop new tunes with an intuitive user interface. It is a key component to the model-driven approach to beamline tuning taken at TRIUMF.

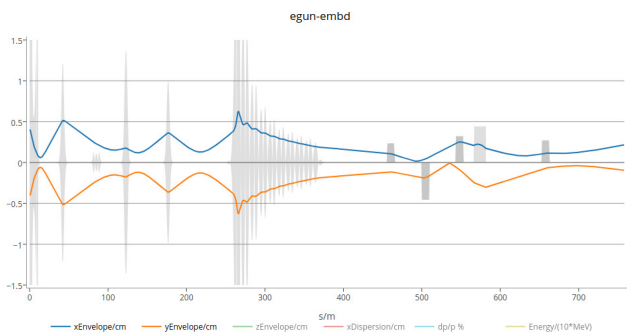


Figure 4: `Envelope` Screenshot depicting an example beam paths horizontal envelope in blue, and negative vertical envelope in orange.

Beam Profile/Tomography

Beam `Profile` and Beam `Tomography` allow diagnostic data to be accessed through the conrol room application web server. The `Profile` app displays profile measurement data with calculated centroid and RMS values for many differ-

ISBN: 978-3-95450-184-7 doi:10.18429/JACoW-IPAC2018-THPML078

ent types of profile monitor types across the site, while the Tomography app reconstructs 2D emittances from measured 1D profiles using the MENT algorithm [6].

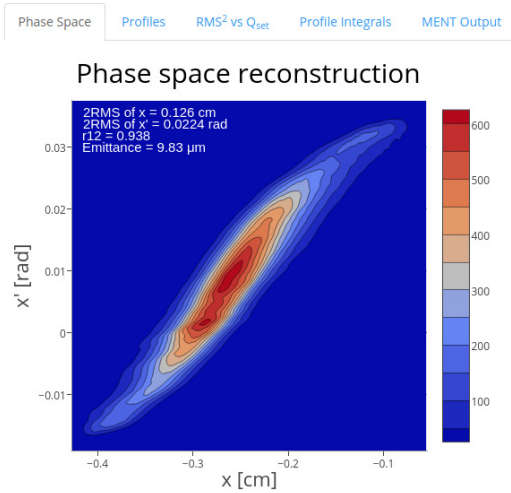


Figure 5: Tomography Screenshot of a horizontal emittance reconstructed from eleven 1D profiles taken while ramping upstream solenoids.

Beam Magnet Degaussing

The Degaussing application is a prototype app that developed organically out of the needs of measurements being performed at TRIUMF's magnet test stand. The goal of this application is to automate the process for applying custom magnet degaussing and measurement routines.

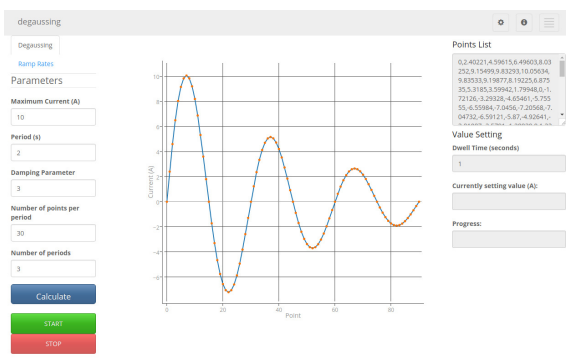


Figure 6: Degaussing screenshot showing the interface for defining custom magnet ramps.

Automatic Tuning of HRS Multipole

An application developed for implementing a novel tuning method for the TRIUMF CANadian Rare isotope facility with Electron Beam ion source (CANREB) High Resolution Separator (HRS) has been affectionately titled ATOM: Automatic Tuning Of (the HRS) Multipole. The multipole corrects for high-order aberrations, increasing the resolution of the HRS. This application will calculate the optimal multipole settings given a user-uploaded emittance measurement which will allow this new tuning algorithm to be available on day one of commissioning [7].

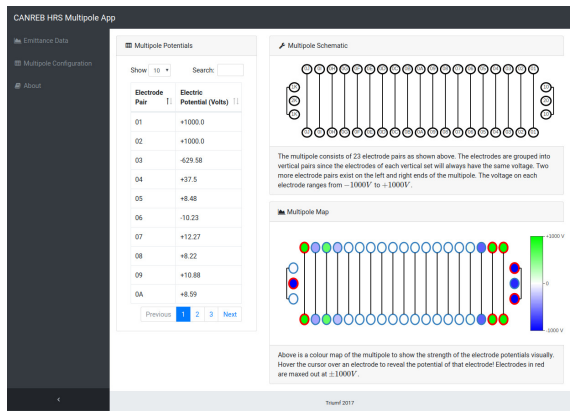


Figure 7: ATOM screenshot of the multipole configuration.

Trim Coil Binder

An electronic version of data that used to be stored in three-ring binders, the Cyclotron Trim Coil Binder application allows users to plot the effect of trim coils as a function of radius or energy. Operators can quickly reference data and even plot custom triplets.

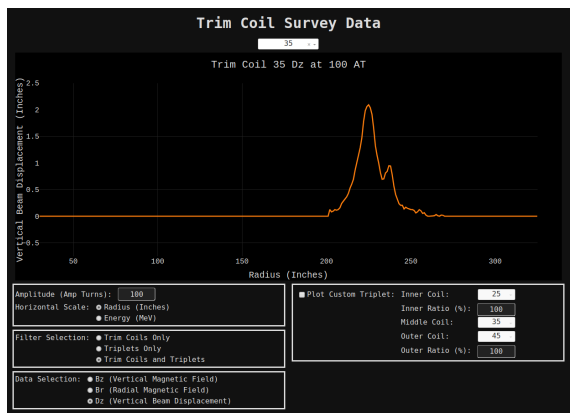


Figure 8: Trim Coil Binder screenshot showing the vertical displacement from trim coil #35 as a function of radius.

CONCLUSION

With a flexible and extensible database at its core, the framework for TRIUMF's web-based control room applications has been designed as a scaleable environment where creativity can flourish and productivity multiply. Utilizing the latest technology available, this platform distinctly couples the development of applications with their operational use, strenthening the circle of feedback and motivating the concept of continuous improvement. More than just setting values quickly and efficiently, web apps are at the apex of control room solutions.

ACKNOWLEDGMENT

The authors thank TRIUMF computing, in particular Kel Raywood and Dan Thomson, as well as TRIUMF controls, in particular Keiko Ezawa, Kyle LeBlanc, and Juan Pon. In addition, the support of Eric Chapman, David Prevost, Friedhelm Ames and Oliver Kester, made this work possible.

ISBN: 978-3-95450-184-7 doi:10.18429/JACoW-IPAC2018-THPML078

REFERENCES

[1] GitLab CI/CD, https://about.gitlab.com/features/gitlab-ci-cd/.

[2] Gitlab Issues, https://about.gitlab.com/features/issueboard/.

[3] Modular Applications with Blueprints, http://flask.pocoo.org/docs/0.12/blueprints/.

[4] Heighway, E. A., and Mark Sybe De Jong. TRANSOPTR-a beam transport design code with space charge, automatic internal optimization and general constraints. No. AECL-6975-REV-A. CM-P00068120, 1984.

[5] Baartman, R. TRANSOPTR: Changes since 1984. TRIUMF, Tech. Rep. TRI-BN-16-06, 2016.

[6] Minerbo, Gerald. "MENT: A maximum entropy algorithm for reconstructing a source from projection data." Computer Graphics and Image Processing 10, no. 1 (1979): 48-68.

[7] D. Sehayek et al."Multipole Tuning Algorithm for the CANREB HRS at TRIUMF", presented at IPAC'18, Vancouver, Canada, May 2018, paper THPML079, this conference.