An Accelerator Physics - Software Engineering Collaboration

On-Line Model for FRIB Commissioning

Johan Bengtsson & Michael Davidsaver

(EPIC Consulting, B. Dalesio)

05/27/16

Abstract

We outline the strategy for and summarize on the result of an effective collaboration for providing an on-line model for the commissioning of the FRIB (Facility for Rare Isotope Beams).

For numerical efficiency, the model is based on the beam envelope formalism for Linacs from the 1980s and an elaborate RF cavity model developed by FRIB.

The following topics will be covered: “Model-Based vs. Equipment Control”, “Why Yet Another Simulation Code?”, “The Merits of Automated Regression Testing”, “Software Requirements: A Use Case Approach”, “The Benefits of a Tightly Integrated Python Interface by Design”, “Rapid Prototyping and Re-Factoring”, “Software Infrastructure, and “Deliverables”.

Outline

- Background, Motivation, and Inspiration.
- Model-Based vs. Equipment Control.
- Why Yet Another Simulation Code?
- Strategy.
- Implementation.
- Deliverables.

Background: Robust Design - NSLS-II

Proceedings of PAC07, Albuquerque, New Mexico, USA MOOAAB01

PHILOSOPHY FOR NSLS-II DESIGN WITH SUB-NANOMETER HORIZONTAL EMITTANCE*

S. Ozaki#, J. Bengtsson, S. L. Kramer, S. Krinsky, V. N. Litvinenko
BNL, Upton, Long Island, NY 11973 USA.

- Robust (conceptual) design ->
- Implementation: An (complex) “Engineering Problem” ->
- Requirements, tolerances, and quality assurance” -> “Turn it on”, and then push the limits (e.g. Swiss Light Source PAC 2001); vs. “make it work”.
- Model Based simulations and design -> predictable results.
- SLS -> sub micron orbit stability (PAC 2003;; thanks to digital BPM front-ends and power supplies):

“To know when to compromise vs. holding your ground, is what distinguishes the good engineer from the crowd.”

“Model Based” vs. “Equipment” Control

Proceedings of ICALEPCS07, Knoxville, Tennessee, USA

NSLS II CONTROL SYSTEM”

Leo Bob Dalesio (BNL, Upton, NY)

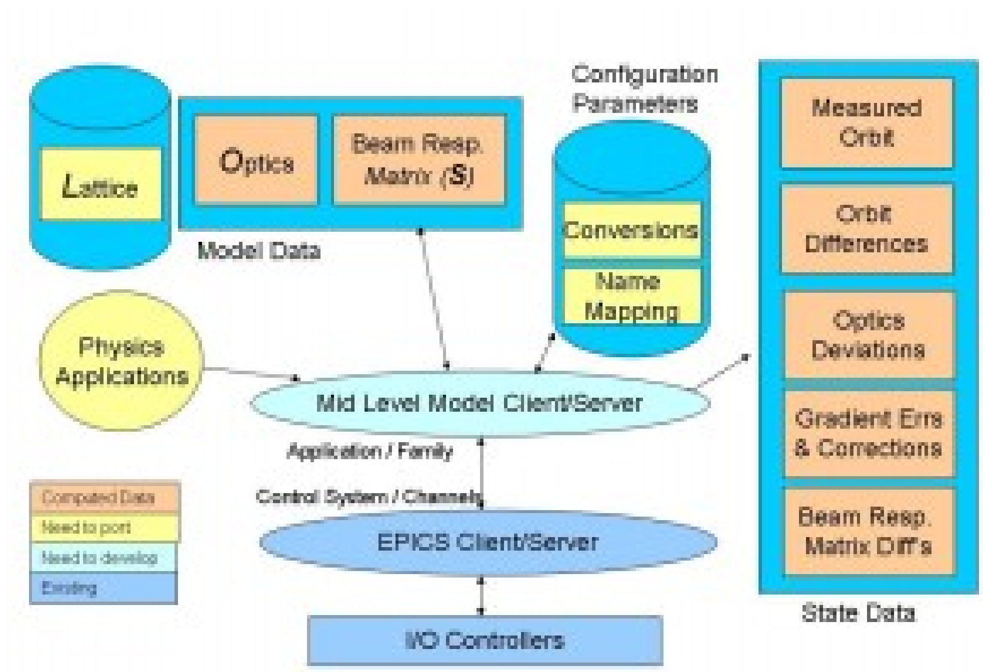


Figure 4: High Level Application Software Architecture.

Background: A Virtual Accelerator via a VIOC

(James Rowland, DIAMOND, EPAC 2006)

THPCH113 Proceedings of EPAC 2006, Edinburgh, Scotland

THE DIAMOND LIGHT SOURCE CONTROL SYSTEM

M. T. Heron, M. G. Abbott, P. H. Amos, K. A. R. Baker, Y Chernousko, T. M. Cobb , C. A. Colborne, P. N. Denison, I. J. Gillingham, A. Gonias, P. Hamadyk, S. C. Lay, M. A. Leech, P. J. Leicester, M. McClory, U. K. Pedersen, N. P. Rees, A. J. Rose, J. Rowland, E. L. Shepherd, S. J. Singleton, K. Vijayan, I. S. Uzun, Diamond Light Source Ltd, Oxfordshire, UK; P. H. Owens, CCLRC Daresbury Laboratory, Warrington, UK; S. Hunt, Alceli Hunt Beratung, Chaletweg 8, 5616 Meisterschwanden, Switzerland.

Virtual Accelerator and Physics Applications

To enable early testing of physics tools through the control system, a virtual accelerator has been implemented to give simulation of the accelerators though the intended PV interface. This was developed by providing EPICS device support to interface to the model using the TRACY II libraries [2]. For physics tools the Accelerator Toolkit [3] for Matlab is used.

James also prototyped and demoed a Python interface for Tracy-2 (the computer model used to guide the robust design for NSLS-II).

Background: NSLS-II Model Server

G. Shen: NSLS-II Virtual Accelerator -> J.M.S. (James Modeling Services, ICALEPS 2013).

Proceedings of ICALEPCS2009, Kobe, Japan THP094

A MODULAR ENVIRONMENT FOR HIGH LEVEL APPLICATIONS

Guobao Shen, BNL, UPTON, NY 11973, U.S.A.

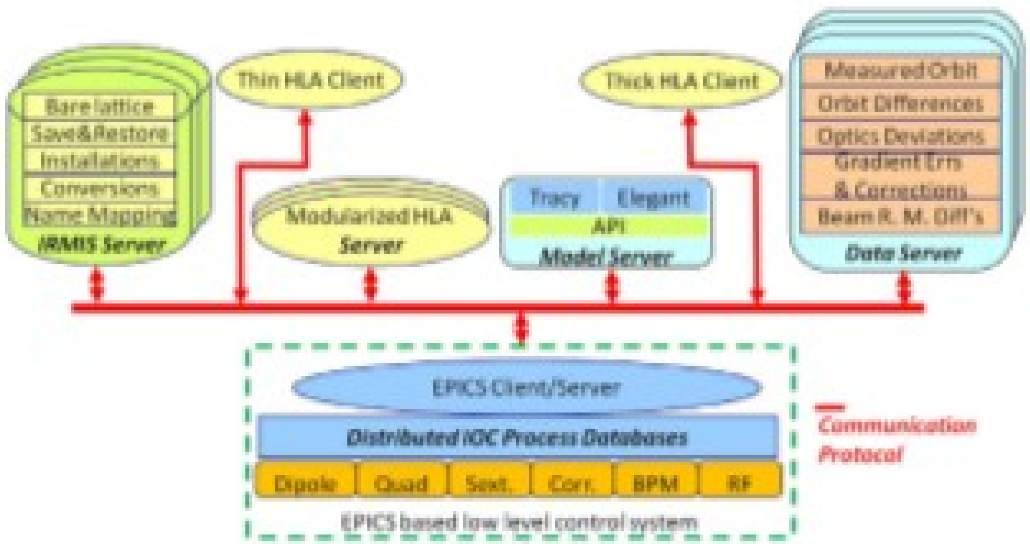


Figure 1: System architecture for a modular environment for high level application.

Scope: FRIB

Facility for Rare Isotope Beams (SRF 2009).

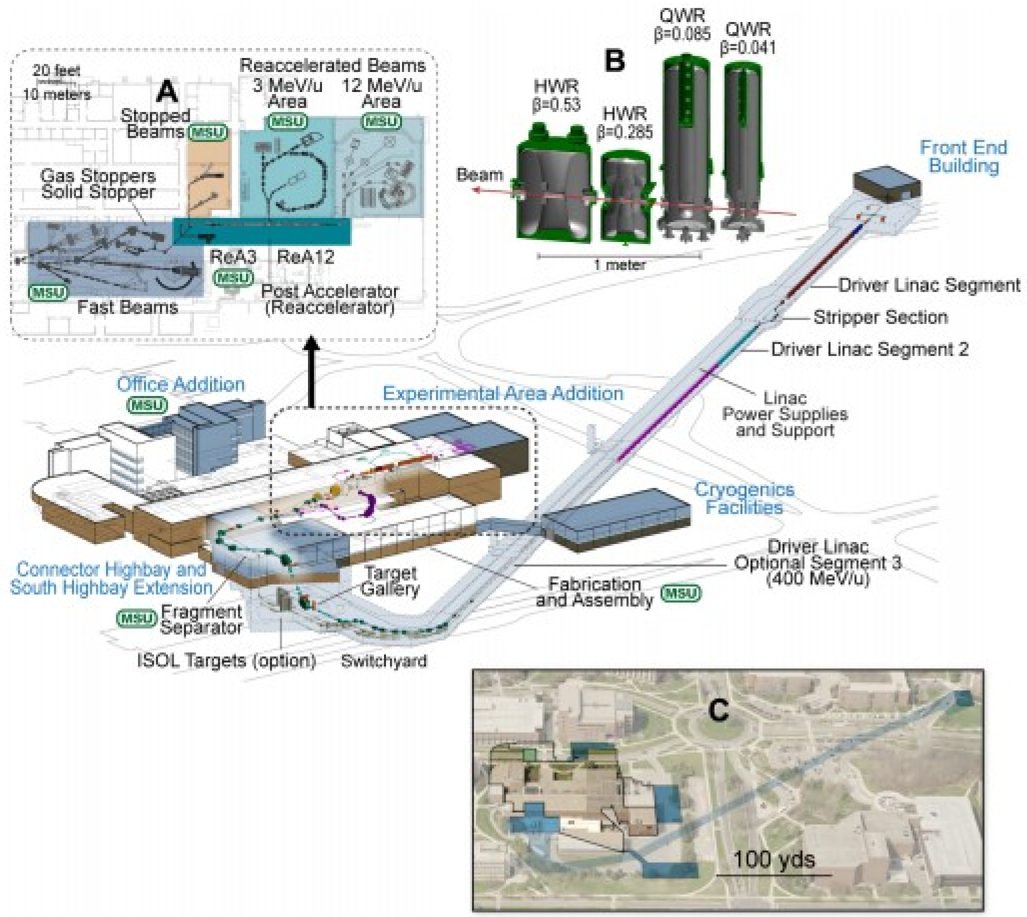


Figure 1: The proposed FRIB facility at MSU showing the driver linac extending from the Front End building through Segment 3. A switchyard will deliver the linac beam to the Target Gallery where the rare isotope beams will be produced and filtered by the Fragment Separator system after which they will be delivered to the experimental area shown in more detail in the magnified region shown in (A). The four superconducting accelerating structures of the baseline design are shown in (B). FRIB overlaid on the MSU campus is shown in (C).

Why Yet Another Simulation Code?

Impact, a PIC (Particle-In-Cell) code was used for the FRIB conceptual design. -> Too slow for the on-line model.

The Beam Envope Formalism for Linacs was developed at CERN and Los Alamos in the 1980s.

By now, many implementations are available; with various levels of legacy coding paradigms (Fortran) and resulting software architecture.

The Matlab “Middle Layer” Toolkit, originated by G. Portmann (PAC 2003), now popular for synchrotron light sources, has demonstrated the utility of providing a scripting environment for accelerator physicists for: commissioning, beam studies, etc.

Python was chosen for NSLS-II (IPAC 2012).

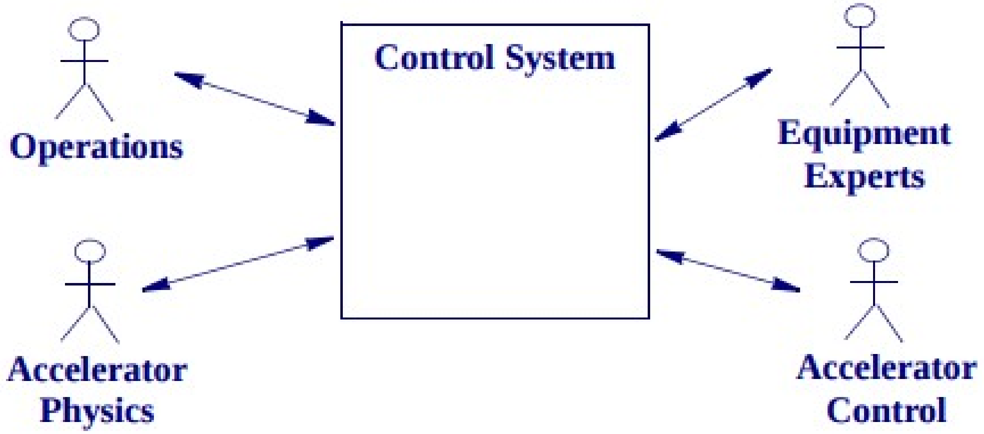
As a result (Virtual Accelerator, Model Server, Python, EPICS v4, etc.) when G. Shen joined FRIB, there was a desire to leverage on these techniques and tools. In particular, there was a need for a (fast, extendable) “Numerical Engine” with a transparent Python interface.

Strategy

- Stage the software project into a prototyping phase-I and implementation phase-II.
- Make a “tightly coupled”, i.e., transparent Python interface a requirement from the start (Mike).
- Increase the “comfort zone” for “breaking” and re-factoring prototype code by implementing Automated Regression Testing from the outset (Mike).
- In other words, enable a “rapid prototyping” and “agile software development” environment.

Strategy (cont.)

- As part of phase-I prototyping, work out the “Use Cases” with the End Users (and stake holders)



- Implement the software Infrastructure (Mike).
- Pursue a “spiral approach” to capture the requirements: instead of “chasing a moving target”, pursue successive refinements to drill down as the implementation evolves.
- Keep re-factoring and benchmarking prototype code.

Deliverables

- Model fidelity: the Beam Envelope Formalism had been benchmarked vs. Impact (a PIC code) by FRIB beforehand.
- Execution time: < 100 msec for two charge states for the entire Linac structure (~160 m).
- Lex/Yacc based (-> extendable) Lattice Parser (Mike).
- Syntax-Free Lattice Grammar (Mike), i.e., it can be extended "recursively": new elements and attributes can be implemented for the beam dynamics model; without requiring code changes to the scanner/parser.
- Code quality.
- Automated regression testing.
- Documentation: beam dynamics model and User Guide.

Lessons ~~Learnt~~ Revisited

- Scope: invariant; time & cost variables.
- Staging: Phase-I (prototype) & Phase-II (implementation).
- Software Requirements: a “moving target”.
- “Use Cases” (End User point of view).
- A “Spiral Approach” (vs. “waterfall”; e.g. Mil Spec).
- Automated Regression testing -> enables “rapid prototyping” and an “agile software development” environment by increasing the “comfort zone” for breaking a re-factoring prototype code.
- Numerical Engine and Infrastructure (C++) with a “tightly”, i.e., transparently, integrated scripting environment (Python); for the End User.

Acknowledgements & References

James Rowland (formerly at DIAMOND): a Virtual Accelerator client/server, implemented via a Virtual IOC (EPAC 2006) -> “end-to-end” testing of controls applications; before e.g. Linac commissioning.

G. Shen:

- “A Software Architecture for High Level Applications (PAC 2009),
- “Development Plan for Physics Application Software for FRIB Driver Linac” (IPAC 2015).

Z. He (FRIB):

- “A Thin-Lens Cavity Model for Fast Linac-Beam Tuning” (LINAC 2012).
- “Beam Dynamics Influence from Quadrupole Components in FRIB Quarter Wave Resonators” (HB 2014).

K. Fukushima (FRIB; joined recently).

B. Dalesio (EPIC Consulting).

R.C. York et al “FRIB: A New Accelerator Facility for the Production of Rare Isotope Beams” (SRF 2009).

Open Source Approach

Thank You!