PXIE/PIP-II Control System

J. Patrick

October 7, 2014

Outline

- PXIE Control System
- Thoughts on PIP-II Control System
- Focus on general infrastructure, not specifics
  - And not much on hardware

Goals for PXIE Control System

- Control/operate the PXIE accelerator
  - Learn what is needed for PIP-II design
- Test bed for for PIP-II
  - Try out new hardware and software
  - Without compromising the first goal

Control System Scope

- High level subsystem control and monitoring:
  - Power Supplies
  - Water
  - Cryo
  - Instrumentation
  - RF
  - Motion Control
  - Machine Protection
- Beam control applications
- Timing system

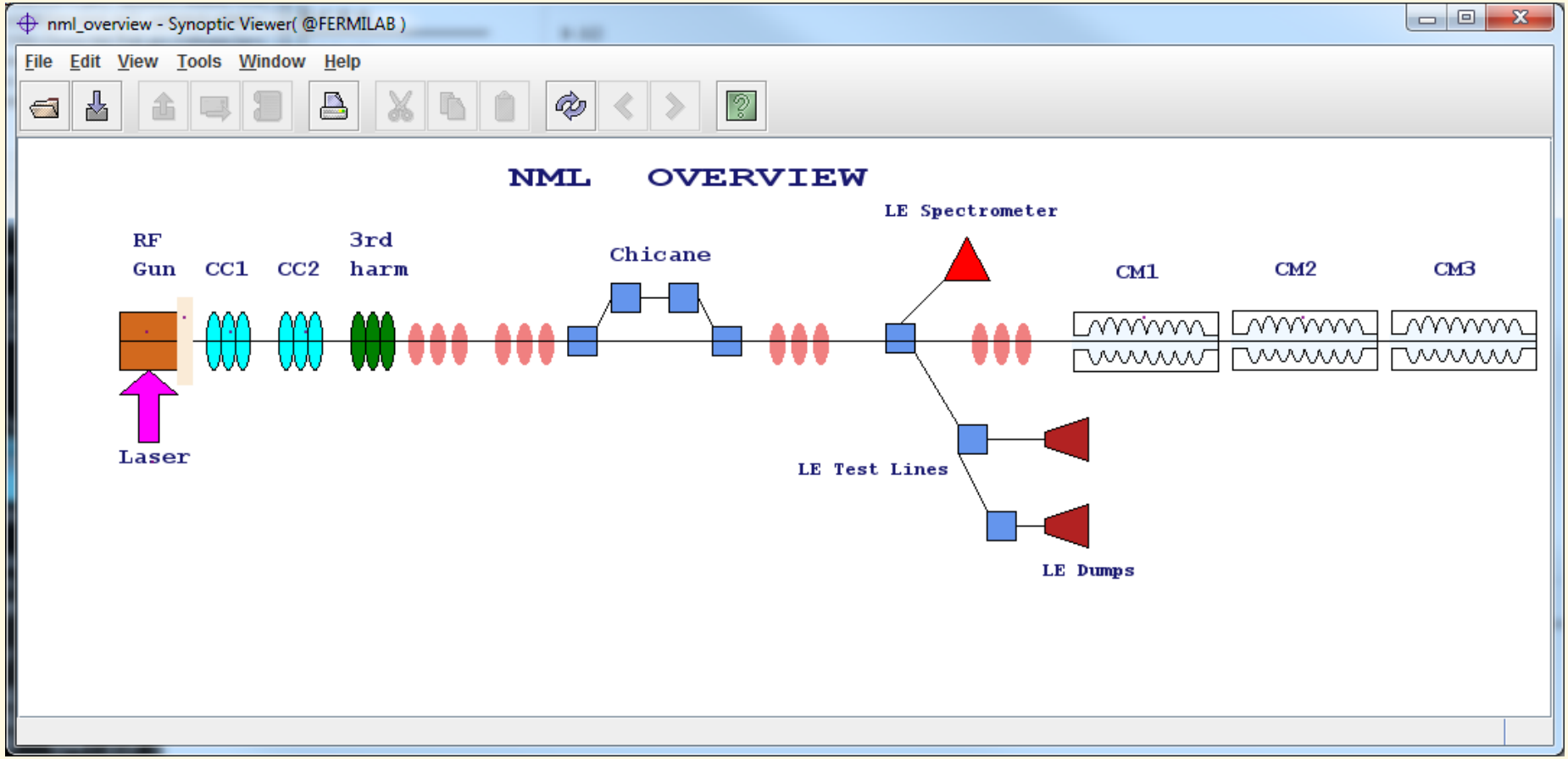
PXIE General Plan

- Based on ACNET
  - It is what we know and use in the main complex
- Similar to NML control system
- NML tries to be forward looking:
  - No CAMAC, C190/290 MADCs, CIA vacuum, IRMs, …
  - Instead VME, PLCs, HRMs
  - Limited console style applications.
  - Instead use of synoptic displays, Java applications, …
- Assume most development by FNAL
  - Need discussion with ANL on HWR subsystem

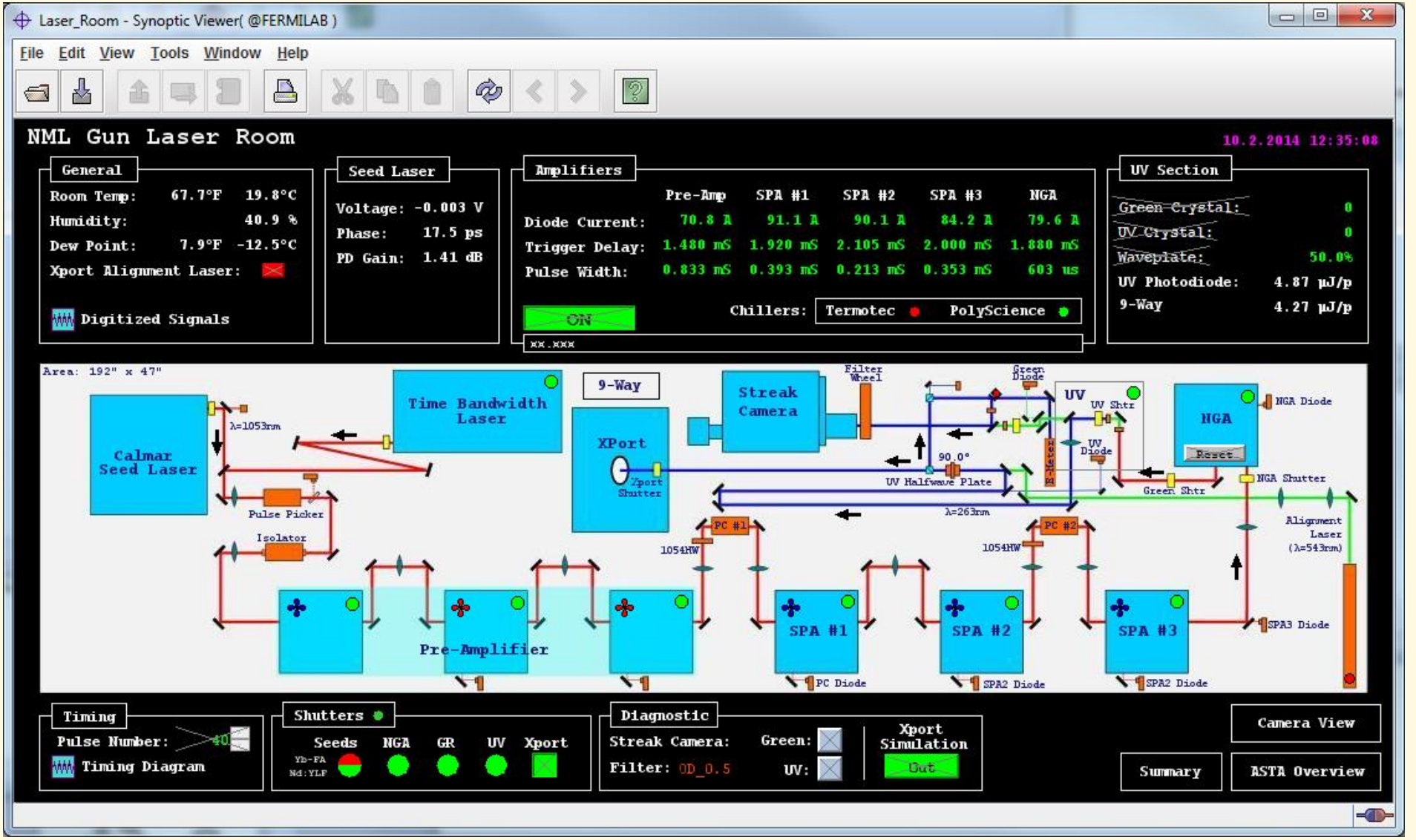
Synoptic Display

- Drag and Drop builder, no code writing required
- Launch via console index page, or from a web browser
- Web display available, no extra work required
  - Viewable on phones
- Supports plots, fetching data logger data
- Can create links to other displays
- Basic functional expressions available without writing code
- Possible to have Finite State Machine back-end
- Possible to attach ACL scripts for more complex things
  - Discouraged for very complicated things
- Heavily used at NML and by cryo in general

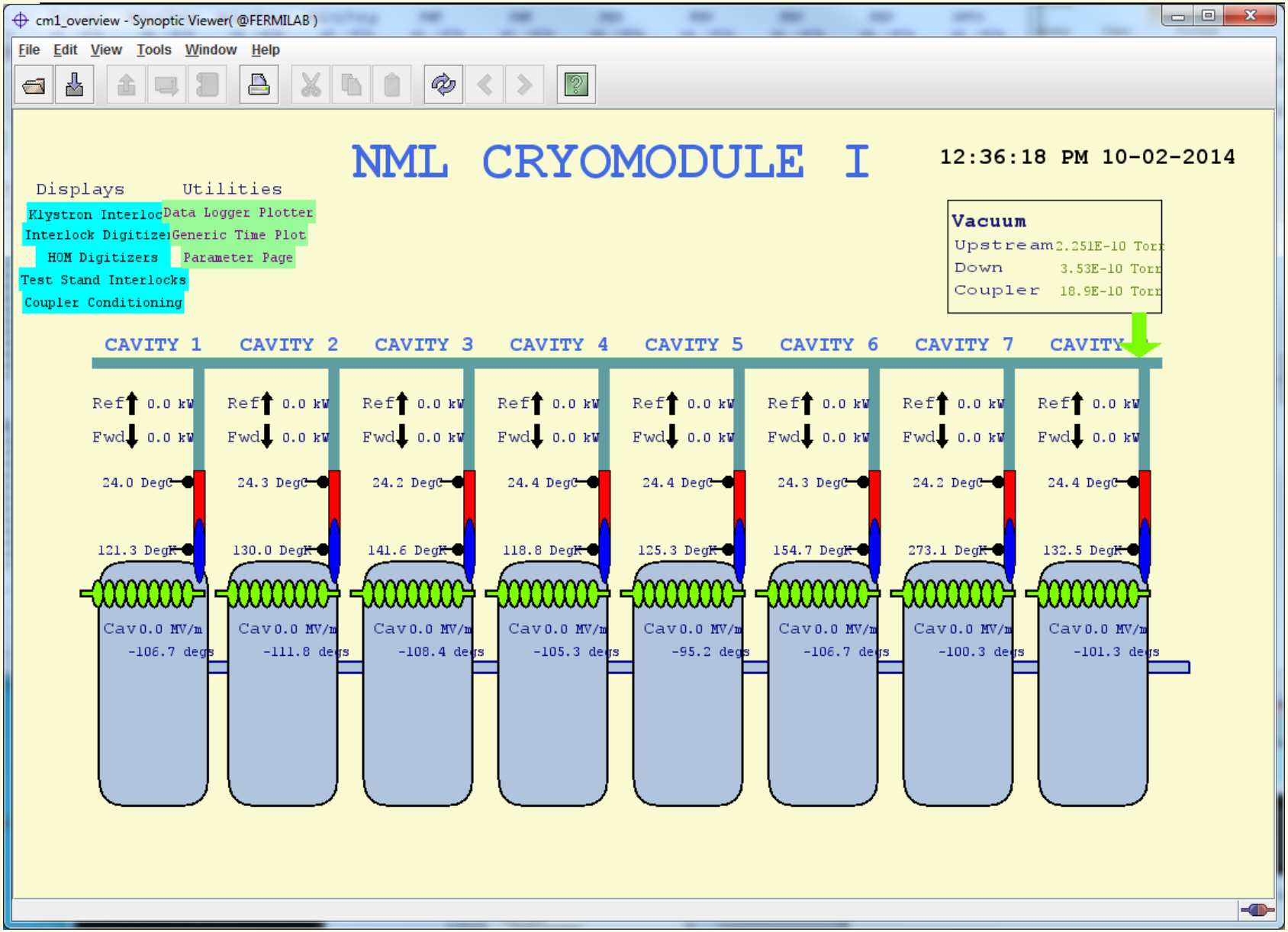
NML Overview Synoptic



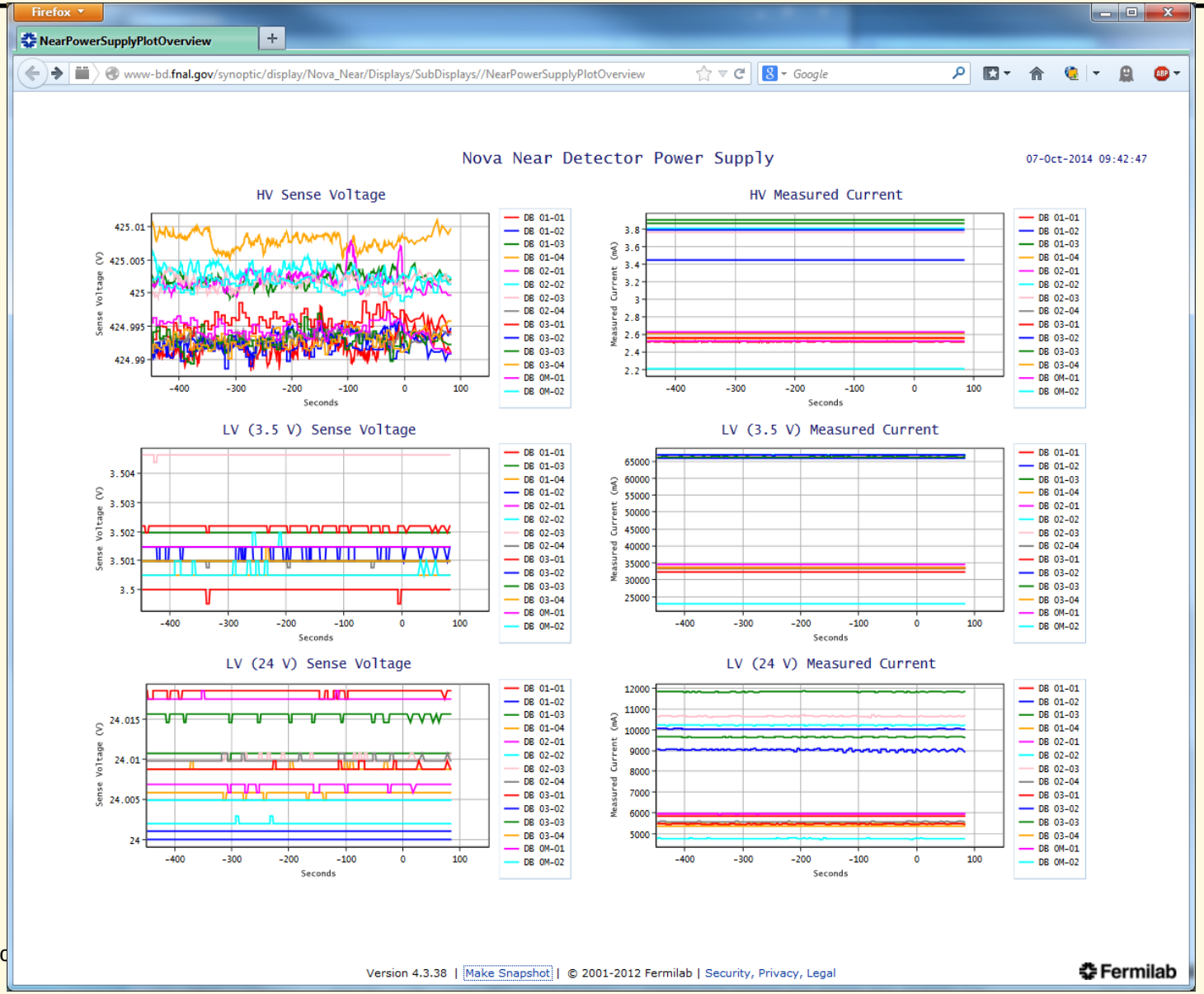
NML Laser Room Synoptic



NML Cryomodule Synoptic

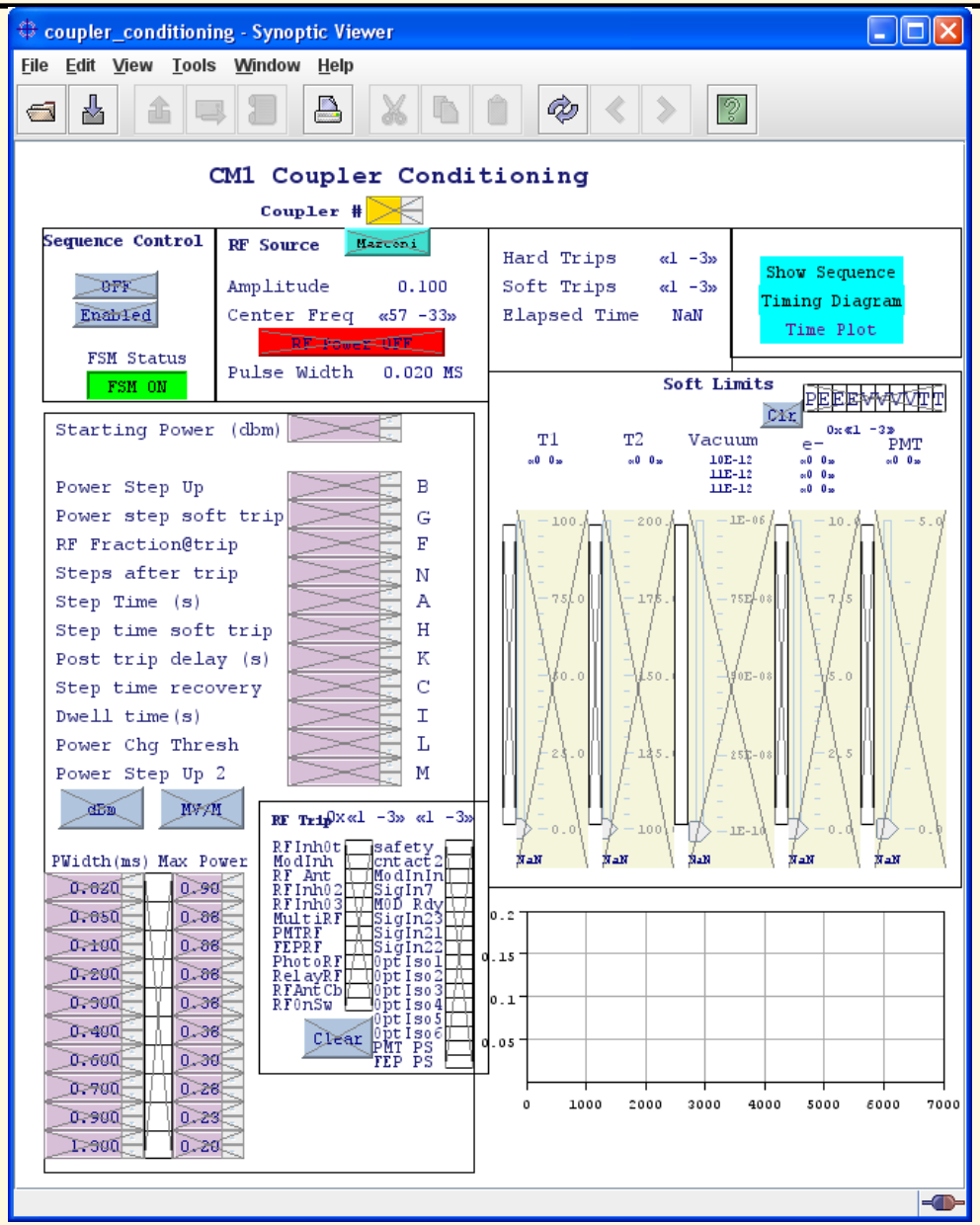


Nova Near Detector (Web version)



NML Coupler Conditioning

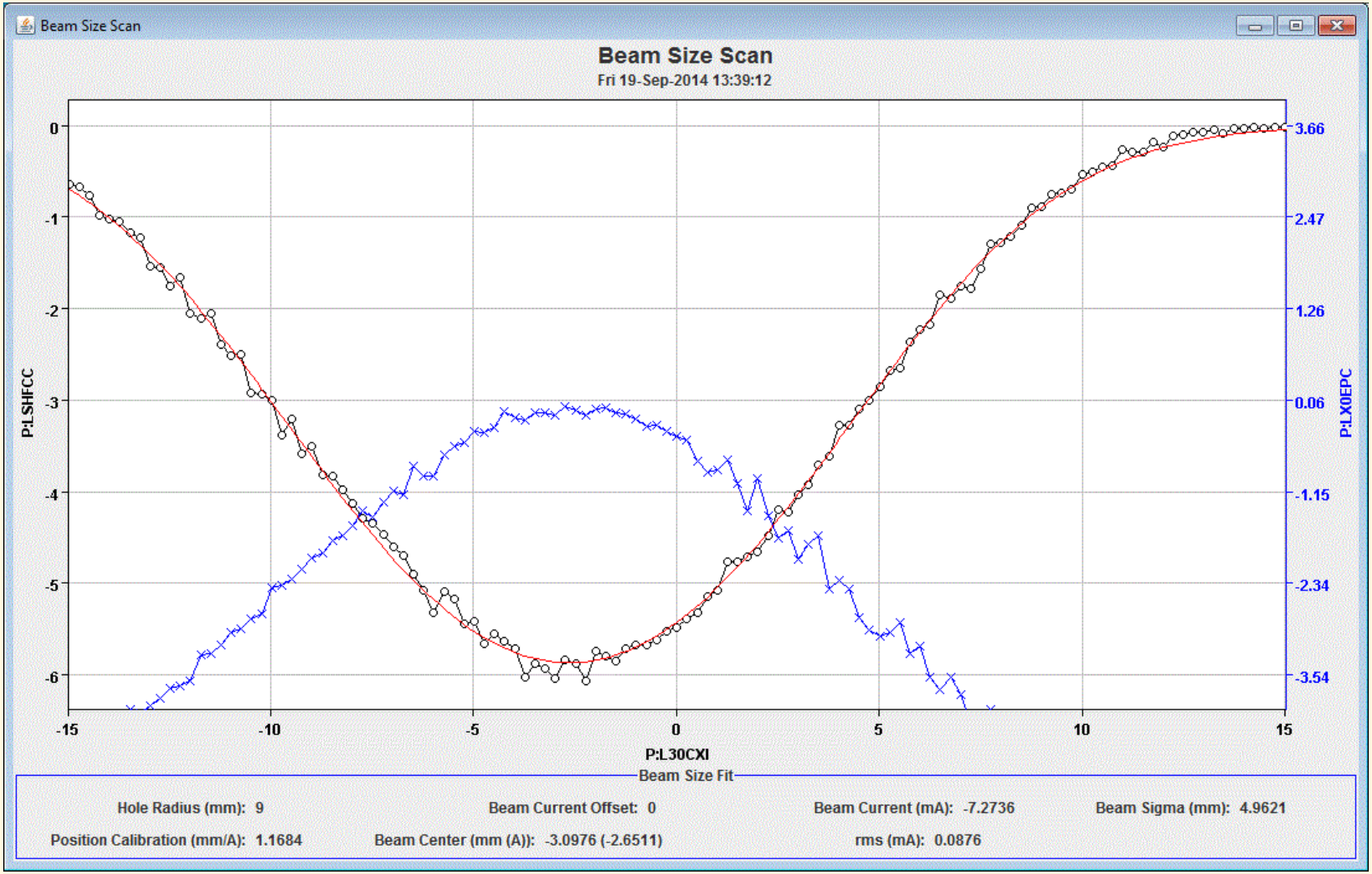
Backed by FSM



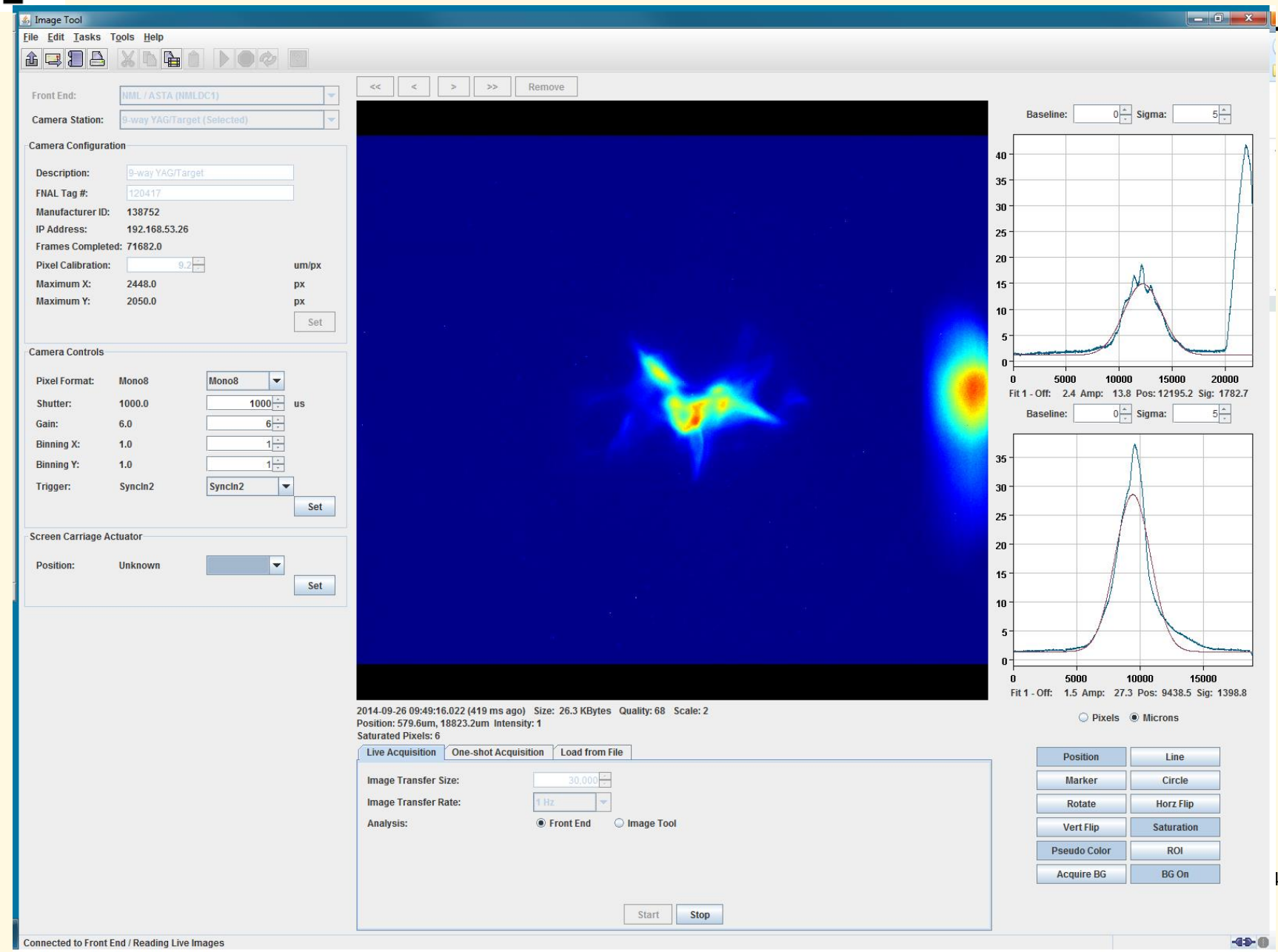
Applications at PXIE

- Currently no synoptic displays at PXIE
- Scan application recently written for use at PXIE
- Other work uses Labview or console plots
- Need to agree on what is needed, how they will be written, and who will write them
- We do have an interface to Matlab if there is interest
  - Currently no code management or launch from console

PXIE Scan Tool



NML Camera Image Tool



PXIE Summary

- Follow NML control system philosophy
- Current control system but with
  - Modern hardware/front-ends
  - Emphasis on synoptic displays
  - Java applications for complicated things
  - A good place to try something else if someone has an idea

Thoughts on PIP-II

- Major Constraint:
  - Large existing complex
  - PIP-II linac must interoperate with it
    - Ideally common a common control system
    - Timing and Machine Protection
  - Not practical to completely replace the current control system
- Major Unknown:
  - Role of India
- The following concerns the general infrastructure
  - Limited discussion of details
  - No discussion of hardware platforms
    - An equally intense religious issue

Current Control System

- Operates a very large and complicated complex
- Largest and most complicated aside from CERN
- Major increase in complexity beginning with start of BNB
  - No fundamental architectural changes required
- Very accessible – anyone can run a console
- Very large code base well managed (mostly…)
- Great flexibility in data acquisition (event+delay)
- Major evolution during Run II
  - Data acquisition engines, greatly expanded data logging
  - VAXes -> Linux
  - Many new hardware systems (BPMs, BLMs, instrumentation…)
  - The very modular nature of the system allowed this to happen

Some Issues

- Console user interface is not very modern
  - Java applications have existed for many years
  - Some have been useful but most were not successful
  - Attempts at modern C++ GUI framework have not been successful
    - However several MI programs use the ROOT C++ package
- Lots of old hardware
- Lots of old software
- Large data packets not well supported (images)
- Structured data supported in ad hoc way
- Limited capacity for logging large arrays
  - Images and waveforms
  - Note cataloging and fetching them out is harder than logging them

Often Asked Question

- Why don't we use EPICS?
- Everyone else does
- Then we just download it and we are mostly done

What is EPICS?

- EPICS is a toolkit, not a complete control system
- It specifies a communication protocol and front-end architecture, these are common among all installations
- It includes a basic set of tools which are fine for getting started or small installations, but not at all suitable for a large complex.
- Not only are machines different, but so are the people who build and operate them.

EPICS cont.

- So every large installation does something different for
  - Core applications (time plots, device viewer, etc.)
  - Synoptic display
  - Application framework and programming language
  - Alarms
  - Data logging
  - …..
- With multiple large projects currently in progress, there is an attempt at more commonality
  - NSLS-II, FRIB, ESS, LCLS-II, ITER, …
  - But there are still very major differences for each
  - And other light source labs are not so much involved in this effort

Other Labs & EPICS

- SNS
- LCLS-I & II
- NSLS-II
- XFEL
- ESS
- FRIB

- Fermilab

- There are many other users, but the theme is similar

SNS

- SNS began operating in 2007, and was the largest EPICS system at the time.
- They developed a new everything on the previous list
  - Synoptic display (EDM)
  - Machine application framework (**XAL**)
    - Including common machine description and online model
  - Device database (and other databases)
  - Alarm system (multiple iterations)
  - Data logging (multiple iterations)
  - E-log (multiple iterations)
  - …
  - Developed Control System Studio after started operations
    - Integrated framework for “core” applications, more later

LCLS(-I)

- Had custom legacy control system + hardware
- About as old as the Fermilab control system
  - Less evolution over the years, less modular system than ours
- General strategy was:
  - Keep old system for the part of the linac used for LCLS
  - Use EPICS for new things
  - Some interoperability between the two
- Several SNS people moved to SLAC
  - Original plan was to reuse SNS code, but in the end not much was
  - SLAC had a long history of doing things in a certain way
  - Different machine model format, different programming languages
- Machine control migrated to EPICS but in a different way
  - Matlab primary application language going forward

NSLS-II

- Developed separately from RHIC – separate accelerator
  - New technologies, initially less influence of past culture
  - But influenced by the tastes of those who developed it
- No reuse from original SNS or SLAC
  - They do pick up Control System Studio from SNS
- New database for machine model and everything else
  - IRMIS-II.
- Different framework for machine applications (python)
- Initiated extending EPICS communication protocol to support structured data (EPICS v4; pvAccess)
- New data logger based on ”big data” technology

FRIB and ESS

- SNS machine application framework and core applications have been refactored to make it more modular, modern, and usable by other places (OpenXAL)
- FRIB and ESS have contributed and expect to use it
- They also expect to use Control System Studio
- They are also jointly working on basically a next iteration of the NSLS-II database architecture (DISCS)

- How much and how long they will stay in sync is not clear

- Note ESS plans to have a small core group and outsource much of the development to CosyLab

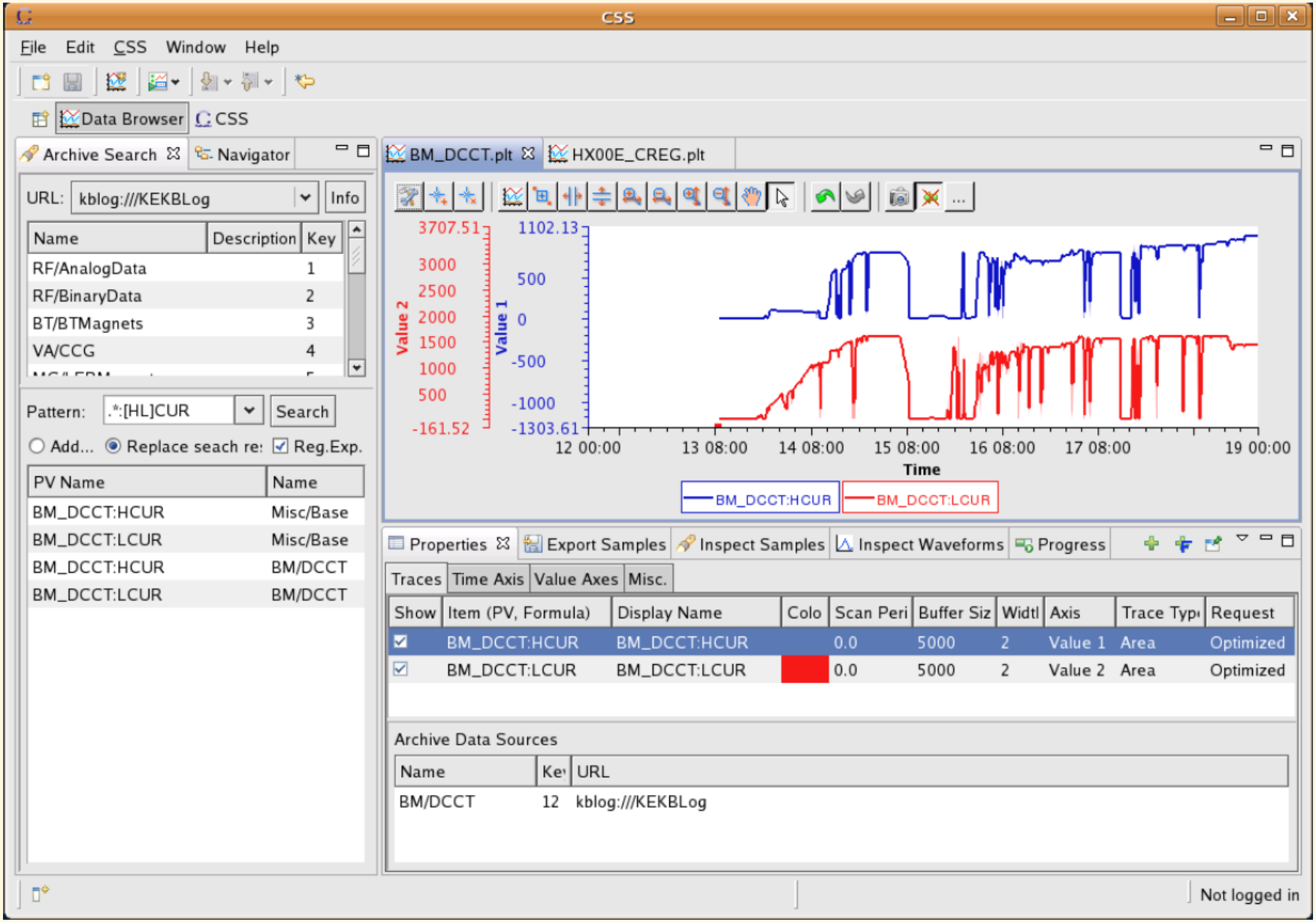
XFEL

- Marriage of four different control systems
  - DOOCS (updated from FLASH)
  - TINE (updated from HERA)
  - EPICS (cryo system)
  - Tango (experimental beamlines)
- Each is a complete control system and has its own user interface framework, logging, alarms, etc. etc.
- User interfaces in one system can access quite a lot on the other systems
  - Several different interface technologies developed
- I don’t consider this a good model…

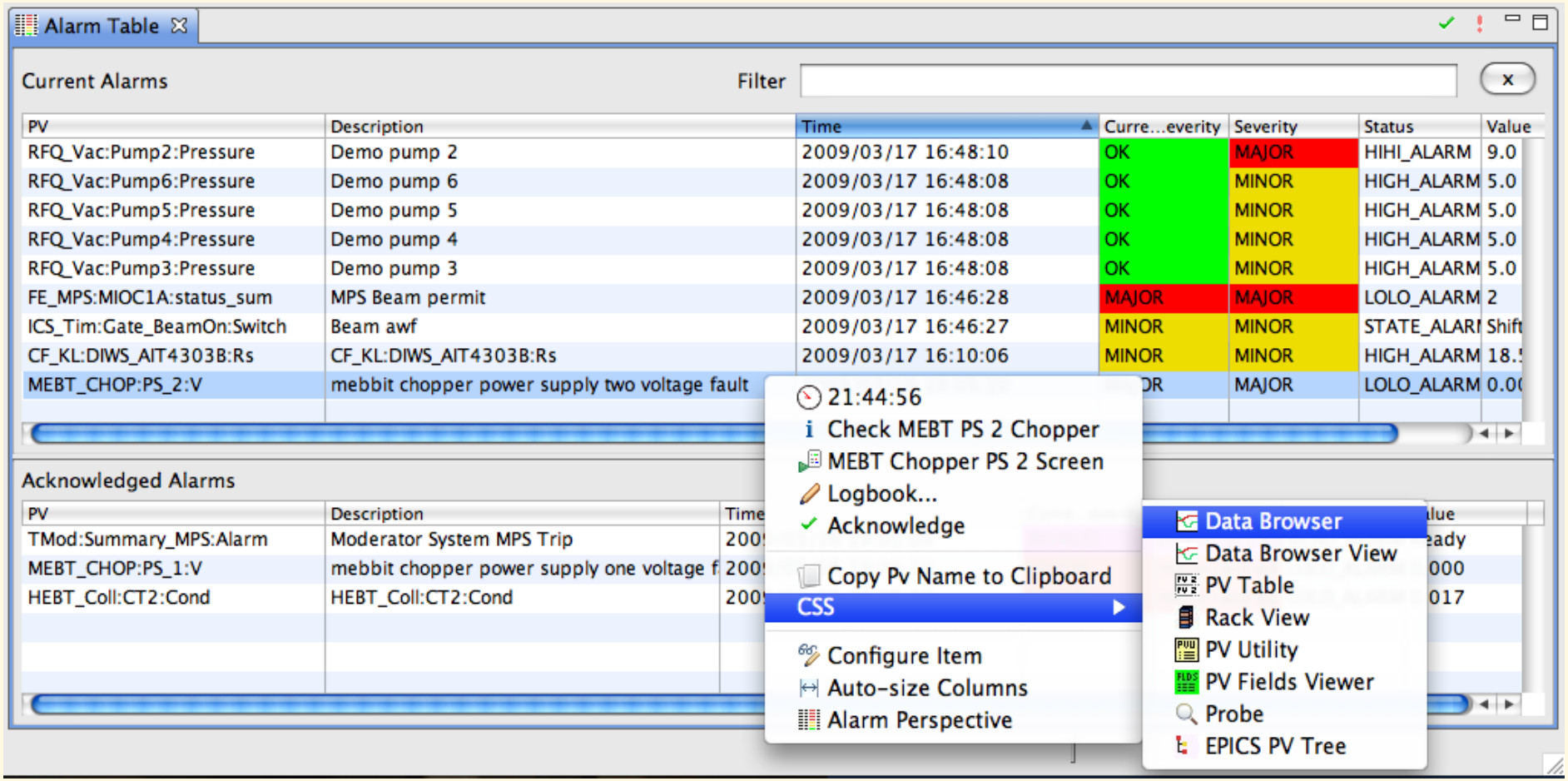
Control System Studio

- Integrated package of “core” applications
- Plotting, alarms display, device viewer, synoptic display, logbook, hardware configuration, ….
- Based on Eclipse development framework
- Plugin based architecture
  - Plugins can communicate context with each other
  - You can create an instance with a selected set of plugins
- Originated at DESY, intended to be system agnostic
  - Picked up by SNS, became mostly EPICS-centric
- In operational use by SNS, NSLS-II
  - XFEL, ESS, FRIB, ITER and others plan to use it
- And here at Fermilab! – Nova Detector Control System

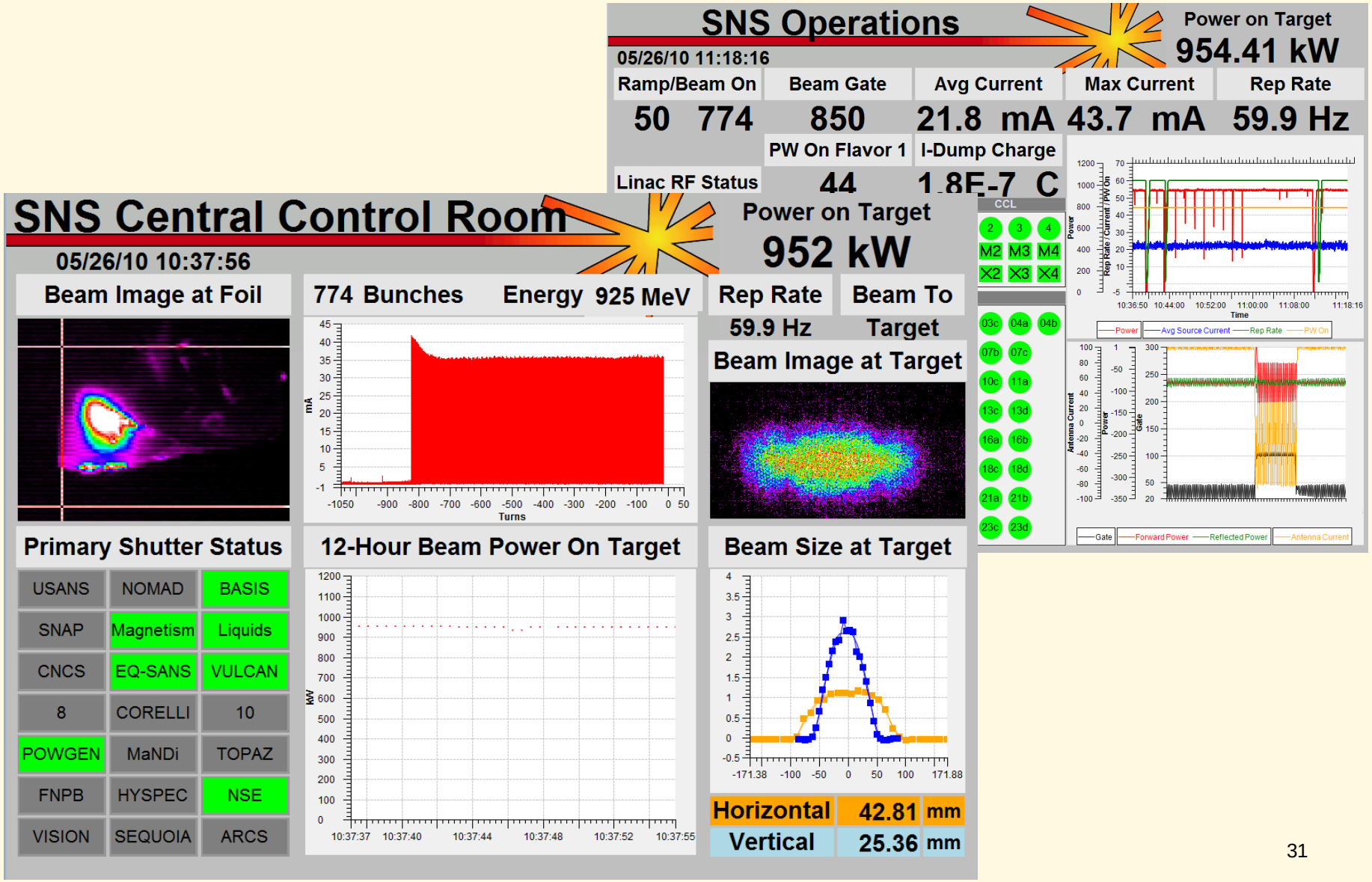
CSS Data Browser



Alarms Display w/Link to Other Tools



SNS Operational Displays



Fermilab

- HINS and NML both started as EPICS only systems
- Used mostly older core EPICS tools
  - Striptool for plots, old alarm system, etc.
- EDM synoptic display from SNS
- Data logger from SNS
- Some python applications written
- Lacked the organization of the main control system
  - Independent code management
  - No central console – run from command line
- To set it up properly was significant work and was not done initially.

Transition at HINS/NML

- HINS mostly transitioned to ACNET, NML did completely
- EDM synoptic displays put in MECCA
- Made launchable from ACNET consoles
- Dual EPICS/ACNET front-end developed
  - Selected instrumentation front-ends + all IRMs
- Possible to access ACNET devices in EDM
- Front-ends eventually all converted to pure ACNET
- Synoptic displays completely replaced EDM displays
- Python applications ported to Java

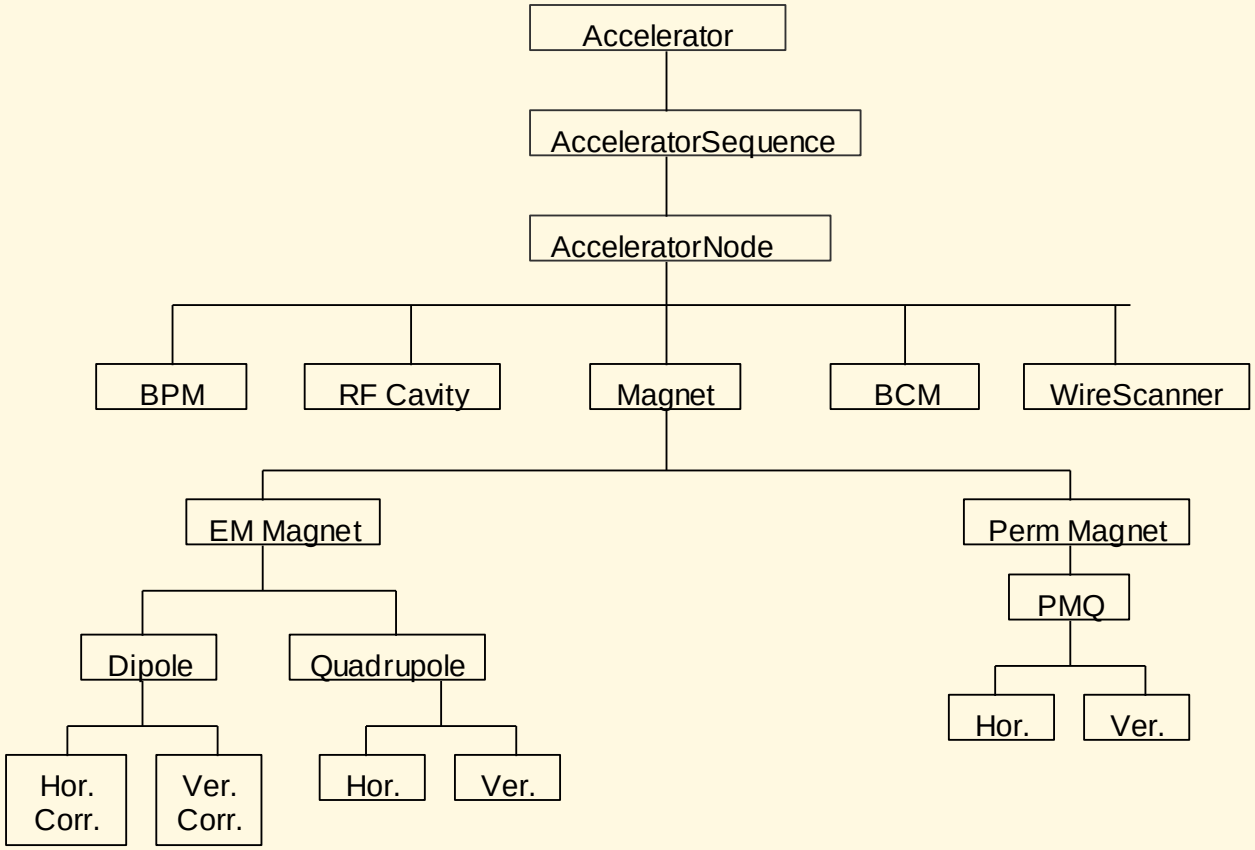
Lessons Learned

- It was interesting to try
  - We all learned a lot
- To do it in a scalable, maintainable way requires work
  - Ideally with some thought before doing the work
- Especially for a small system, there really wasn't any advantage to doing this in EPICS.
  - Expected collaboration with other labs never developed, at least for EPICS related things
  - It did take significant resources to do this. CD was involved also.

Timing System

- PIP-II would benefit from a new system
- New system for pulsed Project X prototyped in 2010
  - 1 GHz bandwidth
  - Much larger number of possible events (TCLK limited to 256)
  - Data payload associated with clock event
    - Removing need for separate MDAT link
  - Counter cycle stamp associated with clock event
    - Aid in correlating data across front-ends
- Should rethink this in context of new accelerator design
- Collect requirements – make suitable for future CW?
- Try out at PXIE?
  - Significant lead time will be required

Accelerator Hierarchy (SNS)



- Accelerator hierarchy from the accelerator physicist point-of-view

XML Accelerator Representation (partial example)

```
<sequence id="MEBT" len="3.633">
  <attributes>
    <sequence predecessors="RFQ"/>
  </attributes>
  <node type="marker" id="Begin_Of_MEBT" pos="0" len="0"/>
  <node type="QH" id="MEBT_Mag:QH01" pos=".128" len=".061" status="true">
    <attributes>
      <magnet len=".061" polarity="-1" dfltMagFld="-34.636"/>
      <align x="0.0" y="0.0" z="0.0" pitch="0" yaw="0" roll="0"/>
      <aperture shape="0" x=".016"/>
    </attributes>
    <ps main="MEBT_Mag:PS_QH01"/>
    <channelsuite name="magnetsuite">
      <channel handle="fieldRB" signal="MEBT_Mag:QH01:B" settable="false"/>
    </channelsuite>
  </node>
```

Summary

- Common control system for PIP-II & existing complex
- Start with what he have and evolve it
  - Can't disrupt operations between now and then
- Modernization of both hardware and software is needed!
  - Plan needed for the existing complex too!
- If parts of EPICS help us get there, we should consider it
  - We have experience in running a mixed system
  - There needs to be a benefit to justify any extra needed resources
- EPICS is different at every large accelerator
  - There is a variety of past experience and personal tastes
  - And EPICS constantly evolves along with computing technology