Application Programming

John Galambos

Accelerator Physics, ORNL

Feb. 12-14, 2002

Outline

- Infrastructure progress
  - Database progress, signal requirements
  - Started an infrastructure, will present programmers with a device oriented view of accelerator
- Time correlation issues
  - Gather time synchronized data across the accelerator
- Virtual Accelerator
- MEBT commissioning applications
- Staff / organization / planning
  - Significant improvement on staffing, organization since last review

Acknowledgements: C. Allen, S. Aleksandrov, S. Assadi, P. Chu, B. Dalesio, K. Danilova, D. Gurd, P. McGehee, J. Patton, N. Malitsky, N. Pattengale, M. Plesko, D. Purcell, T. Pelaia, A. Shishlo, C. Sibley, T. Shea, E. Williams

General Progress Since Last Application Programming ASAC Report

- In Feb. 2001, application programming group had just expanded to 2 people
  - conceptual ideas
  - examining infrastructure possibilities

- Present situation
  - functional group (4 people at ORNL + collaborators)
  - database created
  - programming infrastructure started
  - initial applications being tested

Application Programming Scope

- Application programs are high level programs involving interactions between the machine, instruments and models
  - more than a simple display of data, e.g.:
  - feedback on machine parameters based on input from the machine
  - use of a lattice model

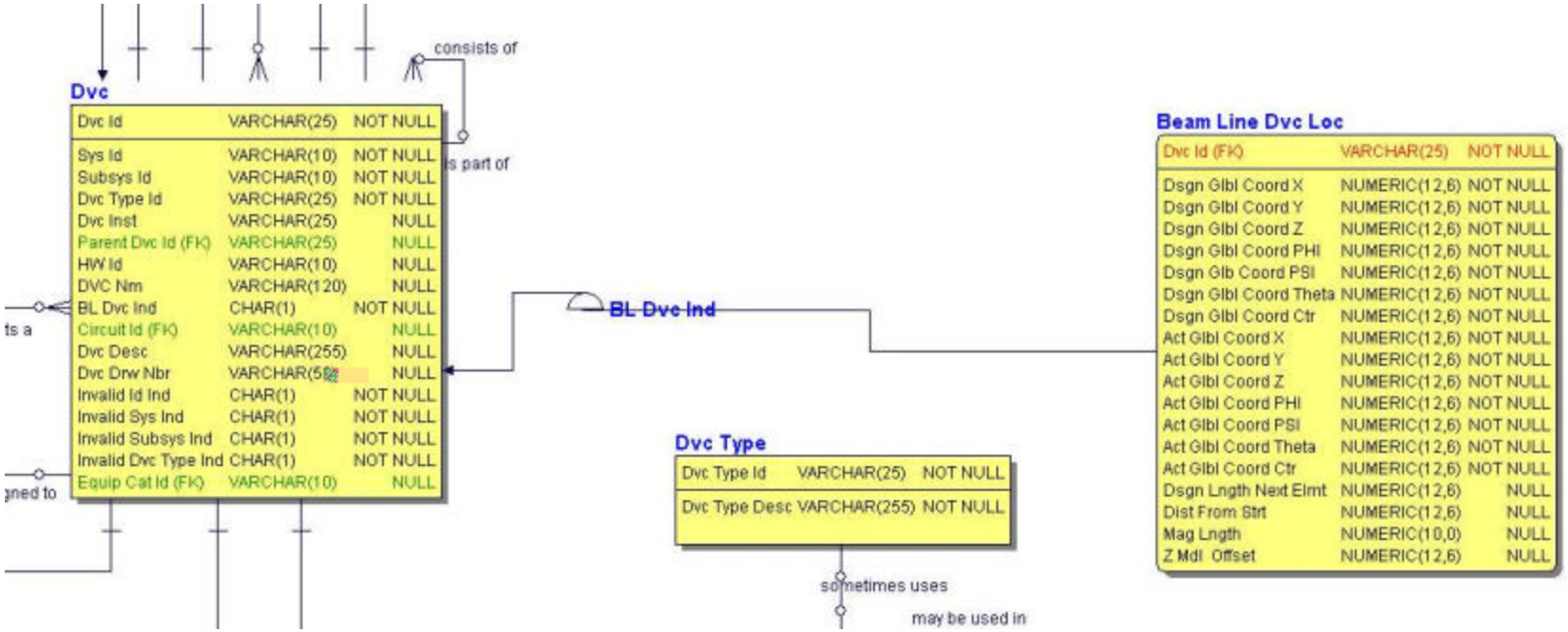
- Collaborative effort with diagnostics and controls
  - (e.g. use controls group version control system)

- Some Application Programming needs to be functional at the time of beam commissioning
  - MEBT at LBNL: March 2002
  - MEBT: Oct. 2002
  - DTL: Feb. 2003
  - Ring/transfer lines:Dec. 2004

Global Database Schemas are Defined, Data Being Populated

- Table schemas, entities, attributes, relationships defined for the beamline devices as well as signals and other accelerator equipment

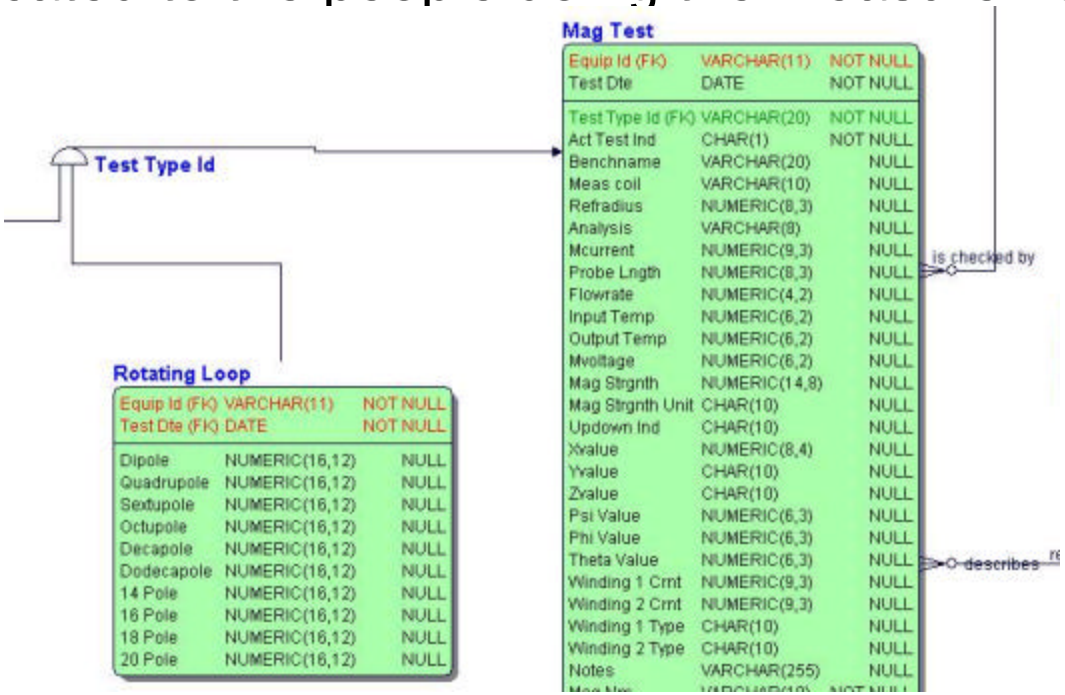
- MEBT device data populated, being used for initial programming tests



Fragment of the beamline device schema – for details see
http://ntser3.cad.ornl.gov/mis/EnterpriseModel/modell.htm

Magnet Measurement Data Requirements Specified

- Database has Tables for storing magnet measurement data
  - Multipole harmonics
  - Excitation curve information
  - Installation position
- Magnet measurement requirements have been written and communicated to the people doing the measurements.



EPICS Signal Requirements are Specified

- EPICS Process Variables expected for application programming are compiled
- Includes signal types for Diagnostics, Magnets, RF, Timing
- Discussions with cognizant groups

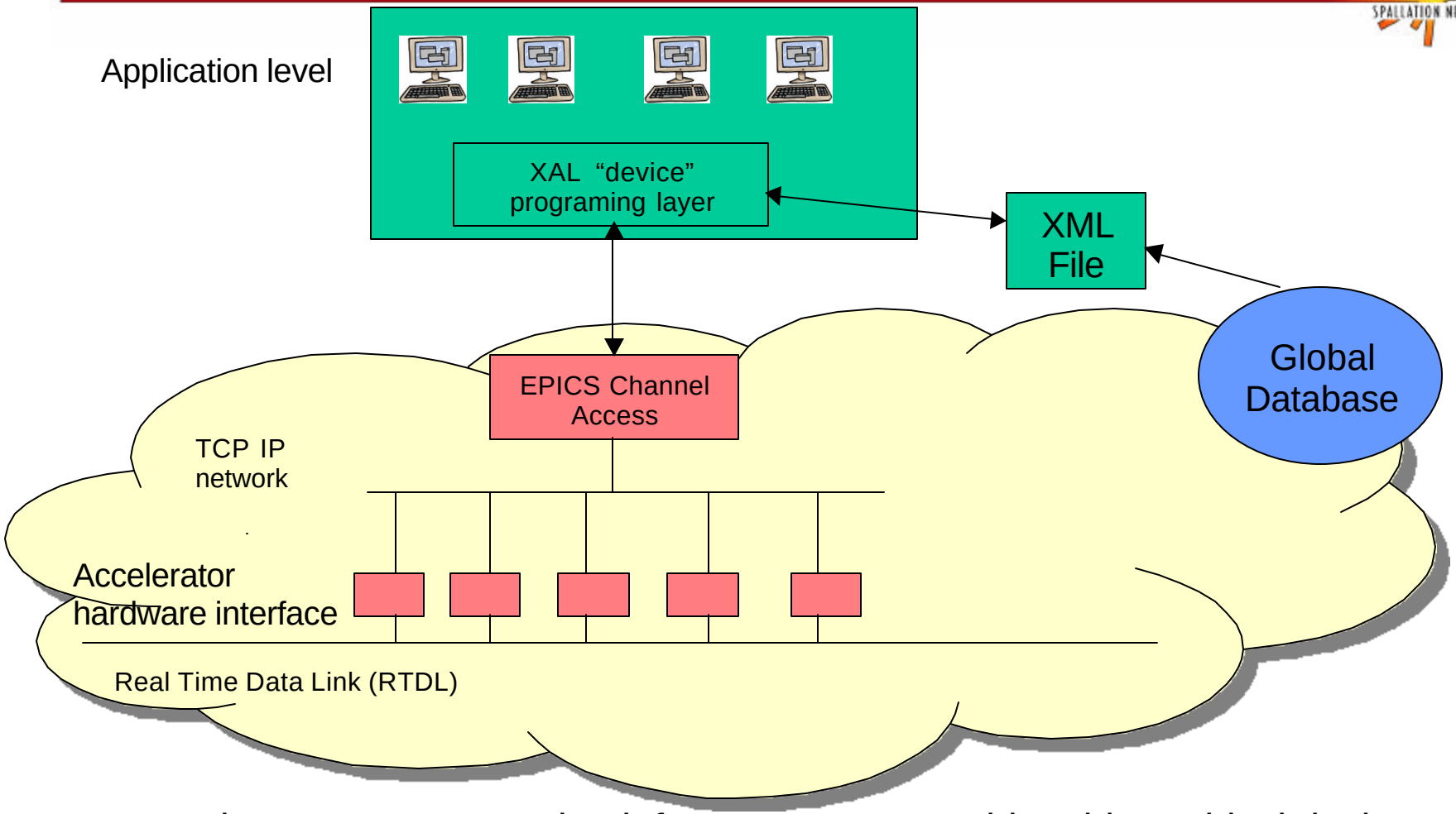
Process Variables	Name	Different PVs for each user type	Single PVs for all user types			
General				Event Triggers		
User type of pulse	pulseType			x	x	
				Snapshot1	*Snapshot2*	*Sna*
BPMs						
X position averaged over the macropulse (mm)	xAvg*	x		x		
Y position averaged over the macropulse (mm)	yAvg*	x		x		
Magnitude averaged over the macropulse (AU)	amplitudeAvg*	x		x		
Phase averaged over the macropulse (deg)	phaseAvg*	x		x		
X position averaged over the macropulse (mm)	xAvg		x	x	x	x
Y position averaged over the macropulse (mm)	yAvg		x	x	x	x
Magnitude averaged over the macropulse (AU)	amplitudeAvg		x	x	x	x
Phase averaged over the macropulse (deg)	phaseAvg		x	x	x	x

See http://www.sns.gov/APGroup/appProg/PVList.pdf

Global Coordinates Are Part of the Database

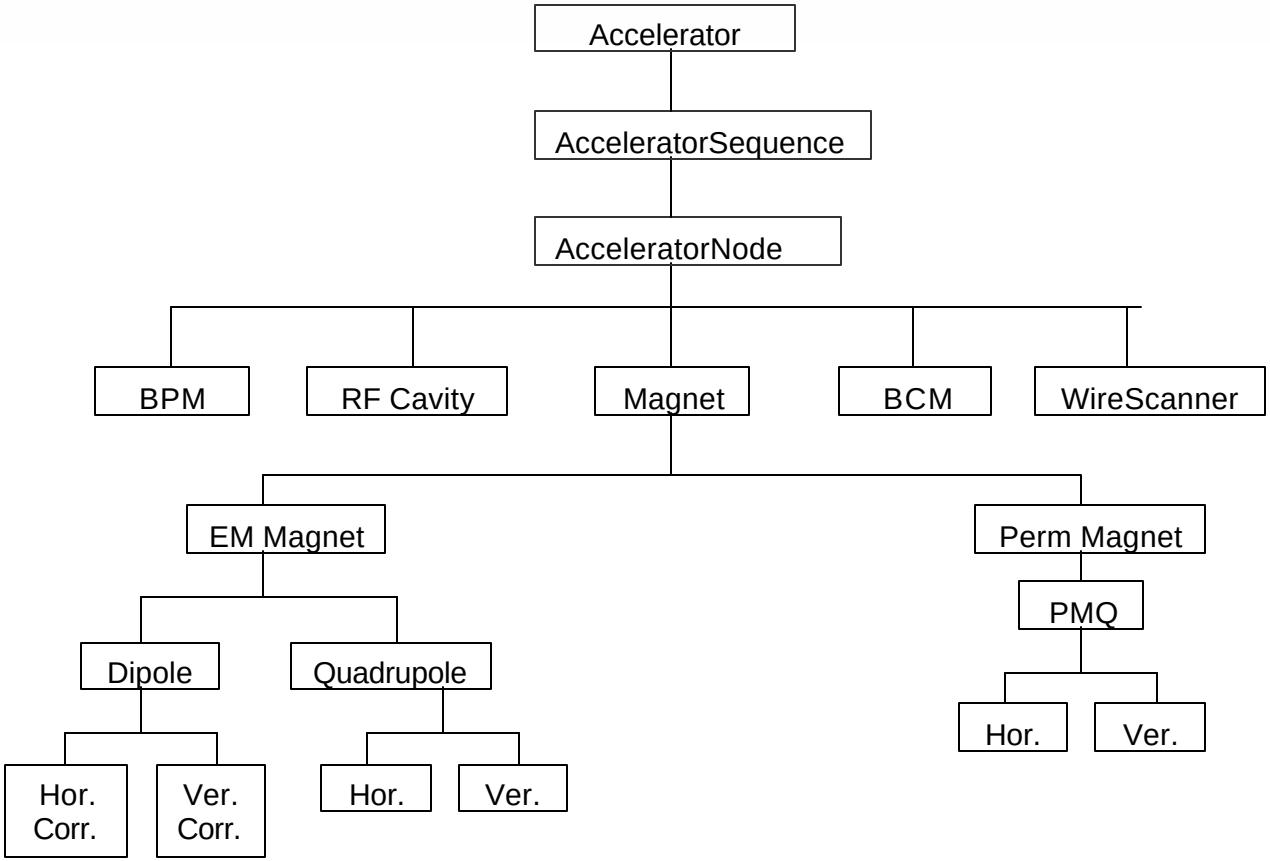
- Global coordinates collected for all primary magnets in SNS (436 dipoles/quads)
  - http://www.sns.gov/APGroup/refTables/GlobalCoordTable2.pdf
- MEBT global coordinates are in the global database
- Data is being collected for correctors, diagnostics, and other beamline devices (foils, collimators etc.)

High Level Programming Infrastructure Started (XAL)



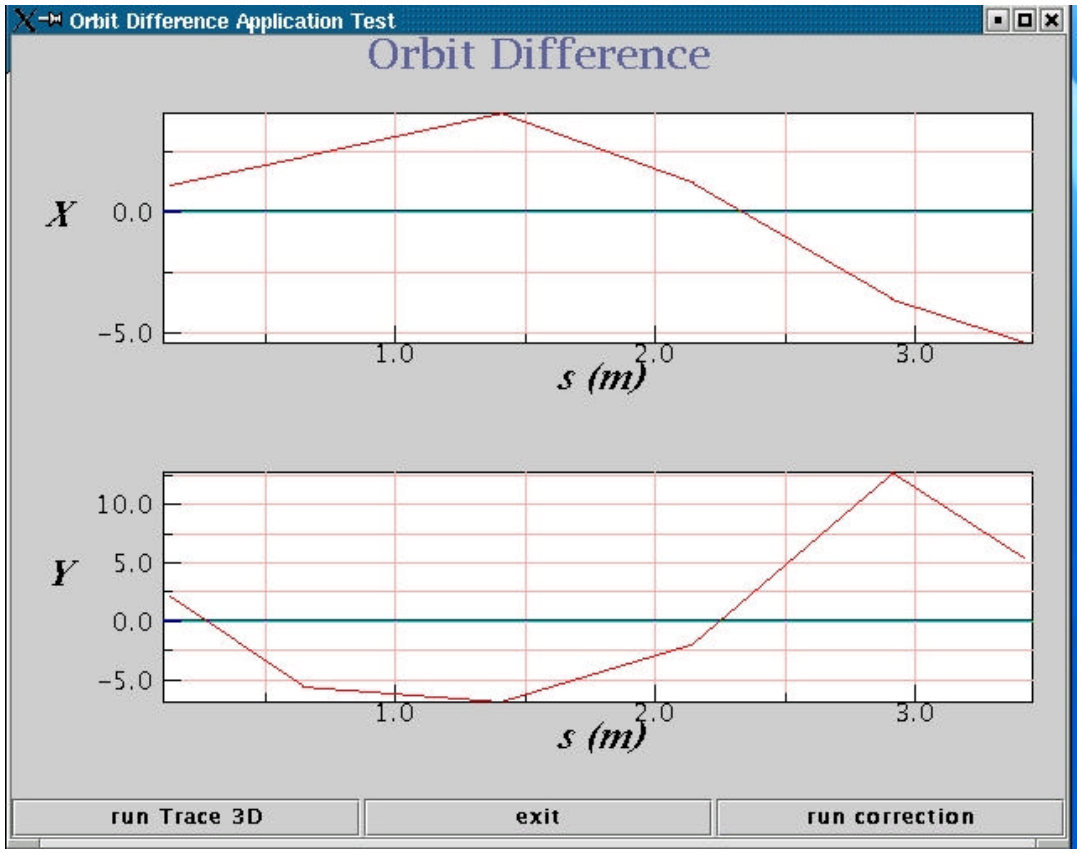
- XAL is a new programming infrastructure to provide a hierarchical device oriented view for application programmers
- Using Java - Interfaces exist to EPICS (java ca) and to database (extensible markup language -XML- file)

XAL Programming Infrastructure Provides Hierarchical Accelerator View



- Class structure that provides a hierarchical “device” view of the accelerator to the application programmers
  - Sequences (MEBT, DTL, CCL, …)
  - Individual Node type classes (Quads, RF, BPMs, etc.)
- Online documentaion: http://sns.gov/APGroup/appProg/xalDoc/

XAL Application: Model – Machine Comparison (*P. Chu*)



MEBT application

Red = BPM,

Blue = model

- Compare difference between two pulses
  - Observe effect of a magnet change
  - For both BPM signals and Model (Trace-3D)
- The beamline-device initialization is from the global database
- Using virtual accelerator to generate machine signals now

Example of Scripting with XAL

- Java compatible python script is tested

```
from xal.smf  import *

# read the accelerator
bldr= XdxfBuilder()
acc=bldr.read("file:./sns_mebt.xml", 0)

# Get the MEBT:
mebt = acc.getSequence("MEBT")

# get the MEBT quads
collect= mebt.getNodesOfType("quadrupole")

nmqs= collect.size()
for i in range(nmqs):
    node = collect.get(i)
    print i, " ", node.getId(), " ", node.getLength(), ' type = ', node.getType()
```

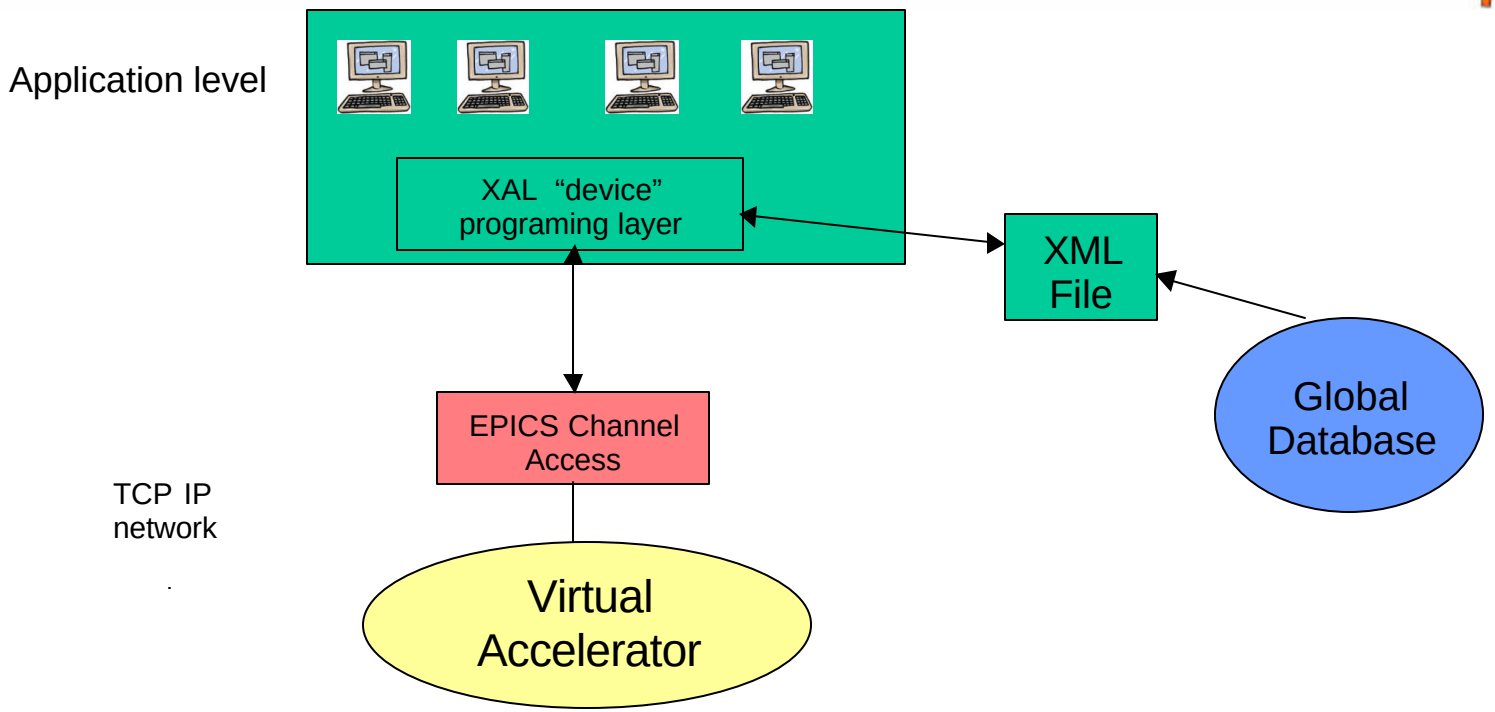
No interface code needed, directly import Java classes!

Get accelerator from XML file

Extract MEBT Sequence

Get MEBT quads

Virtual Accelerator (*A. Shishlo*)



- The virtual accelerator simulates accelerator behavior with a model
- An EPICS channel access server provides an authentic controls system interface
  - The real control system signal names are used
  - Magnet and RF settings, and BPM readbacks etc. are available

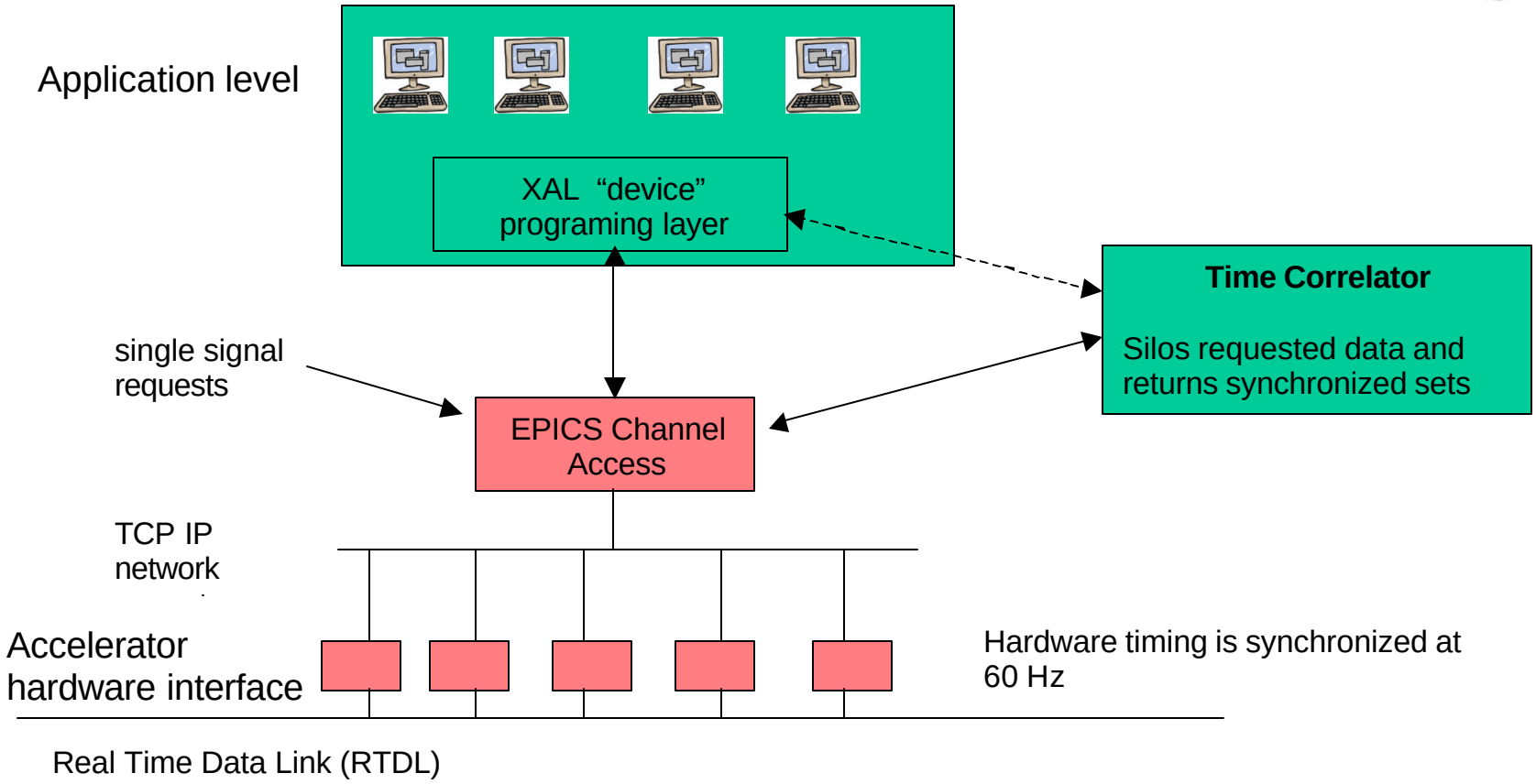
MEBT Virtual Accelerator (*A. Shishlo*)

- Presently two versions exist
  - Trace-3d: fast envelope
  - Parmila: particle tracking (slow) but also simulates BPM responses to arbitrary bunch shapes

- Multiple client applications are being tested with this

- Other beamline sequence implementations are underway

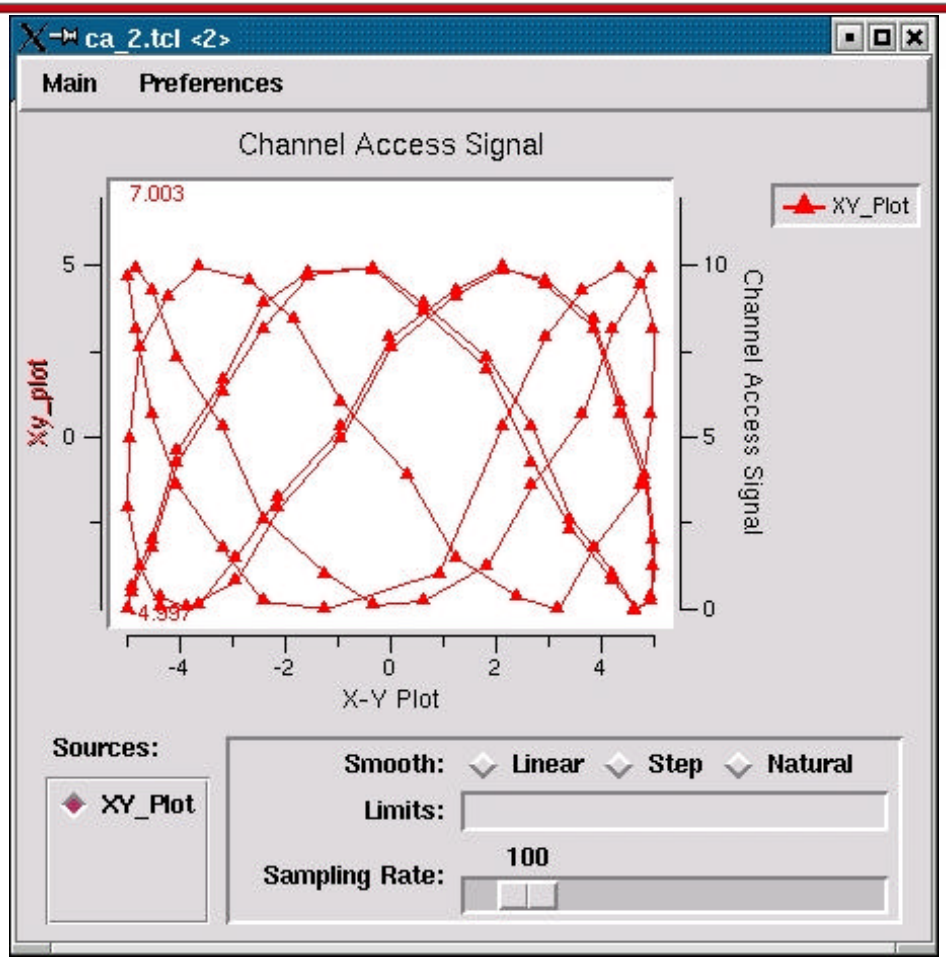
Time Correlator Provides Synchronized Data (*P. McGehee*)



- All data is time-stamped
- Correlator is a callable routine to return sets of EPICS Process Variables from the same macropulse

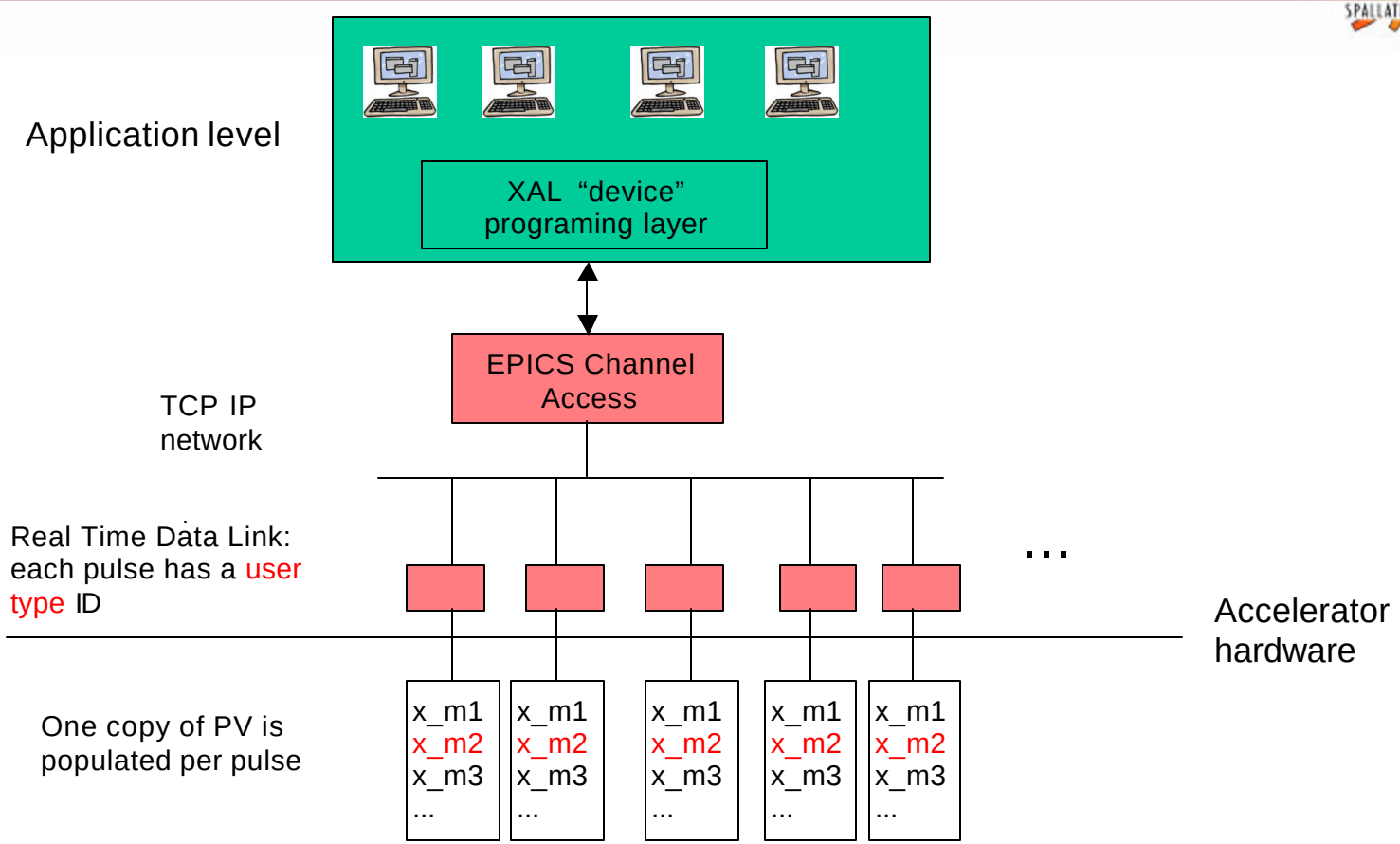
Time Correlator Application (*P. Chu*)

•Specify PV's for correlation



- C++ dataSilo class gathers data from different IOCs and finds the most recent time synchronized data set
- Correlator engine tested with up to 24 PVs at 10 Hz

Multi-mode Signals

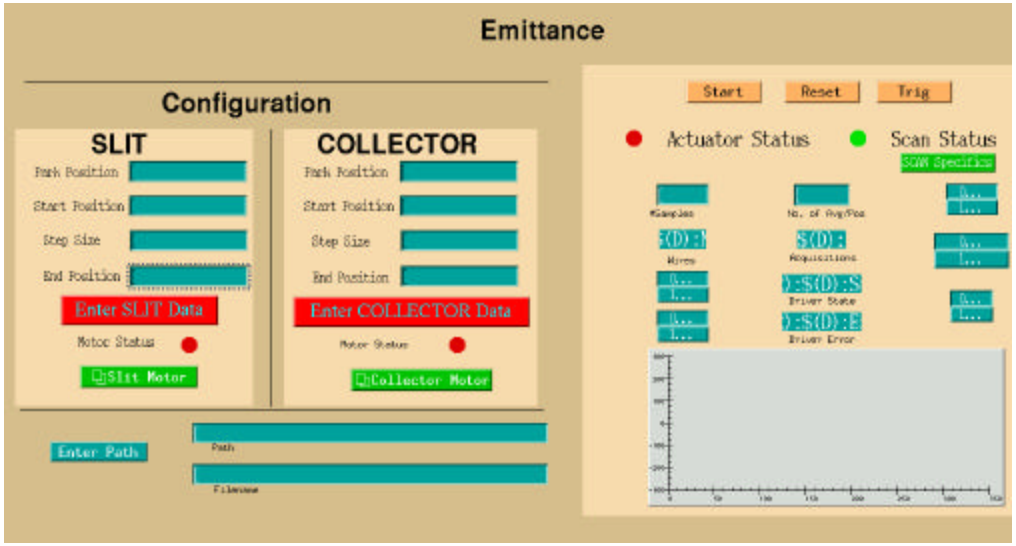


- Diagnostic and RF information have multiple copies of process variables (PV)
- User type value triggers a single PV each macro-pulse
- Allows interspersing occasional "physics-mode" pulses in a stream of "production mode" pulses, and reliably getting this information

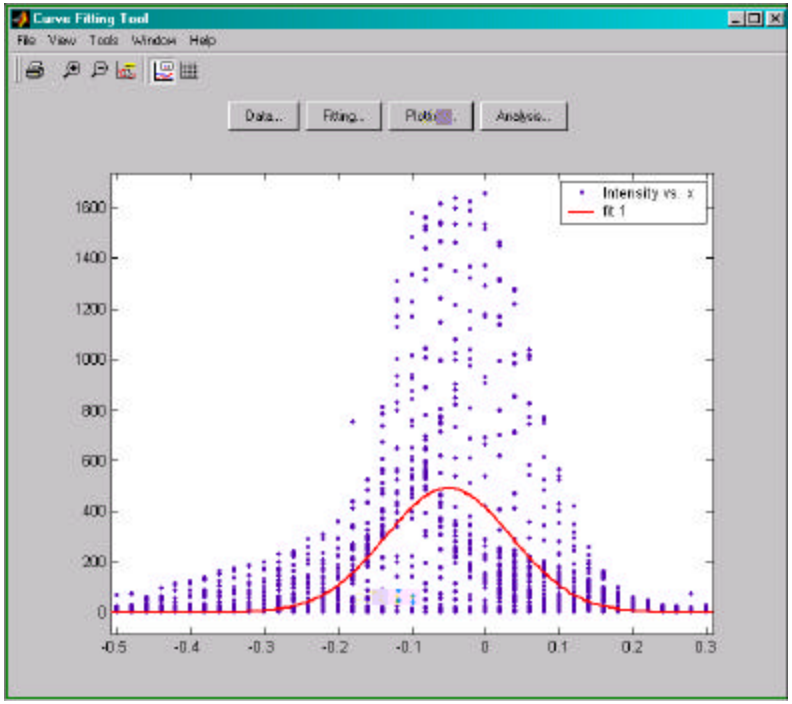
Front End Commissioning Plans

- Front End commissioning has begun at LBNL
- Use MATLAB scripted applications for some of these
  - Will have hardwired signal names etc.
- ORNL will provide application programs
  - Emittance measurement
  - RF phase settings
  - Quad setting
  - Correctors setting
  - Trace-3d model

RFQ Emittance Measurement Application using MatLab



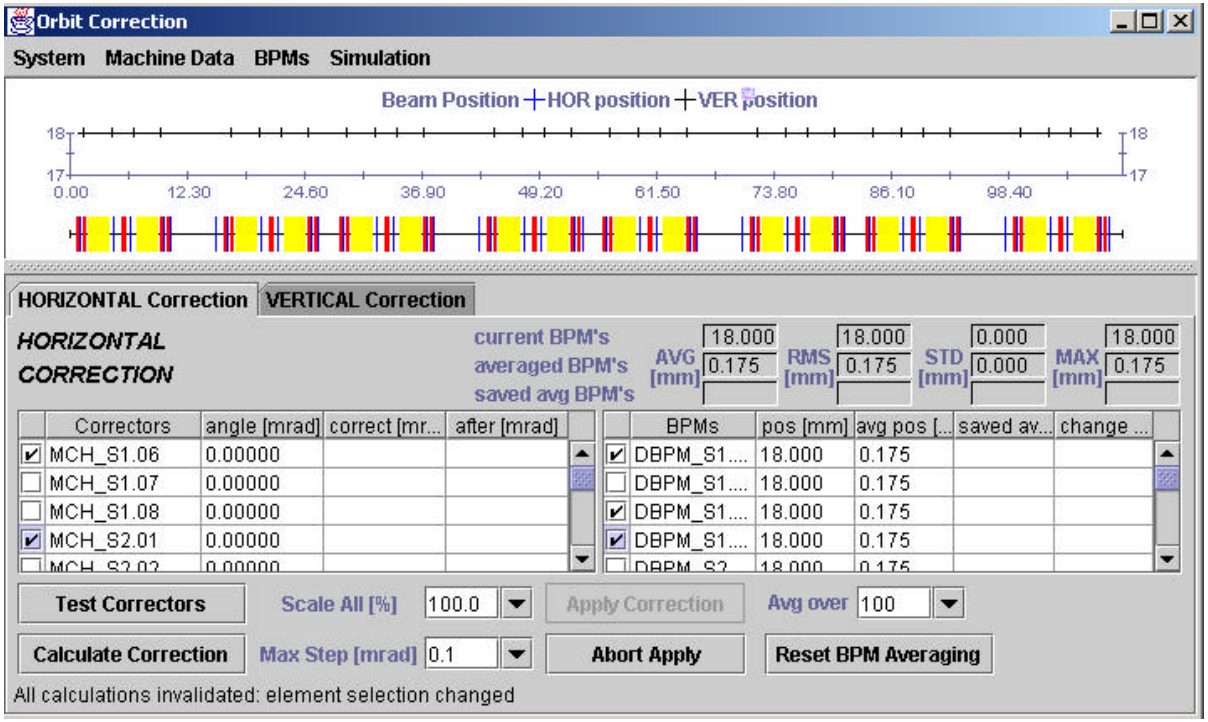
Emittance Application screen layout



Intensity vs. position + fitting

- Being prototyped at LBNL now, S. Assadi (diagnostics), D. Nypaver, E. Williams (Controls)

Collaboration with Cosylab



Example cosylab correction application

- Cosylab produces accelerator applications used ANKA, DESY and Riken
- They have a java based accelerator hierarchy similar to ours, as well as “beans” to quickly create applications
- We have a contract with Cosylab to test the viability of using their tools with our infrastructure

Staff Planning

- Required staff levels have been estimated based on commissioning plans, operational needs and solicited comments.
  - Over 60 applications + general support tools
- Priorities and date required are also compiled
- Time estimates based on 2 prong approach
  - First cut applications for commissioning will not be polished
  - After determining which algorithms work, polished user interfaces will be constructed
- See http://www.sns.gov/APGroup/appProg/AppProgPlan.pdf

Staff Additions – a core team is formed

- Applications programming effort is a collaborative effort between Accelerator Physics, Controls and Diagnostics
- Dedicated personnel: 4 FTE ORNL accelerator phys. + 2 FTE LANL + 1 FTE BNL
- Additional support from diagnostics, controls, and project controls (database).
- Initiated a weekly video-conference.
- Have a viable group now.

Summary

- Database put in place, MEBT parameters being used for testing
- Java based programming infrastructure picked
  - Database and EPICS connections work
  - Initial application ready for testing at LBNL
- Virtual Accelerator being used for preparing applications
- Staff level significantly improved over the past six months
  - Doubled ORNL app programming team
  - Help from staff at LANL and BNL, as well as diagnostics and controls