XAL – An Application Programming Framework *(Java based!)*

ICALEPS2003

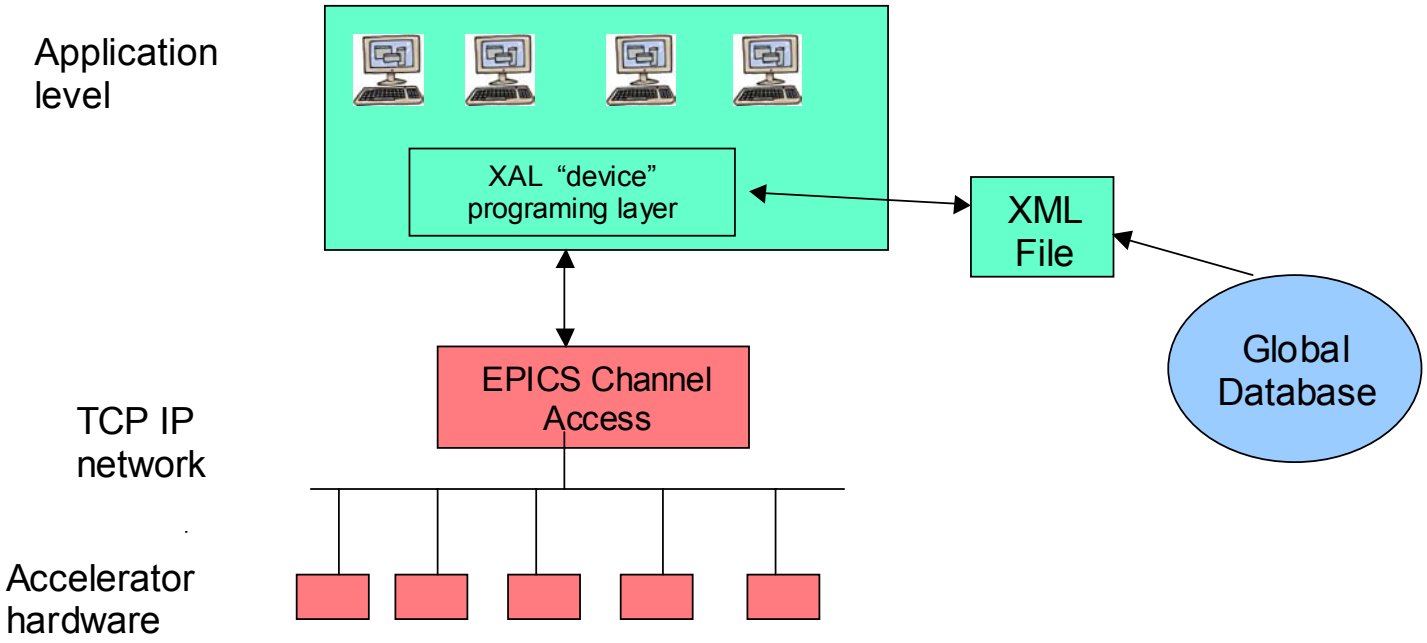
Gyeongju Korea, Oct. 13-17, 2003

J. Galambos, C., M. Chu, W. D. Klotz, T. Pelaia, A. Shishlo (SNS, ORNL), C. K. Allen, C. McChesney (LANL), I. Kriznar, M. Plesko, A. Pucelj (Cosylab)

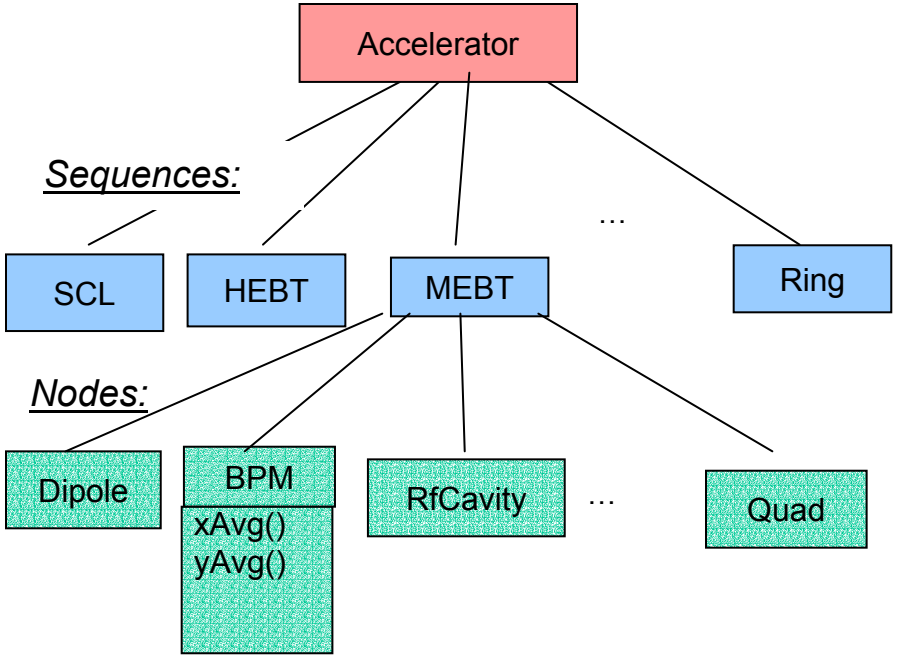
D. Attavio (BNL)

Outline

- XAL
  - Accelerator hierarchy
  - Tools
- Connections
  - EPICS Control System
  - Database
- Applications

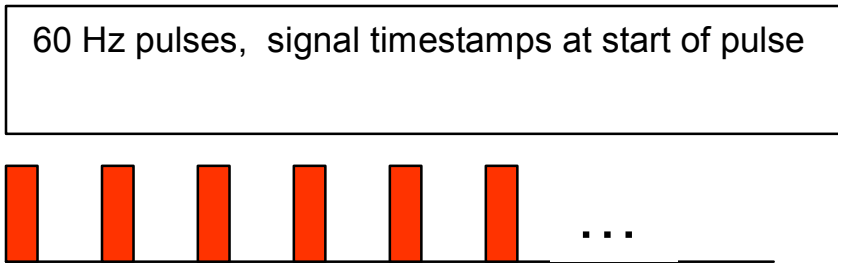


XAL Accelerator Hierarchy



- XAL is Java based
- Includes a class hierarchy describing the accelerator structure
- Methods exist to directly work with accelerator devices

Tools

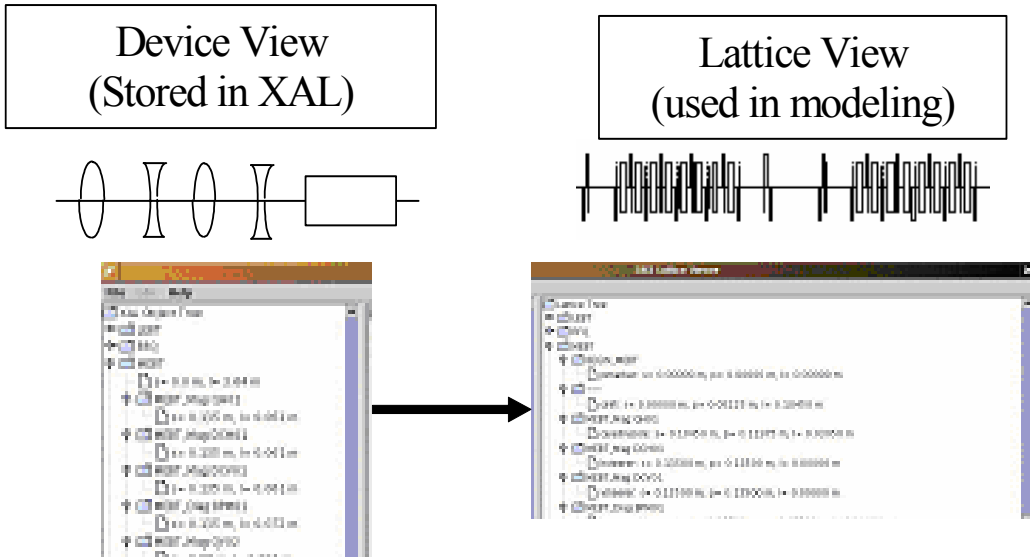


- Channel Correlator
  - SNS is a pulsed device
  - Correlator gathers sets of signals from the same pulse
  - Different modes: stream correlation sets to listeners or periodic posting
  - Can use filters (triggered data acquisition)

- DataTable
  - Provides database like functionality with easy interface
- XML DataAdaptor
  - Easy file i.o. in XML format, for any hierarchy

The Online Model (WE116: A Novel Online Simulator for High-Level Control Applications Requiring A Model Reference, C.K. Allen, et. al.)

- Calculate beam parameters based on machine settings in real time
- A lattice view of the machine is constructed from the “device” structure (via a set of rules
  - Drifts are added, elements are split
- Lattice element values can be updated from the machine, user defined values or design values
- Mostly use an envelope model for linac tracking at present



- Single entry per element

- Only physical devices

- Elements may be split

-Includes drifts

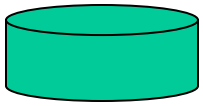
The Control System Connection

- Channel Class
  - An abstract class that provides an interface to the control system
- JcaChannel
  - A concrete class that uses the JCA interface to EPICS channel access protocol.
- Transformations: allows modification of the value of the signal coming from the control system
  - Can map one signal to > one “Channel” object, with and without the transformation
  - Facilitates quick fixes, changes to signals
    - Magnetic field calculated from current
    - BPM polarity fixes

The Database Connection (WE113: "Initial Experience with SNS Database Application", D. Purcell et. al.)

- XAL uses "Beamline Device" tables in the global SNS database
- Creates the accelerator hierarchy
- Serves as a common configuration file for all applications
- Provides the "map" from the flat list of EPICS signals to the accelerator hierarchy
- Use an intermediate XML file for XAL initialization

Database ⟶ XML configuration files



```
<node type="QH" id="MEBT_Mag:QH01" pos="0.135" len="0.061">
  <attributes>
    <align x="10000" y="2000.046" z="19546.440844" pitch="0" yaw="0" …/>
  </attributes>-
  <channelsuite name="magnetsuite">
    <channel handle="fieldSet" signal="MEBT_Mag:QH01:fieldSet" …/>
  </channelsuite>
</node>
```

•On-the-fly editing for quick across the board fixes

http://ntser3.cad.ornl.gov/mis/EnterpriseModel/modell.htm

Scripting Interface to XAL

- Scripting interfaces are available with Jython (www.jython.org) and Matlab
  - No glue code, or extra compile steps etc needed!!! Mix XAL java classes seamlessly with scripting language
- Have written simple applications, but the main use is in providing simple code examples
- http://www.sns.gov/APGroup/appProg/xal/scripts/jythonScripts.html

Jython

```
# read the accelerator
#acc_xml = "file:/home/jdg/xaldev/xal_xmls/sns.xml"
acc_xml = "file:./sns.xml"
acc = XmlDataSource.parseUrlAt(acc_xml, 0)

# get the some primary sequences from the accelerator

mebt = acc.getSequence("MEBT")
dtl1 = acc.getSequence("DTL1")
dtl2 = acc.getSequence("DTL2")

print " There are ", mebt.getAllNodes().size(), "nodes in the
sequence", mebt.getId()
```

MatLab

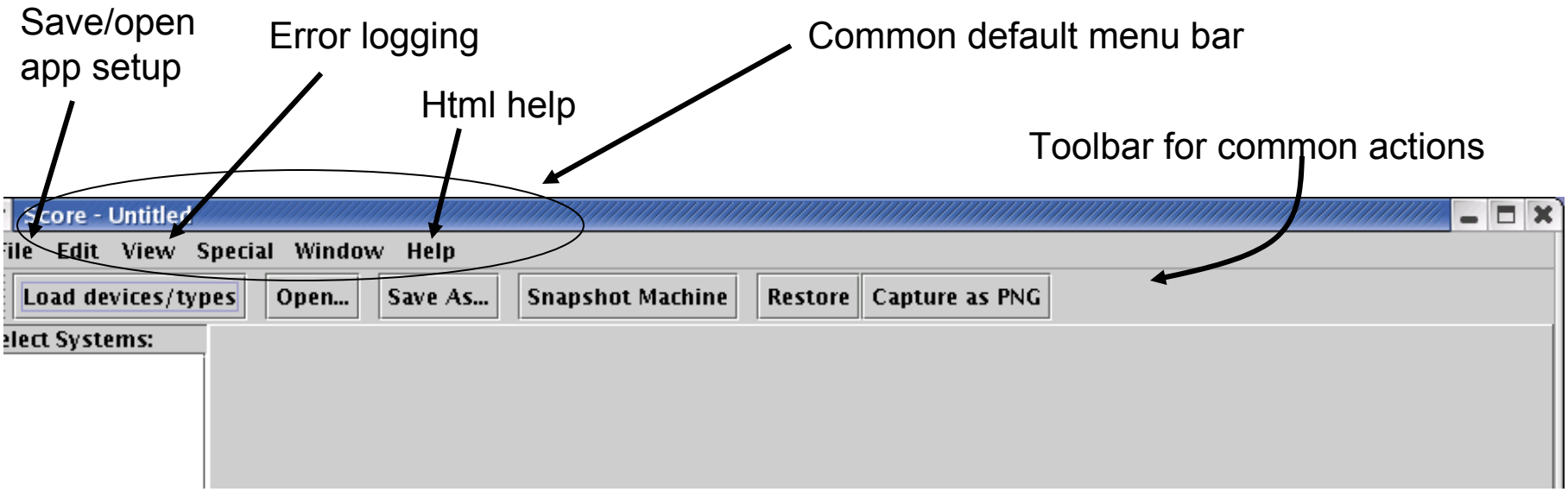
```
%  scan the first quad
% monitor beam positions in the last MEBT BPM

for i =1:10
      fld(i) = field;
            quad.setField(field);
            va_chan.putVal(1); %  for virtual accelerator
            pause(1); %  for virtual accelerator
            xpos(i) = bpm.getXAvg;
            ypos(i) = bpm.getYAvg;
            field = field * 1.015; % increment field value
end

% Plot results
plot(fld, ypos)
```

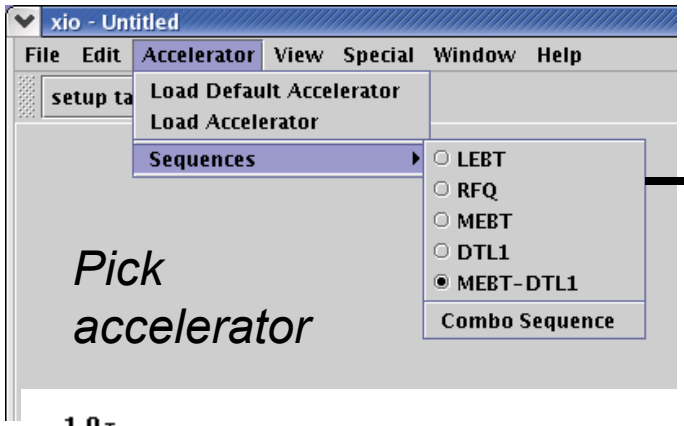
Application Programming Framework

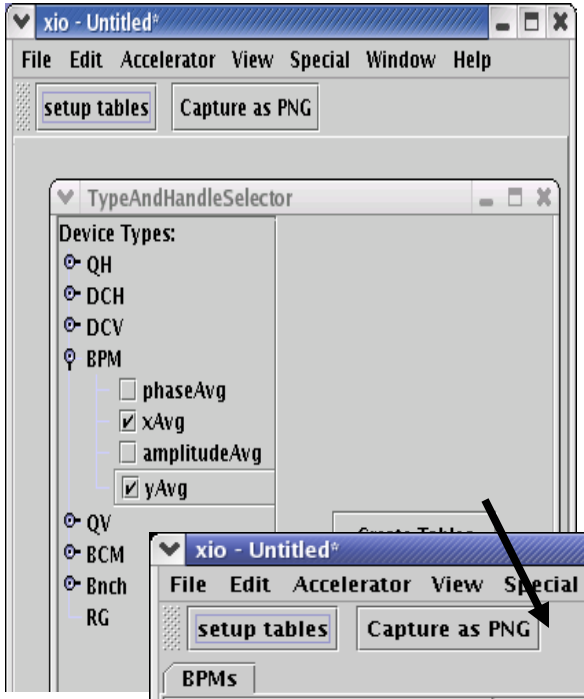
- An Application Framework is developed and used as a common starting point for application programs
  - Provides a common look feel for all apps
  - Quick jump-start for application development
  - Easy retro-fixes across many apps
  - Use familiar “windows” app look & feel
  - Uses a document paradigm



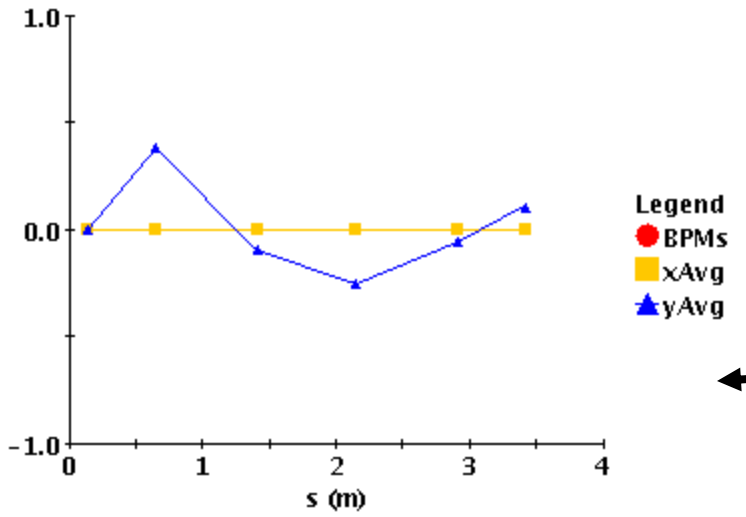
Xio Application

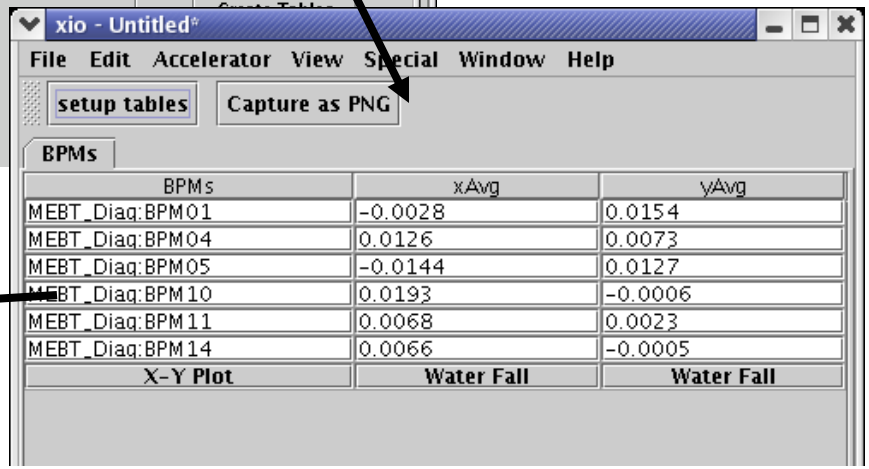
- General purpose value displayer (tables, and or plots)
- Browser the accelerator hierarchy to select what you are interested in





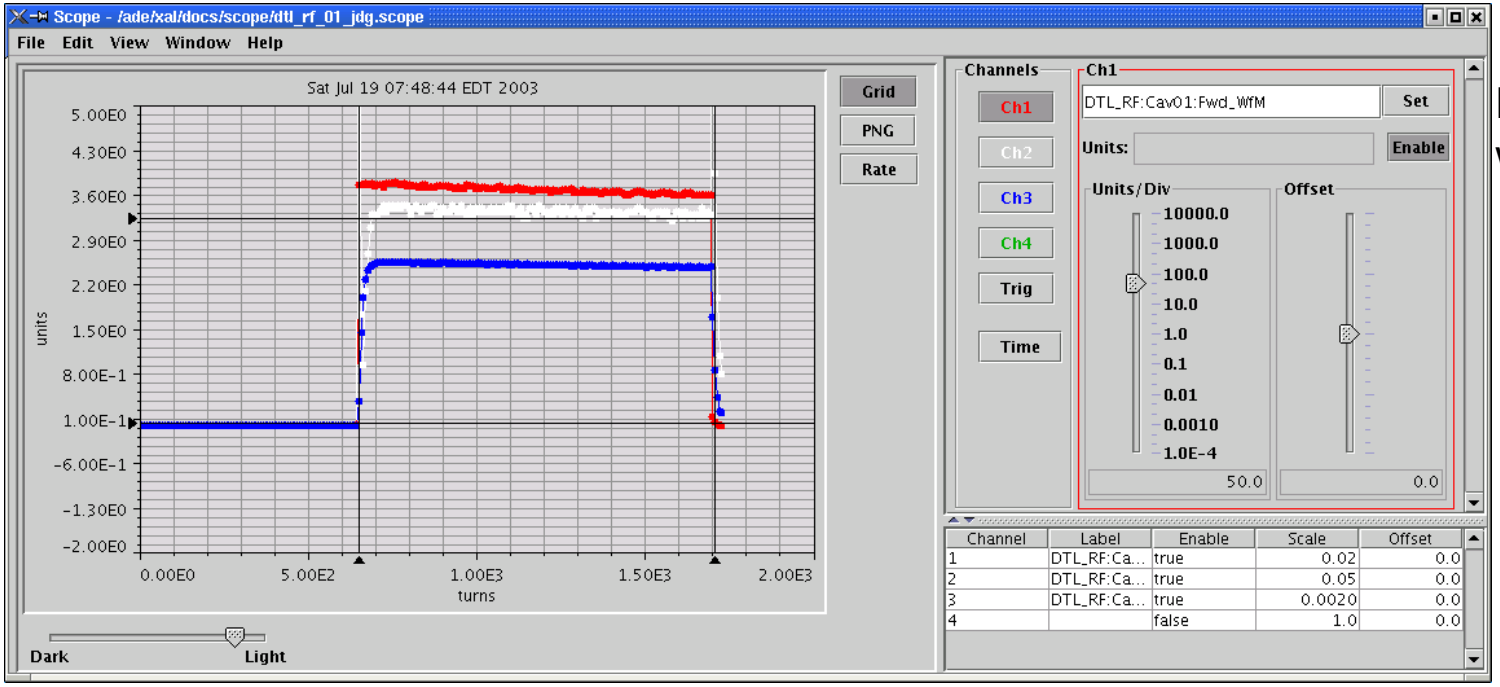
Pick device + signal types





Scope Application

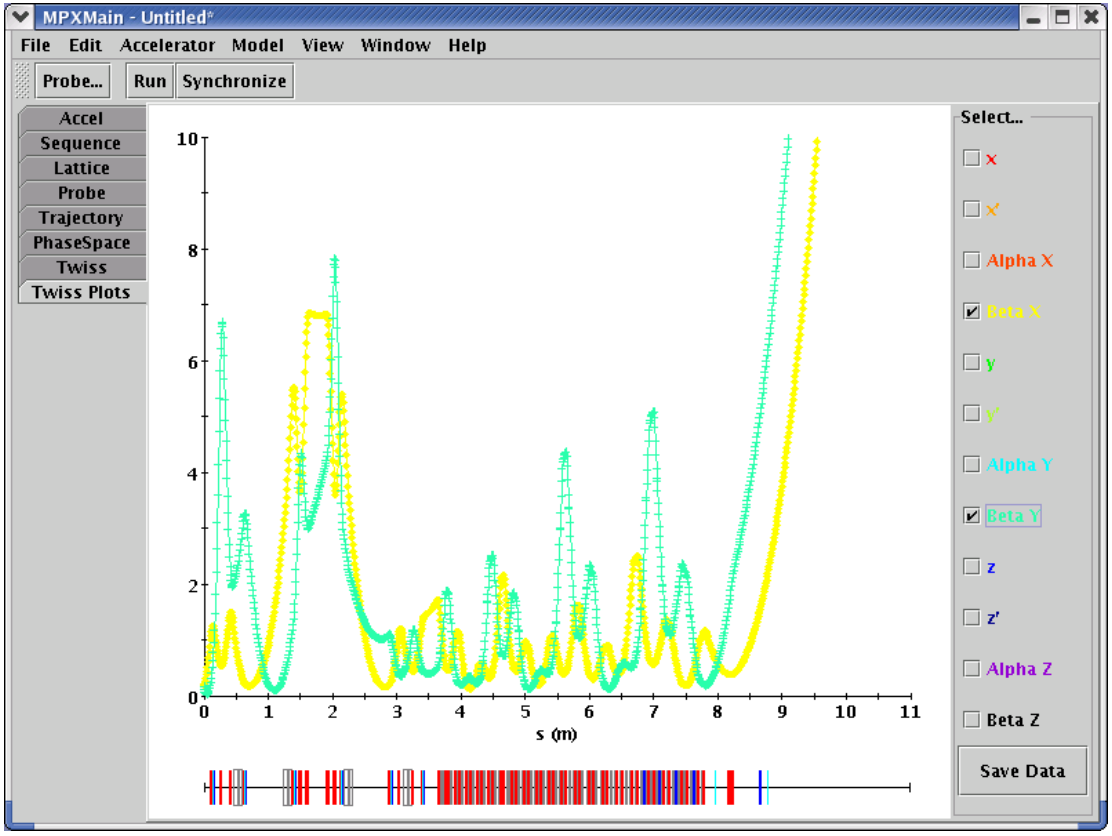
- A Digital Oscilloscope – with a similar user interface as analog scopes
- Displays array waveforms vs. time (NOT vs arbitrary units)
- Uses the time correlator, has built-in math capability, triggered data acquisition+ many other features
- Will use this for comparing waveforms from RF, diagnostics, etc.
- Requires input from signal providers describing how the array information is packaged, and offset from the cycle start



DTL1 RF waveforms

Online Model

- An online model is now available within the XAL framework
- Online model can be run for 1) live values, 2) design values and 3) user defined what-if changes
- Can display or dump beam Twiss output

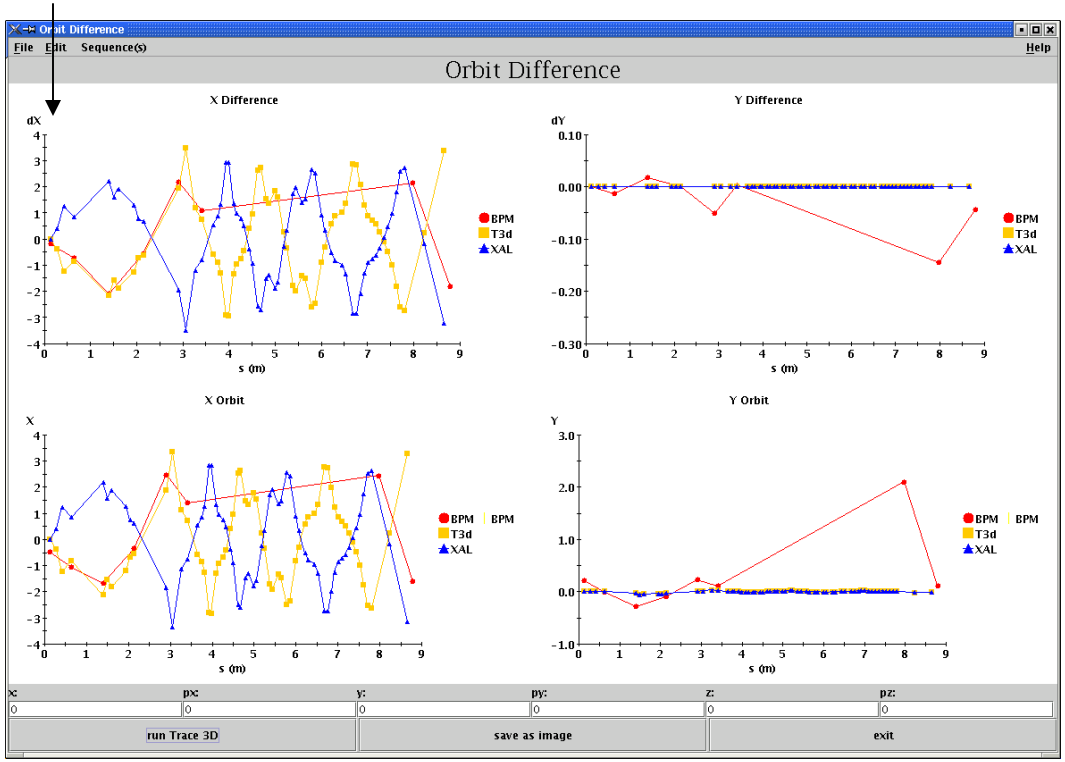


•Vertical and horizontal beta functions through the MEBT, DTL + D-plate for design values

OrbitDifference Application

- Compares differences in beam orbits, for both BPMs and calculated
- Online model is also used in the Orbit Difference Application, in addition to running Trace 3D (external fortran code)

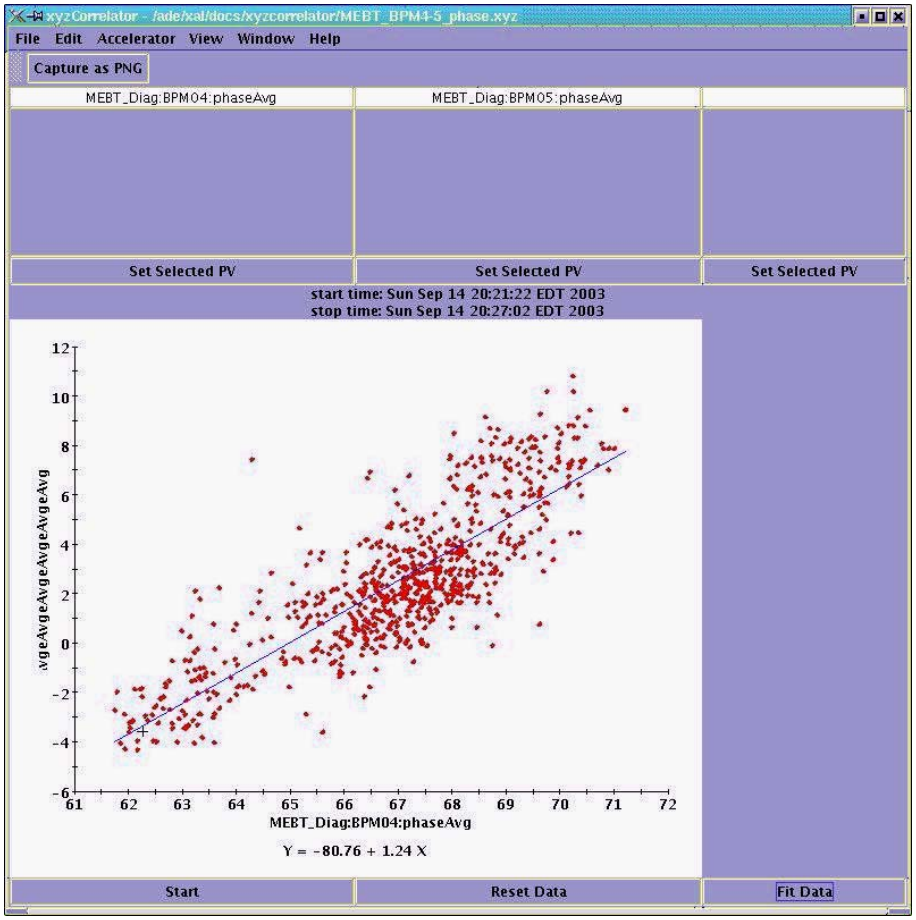
Kick applied here



- Orbit difference example using the online model
- Used to observe orbit difference in the horizontal direction
- Helped resolve sign issue in BPMs

XYZ Correlator Application

- Pick 2 (or three) signals and monitor them together
- Uses the time correlator to ensure signals are from the same pulse
- Can export or fit the acquired data

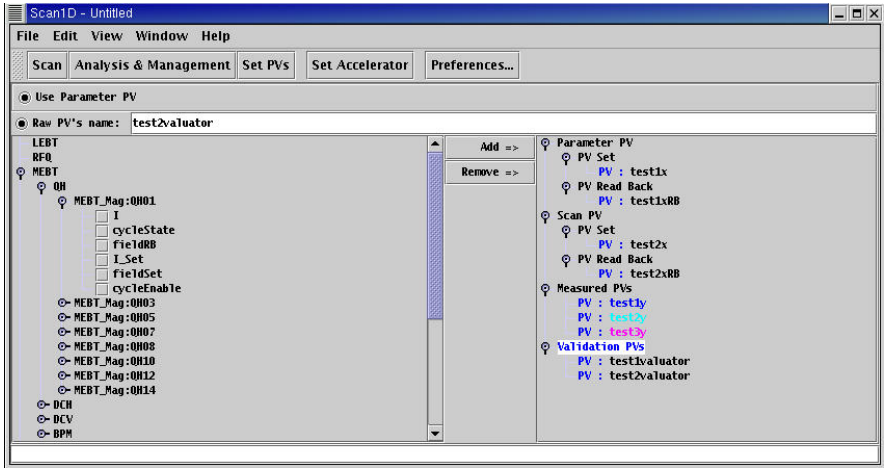


Correlation between phase measurements of 2 BPMs

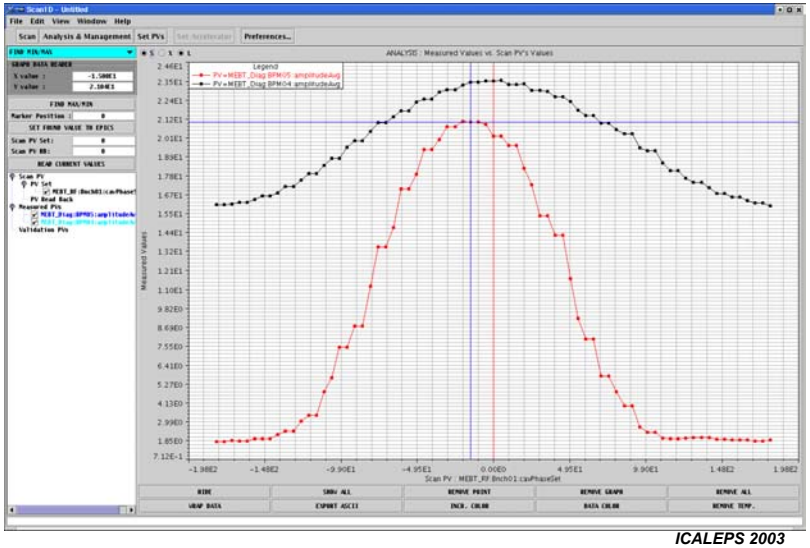
1-D Scan Application

- Provides an easy way to scan one quantity and monitor another
- Can average over pulses, scan multiple times, pause
- Analysis includes fitting, intersection finding, min/max, etc.
- Easy way to do a quick unanticipated experiment

•Can select scan or readback signals from pull down Accelerator hierarchy

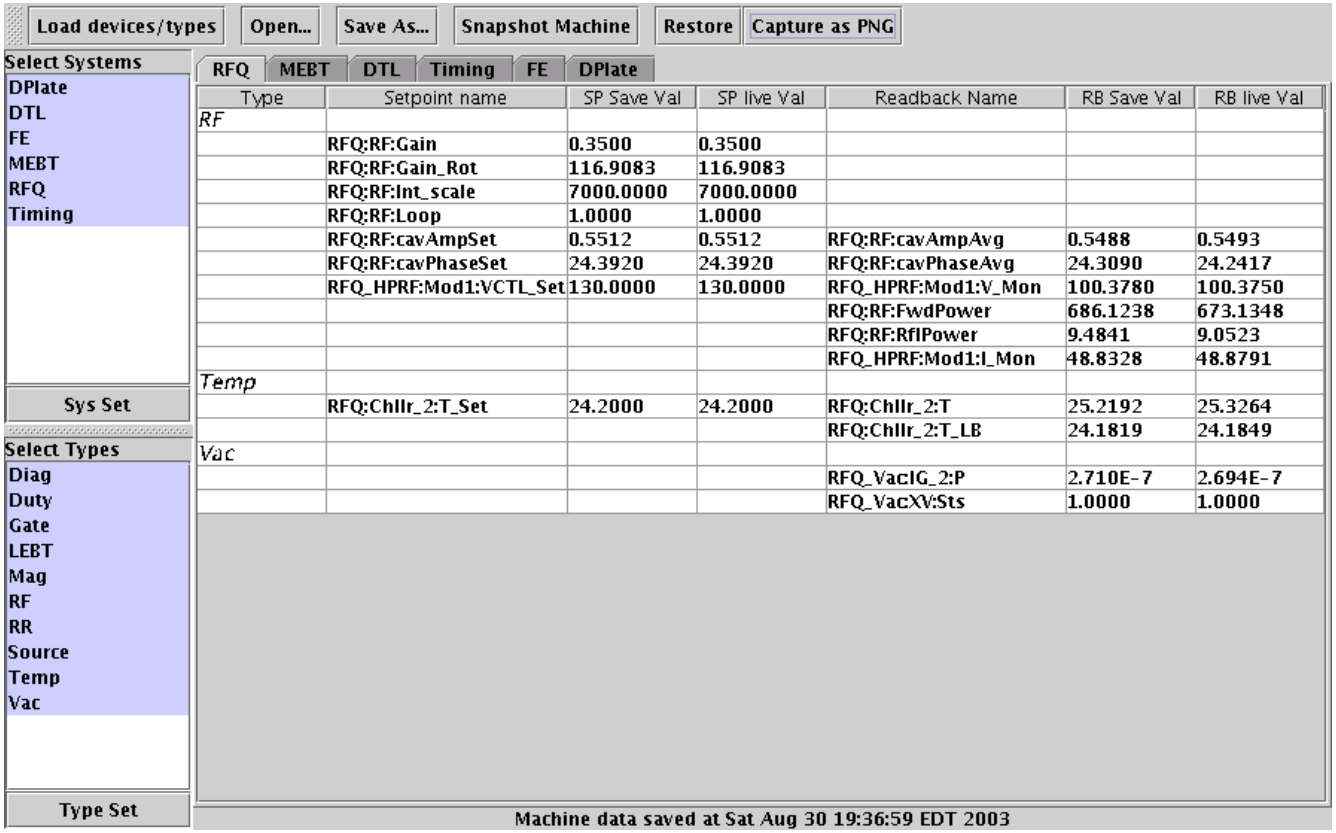


•BPM amplitude vs. MEBT RF phase setting



Save-Compare-Restore (Score) Application

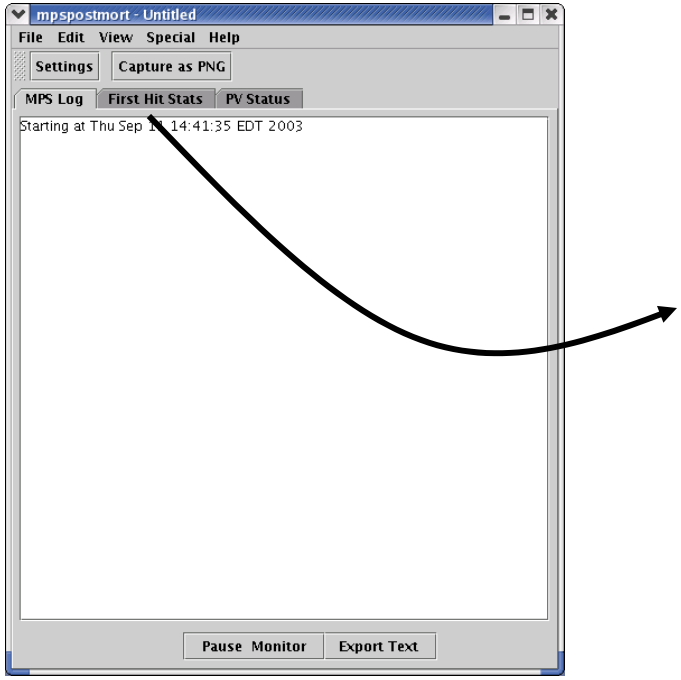
- Provides a means to capture machine setup, compare live values to a saved set, and to restore values to a saved set
  - Grabs settable + readback signals
- Can sort by system and device type
  - Uses DataTable classes for querying

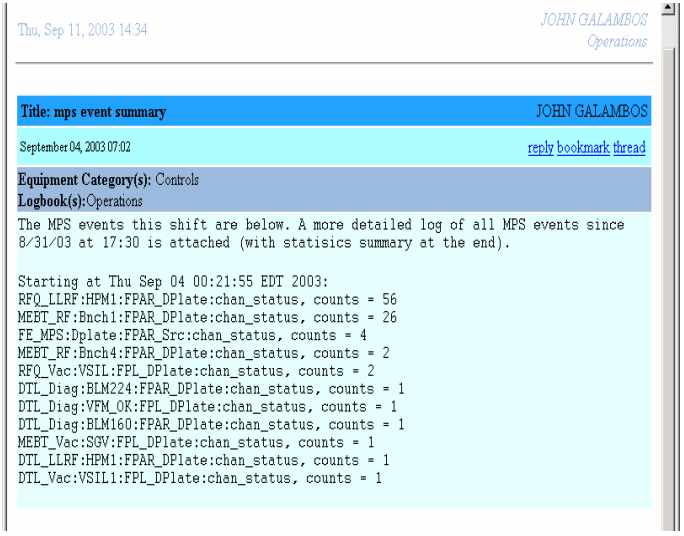


Type	Setpoint name	SP Save Val	SP live Val	Readback Name	RB Save Val	RB live Val
RF						
	RFQ:RF:Gain	0.3500	0.3500			
	RFQ:RF:Gain_Rot	116.9083	116.9083			
	RFQ:RF:Int_scale	7000.0000	7000.0000			
	RFQ:RF:Loop	1.0000	1.0000			
	RFQ:RF:cavAmpSet	0.5512	0.5512	RFQ:RF:cavAmpAvg	0.5488	0.5493
	RFQ:RF:cavPhaseSet	24.3920	24.3920	RFQ:RF:cavPhaseAvg	24.3090	24.2417
	RFQ_HPRF:Mod1:VCTL_Set	130.0000	130.0000	RFQ_HPRF:Mod1:V_Mon	100.3780	100.3750
				RFQ:RF:FwdPower	686.1238	673.1348
				RFQ:RF:RflPower	9.4841	9.0523
				RFQ_HPRF:Mod1:I_Mon	48.8328	48.8791
Temp						
	RFQ:Chllr_2:T_Set	24.2000	24.2000	RFQ:Chllr_2:T	25.2192	25.3264
				RFQ:Chllr_2:T_LB	24.1819	24.1849
Vac						
				RFQ_VacIG_2:P	2.710E-7	2.694E-7
				RFQ_VacXV:Sts	1.0000	1.0000

Machine Protection System (MPS) Post-mortem Application

- Captures MPS events, and sorts the signals in order of occurrence – i.e. determines the root cause of a trip (uses correlator)
- Logs mps events
- Provides statistics





Summary

- XAL is a Java based hierarchal framework SNS is using to write high level applications
- Uses a database to initialize the hierarchy
- Applications are written and being used in the initial SNS commissioning