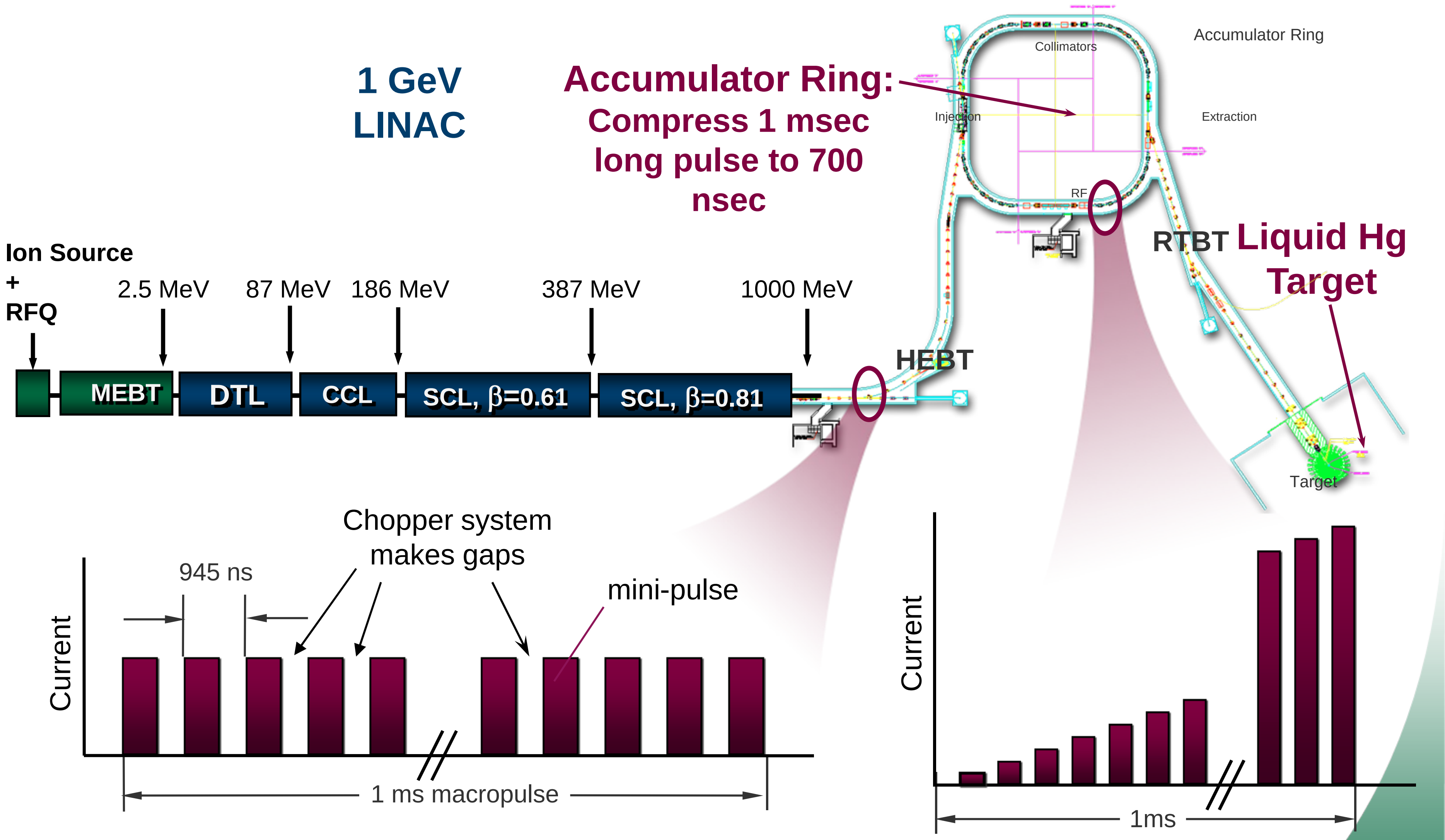
SNS Beam Commissioning Tools and Experience

Andrei Shishlo on Behalf of SNS Team
HB2008, Nashville, TN
August 27, 2008

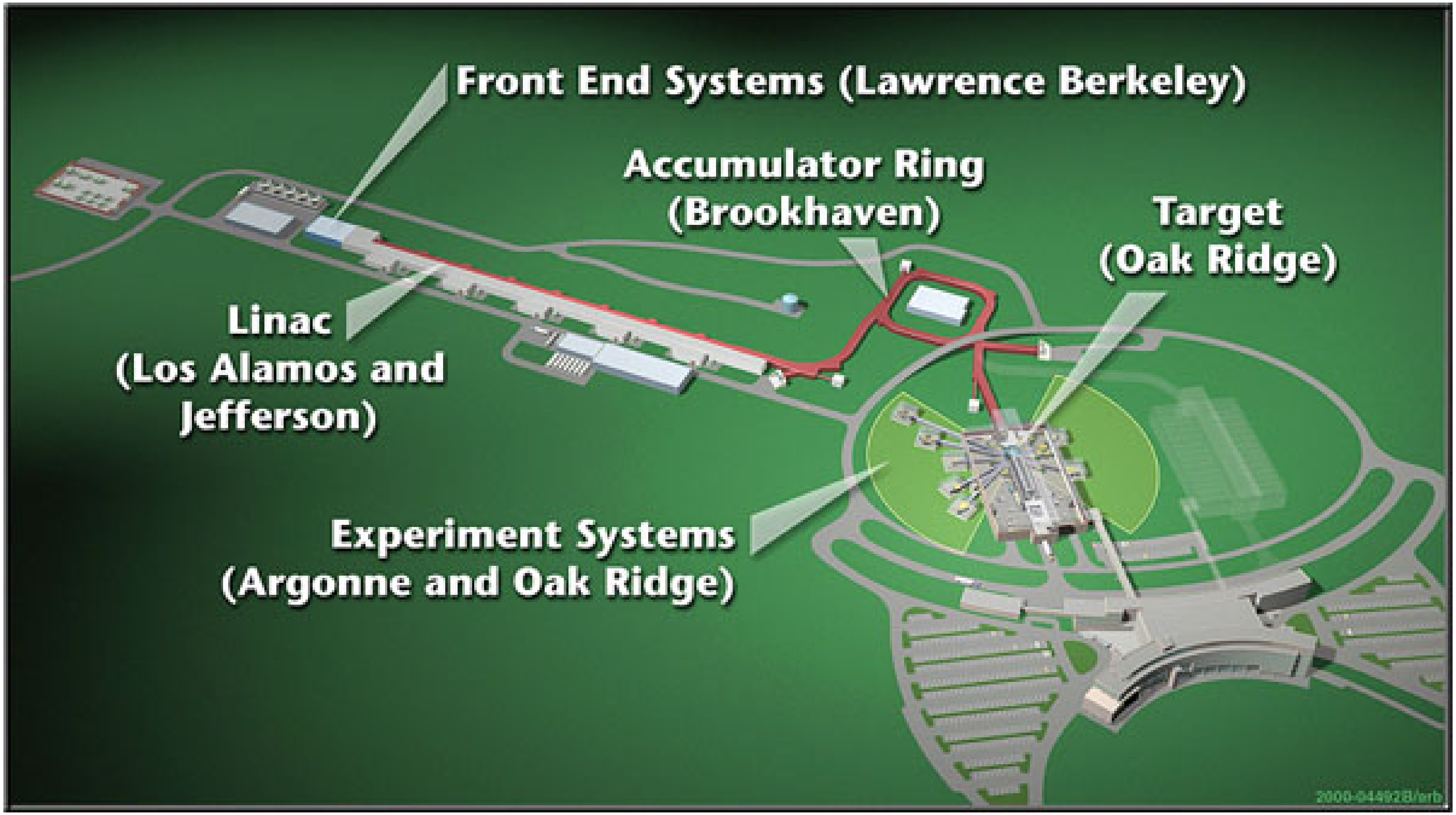
Outline

- **SNS Accelerator Complex**
- **Commissioning and Tools Development Timeline**
- **XAL Structure and Most Useful Applications**
- **Conclusions**

SNS Accelerator Complex

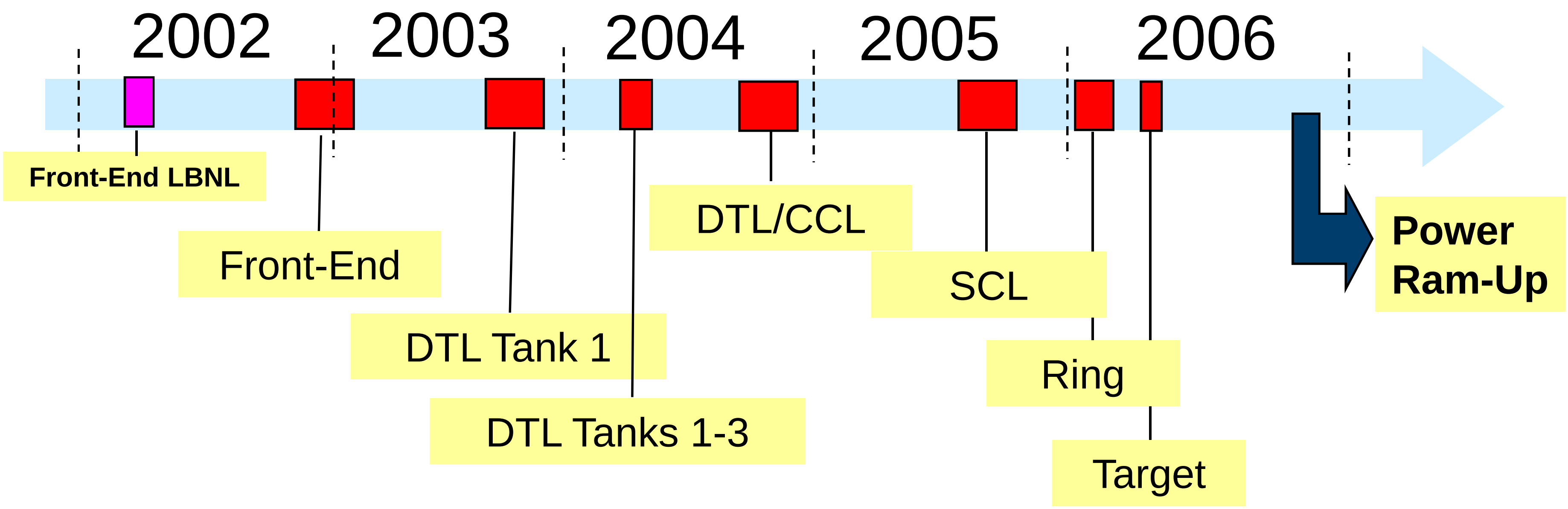


SNS as Collaboration



Accelerator components provided by LBNL, LANL, JLab, ANL and BNL

SNS Beam Commissioning Timeline



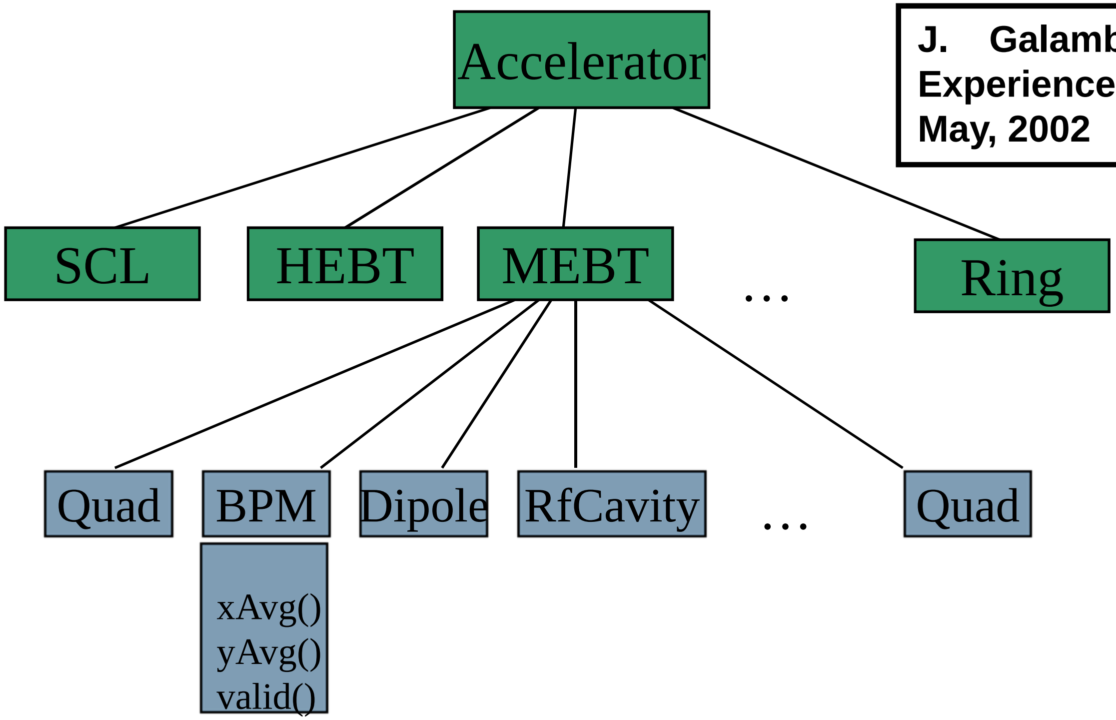
- **Commissioning was squeezed between Installation activities.**
- **Try-and-learn iterations approach to software applications development**
- **Much less time was available for beam commissioning than originally planned.**
- **Pace of commissioning accelerated at the end**

Application Programming Beginning (2000-2001)

- The different technologies were reviewed: FORTRAN applications, MATLAB, SDDS (Self Describing Data Sets), Cdev, Java
- Java
  - Advantages: simple, object oriented, it runs everywhere, GUI, database interaction, client/server application, Java interface to EPICS CA existed, appeal to young physicist/developers
  - Disadvantages (at that time): graphics (contours, error bars, real-time, 3-D, …), mathematical libraries less mature, most AP members used MatLab
- Application programming requirements was formulated, a list of programs was constructed, manpower needed is 43 FTE (Full Time Equivalent) for 3.5 years of commissioning, accelerator physics, controls, and diagnostics groups are involved
- Two versions of applications: for commissioning and for operations. Commissioning versions are streamlined applications with minimal user interface
- The Application Programming Team was created inside Accelerator Physics Group to start development of Java infrastructure and high level physics applications

MEBT and DTL commissioning: MatLab prototypes of some of applications were written first by AccPhy group members, and then they were rewritten in Java to insure a successful commissioning

XAL – A Java based high level programming infrastructure for physics applications



J. Galambos, "SNS Remote Operations Experience + Thoughts on Using Java" talk, May, 2002

- Java class structure that provides a hierarchical "device" view of the accelerator to the application programmers
- Setup from database through XML file, EPICS connections hidden

Other similar frameworks

- Based on UAL2 (http://www.ual.bnl.gov/)
- Cosylab Abeans / databush (www.cosylab.com)

First Test - Remote Testing of Applications

Front End at LBNL

Test control room at ORNL

- **The SNS MEBT was commissioned at LBNL April-May 2002.**
- **3 slots for testing, 5 hrs total beam time, Tested model comparison, orbit correction + general purpose diagnostic app.**
- **Application testing before commissioning is a valuable option**

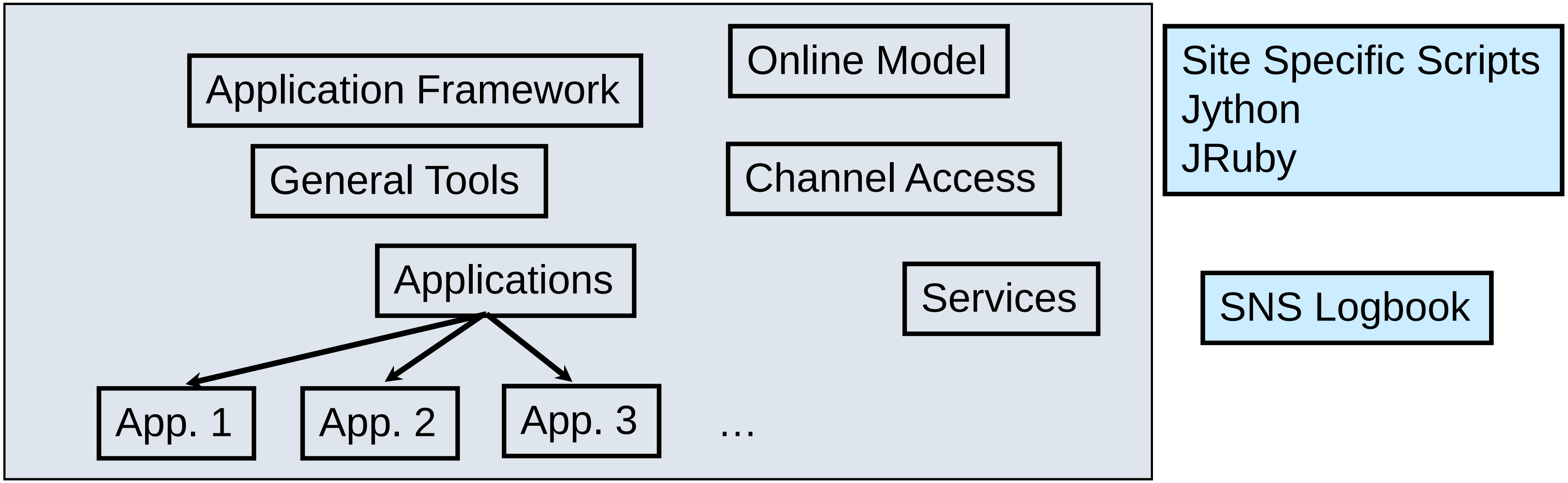
Lessons after First Steps

- **Need to familiarize people with application features before commissioning.**
- **Need GUI interfaced applications for general users.**
- **Have integrated help capability, common look/feel**
- **Testing with Virtual Accelerator before commissioning helped**

Actions:

- **The practice of live lessons for applications become a common practice**
- **The development of the Application Framework initiated**
- **Proceed with the Virtual Accelerator development**

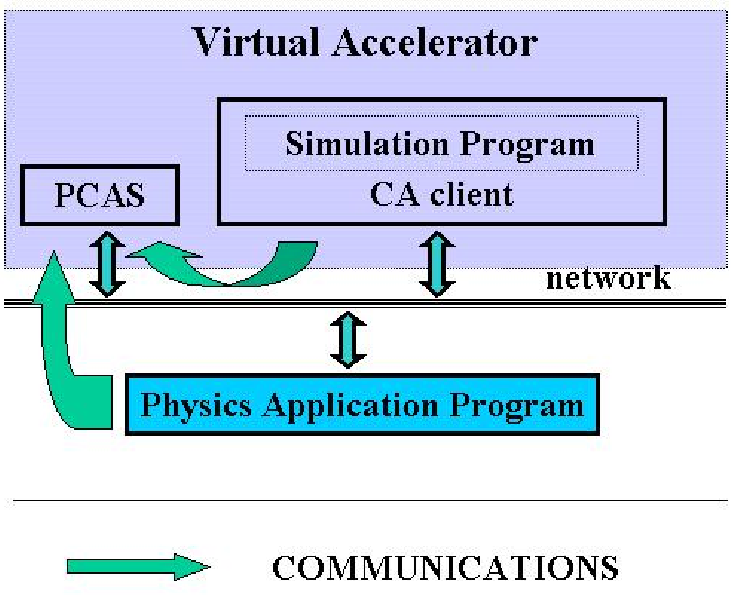
XAL Structure



- **Online Model: simulates charged particle dynamics through specified accelerator sequences; six dimensional phase space propagation; includes space charge**
- **Application Framework: consistent look and feel; standard, familiar menu items; free automatic behaviors; rapid application development**
- **Channel Access: package abstracts channel access; provides some insulation from API changes to underlying access layer**
- **Services: run continuously in the background; provide remote communication with user interfaces**
- **General Tools: solvers, plotting, math etc.**

Web page of Tom Pelaia (XAL project leader):
http://www.ornl.gov/~t6p/Main/XAL.html

Virtual Accelerator



Simulation Program:

- Trace3D
- PARMILA
- XAL Online Model

"Virtual accelerator" is a model imitating the real machine. In the case of EPICS data exchange It looks like a real machine from the EPICS channel access view, because operates with real process variable (PV) names, and produces a reasonable response generated by the simulation model.

- **PCAS - Portable Channel Access Server**
- **Simulation Program – Accelerator Model**
- **CA client – Interface to the Simulation Program + channel access client**
- **Physics Application Program – application under development**

Now it is an XAL Application.
Very useful on early stages and for demonstrations.

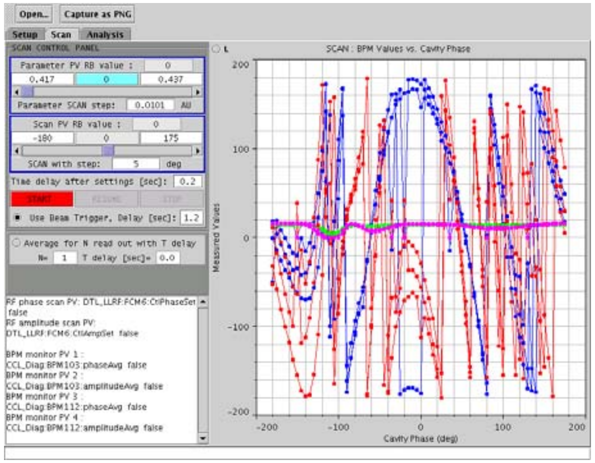
Online Model

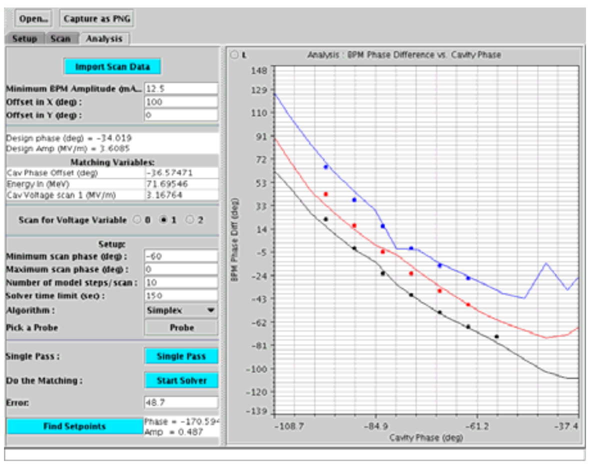
- **Package - gov.sns.xal.model**
- **Simulates charged particle dynamics through specified accelerator sequences**
- **Supports both linear sequences and rings**
- **Calculates Twiss parameters, energy and orbit distortions**
- **Six dimensional phase space propagation**
- **Includes space charge forces for envelop propagation**
- **Optics input can be from design optics, live machine, PV Logger snapshot or custom values (or combination of these sources)**
- **Fast enough to use inside optimization tasks in the interactive mode**

XAL Solver - Optimization Package

- **Optimization using a collection of algorithms**
- **This is a third generation of optimization packages in XAL**
- **Package includes:**
  - **Solver – the primary class for setting up and running an optimization**
  - **Stopper – the object that can stop the optimization process (time, iterations, satisfaction level etc.)**
  - **Problem - the class holds user's problem information: objectives, variables, constraints, hints etc.**
  - **AlgorithmPool – a collection of algorithms that can be used in optimization**
  - **SearchAlgorithm – abstract class for a search algorithms. Now the implementations are random search, random shrinking search, gradient search, simplex algorithm**
- **XAL also has the implementations of linear Least Square Method fitting algorithms and Levenberg-Marquardt method**

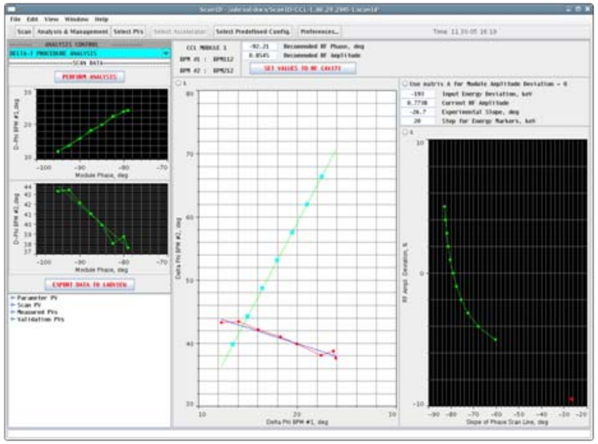
PASTA – Phase/Amplitude Scan and Tuning Application





Application to setup amplitudes and phase of RF cavities

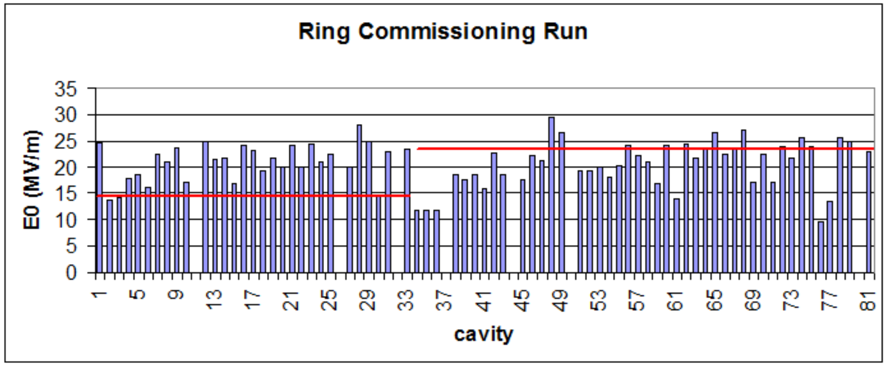
- It scans amplitude and phase of the RF cavity measuring signals from two downstream BPMs
- Solve for incoming beam energy, cavity phase and amplitude by using "phase signature matching".
- It uses the XAL Online Model and XAL Solver for "on-fly" tuning



It replaced the XAL Application based on the "Delta-T" method

To use the "Delta-T you have to find an approximate values for amplitude and phase

SLACS – Superconducting Linac Automated Cavity Setter (XAL Application)



- **As our understanding of the SRF behavior increases, operational settings change, sometimes during a run.**
- **Need to be flexible – Linac output energy is a moving target**

SCL Tune-Up Time:

August 2005: 48 hrs E_{out} = 560 MeV (> 20 cavities off)

Dec. 2005: 101 hrs E_{out} = 925 MeV

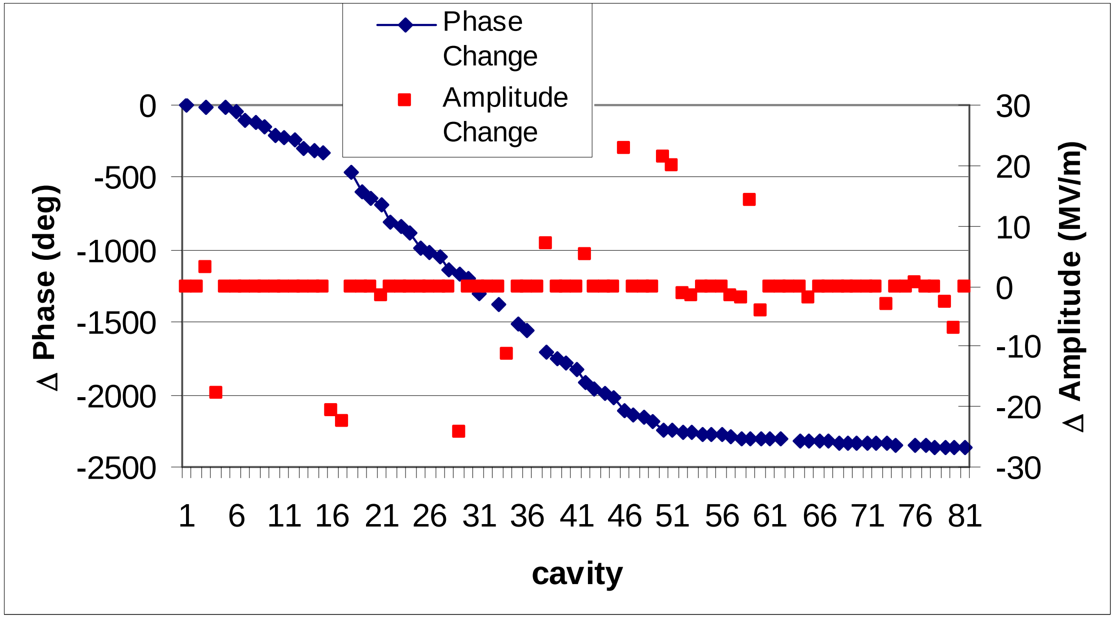
July 2006: 57 hrs E_{out} = 855 MeV

Oct 2006: 30 hrs E_{out} = 905 MeV

Jan. 2007: 6 hrs E_{out} = 905 MeV

Once SCL cavity phase set-points have been established, it is possible rescale downstream cavities using the online model (no measurements needed)

SLACS (Cont.) – SCL Retuning

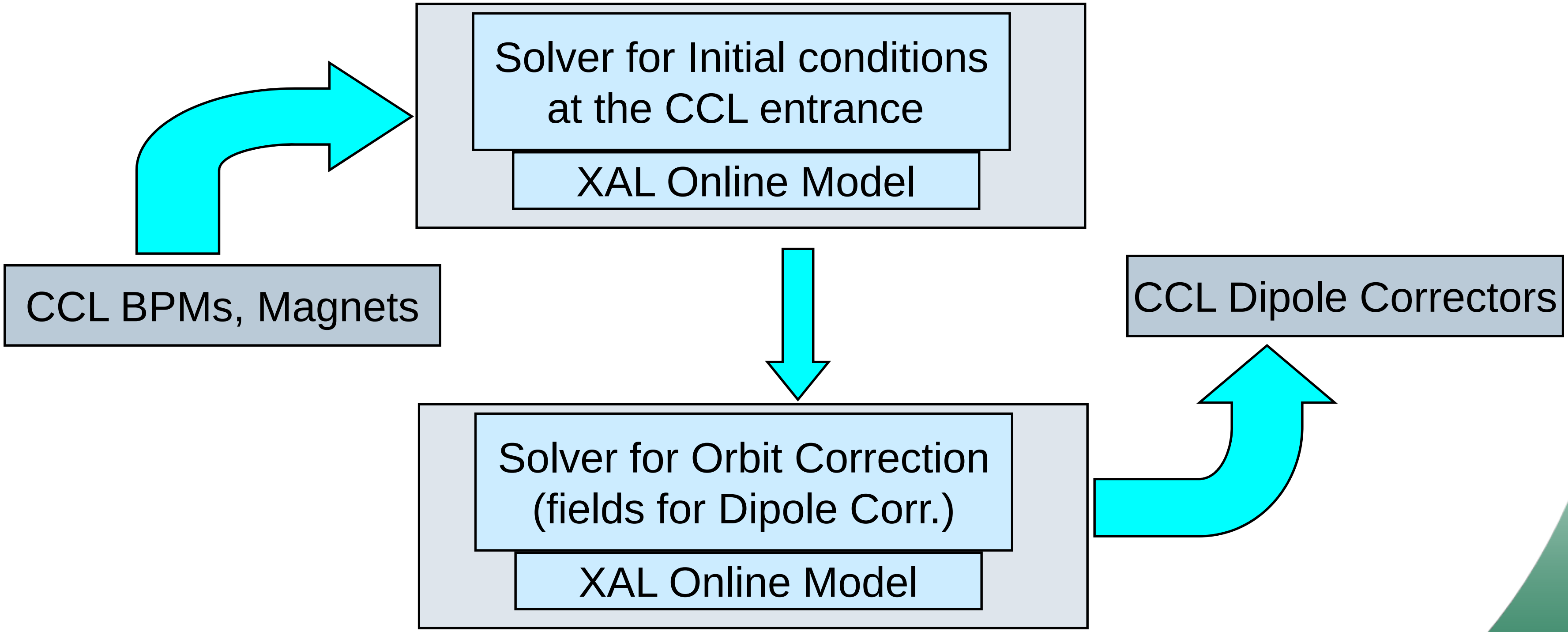


In the transition from 4.2 K to 2 K, 22 cavity amplitudes changed.

- A Model based method is used to predict the changes in cavity phase settings
- Changed over 2000 degrees at the linac end !
- The measured beam energy was within a few MeV of the prediction
- Used this method many times – takes only a few minutes to setup

Model Based Orbit Correction

- **CCL section has many quads and few BPMs**
- **Usual orbit correction with BPMs does not work**
- **Beam based alignment in quads (40 quads, 1 Hz operations freq.) does work, but it takes about 0.5-1 hour to correct orbit**
- **The model based orbit correction was developed. It takes about 30 sec and can be done parasitically**

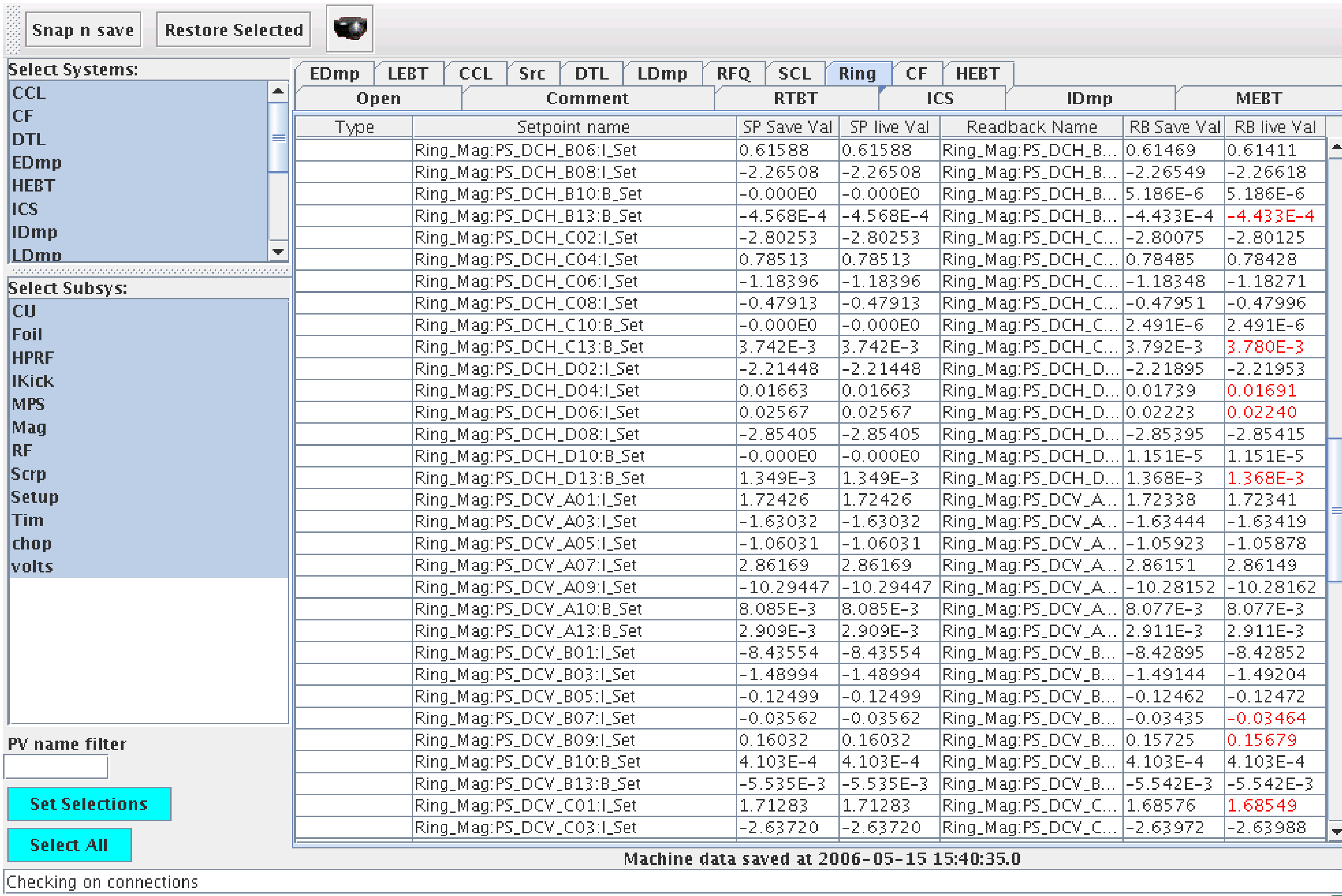


CCL losses were reduced to the acceptable level

PV Logger – XAL Service

- **Runs continuously in the background**
- **Posts sets of data to the database**
- **Posts periodically or “on demand”**
- **Each set has an unique ID**
- **Generalized to allow for custom PV sets**
- **Provides remote communication with any XAL application**
- **Has one directly related XAL Application: PV Log Browser**
- **Has one related XAL tool: PVLogDataSource – source of data for the XAL Online Model**

SCORE – XAL Application to Save - Compare Restore - Accelerator PVs



Type	Setpoint name	SP Save Val	SP live Val	Readback Name	RB Save Val	RB live Val
	Ring_Mag:PS_DCH_B06:I_Set	0.61588	0.61588	Ring_Mag:PS_DCH_B...	0.61469	0.61411
	Ring_Mag:PS_DCH_B08:I_Set	-2.26508	-2.26508	Ring_Mag:PS_DCH_B...	-2.26549	-2.26618
	Ring_Mag:PS_DCH_B10:B_Set	-0.000E0	-0.000E0	Ring_Mag:PS_DCH_B...	5.186E-6	5.186E-6
	Ring_Mag:PS_DCH_B13:B_Set	-4.568E-4	-4.568E-4	Ring_Mag:PS_DCH_B...	-4.433E-4	-4.433E-4
	Ring_Mag:PS_DCH_C02:I_Set	-2.80253	-2.80253	Ring_Mag:PS_DCH_C...	-2.80075	-2.80125
	Ring_Mag:PS_DCH_C04:I_Set	0.78513	0.78513	Ring_Mag:PS_DCH_C...	0.78485	0.78428
	Ring_Mag:PS_DCH_C06:I_Set	-1.18396	-1.18396	Ring_Mag:PS_DCH_C...	-1.18348	-1.18271
	Ring_Mag:PS_DCH_C08:I_Set	-0.47913	-0.47913	Ring_Mag:PS_DCH_C...	-0.47951	-0.47996
	Ring_Mag:PS_DCH_C10:B_Set	-0.000E0	-0.000E0	Ring_Mag:PS_DCH_C...	2.491E-6	2.491E-6
	Ring_Mag:PS_DCH_C13:B_Set	3.742E-3	3.742E-3	Ring_Mag:PS_DCH_C...	3.792E-3	3.780E-3
	Ring_Mag:PS_DCH_D02:I_Set	-2.21448	-2.21448	Ring_Mag:PS_DCH_D...	-2.21895	-2.21953
	Ring_Mag:PS_DCH_D04:I_Set	0.01663	0.01663	Ring_Mag:PS_DCH_D...	0.01739	0.01691
	Ring_Mag:PS_DCH_D06:I_Set	0.02567	0.02567	Ring_Mag:PS_DCH_D...	0.02223	0.02240
	Ring_Mag:PS_DCH_D08:I_Set	-2.85405	-2.85405	Ring_Mag:PS_DCH_D...	-2.85395	-2.85415
	Ring_Mag:PS_DCH_D10:B_Set	-0.000E0	-0.000E0	Ring_Mag:PS_DCH_D...	1.151E-5	1.151E-5
	Ring_Mag:PS_DCH_D13:B_Set	1.349E-3	1.349E-3	Ring_Mag:PS_DCH_D...	1.368E-3	1.368E-3
	Ring_Mag:PS_DCV_A01:I_Set	1.72426	1.72426	Ring_Mag:PS_DCV_A...	1.72338	1.72341
	Ring_Mag:PS_DCV_A03:I_Set	-1.63032	-1.63032	Ring_Mag:PS_DCV_A...	-1.63444	-1.63419
	Ring_Mag:PS_DCV_A05:I_Set	-1.06031	-1.06031	Ring_Mag:PS_DCV_A...	-1.05923	-1.05878
	Ring_Mag:PS_DCV_A07:I_Set	2.86169	2.86169	Ring_Mag:PS_DCV_A...	2.86151	2.86149
	Ring_Mag:PS_DCV_A09:I_Set	-10.29447	-10.29447	Ring_Mag:PS_DCV_A...	-10.28152	-10.28162
	Ring_Mag:PS_DCV_A10:B_Set	8.085E-3	8.085E-3	Ring_Mag:PS_DCV_A...	8.077E-3	8.077E-3
	Ring_Mag:PS_DCV_A13:B_Set	2.909E-3	2.909E-3	Ring_Mag:PS_DCV_A...	2.911E-3	2.911E-3
	Ring_Mag:PS_DCV_B01:I_Set	-8.43554	-8.43554	Ring_Mag:PS_DCV_B...	-8.42895	-8.42852
	Ring_Mag:PS_DCV_B03:I_Set	-1.48994	-1.48994	Ring_Mag:PS_DCV_B...	-1.49144	-1.49204
	Ring_Mag:PS_DCV_B05:I_Set	-0.12499	-0.12499	Ring_Mag:PS_DCV_B...	-0.12462	-0.12472
	Ring_Mag:PS_DCV_B07:I_Set	-0.03562	-0.03562	Ring_Mag:PS_DCV_B...	-0.03435	-0.03464
	Ring_Mag:PS_DCV_B09:I_Set	0.16032	0.16032	Ring_Mag:PS_DCV_B...	0.15725	0.15679
	Ring_Mag:PS_DCV_B10:B_Set	4.103E-4	4.103E-4	Ring_Mag:PS_DCV_B...	4.103E-4	4.103E-4
	Ring_Mag:PS_DCV_B13:B_Set	-5.535E-3	-5.535E-3	Ring_Mag:PS_DCV_B...	-5.542E-3	-5.542E-3
	Ring_Mag:PS_DCV_C01:I_Set	1.71283	1.71283	Ring_Mag:PS_DCV_C...	1.68576	1.68549
	Ring_Mag:PS_DCV_C03:I_Set	-2.63720	-2.63720	Ring_Mag:PS_DCV_C...	-2.63972	-2.63988

Conclusions

What we did right:

- **Early staged commissioning approach**
- **Iterative Approach for Commissioning Tools**
- **Using physicists (i.e. commissioners) to write applications (Need a core group of “mentor” programmers)**
- **Educational efforts**

In XAL Development:

- **Choose Java**
- **Initialization files created from a database**
- **Online Model**
- **Application Framework**
- **Scripting (Jython/Ruby)**

What we did wrong:

- **Most applications and some of tools are SNS specific**
- **Lack of documentation**
- **Did not implement service daemons to reduce EPICS traffic**
- **We used commercial plotting package (JClass) in the open source software (XAL)**

Backup Slides

XAL - Open Source Environment for Creating Accelerator Physics Applications and Services

Features

- **Open Source collaboration with dozens of developers among several sites SNS, SLAC, BNL, JPARC and others**
- **Pure Java for cross platform development and deployment**
- **Application Framework for rapidly developing modern applications**
- **Toolbox of Java packages**
- **Collection of applications (over four dozen) and services**
- **EPICS Channel Access support**
- **Ant based build system independent of IDE**