

Beamline Control Software and Software Infrastructure

Claude Saunders, SSG

Pete R. Jemian, BCDA

2010-07-14

APS Building 401, Room A1100

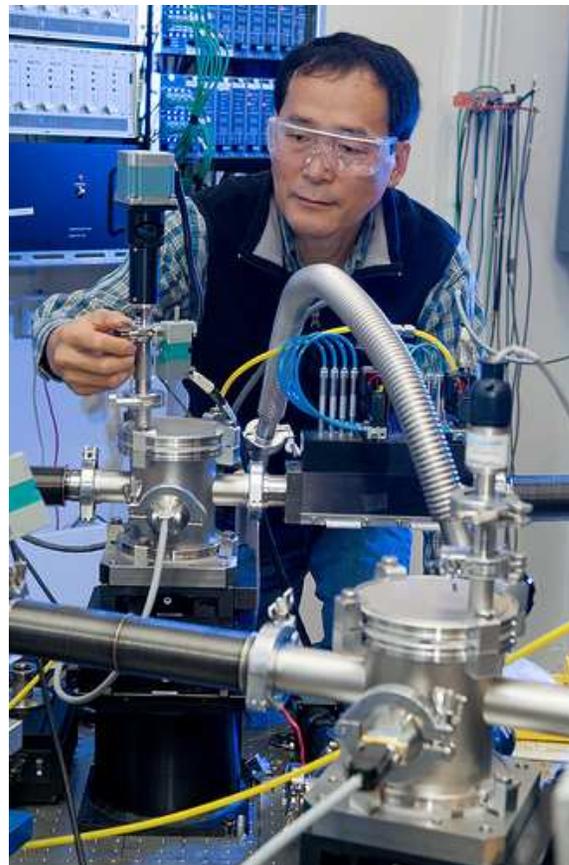
Presented as part of the APS Upgrade Technical Seminar series

ICMS: APS_1411110

Why is software important to the APS Upgrade?

- The goal of software infrastructure development for the APS Upgrade is to develop software that improves the efficiency and effectiveness of the APS beamlines and allows completion of experiments and analyses that are not possible with currently available software tools.
- Our emphasis on beamline software reflects the great room for improvement identified by our diverse user community.

A major component of any modern scientific instrument is its suite of support software



Outline

1. What are the requirements?
2. What tools do we have now to meet the requirements?
2. What will we need to meet the requirements?
2. How will we meet the requirements?
 - 3.



1. What are the requirements?

- Everything we do now, plus ...
- Improved support for detectors at beamlines
 - Many new area detectors coming
 - plus other n -dimensional detectors (e.g. hemispherical analyzer)
- Improved software
 - “We really need better and more uniform user interfaces, and thus they need to be easy to develop” – Mark Rivers, U. Chicago
- High-performance on-demand computing for some instruments
- Improved capabilities for motion control
 - Advanced motor controllers with more capabilities
 - Coordinated axes
 - Scanning multi-axis trajectories
 - Coordinating undulator scans, motion in multiple controllers, etc.
 - Provide software for advanced motion control (similar to what was done for detectors)



We know these requirements because ...

- 2006 Scientific Software Workshop:
 - Workshop Summary: ICMS: ANL-APS TB-51 aps_1193250 (<http://www.aps.anl.gov/Science/Publications/techbulletins/TB-51.pdf>)
 - Pre-Workshop Survey: ICMS: ANL-APS TB-52 aps_1191758 (<http://www.aps.anl.gov/Science/Publications/techbulletins/TB-52.pdf>)
- 2008 APS Renewal Program Workshop
 - http://www.aps.anl.gov/Upgrade/Meetings/2008_Workshop/
- 2010 APS Users' Meeting: workshop "Beamline 2.0"
 - http://www.aps.anl.gov/Users/Meeting/2010/Workshops/wk2_beamline.php
 - "The potential of an APS upgrade gives the community the opportunity to reexamine the paradigm on which beamlines are designed and integrated. The workshop will examine storage ring beamlines from the standpoint of complete integrated instruments from the particle beam through the final experimental setups. Future directions in state-of-the-art synchrotron radiation instrumentation technology will be explored. Speakers will present their individual topics within the framework of what is required of a completely integrated beamline instrument."
- Monthly meetings of BCDA, SSG, & IT, with XSD beamline staff
- Scientific interest groups
- Discussions with the users



Survey: IT & Software Infrastructure to improve Scientific Productivity of APS Beamlines

- <http://www.surveymonkey.com/s/DF6R7P2>
- "What improvements can be made in the IT and Software Infrastructure to assist in improving the scientific productivity of the beamlines?"
- Questions cover:
 - Data Rates, Data Volumes, and Computational Resources
 - Data Storage, Retention, and Transfer
 - Work Flow, Tools, and Software Expertise
 - Network Upgrade Priorities
- Additional comments are requested



2. What do we have now to meet the requirements?



Brief History

- EPICS and APS + broad, international collaboration
 - GUI tools (such as MEDM) based in X11
 - synApps: software to operate a beamline using EPICS
 - spec: software to operate diffractometers
 - asyn: communications with other systems
 - areaDetector: integrate detectors with EPICS
 - softGlue (FPGA with EPICS interface)
-
- High-level software has been left to the users...



APS computing support groups

Most computing support for the APS is centralized in AES

- AES/Beamline Controls & Data Acquisition
 - Operation of beamline equipment and experiments
- AES/Controls
 - Operation of the APS accelerator and storage ring
- AES/Information Solutions
 - Database applications for APS business processes
- AES/Information Technology
 - Networking, cybersecurity, computers, and storage
- AES/Software Services
 - Common software for beamlines, the accelerator
 - Some technique-specific software such as tomography
- XSD/Optics and Detectors
 - Detector support, works closely with AES groups
- XSD/Theory and Software
 - Also develops scientific applications

Accelerator
Systems (ASD)

APS Engineering
Support (AES)

X-ray Science (XSD)



How to get help at XSD beamlines

- BCDA sector contact
 - Contact is the steward for support
 - Other staff may do the work
 - <http://www.aps.anl.gov/bcda/group/CATreps>
- IT sector contact
 - Use Support Request system
 - http://www.aps.anl.gov/it/General_Support/staff.htm
- “tunneling” like you’ve always done it
 - Direct contact with known support
 - Keep sector contact informed
 - Other groups fit here:
 - Detector Group
 - Software Services Group
 - Theory and Software Group
- “Responsibilities for support of beam line controls, data acquisition, and data analysis”
 - ICMS Content ID: APS_1303193
 - http://www.aps.anl.gov/bcda/aps_beamline_controls_responsibilities.pdf

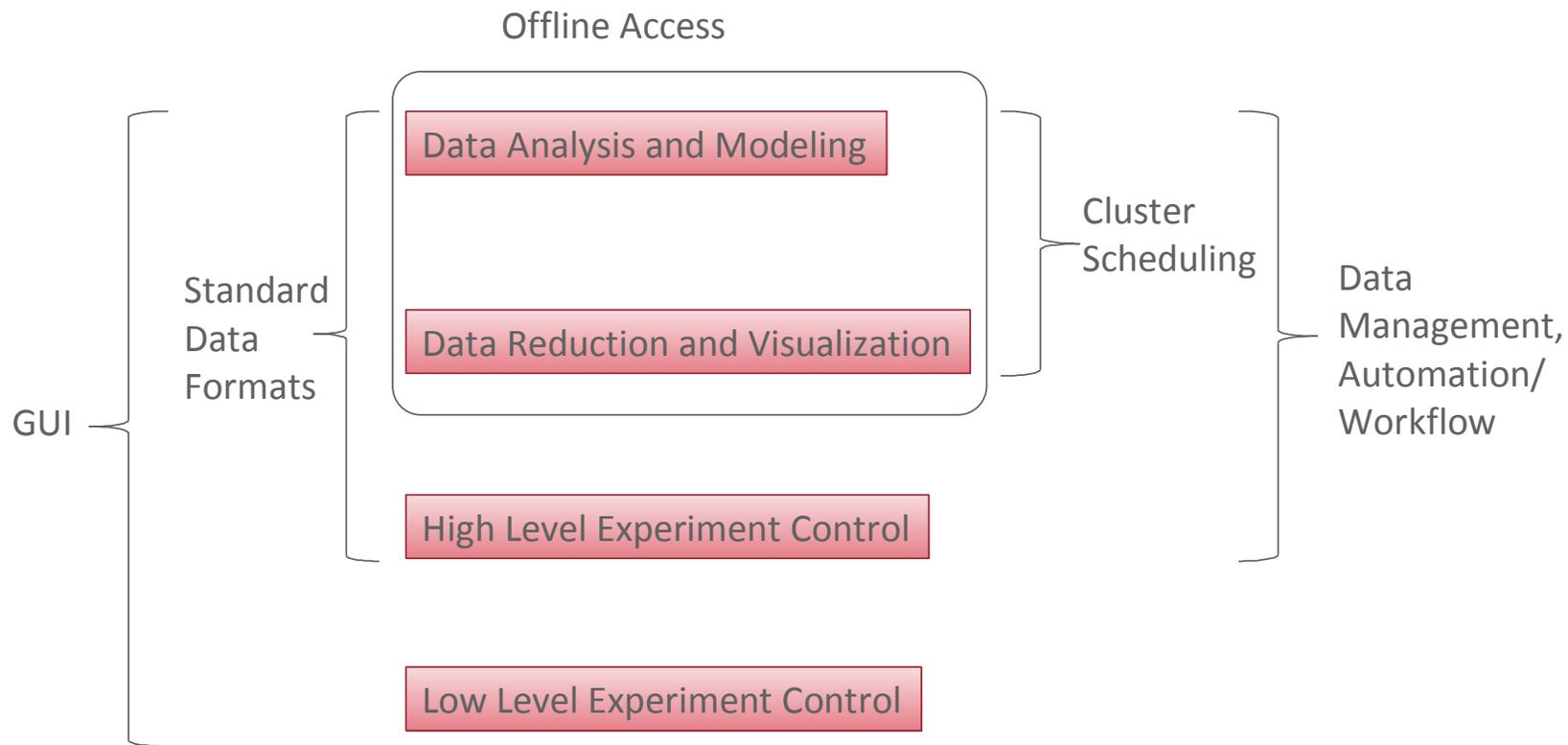


3. What will we need to meet the requirements?

- Other facilities have faced similar challenges.
- The ideas presented here are a combination of these considerations:
 - They meet the requirements of our users
 - They build upon methods chosen by other facilities to meet similar requirements from their users.



Software Infrastructure Categories

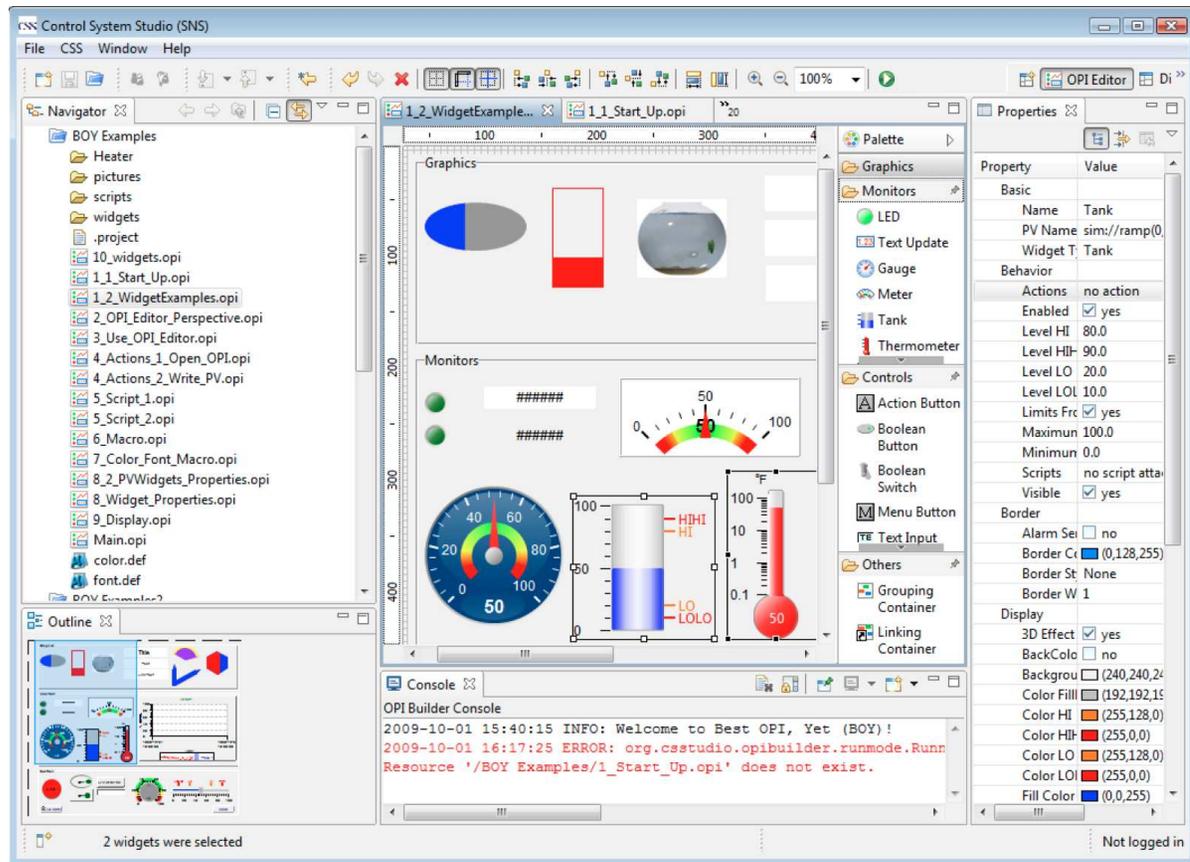


GUI (Graphical User Interfaces)

- **Current State:** Matlab, IDL, IGOR, Python WxWidgets, Java Swing, Eclipse, Tcl/TK, Qt, X-Windows (MEDM), etc...
 - For the most part – “User interface follows function”
- **APS Upgrade:** Select and develop best-of-breed, which is Eclipse, and make this the standard window onto other infrastructure functionality.
- A common starting point and center...
 - “The Scientific Workbench Concept”
 - Common look-and-feel, help mechanism, distribution and upgrade mechanism, etc.
- ... but not the only provider
 - We will accommodate existing MEDM screens
 - We will accommodate pure python code, including python-rendered GUI
 - We will accommodate execution of external tools, such as SPEC
 - Other...



Eclipse - running CSS-BOY



Eclipse - running GDA

The screenshot displays the Eclipse IDE running the Data Acquisition Client (GDA) software. The interface is divided into several panels:

- Experiments Panel:** Shows configuration parameters for tomography, flat field, and dark field.
 - Tomography Parameters:** Start Angle (149 degrees), End Angle (289 degrees), Projection Exposure time (1 ms), Number Of Projections (6027).
 - Flat Field Parameters:** Number of images taken per flatfield (100), Number of Flat Fields taken (10).
 - Dark Field Parameters:** Number of images to be taken (10).
- Plot 1 Panel:** Displays a graph of detector energy (edxd_energy) on the X-axis (0 to 160) and Y-axis (0 to 10000). The plot shows a sharp peak at approximately 60 units and a smaller peak at approximately 70 units. Below the plot, there are two checkboxes labeled "Active", both of which are checked.
- Console Panel:** Displays the EDNA Console output, showing a successful test result:

```
[SUCCESS] [ 1 ] [ EDTestCaseEDFactoryPluginTest.testGetProjectRootDirectory ] [ 0. ]
```

=====
[UnitTest]: #####
[UnitTest]: EDTestSuiteKernel summary report:
[UnitTest]: Total TestCases: 7
[UnitTest]: Total TestCases [SUCCESS]: 7
[UnitTest]: Total TestCases [FAIL]: 0
[UnitTest]: [Total TestMethods]: 22

The status bar at the bottom indicates "Not logged in", "ssg37927", "Baton held", "No Scan running", and "No S...ning".

Eclipse - running Fable

The screenshot displays the Fable software interface. The main window is titled "Fable" and contains several panes:

- Left Pane:** A file tree showing the project structure, including folders like "FastSweepCu", "SimAl", "GrainSpotter", "ImageD11", "PolyXSim", and "Structure Files".
- Transformation Options Pane:** Contains fields for "Filtered peaks file", "Fit" parameters (tth min: 0, tth max: 180), and "Save files" options (G-vectors file name, Filtered peaks file name).
- Plot Pane:** A scatter plot titled "exingExamples/SimAl/ImageD11/simAI20081" showing a circular pattern of blue dots. The x-axis is labeled "fc" (ranging from 100 to 900) and the y-axis is labeled "sc" (ranging from 0 to 1000).
- Plot Options Pane:** Includes dropdown menus for X and Y axes, a "Data" dropdown, and buttons for "Remove Data", "Data", "Switch X Y", "Update", "Clear", "Keep", and "Remove".
- Console Pane:** A text window showing the following output:

```
Fable console
Mon Jul 12 14:04:55 CDT 2010 c:\FableData\GrainIndexingExamples\SimAl\PolyXSim\data\simAl0117.tif 0.01/s T=50 n=3 ; 0.08/s 0.09/s
Mon Jul 12 14:04:55 CDT 2010 c:\FableData\GrainIndexingExamples\SimAl\PolyXSim\data\simAl0118.tif 0.01/s T=50 n=4 ; 0.07/s 0.08/s
Mon Jul 12 14:04:55 CDT 2010 c:\FableData\GrainIndexingExamples\SimAl\PolyXSim\data\simAl0119.tif 0.00/s enqueue 0.00/s 0.00/s
Mon Jul 12 14:04:55 CDT 2010 c:\FableData\GrainIndexingExamples\SimAl\PolyXSim\data\simAl0119.tif correct 0.10/s enqueue 0.00/s 0.10/s
Mon Jul 12 14:04:55 CDT 2010 Corrector thread stopping
Mon Jul 12 14:04:55 CDT 2010 c:\FableData\GrainIndexingExamples\SimAl\PolyXSim\data\simAl0119.tif 0.01/s T=50 n=6 ; 0.07/s 0.07/s
Mon Jul 12 14:04:57 CDT 2010 PeakSearch has finished.
Mon Jul 12 14:04:57 CDT 2010 The output directory should be:
Mon Jul 12 14:04:57 CDT 2010 C:\FableData\GrainIndexingExamples\SimAl\ImageD11
Mon Jul 12 14:04:57 CDT 2010 For each search, one filtered Peak file (.flt) per threshold and one 2D Peak
Mon Jul 12 14:04:57 CDT 2010 file (.spt) per threshold have been created in your output directory.
Mon Jul 12 14:04:57 CDT 2010 You can either calibrate the filtered Peak file in Transformation, plot
Mon Jul 12 14:04:57 CDT 2010 spots using the PeakSearch/Open 3D Peak file(.flt) menu, or display 2D
Mon Jul 12 14:04:57 CDT 2010 peaks on images using the PeakSearch/Open 2D Peak file(.spt) menu.
Mon Jul 12 14:10:31 CDT 2010 Welcome to transformation 1.0.5
```



Eclipse - running pyDev (Python editor)

The screenshot shows the Eclipse IDE interface with the pyDev plugin. The main editor displays the Python code for `scanTimeCalc.py`. The code includes a `pv_monitor_handler` function for handling EPICS events and a `main` function that sets up the GUI and connects to the EPICS database. The Package Explorer on the left shows the project structure, including a `python` sub-project with various Python files. The Outline view on the right shows the class structure of the `USAXS scan time calculator` GUI.

```
def pv_monitor_handler(epics_args, user_args):  
    '''EPICS monitor event received for this code'''  
    stcTool.monitor_count += 1  
    value = epics_args['pv_value']  
    pv = user_args[0]  
    name = XREF[pv]  
    db[name] = value  
    msg = "%s %s: %s(%s)=%s" % ('pv_monitor_handler', stcTool.monitor_count, pv, name, value)  
    stcTool.postMessage(msg)  
    return True  
  
def main():  
    '''  
    this routine sets up the GUI program,  
    starts the EPICS connections,  
    runs the GUI,  
    then buttons things up at the end  
    ...  
    '''  
    global stcTool  
  
    # start wx  
    app = wx.PySimpleApp()  
  
    # prepare ChannelAccess support  
    if pvConnect.IMPORTED_CACHANNEL:  
        capoll_timer = pvConnect.CaPollWx(0.1)  
        capoll_timer.start()  
  
    # build the GUI  
    stcTool = scanTimeCalcToolFrame(None)  
    stcTool.Show(True)
```

The `USAXS scan time calculator` GUI is shown in the foreground. It features a table of user parameters and a table of calculated values.

copy EPICS Pvs to table		
other user parameters		
AR step-scan speed	0.02	degrees/second
AR return speed	0.4	degrees/second
AR acceleration time	0.2	seconds
delay time/point	0.5	seconds
tuning and dark current time	150	seconds
calculated values		
AR motor step time	512.016	seconds
counting time	750	seconds
delay time	75	seconds
AR motor acceleration time	60	seconds
AR motor return time	25.6008	seconds
one sample scan time	1422.62	seconds/scan
total time complete series	1572.62	seconds/series

parameter	value	percentage
AR motor step time	512.016	35.99%
counting time	750	52.72%
delay time	75	5.27%
AR motor acceleration time	60	4.22%
AR motor return time	25.6008	1.80%
one sample scan time	1422.62	0:23:42
total time complete series	1572.62	0:26:12

saved settings in: C:\Users\Pete\scanTimeCalcrc



Eclipse - running KNIME workflow

The screenshot displays the KNIME software interface within an Eclipse environment. The main workspace shows a workflow with five nodes: File Reader (Node 1), k-Means (Node 2), Color Manager (Node 3), Scatter Plot (Node 4), and Interactive Table (Node 5). The k-Means node is highlighted with a yellow border. A 'Node Description' window for the k-Means node is open on the right, showing the title 'k-Means' and a description: 'This node outputs the cluster centers for a predefined number of clusters (no dynamic number of clusters). K-means performs a crisp clustering that assigns a data vector to exactly one cluster. The algorithm terminates when the cluster assignments do not change anymore. The node can be configured as follows:'. Below the description is a 'Dialog Options' section with a 'number of clusters' field. A 'Scatter Plot' window is also open, displaying a scatter plot of data points. The plot has a white background and a blue border. The x-axis ranges from 4.3 to 6.1, and the y-axis ranges from 3.2 to 4.4. The data points are colored red, blue, and green, representing different clusters. The plot has a 'File' menu, a 'HiLite' button, and a 'Show/Hide' button. Below the plot are three tabs: 'Default Settings', 'Column Selection', and 'Appearance'. The 'Default Settings' tab is active, showing a 'Fit to size' button and a 'Background Color' button. The 'Console' window at the bottom shows the following text: 'KNIME Console', '*****', '*** Welcome to KNIME v2.1.', '*** Copyright, 2003 - 20', '*****', and 'Log file is located at: /Us'. The 'Outline' window at the bottom left shows a small version of the workflow diagram.

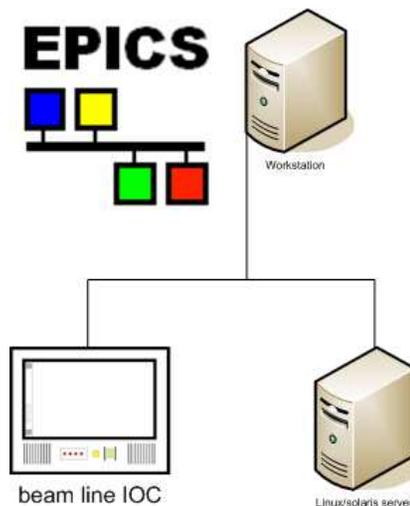
4. How will we meet the requirements?

- The challenge will be to fill common needs, while still preserving the independence and technique-specific requirements of the individual beamlines, and support continued innovation in both measurement and data analysis methodology.
- Your input and advocacy will be integral to the success in meeting the scientific goals of the APS Upgrade.



Low Level Experiment Control

- EPICS and VxWorks on VME to remain primary model
- Continue with synApps, asyn, areaDetector
- Continually work with instrument scientists to support new devices
- Develop motor control
 - DeltaTau and other advanced motor controllers
 - Multi-axis coordination
 - Trajectory scans
- Version control of all IOCs



High Level Experiment Control

- Many solutions exist for sample loading, centering, shutter control, scan setup and execution, and acquisition
 - EPICS and synApps with SPEC, Matlab, Python, etc. layered on top
- **APS Upgrade:** Leverage Eclipse to provide common, familiar environment, but support multiple modes of control
 - Low-level MEDM-style control panels
 - Converted MEDM or native CSS screens (Control System Studio- Eclipse based)
 - Detector and motor abstractions and python (JPython) scripting environment
 - GDA (Generic Data Acquisition from Diamond Light Source – Eclipse based)
 - Scan setup and execution tool
 - Work in progress among multiple facilities
 - Graphical workflow editor and executor
 - Possibly KNIME (Eclipse based)



Data Reduction and Visualization

- Graphical User Interfaces for:
 - Configuring areaDetector plug-ins
 - 1-D and 2-D live and offline detector image display
 - Visual ROI selection
 - Configuring and executing image processing pipeline (outside areaDetector)
 - Defined mechanism for contributing processing modules
 - Could share same workflow engine used for High Level Experiment Control
 - See later slide on Automation/Workflow
 - Custom ATCA-based detectors and FPGA-based data reduction



Data Analysis and Modeling

- Participate in cross-disciplinary, technique-specific working groups
 - Educate and support MVC (Model-View-Controller) design in analysis/modeling/simulation codes.
 - Educate on other design patterns to improve modularity in codes.
 - Recommend and support libraries for data formats (NeXus, others), parallel processing (MPI, openCL?), etc.
 - Support execution environment (Condor?) to allow for both single-host and cluster-based execution.
 - Support for writing new codes or re-factoring existing codes.
 - Support for parallelizing sequential codes.
 - Support for wrapping codes for inclusion in workflow engine.
 - Support for making use of relational databases for metadata storage and query



Standard Data Formats

- We recognize that there are custom/proprietary formats and other formats consumed/produced by long-established applications that must be accommodated. For example: mda and CIF
- For newly developed tools and platforms, however, we focus on a single format (acknowledging that conversion to/from other formats may be optimal for some experiments).
 - NeXus
- We further acknowledge that there is considerable work remaining to fully support NeXus at APS.
 - NeXus API not well supported across all architectures
 - Existing hodge-podge of libraries
 - Interoperability not there yet
 - Technique-specific data dictionaries need to be settled on



Data Management

- Experiment Data Catalog
 - Implemented using relational database
 - Not for storing all data, but as index to file-based data
 - For each sample put in beam, track:
 - Sample id
 - Acquisition parameters
 - Originating proposal, scheduled experiment, and ESAF
 - Data reduction and analysis history
 - Analysis execution logs
 - Data file generation and movement
- Generic web portal to locate experiment data
- Beamline Notebook
 - View and annotate context of an entire experiment
- Data Transfer
 - Agents such as GridFTP for data transfer
 - Within APS facility (ex. From beamline data acquisition disks to HPC cluster)
 - Between APS and MCS and ALCF
 - Between APS and experimenter's home institution



Cluster Scheduling

- 2 Modes

- **Reservation-based Scheduling**

- Compute cores and disk space set aside for dedicated use
 - For on-demand computations during experiment
 - For on-demand computations before/after experiment
 - This is current mode for existing APS HPC clusters
 - Batch scheduling queue fronting n cores dedicated to a sector
 - Expand to support reserving cluster resources using beamline scheduling system

- **Opportunistic Scheduling**

- For computations outside scheduled experiment
 - Compete for unused resources
 - Can include single-host systems (ie. Using Condor)
 - Will evaluate and deploy cluster scheduling that meets both needs.
 - Also will support job submission to Argonne MCS and ALCF clusters.



Offline Tools

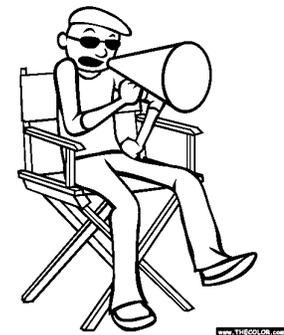
- Support for packaging up
 - Data
 - Applications
 - Execution Logs
- For use at home institution
- This will require:
 - Offsite software distribution mechanism
 - Eclipse update site
 - Packaging code into one click installers where possible
 - Designing (and testing) applications so they don't depend on APS network environment



Automation/Workflow

- Encourage creation of components that can be integrated into a workflow
 - Define mechanism(s) for designing or wrapping codes so they may be considered as potential nodes in a flexible workflow editor/executor
 - Given reduction and analysis sequence documented explicitly
 - And can be stored in experiment data catalog
 - Analysis workflow can be modified dynamically
 - Palette of analysis steps available to all

- What we have experience with thus far:
 - Control-flow style
 - Scripting using traditional language control statements (for, if-else, while)
 - Condor DAGman (Directed Acyclic Graph manager)
 - Data-flow style
 - Taverna
 - KNIME (just beginning evaluation)



Summary

- Software support is a continually-evolving process
- New software is needed to make the “integrated beamline” a reality
- EPICS gives us excellent tools for low-level high-performance data acquisition
- Constant communication is necessary
- Can start right away!
 - Not necessary to wait for APS-U CDn phase to start work on software
- Low-level
 - EPICS + synApps + asyn
- High-level
 - SPEC, at least for the short-term future
 - Eclipse + pure Python
- Data storage standard
 - NeXus



Thank you for your attention!

