Argonne NATIONAL LABORATORY

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

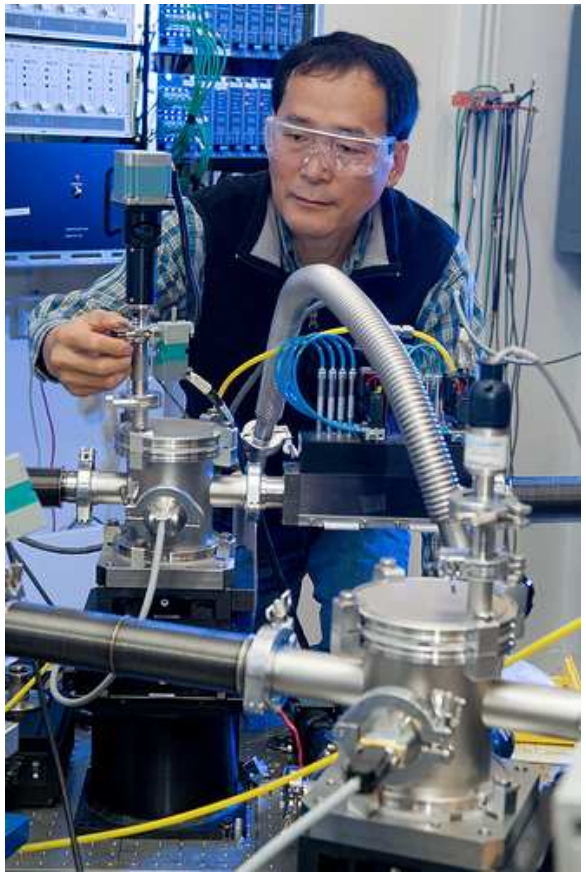
U.S. DEPARTMENT OF ENERGY

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

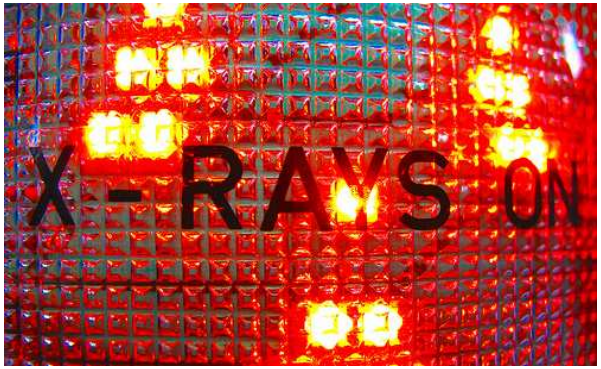
- http://www.surveymonkey.com/s/DF6R7P2
- "What improvements can be made in the IT and Software Infrastructure to assist in improving the scientific productivity of the b eamlines?
- Questions cover:
  - Data Rates, Data Volumes, and Computational Resources
  - Data Storage, Retention, and Transfer
  - 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)

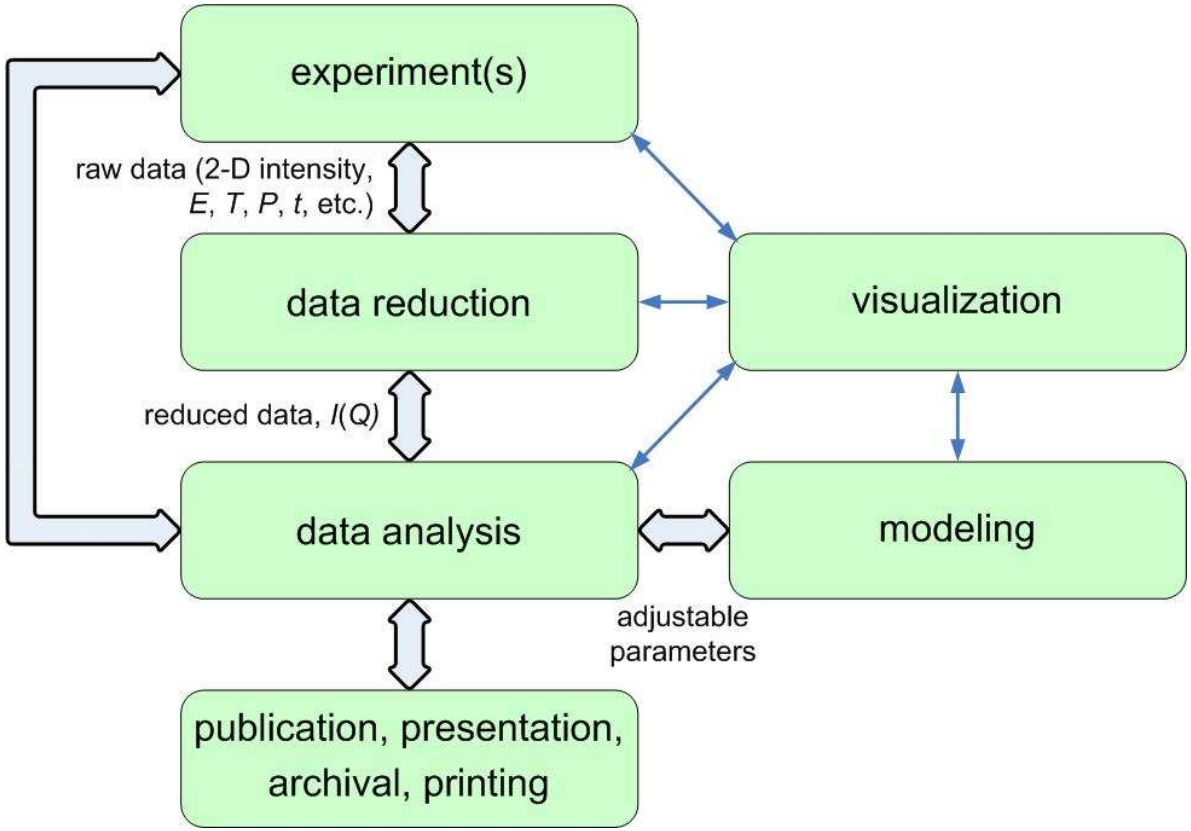
How to get help at XSD beamlines

- 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

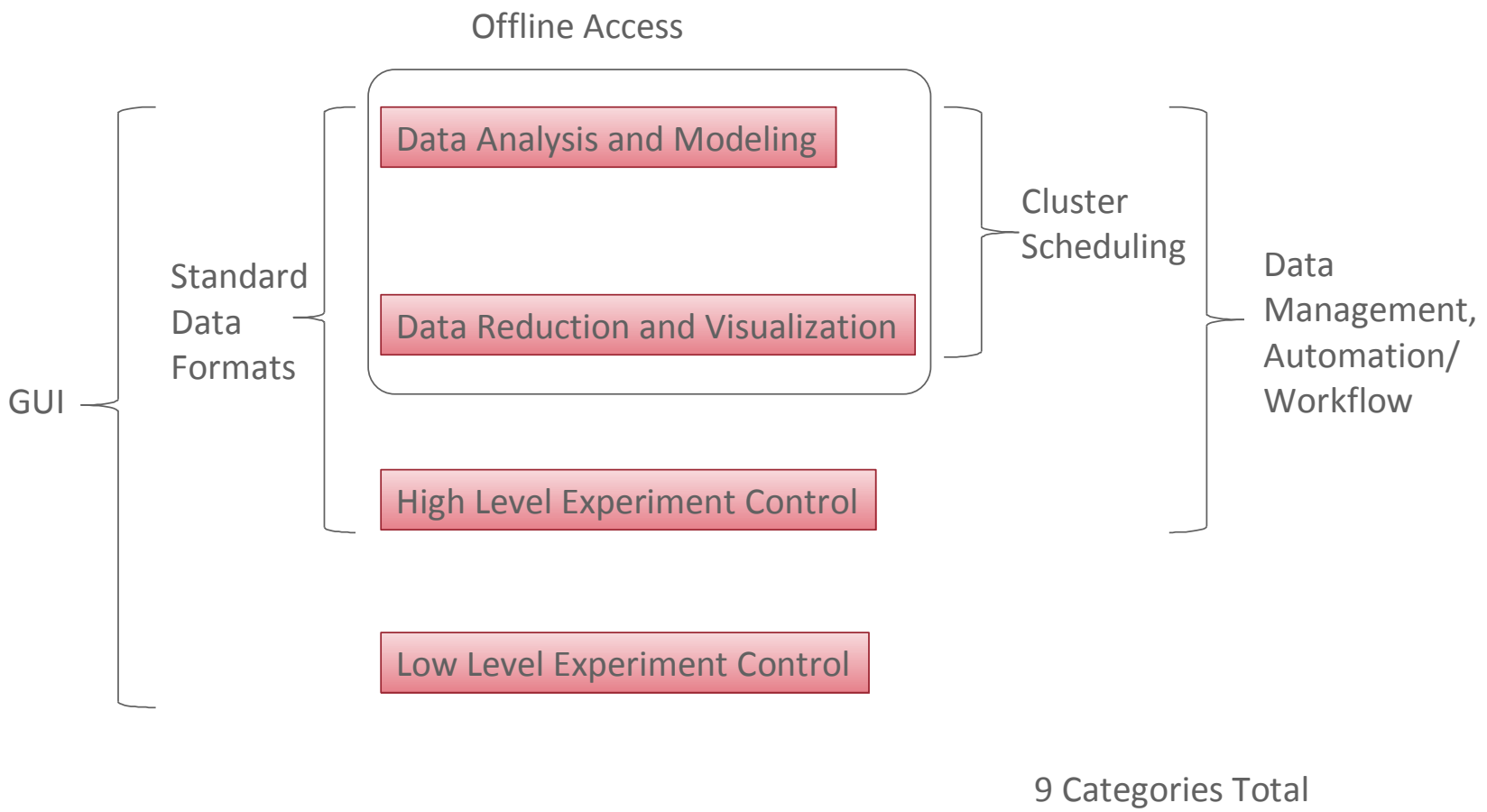
3. What will we need to meet the requirements?

- 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.

Workflow of the "Integrated Beamline"



Software Infrastructure Categories

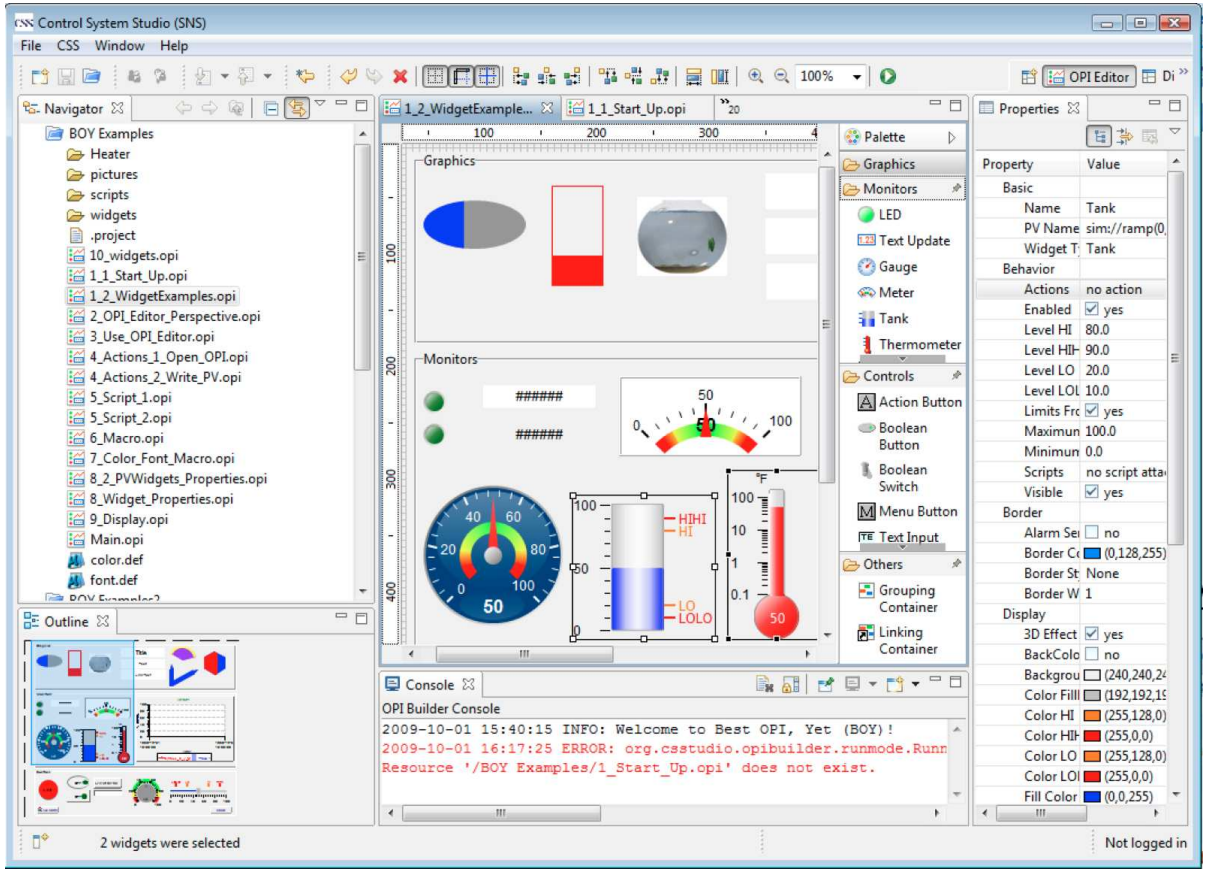


9 Categories Total

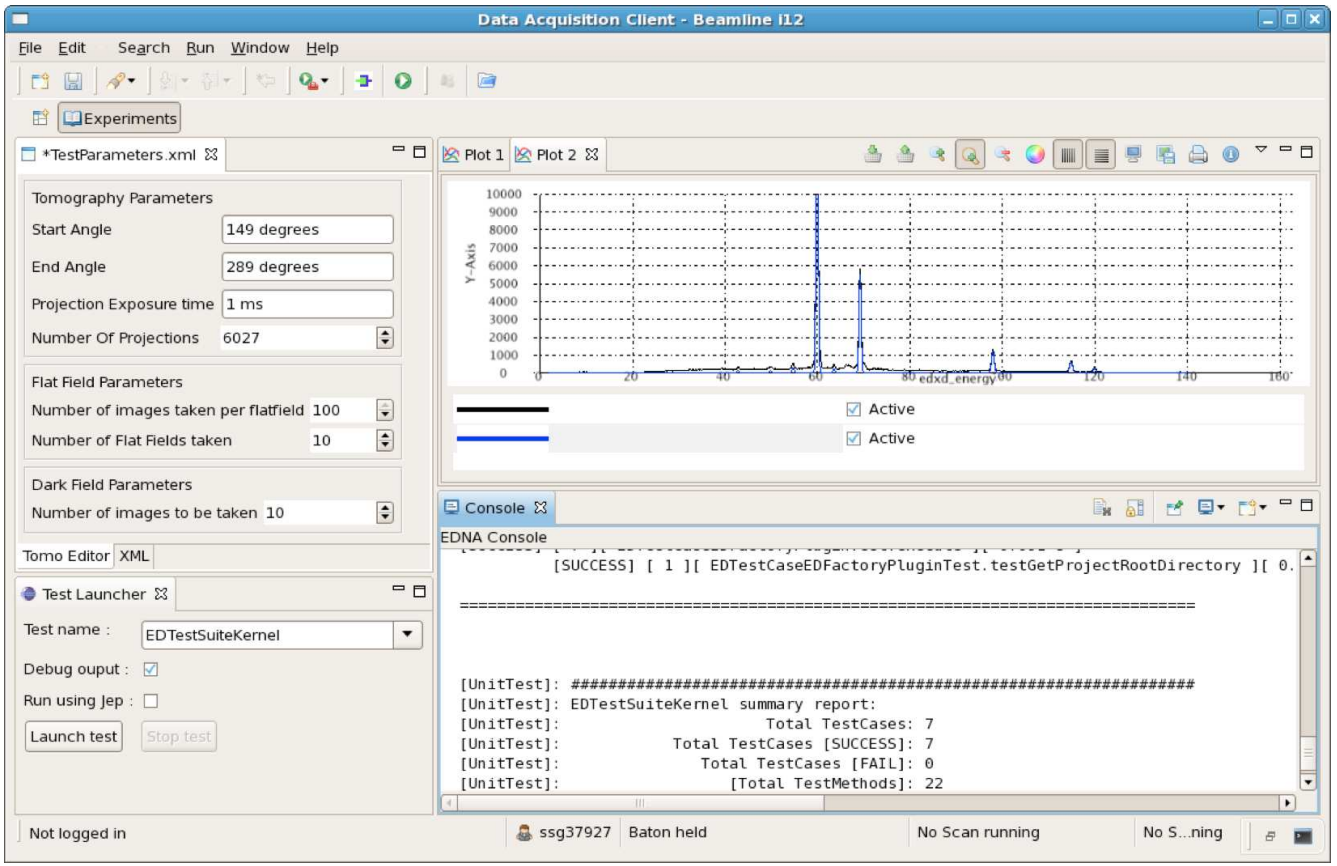
GUI (Graphical User Interfaces)

- 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…

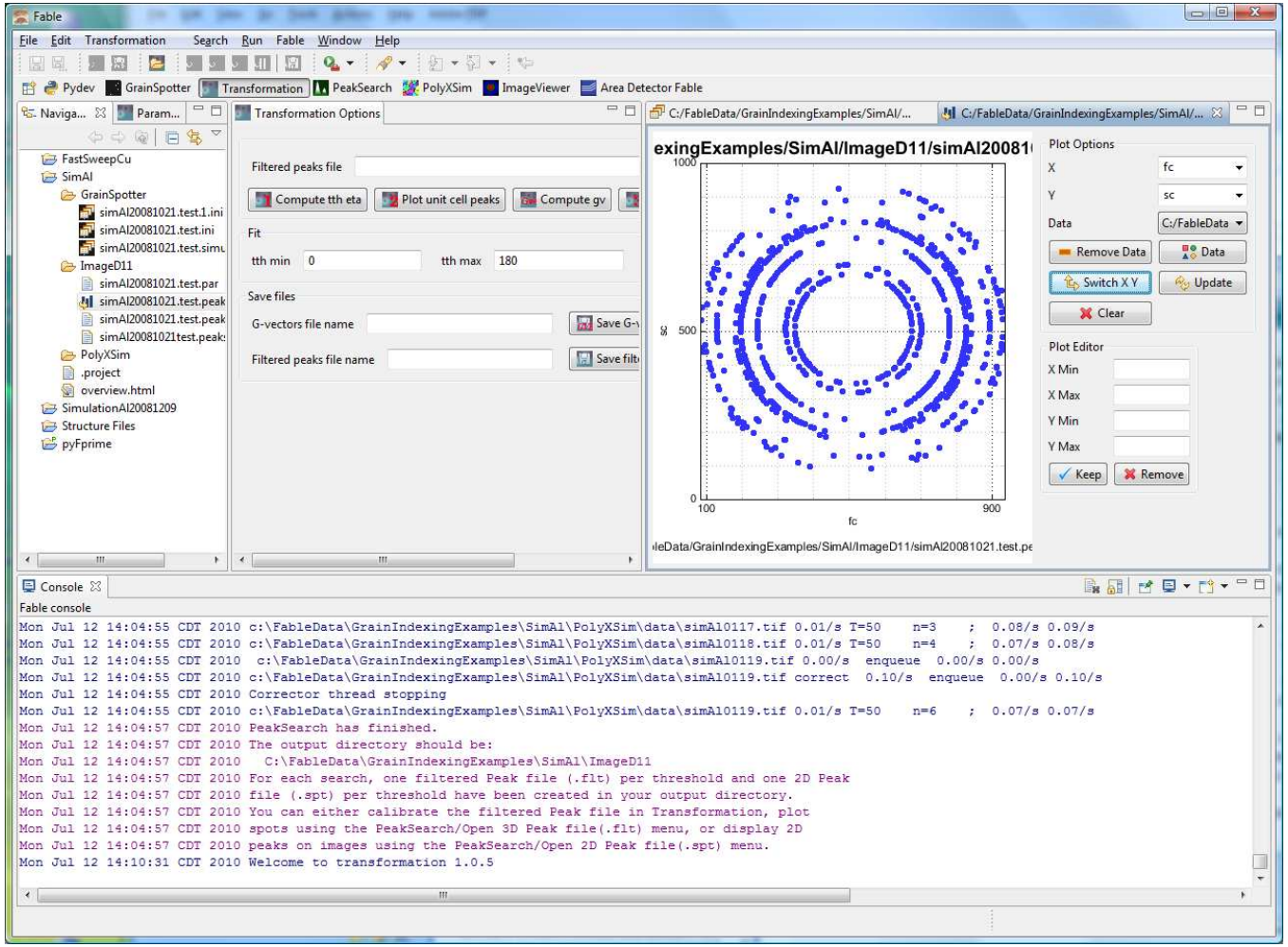
Eclipse – running CSS-BOY



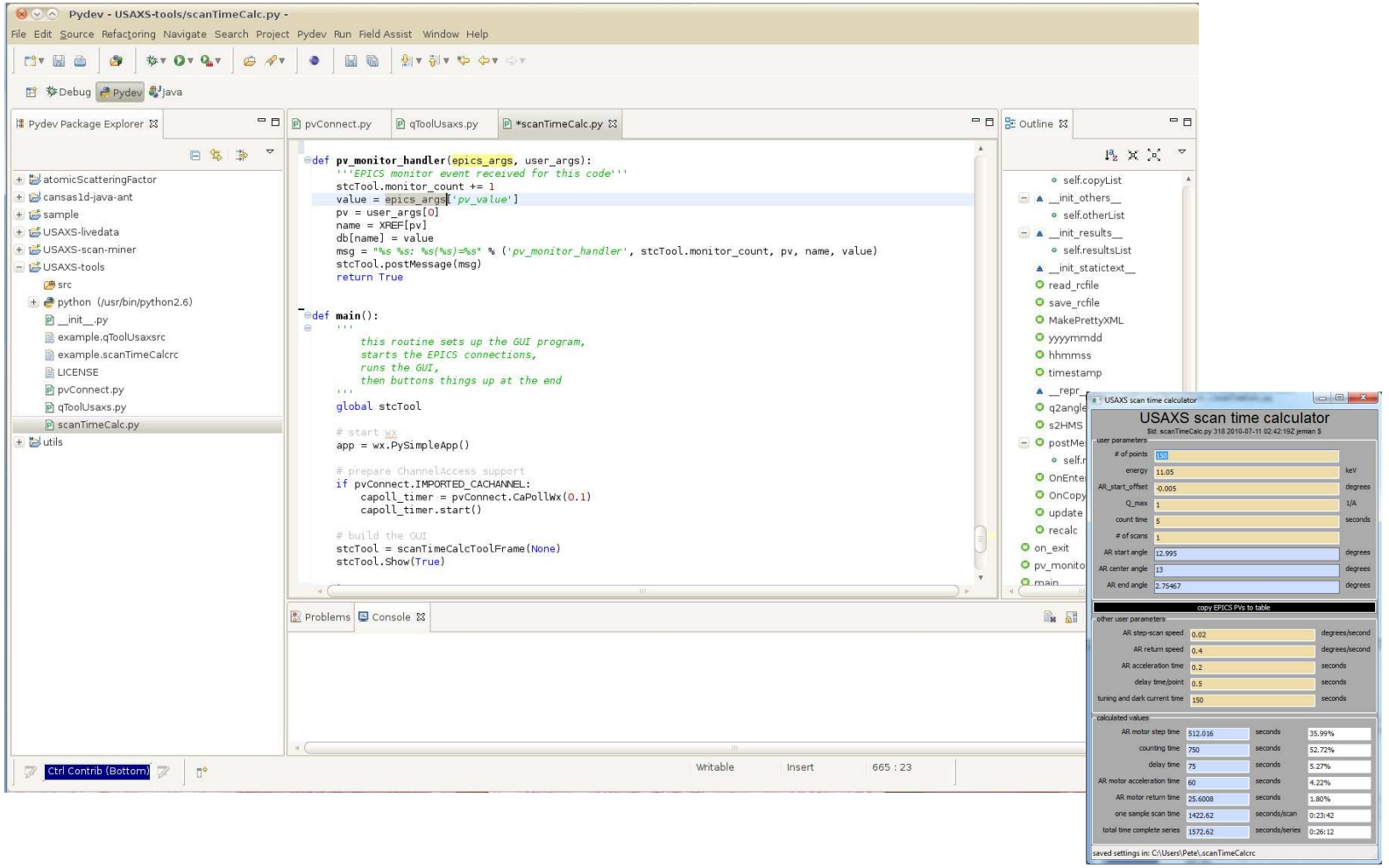
Eclipse – running GDA



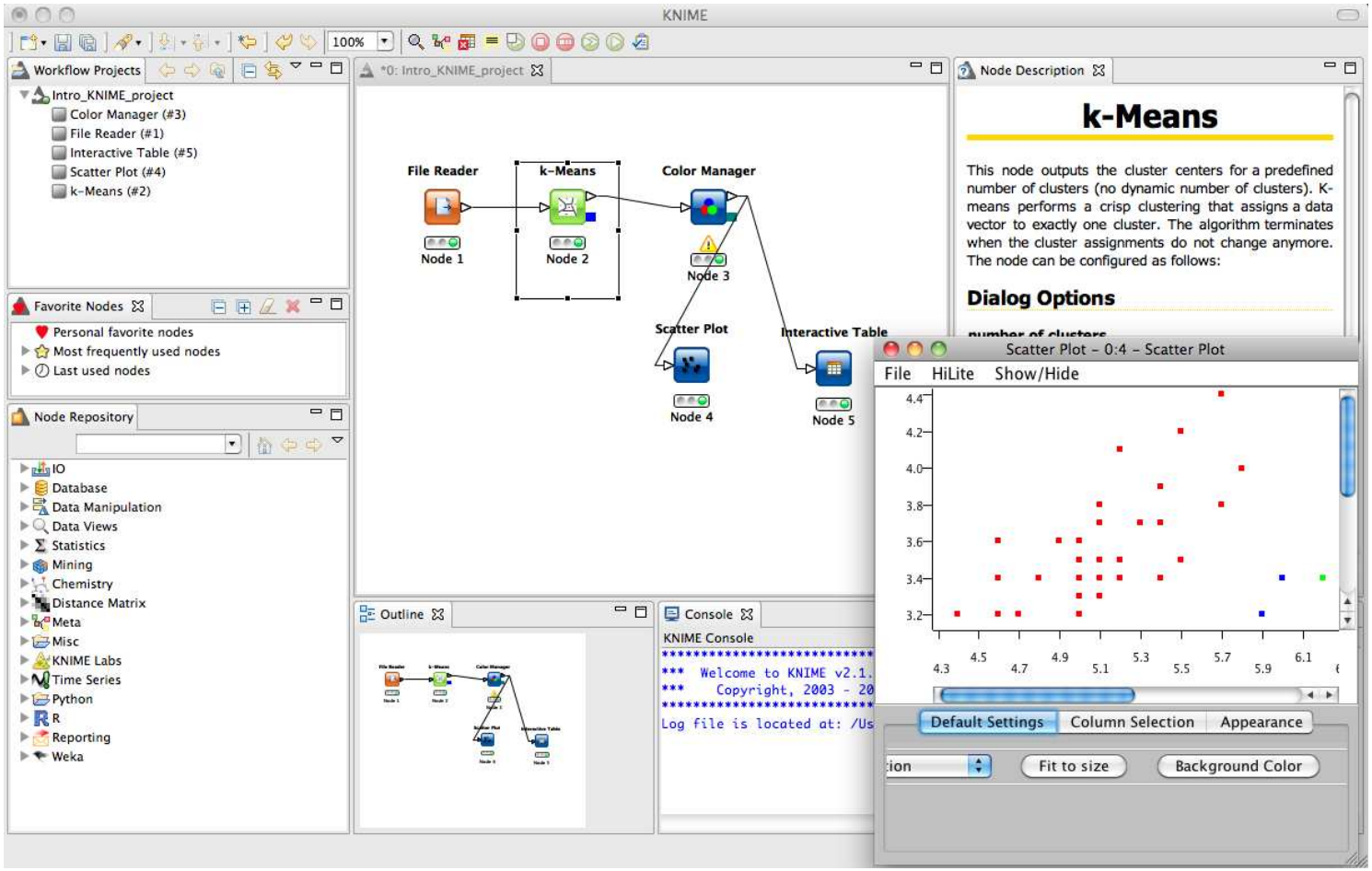
Eclipse - running Fable



Eclipse - running pyDev (Python editor)



Eclipse – running KNIME workflow



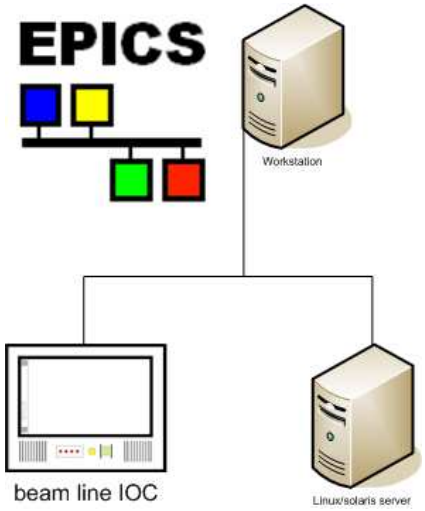
4. How will we meet the requirements?

- 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

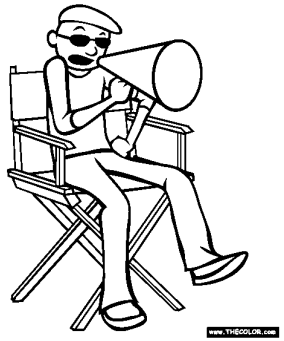
- Support for packaging up
  - Data
  - Applications
  - Execution Logs
- For use at home institution

- 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 CD*n* 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!