Outline of talk

• 6 GeV SOS/HMS spectrometer experiments in Hall C: Detectors and software.

• Describe expected initial set of 12 GeV Hall C experiments
  - Include typical/extreme rates and PID requirements
  - List online analysis requirements

• Describe Hall C 12 GeV SHMS and HMS detectors, electronics and DAQ

• Expected network needs
• Expected data storage and offline analysis needs

• Slow controls for magnets, cryogenics, target, vacuum and HV systems

• Goals of 12 GeV software
• Hall C Simulation software (for detector design, expt planning and data analysis)
• Management structure for 12 GeV software
• Timeline

Goals of Hall C 12 GeV Software

- Main goal is to have online/offline software when ready for start of experiments. To achieve this goal decided on a two-track approach
    1. Supplement present Fortran/CERNLIB based analysis code to include the SHMS and updated the HMS sections of code.
    2. Develop a C++/ROOT based analysis code based on the existing Hall A code.
- Add SHMS to the Hall C Simulation Code (SIMC). This has already been done, since it is a necessary part of planning for 12 GeV experiments.

Reasons for Fortran code

- Detectors in SHMS similar to HMS. Much of the HMS code can be copied and reused
- Hall C staff has expertise in Fortran programming
- Use as a cross check on the C++/Root code
- Drawbacks:
  - CERNLIB is no longer supported
  - Students not familiar with Fortran.

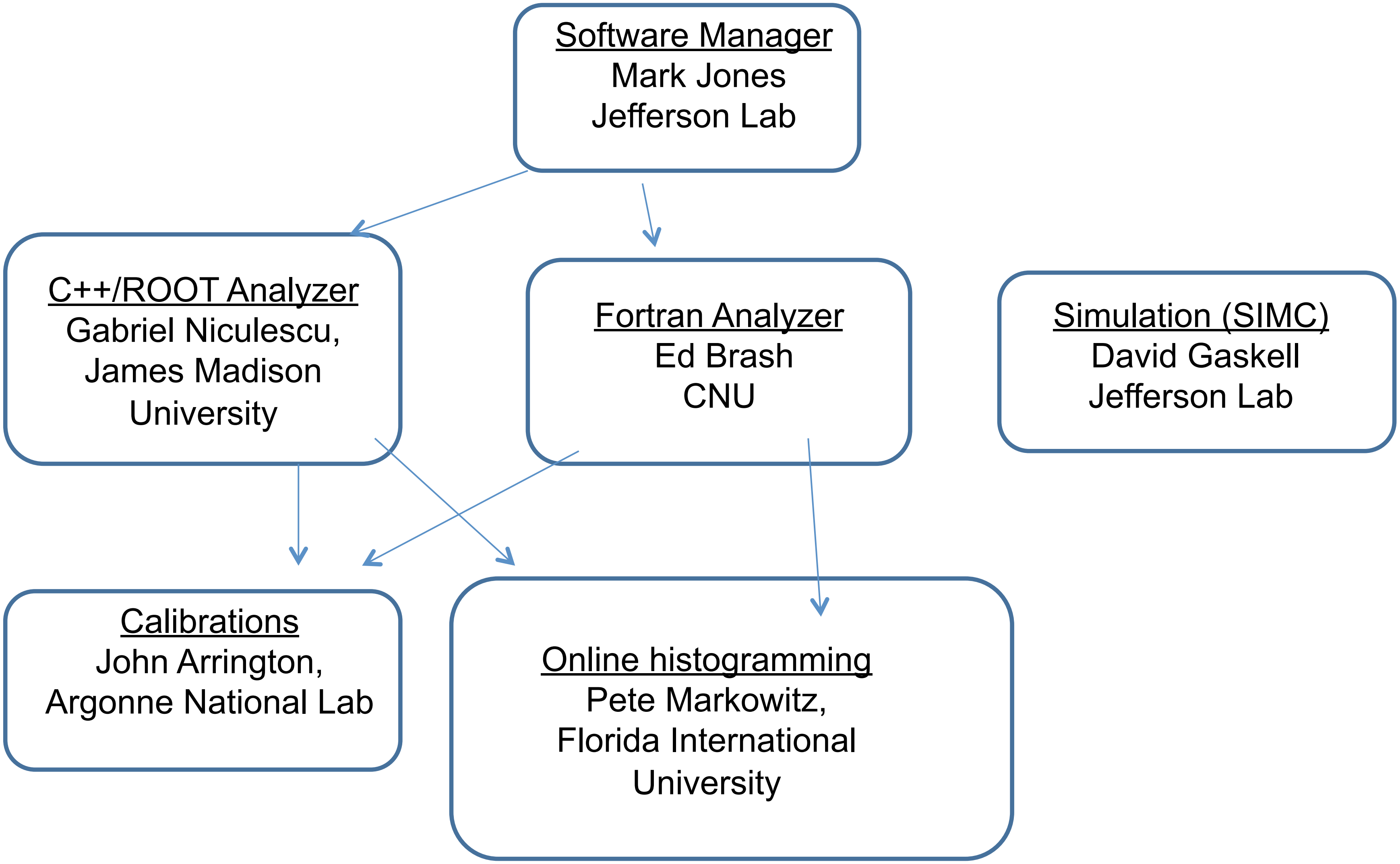
Reasons for new C++ analysis code

- To have a modern object oriented language. New students will be more familiar with C++.
- To have histogramming and data storage in the ROOT libraries which are support by CERN and the world-wide community.
- To have similar style codes in Halls A and C which both use spectrometers so that users can minimize the cross Hall learning curve.
- To share code development and documentation with Hall A and to take advantage of ROOT and C++ software developed elsewhere in the world.
- To have a straight forward mechanism for adding third arm detector setups to the code. Hall A has had great success with adding BigBite and other third arms to their software package during the 6 GeV era.
- Drawbacks:
  - Lack of C++ expertise in Hall C staff
  - Major manpower effort.

Slow Controls Software

- EPICS based
- Spectrometer controls developed by Hall C Engineering group. Part of 12 GeV project.
- Target controls maintained by target group.
- Most other beamline objects on accelerator controls, maintained by accelerator software group
- Only major new controls needed is for high voltage for SHMS. Plans TBD after selection of HV system is made. (Not critical for commissioning as modern HV systems have full featured interfaces.)
- Other controls and screens done on adhoc basis for specific experiments.

Management Structure



Milestones

Month/Year	Milestone
5/2012	Define management structure. MOUs defined. (Just JMU?)
6/2012	Official Simulation Monte Carlo code ready (Done, but doc update needed)
7/2012	Code management system deployed (Done, but workflow usage not defined)
7/2012	Define set of reference HMS data for testing ROOT analyzer
9/2012	Preliminary definition of DAQ hardware/Data format (Brad)
10/2012	DAQ decoding software more "object oriented" in Hall A ROOT analyzer
11/2012	Ability to decoding old Hall C data in ROOT analyzer
12/2012	ROOT analysis of HMS hodoscopes verified with old data set
1/2013	Complete definition of software functionality (document what analyzer is supposed to do)
4/2013	Decoding of new DAQ hardware ready in Fortran Analyzer
6/2013	SHMS added to Fortran Analyzer
7/2013	Full analysis of HMS from old data set
9/2013	SHMS Code (C++) ready for shower counter tests
12/2013	Full ROOT analysis of legacy HMS data verified
1/2014	Decode and basic analysis of BCM/BPM data in ROOT Analyzer
2/2014	Calibration codes ready
4/2014	Both ROOT and Fortran analyzers ready for cosmic tests of individual detectors
7/2014	Demonstrate focal plane analysis with cosmics (HMS)
9/2014	Basic Coincidence analysis, First beam
12/2014	Data driven bug fixing/code improvements

- Following slides probably for breakout session.

(S)HMS Analysis Overview

- Overview: follow existing engine approach
  - Raw hit processing/decoding [need updates for new ADC/TDCs]
    - Option to dump information needed for detector calibration
  - Track-independent detector quantities
  - Tracking
  - Track-dependent reconstruction
  - Calculate efficiencies for each detector, reconstruction step
    - Robust algorithms which yield reliable measure of performance. Not tuned to specific experiment (e.g. extreme rates or backgrounds may require modified approaches)
  - Calculate 'basic' physics quantities for each event
  - Dump all information on luminosity, efficiencies, deadtimes, etc… for run
- Heavy emphasis on experiment-independent issues
  - Each experiment…
    - Must provide higher-level physics reconstruction
    - Must decide if they want to use more specialized efficiency calculations
    - Must determine efficiency of experiment-specific cuts
  - Most are fairly well standardized, with multiple 'default' options

(S)HMS Analysis Algorithms

- Algorithms
  - Current codes identical for HMS and SOS, with detailed detector layout defined by position/geometry parameter files
  - SHMS differs only by geometry; can use identical code
    - New code needed to implement additional functionality, e.g. w.r.t. multihit TDCs, flash ADCs, special benefits of quartz hodoscope.
  - Default efficiency calculations built in
  - Option to dump information for diagnostics, detector calibration
    - For some detectors, also have built-in calibration options

(S)HMS Tracking

- Tracking algorithm
  - Identify pairs: hits in overlapping wires (non-parallel planes)
  - Identify combos: group pairs that are within R~1.2 cm
  - Generate spacepoints: combine combos within R~1.2 cm
  - Generate stubs (single-chamber tracks) for all spacepoints
  - Find tracks: link stubs between two chambers
  - Apply drift time offset, determine L/R
    - Between planes where offset
    - Best chisquared for unmatched planes
  - Calculate track-dependent quantities for all surviving tracks
  - Select 'final' track

  - Various places where cuts can be applied. By default, only raw timing on DC TDC hits

Beyond single event reconstruction

- Internal efficiency calculations (detector, tracking…)
  - Expt.-specific modifications (e.g. tracking efficiency vs. particle type, high-background data, singles vs. (rare) coincidences)
- Scalar analysis (defaults, dump of E/x/y vs. t, beam on/off)
- Diagnostics (raw detector dumps, online monitoring)
- Calibrations
  - Internal (including continuous)
  - Data dump for external calibrations
- Corrections (beam drift, FR, special[coin/cer block...])
- Cuts, normalized yields in arbitrary bins.
- Simulation: ratio method, include all 'physics' event-by-event, apply global corrections (mean eloss) as in analyzer

- At present, engine has robust default estimates for efficiencies, calibrations, etc… with options for specific conditions (or user-provided experiment-specific changes). New framework: may want multiple (or selectable) versions to allow simpler selection of optimized approach.