SBS Software Status

Andrew Puckett
University of Connecticut
SBS Summer 2019 Collaboration Meeting
August 5, 2019

UCONN Jefferson Lab

Seamus' Software Milestones (presented at Feb. 2019 Collab. Meeting)

- Nov 2016 - Software Review
- Jan 2017 - Start Digitized Simulation Output
- Apr 2017 - Decoders for all DAQ modules written
- Jul 2017 - Each detector system in analyzer, experiment configurations, basic
- reconstruction algorithms
- Can analyze channel-level raw data at this point
- Jan 2018 - Start simulated analysis for detector reconstruction
- Sep 2018 - Simulation Interfaced to analysis
- Spring 2019 - Get analysis scripts together
- End of 2019 - GMn tracking to 80% efficiency, 8 Hz, ready for nFF
- June 2021 - GEp tracking to 80% efficiency, 3 Hz
- Jan 2022 - SIDIS ready
- Spring 2021 likely earliest start of neutron experiments
- Fall 2022 likely earliest start for GEp

- **Big picture/bottom line:**
  - **We are more or less on track with this list of milestones**
  - **But there are many issues and challenges**

Major Issues/Challenges

- Leadership transition from Seamus Riordan to Andrew Puckett and the UConn group.
- Current code exists, and is somewhat functional, but:
  - Code is highly disorganized, not "production-ready", no stable working release
  - Many things hacked together sloppily for "quick-and-dirty" analysis to produce simulation analysis results quickly—not currently flexible enough for general use/application.
  - Algorithms under rapid development and testing
  - Many analyses presented but not documented.
- In current state, the code is not user-friendly
- Documentation is basically non-existent. The lack of documentation and the lack of a stable, "production-quality" release makes it difficult for groups outside the core software team to contribute—slows progress of development
- Build system is too narrowly tailored, with simplistic Makefiles, only works for Linux/JLab batch farm, and sometimes not even there.
  - Near future: migrate all SBS-related code to CMake build system with Podd analyzer version 1.7. Support for Linux and Mac OS X envisioned. No Windows support envisioned.
- Addressing the above issues is at the top of the near-term priority list for the software team leadership (Andrew/Ole).

Existing SBS Software

- Final SBS online/offline analysis software will be based on Podd, the standard C++/ROOT-based Hall A analysis framework, and use the highly successful ROOT-based "OnlineGUI" for online monitoring plots for shift workers, etc.
- Existing repositories:
  - **SBS-offline**: (principal author Seamus Riordan) https://github.com/JeffersonLab/SBS-Offline Main software repository of SBS-specific libraries, source codes, database files and replay scripts. Includes raw data decoders for all currently planned SBS DAQ modules and basic skeleton classes for all major SBS subsystems, within Podd/analyzer framework. Status of testing uncertain. Not yet in widespread use. Already in use in HCAL testing.
  - **Libsbsdig**: (principal author Eric Fuchey) https://github.com/JeffersonLab/libsbsdig Main library for digitization of simulation output; translates *g4sbs* output (hit time, position, energy deposit) into simulated raw detector signals ("pseudo-data"), populates "hit" data structures used by reconstruction (ADC, TDC, crate, slot, channel, etc); purpose is to test and develop reconstruction algorithms on simulated events using identical algorithms to those used for real data.
  - **Libsolgem/libsbsgem:** (principal author Eric Fuchey; adapted from SOLID gem digitization) https://github.com/JeffersonLab/libsolgem State of the art GEM digitization, "tuned" to real data. Converts simulated GEM hits (position, time, energy deposition) to pseudo-raw data (strip ADC samples, with realistic superimposed backgrounds)
  - **G4sbs:** (principal authors Andrew Puckett and Seamus Riordan) https://github.com/JeffersonLab/g4sbs GEANT4-based simulation of all of the SBS experiments. Documentation at https://hallaweb.jlab.org/wiki/index.php/Documentation_of_g4sbs

SBS software team—core group

Name	Institution	Role
Andrew Puckett	UConn	Software team lead, simulation, online and offline analysis software. Development, maintenance, version control, user support
Eric Fuchey	UConn	Simulations, software development, GEM clustering/tracking development and testing
Ole Hansen	JLab	Podd/Hall A analyzer framework development, maintenance, and support
Alex Camsonne	JLab	Hall A DAQ/raw data decoding
Juan Carlos Cornejo	CMU	HCAL, digitization development, SBS-offline development, database organization

SBS software team—Contributors, experiment and subsystem contacts (I)

Subsystem/software topic	(Software) Contact	Supporting Groups	Notes
General software	Ole Hansen	UConn, JLab	Podd, analyzer framework, repository maintenance and write access control
DAQ	Alex Camsonne	JLab DAQ group, subsystem PIs, Jones	Data acquisition and raw data decoding
HCAL	Juan Carlos Cornejo/B. Quinn	CMU, JLab (Barcus et al.), INFN	HCAL calibration/gain matching, decoding, clustering, hit position/energy/timing reconstruction
GEMs	Puckett	UVA, INFN, JLab, Glasgow	GEM clustering/2D hit reconstruction, tracking, recoil nucleon polarimetry
GRINCH	Averett	W&M, UConn	PMT clustering, timing, track and calorimeter matching, BigBite trigger logic upgrade (future)
BigBite calorimeters	Puckett	JLab (Doug and friends), UConn	Trigger logic, clustering and track matching, pion rejection, calibration, gain matching
BigBite timing hodoscope	Rachel Montgomery	Glasgow	Precise timing reconstruction, track matching, calibration, gain matching, etc.
Magnets/optics calibrations/field maps/spin transport	Puckett	UConn, JLab (Bogdan, Pentchev, Jones et al.)	TOSCA field map calculations, BigBite and SBS optics models, spin transport calculation, calibration procedures and tools with sieve slit/multi-foil targets
ECAL for GEP	Puckett	JLab	Clustering, position, timing, energy reconstruction, calibration and gain matching, online monitoring, trigger
RICH	Puckett	UConn	Ray tracing, PID likelihood analysis

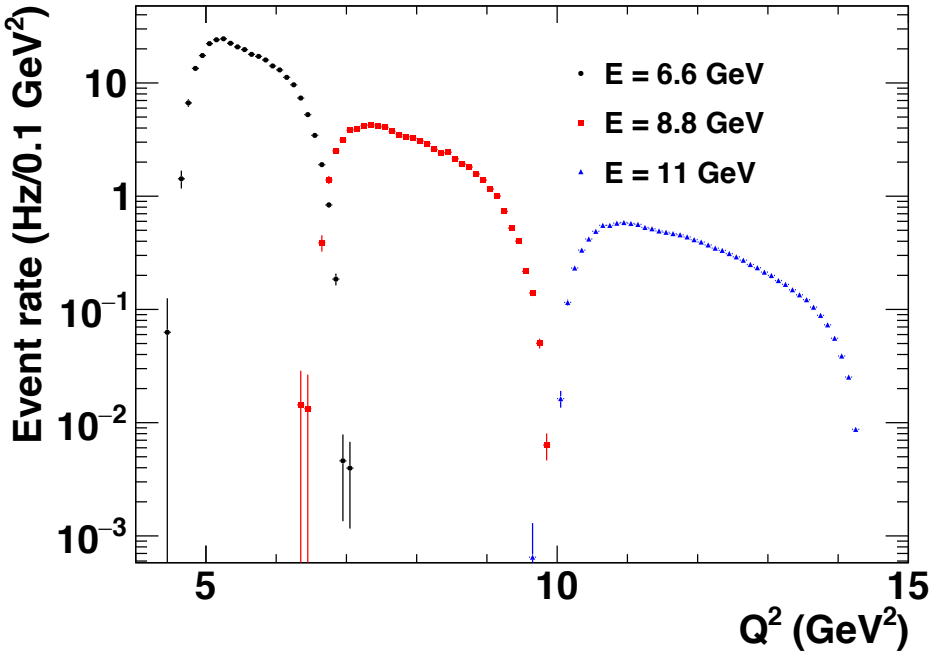
SBS software team—Contributors, experiment and subsystem contacts (II)

Subsystem/software topic	(Software) Contact	Supporting Groups	Notes
CDET	Monaghan/Brash	CNU, JLab, UConn	Calibration, gain matching, threshold calibration, GEP electron arm reconstruction
General slow controls	?	JLab	HV and LV control and monitoring, etc.
GMN experiment analysis	Brian Quinn	CMU, UConn, INFN, JLab	BigBite electron track reconstruction, timing, and PID, q-vector determination, HCAL clustering, quasi-elastic event selection, efficiency analysis
GEN experiment analysis	?? (formerly Riordan)	?? UConn, JLab, UVA	Similar to GMN, plus beam and target polarimetry, asymmetry extraction, etc.
GEP experiment analysis	Cisbani/Puckett	UConn, UVA, JLab, INFN	Electron arm reconstruction (ECAL+CDET), GEM tracking, recoil polarimetry
SIDIS experiment analysis	Puckett	UConn, JLab, INFN, CMU, UVA	Polarized 3He target, SBS with HCAL+GEMs+RICH, BigBite
GEN-RP experiment analysis	Hamilton	Glasgow, JLab, UConn, Hampton	Charge-exchange polarimetry, GEN-RP GEM and polarimeter reconstruction
TDIS experiment analysis	Dutta	MSU, JLab, UConn, Glasgow	MTPC, SBS e-mode with RICH+GEMs+LAC

G4sbs: SBS Monte Carlo Simulation

- ***G4sbs* is a success story of the overall SBS software effort:**
  - Many users and contributors from inside and outside the core collaboration; **successful use in new proposal development and approval, assisting in detector design, and even in reanalysis of Hall A GEN data (E02-013).**
  - Self-contained GEANT4 application with self-documenting ROOT output including version control
  - Github version control and code maintenance
  - CMake-based build system works "out of the box" on most Linux and Mac OS systems
  - Minimal external dependencies (only ROOT, GEANT4, and python required), and planning to keep it that way!
  - Self-documenting output
  - Extensive features and capabilities, and also highly extensible
  - Thorough documentation, maintained by the UConn group
  - Flexible geometry configuration/event generation machinery
  - Straightforward addition of new detectors/geometries with standardized ROOT outputs using pre-defined "sensitive detector" types that can handle many use cases without modification.
  - **STL vectorization of array-valued output ROOT Tree branches**—fully dynamically sized arrays, easier to read back with far less "memory waste" during analysis.
  - Built-in event generators for main processes of interest for SBS program
  - Flexible interface to generic external event generators—separate event generation from detector simulation, re-use same events in many detector configurations.
  - Anyone with reasonable C++/GEANT4/ROOT proficiency can contribute (and many have!)
- **Plan is to replicate successful *g4sbs* design approach for SBS-offline, within Podd framework**
- **See more details from Eric Fuchey's simulation, tracking talks**

New: GEP event rate and uncertainty projections for PAC47 Jeopardy Update



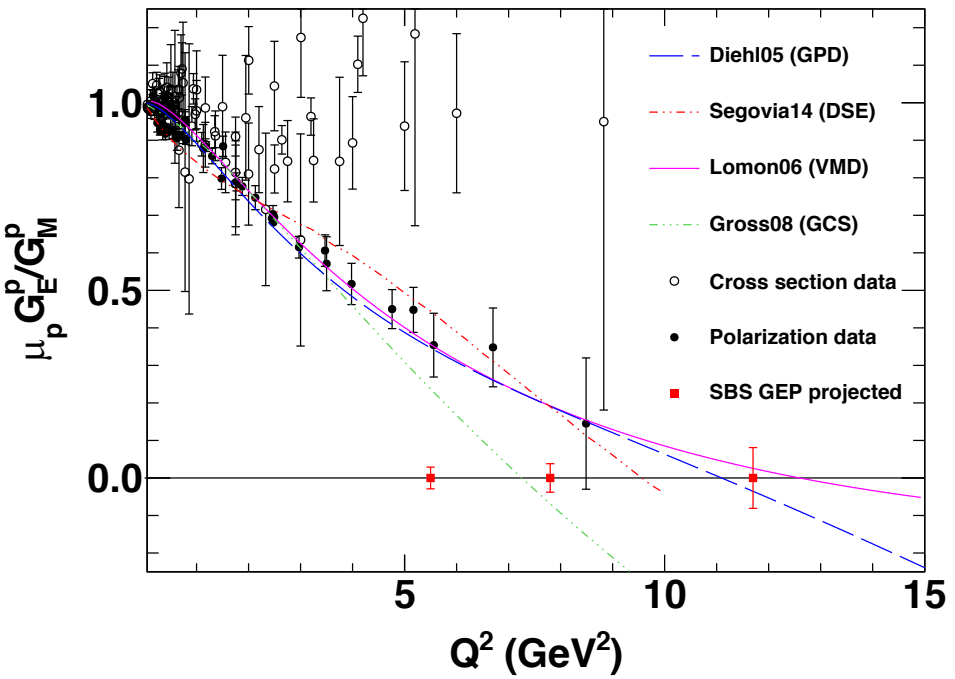
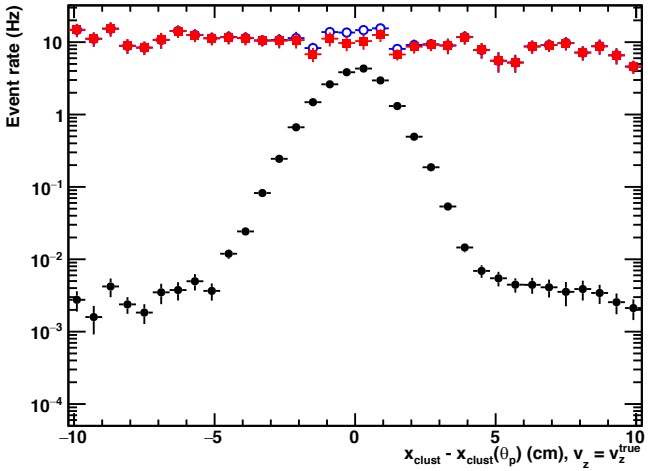
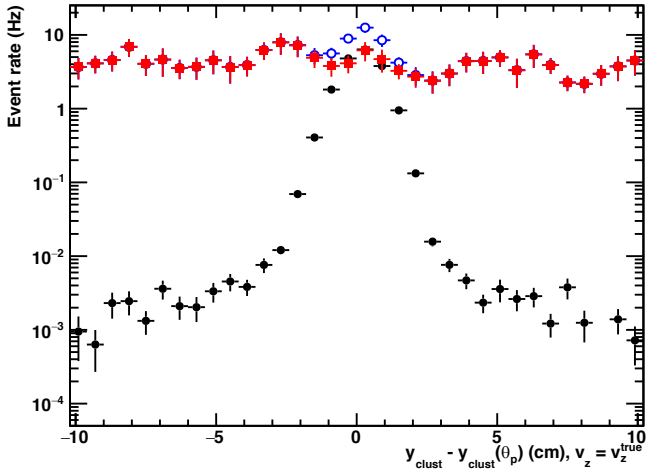


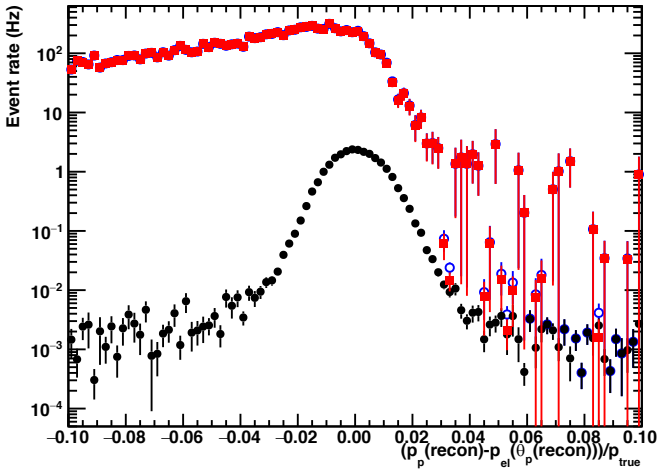
TABLE I. Kinematics, projected accuracy and beam time allocations. The projected statistical uncertainties in the form factor ratio include the assumption of 70% overall event reconstruction efficiency due to the combined efficiencies of the individual detectors, including DAQ dead-time.

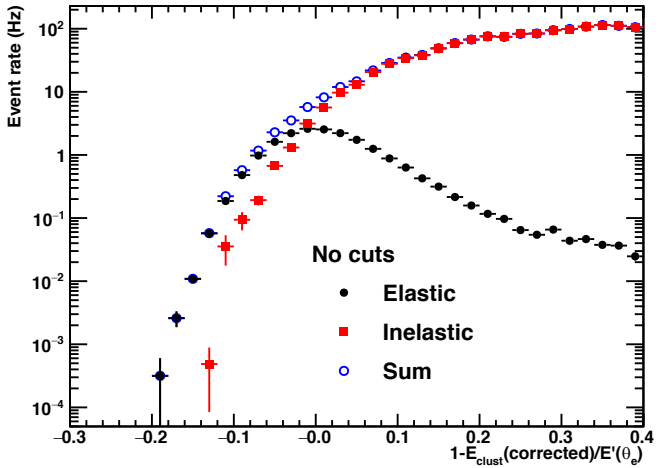
E_{beam}, GeV	Q^2 range, GeV^2	$\langle Q^2 \rangle$ GeV^2	θ_{ECAL} degrees	$\langle E'_e \rangle$, GeV	θ_{SBS} degrees	$\langle P_p \rangle$ GeV	$\langle \sin\chi \rangle$ degrees	Event rate Hz	Days	$\Delta(\mu G_E/G_M)$
6.6	4.5-7.0	5.5	29.0	3.66	25.7	3.77	0.72	291	2	0.029
8.8	6.5-10.0	7.8	26.7	4.64	22.1	5.01	0.84	72	11	0.038
11.0	10.0-14.5	11.7	29.0	4.79	16.9	7.08	0.99	13	32	0.081

New: GEP inelastic background estimation



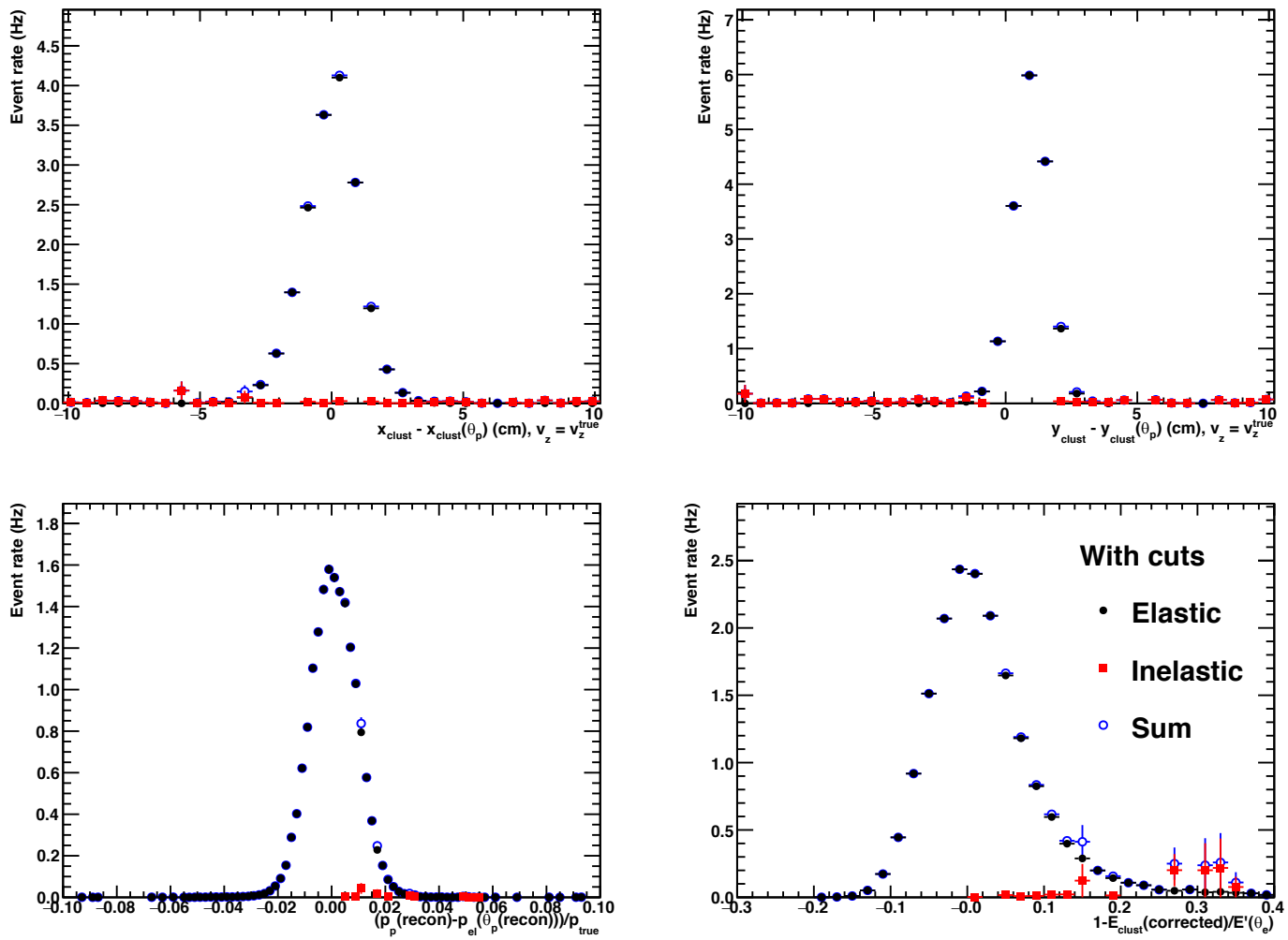






- GEP exclusivity cut variables, **elastic** and **inelastic** events, with realistic simulated detector resolutions and optics, **BEFORE CUTS**

New: GEP inelastic background estimation



- GEP exclusivity cut variables, **elastic** and **inelastic** events, with realistic simulated detector resolutions and optics, **AFTER CUTS; residual inelastic contamination < 1%**

Software: What we need from subsystem group leaders

- Subsystem group leaders need to contribute software for online analysis/monitoring and offline reconstruction.
- Software group is ready to support development and incorporation of your source code and calibration scripts into Podd/analyzer framework, but:
- **We need information about your detector, and what you need/want in terms of software capabilities/features/support!**
- Request: each subsystem group should provide a short software "white paper" *a la* "Technical Specifications Documents" from 2014 DOE Review (for examples, see: https://hallaweb.jlab.org/wiki/index.php/Preparations-DOE2014-Nov-Review) with the following information relevant to online/offline monitoring, calibration and commissioning, and reconstruction software in each of the following categories:
  - **Essential geometry information for reconstruction**: description of detectors, local coordinate systems/fiducialization points, local ← → global coordinate transformations (as installed in final spectrometer layout), assumed final layout in experiment(s)
  - Key performance specifications and plans to achieve them during commissioning
  - **Calibration and commissioning procedures with beam**
    - **"External dependencies"** of calibration/commissioning procedures: other detectors/targets/beam energies/reaction channels/sieve slits/triggers/etc required for commissioning
  - Desired low-level histograms/plots for online analysis/data quality monitoring/calibration
  - Brief description of what raw information will be recorded/available from each event in your detector
  - Required database parameters—geometry information, calibration constants/parameters/etc.
  - Specialized reconstruction algorithms/external software libraries/dependencies (if applicable)

Summary and Conclusions

- The loss of Seamus as a collaborator is a major setback—expert manpower availability is a challenge, but manageable
- The UConn group will take charge of the SBS analysis software development effort
  - Andrew Puckett and Eric Fuchey—training up two new Ph.D. students on SBS who will have a major role in software development.
- As of today, we are still more or less on track with Seamus' software milestones as presented at most recent software review (Nov. 2018).
- Next major push: finish "end-to-end" simulated experiment analysis for GMN for the highest Q^2, and also for GEN-RP.
  - All major ingredients are basically done—only missing ingredients are digitization and background superimposition for non-GEM detectors and integration of all detectors and spectrometer-level analysis into SBS-offline.
- Final design choice is to develop analysis software within existing Hall A Podd/analyzer C++/ROOT—based framework
  - Podd should be adequate for SBS reconstruction/calibration needs, and is the path of least resistance to user-friendly "production" replay code given available software expertise within Hall A and SBS collaborations.
- Replicate successful aspects of *g4sbs* approach in development of offline analysis/reconstruction:
  - Minimize reliance on external libraries/dependencies other than "standard" JLab software tools: ROOT/EVIO/GEANT4/etc.
  - Keep it simple!
- Weekly software, simulation and DAQ working group meetings:
  - Currently Mondays, 2-3 PM.
  - Recent attendance poor!
- Mailing list: sbs_software@jlab.org
- We need input (code, detector geometry/database information, and documentation!) from subsystem PIs!

Recent simulation developments/highlights

- CMake build system cleaned up/improved and convenience features added (currently only in *uconn_dev* branch):
  - Now installs g4sbs executables, scripts, root macros for analysis, data files such as PDF and fragmentation function grids, detector database files and field maps to standard directory structure under CMAKE_INSTALL_PREFIX.
  - Creates an environment setup script "g4sbs.sh" or "g4sbs.csh" that can be sourced by the user's login file, allowing execution of *g4sbs* from any directory (simplifying batch farm submission)
  - *G4sbs* can now automatically locate configuration/execution scripts, in either ".", "./scripts" or "$G4SBS/scripts"
  - Creates rootlogon.C and .rootrc files that automatically load libraries for SBS-specific ROOT-based classes (inheriting from TObject) when a ROOT session is started from the directory $G4SBS/run_g4sbs_here
  - Also adds g4sbs analysis root macros to ROOT's macro search path; load and execute from any directory
- New "G4SBSTrackInformation", "G4SBSTrackingAction", and "G4SBSSteppingAction" classes added to keep track of associations between hits in sensitive volumes and primary and/or secondary particle tracks at interesting stages of event simulation, and efficiently store this info in the ROOT Tree, without relying on the memory-intensive creation of "G4Trajectories" for every primary and secondary track created in the event.
  - Useful for evaluation of recoil nucleon polarimetry and charge-exchange polarimetry
  - Simplifies the evaluation and bookkeeping of "Monte Carlo truth" information for subsequent analysis of simulation output and benchmarking of reconstruction algorithms.
- See more from Eric's simulation talk.