Hall A Software Overview

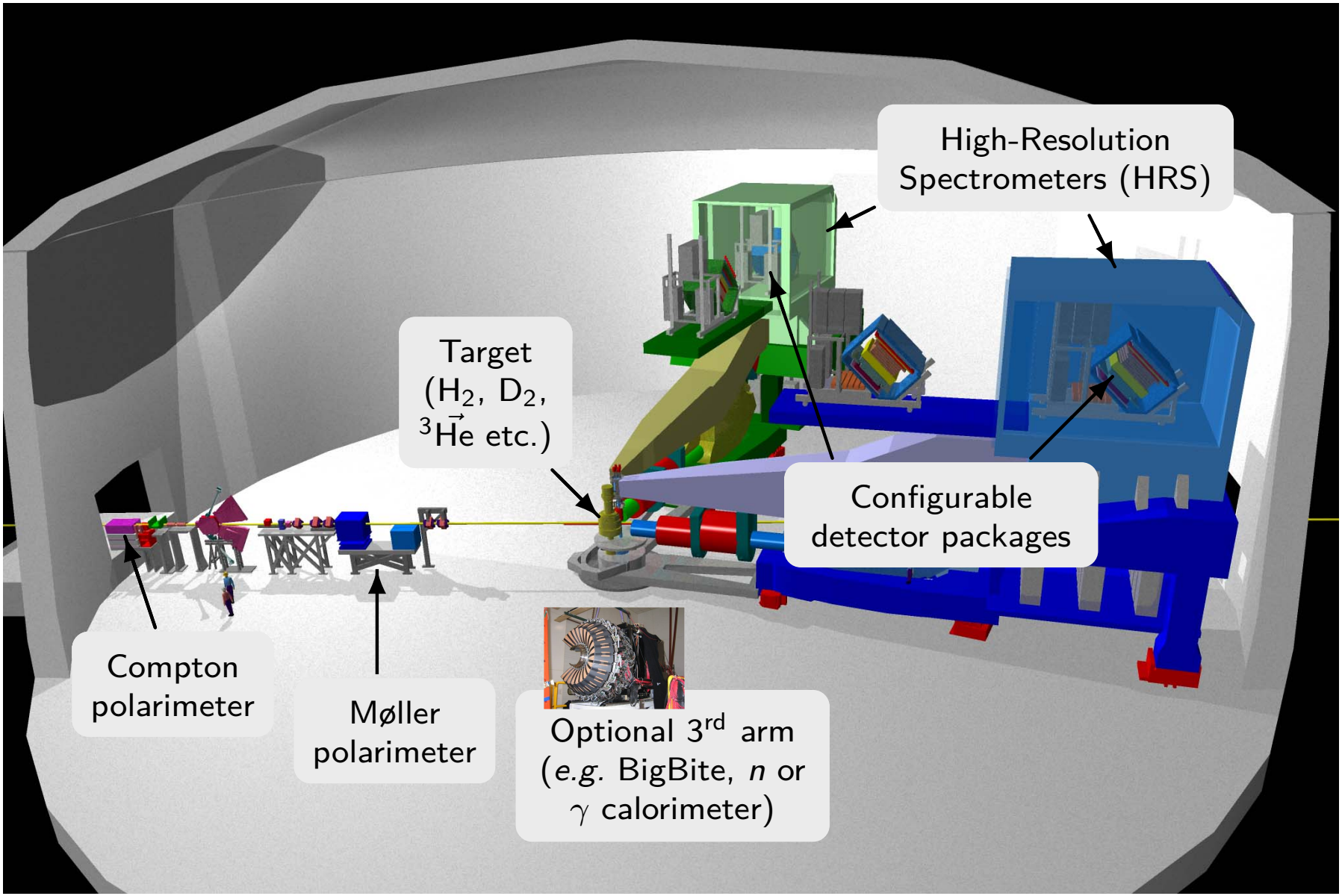
Ole Hansen

Jefferson Lab

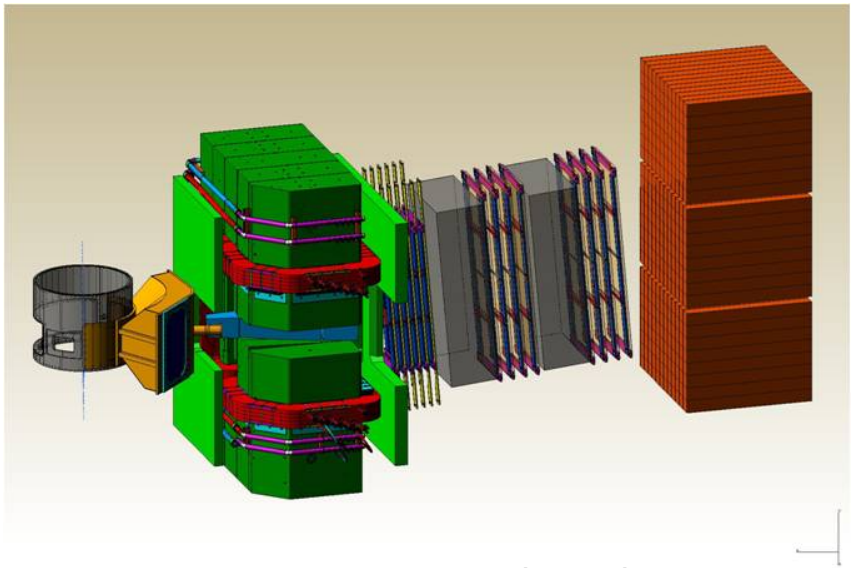
JLab 12 GeV Software Review IV
November 10, 2016

Jefferson Lab
Thomas Jefferson National Accelerator Facility

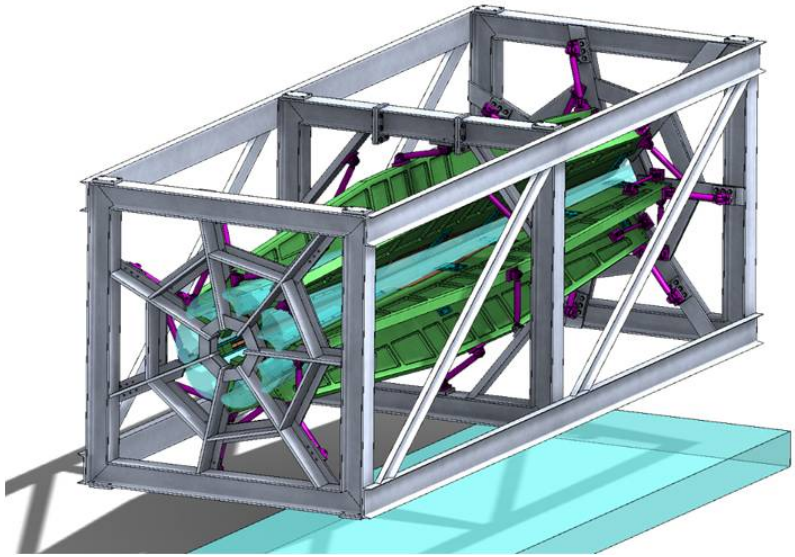
Hall A Core Experimental Equipment



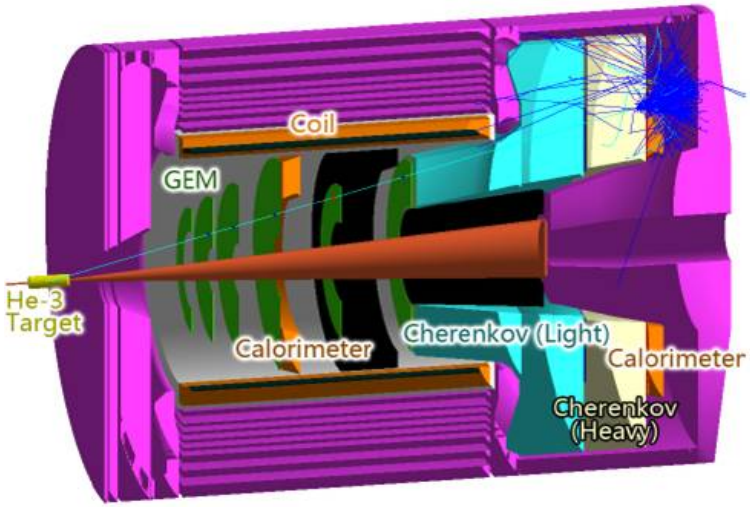
Future Hall A Projects



SuperBigBite (SBS)



Møller (standard model test)



SoLID (SIDIS, PVDIS, J/ψ, . . .)

Hall A Reconstruction Software Overview

- Hall A: "dynamic" environment
  - Frequently changing experiments & equipment configurations
  - Requires flexible, modular software
- Standard Reconstruction Software ("C++ Analyzer", "Podd")
  - ROOT-based analysis framework, in production use since 2003
  - Plug-in architecture, highly configurable
  - Code for all core systems available (standard HRS detectors, beamline)
  - Plug-ins for non-core equipment provided by individual experiments
  - Supported on Linux and Mac OS X
  - Basis for Hall C analyzer
- Historically relatively low computing resource requirements
- Future ($\geq$ 2018)
  - SBS and later projects will be much more resource-intensive
  - SBS can reuse existing analysis framework, but requires development of new software components (see later)

Projected Hall A Schedule & Experiment Requirements

as of 11/2016 (schedule past 2017 is tentative)

	2017		2018		2019	
	Spring	Fall	Spring	Fall	Spring	Fall
Experiment	^{3}H/^{3}He	^{3}H/^{3}He	*APEX*	*PREX* A_1^n *?*	*SBS commiss*	*SBS-GMn*
Trigger Rate (kHz)	5	5	10	20	2	2
Event Size (kB)	2	2	1	2	20	20
Data Rate (MB/s)	10	10	10	40	40	40
DAQ Volume (TB)	30	15	40	160	100	160
Sim+Cal+Prod cores	85	85	85	130	85	100
Tape Volume (TB)	50	40	60	235	120	200
Equipment/ Software	HRS ✓	HRS ✓	HRS(HR) ✓	HRS ✓	SBS ✎ BigBite ✎	SBS ✎ BigBite ✎

APEX: dark-photon search, PREX: neutron skin of ^{208}Pb
HRS(HR): High-Resolution Spectrometers (HRS) with VDCs under high-rate conditions

- Software for HRS systems exists and has been extensively tested ✓
- "High-rate" HRS reconstruction code essentially finished ✓
- First non-core equipment will be SBS. Software under development ✎

Reconstruction Software Readiness

Experiment	Base Software	Required Analyzer Extensions	Status / Required By
$^3H/^3He$	C++ Analyzer	-	Ready
A_1^n	C++ Analyzer	-	Ready
APEX	C++ Analyzer	High-rate VDC track reconstruction	90% Done $\geq$ Spring 2018
PREX/CREX	PAN & C++ Analyzer	-	Ready
SBS	C++ Analyzer	GEM Tracking, Calorimeters, Coord. Detector, etc. → breakout talk	In progress $\geq$ Spring 2019

Purple: First implementation Red: Incomplete

Progress on 2015/16 Milestones

Task	Status
Analyzer Release 1.6	Beta Version (80%)
Automated builds & results validation ("make test")	Done
Detailed SBS requirements & development timeline	Done
Code validation procedures integrated (cppcheck, valgrind)	cppcheck: done valgrind: not yet
Analyzer parallelization	Not yet (lack of manpower)
Metadata class, self-documenting output	In progress (50%)
Optimized HRS high-rate track reconstruction for APEX	Essentially done (90%)
High-background SBS track reconstruction simulations	In preparation (10%)

Response to General Recommendations

Recommendation	Response
Consider additional code analysis tools (**`clang`** scan-build, Coverity)	Not yet addressed. Low priority. No Coverity license available at this time.
Set up nightly builds	Using Travis-CI continuous integration system https://travis-ci.org/JeffersonLab/analyzer/builds (automatic rebuild for each GitHub commit)
Implement bit-level checks of output	Done as part of "make test" test suite
Automatically generate and store online monitoring histograms as well as expert-level output	• Histograms still manually generated, but part of shift routine. (Conditions change too frequently for easy automation) • Expert-level slow control data are automatically logged

Response to Hall A-Specific Recommendations

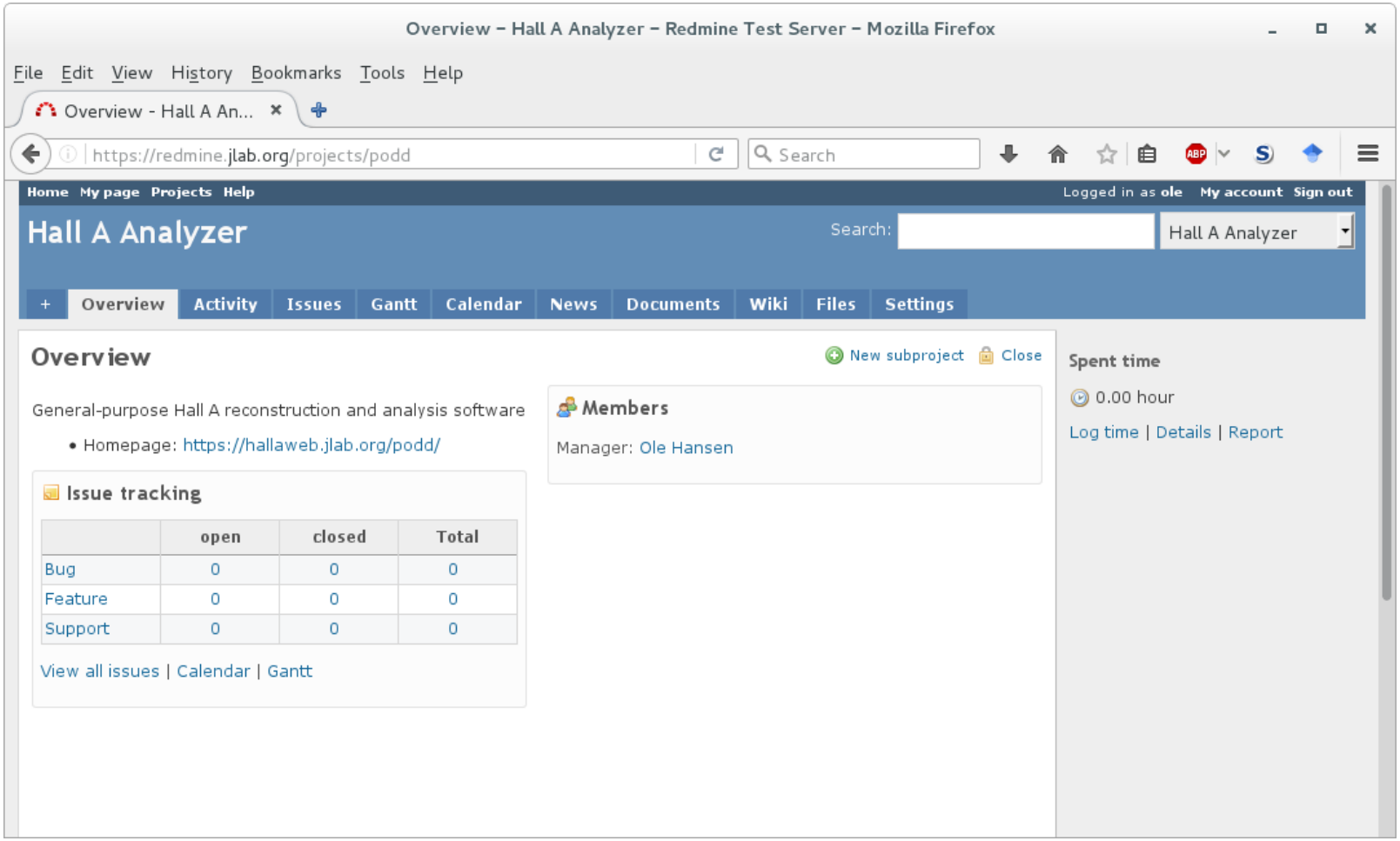
Recommendation	Response
Clarify the role of timestamps and run numbers. Prefer run numbers over timestamps unless conditions change too rapidly.	Historically, Hall A experiments have not kept globally unique run numbers. Each experiment usually starts a new sequence. Additionally, old Hall A databases are indexed by timestamps, and existing tools generate database files using timestamps. Thus, switching away from timestamps will involve considerable effort. On the other hand, all existing and new Hall C data do have unique run numbers. Hall C databases are keyed by run number only. Thus, run number indexing will need to be supported in the shared analysis software eventually. We plan to address this issue as part of the refactoring of our database API, which is currently in progress. For the time being, Hall C uses their own database front and backends, which supports run numbers.

Analysis of Current Experiments (2014–2016 run periods)

Experiment		G_M^p	DVCS
Simulations		SIMC ✓	Geant4 ✓
Reconstruction		C++ Analyzer ✓	C++ Analyzer ✓
Custom Libraries		Yes (FPP) ✓	Yes (γ calorimeter) ✓
Total # events		5×10^8	10^9
DAQ Volume (TB)		2	60
Recon phys RAM (GB)		0.5	1
Offsite production		No	Yes (60 TB txfr)
Prod time w/100 cores		1 day	6 weeks
Manpower:	PhD students	4	5
	postdocs	1	-?-
	staff	1–2	-?-
Meetings:	analysis	Weekly	Weekly
	collaboration	-?-	Annually
First publication ETA		-?-	1 year after end of exp.

Project Management

Considering Redmine, OpenProject



Summary & Conclusions

- Hall A has a mature software framework. In production use for well over 10 years.
- Hall A reconstruction software ready for early 12 GeV experiments through 2018
- Productive collaboration with Hall C underway
- Future SBS project requires extensive software development, but will likely not run before early 2019