|  |
| --- |
|  |
| Research Management Plan for the Hall C 12 GeV Detector Software  |
|  |
|  |
|  |
| **5/27/2012** |

|  |
| --- |
|  |

# Introduction

This document describes plans for the development and implementation of the data analysis software for Hall C experiments for the initial period of 12 GeV running. The data analysis software will be based on the set of Fortran code (Hall C ENGINE) developed for experiments using the High Momentum Spectrometer (HMS) and the Short Orbit Spectrometer (SOS) in the 6 GeV era. The SOS will not be used for the 12 GeV experiments. The Fortran code relies on the CERNLIB histogramming and data storage libraries which are no longer supported by CERN. The Super High Momentum Spectrometer (SHMS), which is being built, and the HMS will be used for the initial set of 12 GeV experiments. The detectors used in the SHMS are similar to those in the HMS and therefore the software needs are also similar. The simplest approach would be to copy and modify the existing HMS Fortran code to be used for the SHMS detectors. This will be the first part of a two part strategy for having the data analysis software ready and well tested before the 12 GeV experiments start. The second part of the strategy is to develop a C++ ROOT analysis software. Hall A has already developed a C++ ROOT analysis software which has been used for many years and Hall C would build upon the Hall A software. With the downtime between the end of 6 GeV running and the start of the 12 GeV experiments, the time and manpower for changing to a different programming style is available. The decision to develop a C++ ROOT analysis software is motivated by the need:

* 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.

Monte Carlo simulation is a necessary part of the analysis to produce final results from the experimental data, but a full blown GEANT based Monte Carlo is not needed. The Hall C Simulation Monte Carlo, SIMC, was used for HMS-SOS coincidence experiments in Hall C during the 6 GeV era. SIMC has been modified to include code for the optics, apertures and detectors of the SHMS.

# Goals

The main goal is to have data analysis software ready when the experimental program starts so that data can be analyzed online and offline and quickly have publishable results. In addition, the Hall C Simulation Monte Carlo has been expanded to include the SHMS. A meeting of interested users and staff has been held. A set of milestones have been established (see Table 1) and volunteers are been assigned to manage different aspects (see Table 2). For the development of the C++ Analyzer , the initial goal is to be able analyze the scintillator detectors from old HMS 6 GeV data to compare with the original Hall C Fortran ENGINE Analyzer by the end of 2012. Then by mid 2013 , the C++ code would be ready to begin a full comparison to the ENGINE analyzer with the C++ code verified by the end of 2013.

In parallel to the C++ code effort, the Hall C Fortran ENGINE Analyzer will be updated to include the SHMS. First the ENGINE code will be documented to aid in the implementation in the new C++ Analyzer. Then code will be added for the decoding of the new ADC and TDC modules. The final step will be to add the code for the SHMS detectors.

# Milestones

* Present
	+ Set-up Management structure
	+ Monte Carlo simulation is ready
	+ Decided on Git for code management of C++ analyzer
* 2012
	+ July : Define reference HMS data for testing code
	+ Sep : Documented non-tracking HMS detectors code in Fortran Analyzer
	+ Oct : Make DAQ decoding in C++ Analyzer object-oriented
	+ Oct : Ability to analyzed Hall C data at the raw data level in C++ Analyzer
	+ Dec : Documented the drift chambers and tracking code in Fortran Analyzer
	+ Dec : Verify HMS hodoscope analysis in C++ Analyzer
* 2013
	+ Jun : SHMS code added to Fortran Analyzer.
	+ July : Full analysis of HMS data with C++ Analyzer ready
	+ Sep : C++ Analyzer ready for SHMS calorimeter tests.
	+ Dec : Full analysis of HMS data with C++ Analyzer verified by comparison to Fortran analyzer.
* 2014
	+ Jan : Scalar and BPM analysis code in C++ analyzer
	+ Feb : Calibration codes ready.
	+ Jul : Analyze cosmic ray data in SHMS with both Analyzers
	+ Sep : First beam, analyze data with both Analyzers

# Institutional Responsibilities

The goals for the development of the Hall C software have been defined and Table 1 lists the leaders of the various groups. Members of the various detector groups will be involved in the development and testing of the software.

Table 1

|  |  |
| --- | --- |
|  | Task |
| Mark Jones, Jefferson Lab | Software Manager |
| Gabriel Niculescu, James Madison University | C++ Root Analyzer Coordinator |
| Ed Brash, Christopher Newport University | Fortran Analyzer Coordinator |
| John Arrington, Argonne National Lab | Calibrations and Workflow |
| Pete Markowitz, Florida International University | Online Histograms  |
| Dave Gaskell, Jefferson Lab | Monte Carlo Simulation |