org.lcsim: Event Reconstruction in Java

Norman A. Graf
SLAC National Accelerator Laboratory, 2575 Sand Hill Road, Menlo Park, CA, 94025, USA
E-mail: Norman.Graf@slac.stanford.edu

Abstract. Maximizing the physics performance of detectors being designed for the International Linear Collider, while remaining sensitive to cost constraints, requires a powerful, efficient, and flexible simulation, reconstruction and analysis environment to study the capabilities of a large number of different detector designs. The preparation of Letters Of Intent for the International Linear Collider involved the detailed study of dozens of detector options, layouts and readout technologies; the final physics benchmarking studies required the reconstruction and analysis of hundreds of millions of events.

We describe the Java-based software toolkit (org.lcsim) which was used for full event reconstruction and analysis. The components are fully modular and are available for tasks from digitization of tracking detector signals through to cluster finding, pattern recognition, track-fitting, calorimeter clustering, individual particle reconstruction, jet-finding, and analysis. The detector is defined by the same xml input files used for the detector response simulation, ensuring the simulation and reconstruction geometries are always commensurate by construction. We discuss the architecture as well as the performance.

1. Introduction
Detectors designed to exploit the physics discovery potential of lepton collisions at the Terascale will need to perform precision measurements of complex final states. One needs to fully reconstruct hadronic final states, with the ability to tag quark flavors with high efficiency and purity. Exceptional momentum resolution is required, leading to either large volume gaseous or low-mass silicon trackers. The excellent vertexing capabilities demanded by the flavor-tagging requirement point to a multi-layered giga-pixel vertex detector with micron point resolution. Calorimetry capable of very high di-jet mass resolution points towards highly segmented imaging calorimetry or to dual-readout total absorption crystals. The mission of the Linear Collider Detector (LCD) Simulation and Reconstruction group is to provide full simulation capabilities for the Linear Collider physics program, from physics event generation, to detector design simulations, through to event reconstruction and analysis. Its goal is to facilitate contributions from physicists in different locations with various amounts of available time. We do so by using standard data formats, providing a general-purpose framework for physics software development, and providing a suite of reconstruction and analysis code which is easy to install and use.

2. Modeling the Detector Response
As part of the American Linear Collider Physics Group's (ALCPG) simulation and reconstruction effort, we set out to provide a full detector response simulation program which would free our end users from having to write code or to be expert in the details of the physics simulation to design and study a detector. The system should be powerful, yet simple to install.
and maintain and be flexible enough to accommodate new detector geometries and technologies. All of the detector properties, not just the geometry, should be definable at runtime with an easy to use format. The output simulated detector response should be made available in a simple, well-documented format, allowing further processing or event reconstruction to be undertaken with a minimum of effort. The solution uses the Geant4 toolkit [1] to model the interaction of particles with matter and fields. A thin layer of non-Geant4 C++ code provides access to the event generator, the detector geometry description and the detector hits output. We selected XML as the textual file format for the geometry description, and extended the existing Geometry Description Markup Language (GDML) [2] for pure geometry description to incorporate missing features required for a full detector description. The simulation output is a simple binary format with bindings for multiple languages.

2.1. Geometry and Conditions Handling
GDML is an application-independent geometry description format based on XML developed at CERN for high energy physics applications. It can be used as the primary geometry implementation language or it can provide a geometry data exchange format for existing applications. It provides expressions, materials, solids, volume definitions and a geometry hierarchy. We adopted this as the basic geometric description and extended it to include detector identifiers, definitions of sensitive detectors, regions, physics limits and cuts, physics list definitions and electromagnetic fields, either as simple parameterizations (e.g. solenoidal or dipole fields) or as field maps. Conditions associated with a detector, such as sampling fractions, track finding strategies, etc. are bundled with the geometry description, stored in a .zip file and made available by detector name on the web.

2.2. Full Simulation using Geant4
The Geant4 toolkit is the de facto high-energy physics standard for simulating the interaction of particles with fields and materials. However, the end user is normally required to write their own C++ program to access the libraries, and the learning curve for setting up the detector geometry and defining sensitive elements and readout can be quite daunting. We have developed the detector response package, slic [3], which is based on the Geant4 toolkit but defines generic input and output data formats, and uses the xml file format described in the previous section to define the complete detector. This allows the end user to fully describe the detector geometry and readout at runtime using a plain text file. We provide executable programs for Windows, Mac OSX and several flavors of Linux. For persistence of the detector response, we developed LCIO [4], a simple event data model and persistency framework. The output from the detector response simulation consists of collections of generic tracker and calorimeter hits along with the complete Monte Carlo particle hierarchy, including secondaries produced in the simulation. LCIO is performant, with on-the-fly data compression and random access, well documented, with C++, Java, python and FORTRAN bindings.

3. Event Reconstruction
The Java-based reconstruction and analysis package org.lcsim [5] is based on a plug-and-play architecture which employs Drivers to perform discrete actions. Drivers interact with the Event by fetching objects or collections from the Event, processing them, and optionally adding results back to the Event. The Driver class interface provides the following methods:

- startOfData()
- getConditionsManager()
- process(EventHeader event)
- detectorChanged(org.lcsim.geometry.Detector detector)
although, in practice, most user classes only override the `process(EventHeader event)` method. The code runs either within the Java Analysis Studio (JAS) IDE for interactive use or standalone for batch or Grid production. The “write-once, run anywhere” feature of Java means that the exact same libraries run on all platforms (Windows, Mac, Linux(es) using the Java Virtual machine. A number of packages provide such functionality as: overlaying beam backgrounds at the detector hit level (including time offsets), digitization of readout electronics (CCD pixels, silicon microstrips, TPC pad hits), *ab initio* track finding and fitting, multiple calorimeter clustering algorithms, and individual particle reconstruction, also known as Particle Flow Analysis (PFA). Data analysis tools include an event browser and the Wired 3D interactive event display, whereas physics analysis tools include jet finding, event shape, vertexing and particle ID algorithms.

### 3.1. Monte Carlo Hit Digitization

Because we are interested in studying the effects of many tracking detector parameters, we chose not to perform the detector digitization during the Geant4 response simulation. Instead, we write out the simulated deposition of energy in sensitive detectors, and allow the end user to apply all digitization (e.g. development of the signals and electronic response) at the reconstruction or analysis stage. This allows us to rapidly and efficiently study the effects of pixel size and strip pitch as well as various readout technologies without having to rerun events through the full Geant4 simulation. After the effects of drift and diffusion of deposited charge is simulated, the electronics simulation adds noise, propagates the signal to the readout, applies a threshold and gain, and provides “raw” data in the form of channel IDs and ADC counts. Various clustering algorithms are provided which perform 1D clustering for strips and 2D clustering for pixel detectors. The clustering provides hit centroids and cluster-size dependent uncertainties which then serve as input to track finding and fitting.

### 3.2. Track Finding and Fitting

A number of track finding strategies are available within org.lcsim, supporting standalone pattern recognition for 1D (Si micro-strip) and 2D (Si pixel) hits. The efficiency for finding tracks is very high, even in the presence of backgrounds and at low momenta. Conformal-mapping pattern recognition is also available for 3D hits, applicable to a TPC. Pattern recognition initiated by extending track-stubs found in the highly segmented calorimeters into the trackers is also provided. Track fitting incorporating multiple scattering and energy loss via a weight matrix approach is currently used, with a full Kalman Filter approach becoming available.

### 3.3. Calorimeter Reconstruction

A number of clustering algorithms have been implemented which are used to support the Particle Flow Algorithm (PFA) approach to event reconstruction. One attempts to unambiguously associate all clusters in the calorimeter to their originating particle. For charged particles, one then uses the measured track momentum to replace the much less precise energy measurement. Clusters unassociated with tracks are either photons, measured reasonably well in the electromagnetic calorimeters, or neutral hadrons, which only represent a fraction of the total event energy. Using this technique, individual jet energy resolutions of three to four percent can be achieved.

### 4. Job Control

Exploring large regions of detector design phase space requires an easy way to configure not only the detector geometry, but also the reconstruction processors and their associated control
parameters. Again, we have adopted an xml format for our run control steering files. In it we can not only define the input and output files, but also the number of events to process, the Drivers to run, along with arguments to any of the algorithms.

5. The ILC LOI Exercise

The set of applications described here has been successfully used to design the Silicon Detector [6], one of the two concepts currently being investigated to study the physics of high energy electron-positron collisions at the International Linear Collider (ILC) [7]. The optimization process for the tracker design started out with a number of simplified geometries, with the complexity of the simulations increasing as the designs became more mature. The current model is quite sophisticated, including most of the details of the engineering models for the support and assembly of the detectors, as well as the electronic readouts currently being considered. Figure 1(a) shows a cross section of a CAD model of the central tracker. The corresponding Geant4 model is shown in cutaway in Figure 1(b).

The Letter of Intent (LOI) process required a number of physics analyses to be conducted with full-detector simulation, ab initio event reconstruction, and analysis. The physics benchmark processes were deliberately chosen to highlight the intrinsic detector performance, to facilitate comparisons between the concept designs. Although still far from real, the physics benchmarking requirements presented the community with a large-scale, end-to-end exercise which stressed most aspects of the software systems, including:

- Event Generation
- Detector Simulation
- Event Reconstruction
- Physics Analysis

On the order of a hundred million events were fully simulated and reconstructed as part of the data challenge. Extensive use was made of the LCG (primarily DESY, RAL Tier 1 and IN2P3) and OSG (primarily on the FermiGrid) grids. In general, no problems were encountered with the concept software. All the classes were bundled into a single jar file, and the JVM was shipped to grid nodes if needed. A number of issues with Grid job submission, monitoring and file transfers were encountered, since the Grid was still high-maintenance and very LHC-centric.

6. Interoperability and User Base

Although the org.lcsim software suite is fairly fully featured, there exist tools written in C++ which offer additional functionality, or which could be used to cross-check results. The common event model and persistency format allow tools developed in other regions, other languages or
other analysis frameworks to be used to process events. LCIO files can be analyzed using root by importing a root LCIO dictionary. The org.lcsim package was developed by and for the ILC physics and detector community, but is being adopted by a wider community. Recently it has been adopted by CERN as one of two frameworks for detector simulations for the CLIC physics CDR [8]. It has also been adopted for the Fermilab-based Muon Collider physics and detector studies [9]. Two recent heavy-photon search proposals at the Thomas Jefferson National Accelerator facility also used this package for their initial detector response simulations. The Fermilab dual-readout crystal calorimetry group has contributed a number of improvements to the handling of optical processes as part of their detector studies, and a group at SLAC National Accelerator Laboratory has used org.lcsim for ATLAS pixel upgrade simulations.

7. Summary
The ALCPG simulation group supports an ambitious physics and detector simulation, reconstruction and analysis effort. The goal is flexibility and interoperability which is not technology or concept limited. It provides a complete and flexible detector simulation package capable of simulating arbitrarily complex detectors with runtime detector description. The reconstruction and analysis framework exists, core functionality is available, an individual particle reconstruction has been developed, and various analysis algorithms have been implemented. It has been used extensively in support of studies demonstrating the scientific merit and feasibility of detectors at the International Linear Collider, specifically the successful validation of the Silicon Detector (SiD) Concept. In addition, it is being adopted by an increasingly larger group of users and being adapted to a broader range of applications.

Acknowledgments
I would like to thank the members of the ALCPG simulation and reconstruction team for their contributions, especially Jeremy McCormick. This work was supported in part by the US Department of Energy, Office of High Energy Physics.

References
[1] http://geant4.cern.ch
[2] http://gdml.web.cern.ch/GDML
[3] http://www.lcsim.org/software/slic
[4] http://lcio.desy.de
[5] http://lcsim.org
[6] http://silicondetector.org
[7] http://www.linearcollider.org
[8] http://clic-study.org/
[9] http://www.fnal.gov/pub/muon_collider/