Development, validation and maintenance of Monte Carlo event generators & generator services in the LHC era

M. Kirsanov

(INR Moscow)

LCG Generator

GOAL: to guarantee the event generator support for the LHC

WP1: GENERATOR SERVICES LIBRARY (GENSER)

WP2: EVENT FORMATS AND EVENT INTERFACES

WP3: SHARED EVENT FILES: FRAMEWORK & DATA BASE (MCDB)

WP4: TUNING AND VALIDATION

CERN (Library)
FNAL, CERN (Event Interfaces)
LCG-Russia (Library, Data Base)
Collaboration with independent projects: CEDAR, HepForge (Validation, New MCs)
Contact persons/Collaborators in MC Projects and LHC Experiments

Started May 2003
Long Term Project

Subproject leader: A. Ribon (CERN)
Monthly meetings and 1- 2 times per year planning meetings

WP1. The LCG Generator Library (GENSER)

GOAL: to replace the obsolete CERN Library for what concerns the Generator Services

→Mandate:

- **To collaborate with MC authors to prepare LCG Compliant Code**
- **To maintain older MC packages on the LCG supported platforms**

→Clients:

- **Addressed to LHC experimentalists and theorists both at CERN and in external laboratories (Other users are welcome!)**

- ✓ CVS Repository, AFS Distribution
- ✓ At version 2 now: GENSER2
- ✓ MC Packages,
- ✓ Test, Validation Packages
- ✓ Tested by the LHC experiments
- ✓ Continuous Release Scheme

ATLAS, CMS & LHCb PRODUCTIONS RELY ON GENSER

Documentation: http://lcgapp.cern.ch/project/simu/generator
Savannah Portal: https://savannah.cern.ch/projects/genser
AFS: /afs/cern.ch/sw/lcg/external/MCGenerators

WP1. The GENSER Team

- GENSER Integrators & MC Experts ~3FTE
  - M. Kirsanov INR (Moscow) Master Integrator
  - O. Zenin IHEP (Protvino) Integrator
  - A. Ryabov IHEP (Protvino) Intergator
  - S. Bityukov IHEP (Protvino) Integrator

WP1: GENSER. History.

- GENSER1 was based on SCRAM
  - Insufficiently flexible
  - Unfamiliar to many users
  - Quarterly release scheme insufficiently flexible
  - Decided to move to GNU Make build systems
- GENSER2 Since the end of 2006
  - GNU Make build system is provided by GENSER for generators distributed just as a single source file (pythia6, herwig6). "configure" script is also provided being made compliant to the adopted standards (arguments –help, --with-package, treatment of *FFLAGS env. Variables etc.) as far as possible . Directory /examples with a similar build system
  - GENSER also provides a build system for some generators with insufficiently developed author's build system (agreed with authors)
  - Author's build system used if sufficiently developed
  - GENSER installation scripts kept in CVS for each package, invisible for users

WP1: GENSER

- GENSER2
  - Continuous release scheme: a new version of a generator is installed as soon as released by authors
  - Distribution of generators is performed through the tarballs kept on CERN AFS in MCGenerators/distribution/. For each generator the source tarball (with the author's or GENSER-provided build system) and binary tarballs for each LCG-supported platfom are provided. Binary tarballs usually contain minimum that is necessary to work with a generator without recompiling it and the directory /examples
  - Security: old GENSER versions are never removed. Normally, new generator versions are installed into dedicated AFS volumes. After internal testing the volume becomes read-only. Only one person is allowed to create new volumes and set their permissions.
  - Standard LCG platforms slc4_ia32_gcc34 and slc4_amd64_gcc34 are fully supported. More about platforms later in this talk.
  - 28 MC generators and auxiliary packages are currently in GENSER2

WP1. Testing

Large scale MC productions involve a lot of manpower, they are are expensive

A bug in MC generator, depending on its severity, can make a production mostly useless

Test thoroughly!
Test quickly!

Detailed validation of MC, comparison of distributions with previous versions and data, can take a lot of time.

Several levels of tests and validations can help

WP1. Testing

- Level 0 test. The table of generators for the GENSER web page is created automatically with a special script that also checks the existence of libraries and tarballs
- Level 1 test. Dedicated package in GENSER. It compiles one or several applications for each generator to be tested and runs them one by one. Each application computes several observables, their values are compared to the reference ones stored in a file. Problems and statistically significant deviations are reported

Next tests are outside WP1. Additional test levels can be used

- Level 2 tests. Dedicated projects (WP4)
- Level 3 tests. Tests inside the experiments

WP1. Level 1 testing, more details

- Dedicated package in GENSER: TESTS, additional dependencies, like ROOT, moved here
- The results are often correct, but not identical, even if random numbers sequence is repeated: it is Monte Carlo! Need to look for significant changes, taking into account statistical errors.
- Comparison is made by a script. It detects missing results in case of test crash and continues
- All MC generators and most important auxiliary packages (like lhapdf) are now involved in this testing. However, the complexity of the tests for different generators is still very different: from several hundred observables, some of them obtained after a complicated event analysis to only one number, e.g. cross section. The work is to be continued

WP1. Level 1 testing, more details (2)

- The testing of all GENSER (last versions) takes about 6 hours on 3000 MIPS machine (shared and archive libraries are tested separately, generators subversions with HEPEVT common block sizes 10000 and 4000 also separately)
- Several bugs were found with level 1 testing this year
- GENSER also included some tests adopted in LHC experiments into Level 1 test suite. This process will also continue

WP1. Package management

Currently we are testing mature *pkgsrc* package management system (comes from Berkeley UNIX, runs on **all** POSIX platforms: GNU/Linux, *BSD, Mac OS X, Solaris, HP-UX, AIX, Cygwin/NT, etc.) to be adopted as an OS-independent base for GENSER.

- Transparently fetch, extract, patch, configure && build && install MC generators and related packages
- Transparently prepare patched source tarballs that can be used in a standalone mode
- Transparently account for some system-dependent details: can help if a package native build system not fully portable
- Automatic dependency tracking
- Can automatically build entire GENSER on a standalone site
- Several most important generators migrated
- Will be GENSER3 if adopted

WP1. Package management (2)

We are evaluating AutoTools as a GENSER – provided build system for generators distributed without one.

- Works not only on Linux, but on many other OS (MacOSX etc.)
- Supported, will include new OS if they appear
- Is more complicated (the “configure” script can be bigger than all the generator code:)
- Sometimes not easy to make it flexible enough
- Some caveats with portability of libraries (moving them to another location) having to do with -rpath

WP2. Event Formats and Event Interfaces

GOAL: standardize interfaces, support the new OO MCs

- **HEPMC – the MC Truth Interface**
  - → **January 2006: installed in LCG external, based on the version 1.26 from M. Dobbs. Interfaces to generators split in a separate library. Further development started**
  - → **Version 2: made independent of CLHEP**
  - → **This year all LHC experiments moved to version 2**
  - → **At version 2.04.00 now after the discussion at the dedicated GENERATOR meeting and revision**
- **HEPML proposal (XML - Les Houches Agreement I compliant)**
  - → **Meta-data format facilitating automated documentation**

WP3. Monte Carlo Data Base (MCDB)

- **Motivations**
  - **To Provide Configuration, Book-keeping, Documentation, Storage for the Shared Event Files**
  - **To keep track of the full generation chain, Exploiting the Competences of Monte Carlo Experts and Monte Carlo Authors**
- **In Production. Only minimal support now. Used by CMS**

WP4. Validation and tuning

- RIVET – Robust Independent Validation of Experiment and Theory, developed by the CEDAR collaboration as a C++ replacement for HZtool
- Three examples of Experiment to MC generator comparison were created by the GENSER team
- Experimental results are taken directly from the Data Base

WP4. Validation and tuning

- MC-TESTER is now being implemented as one of the elements of GENSER validation
- Compares two generators or two versions of a generator as to how they simulate kinematical distributions for various particle decays
- Some results already can be seen on the GENSER WEB page

Conclusions

- **GENSER2 and MCDB in production**
- **Level 1 validation package TESTS in GENSER tests all available generators**
- **RIVET and MC-TESTER are also used for the generators validation**
- **New cross-platform tools (PKGSRC, Autotools) are being evaluated**
- **Feedbacks, bug reports, requests from LHC experiments and external users are welcome**