GENSER

(part of LCG Generator Services project)

Witek Pokorski

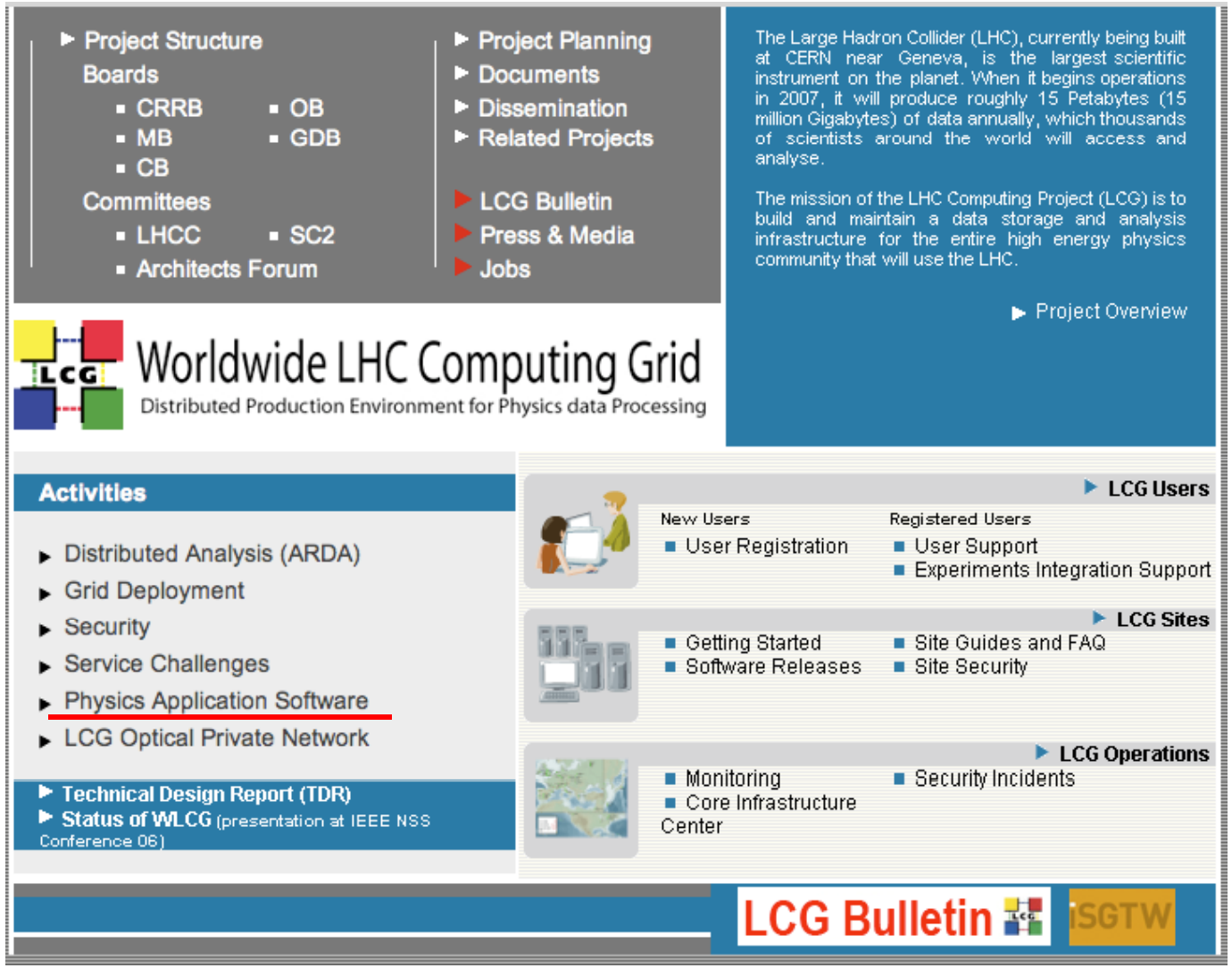
01.11.2007

HERA-LHC workshop, DESY

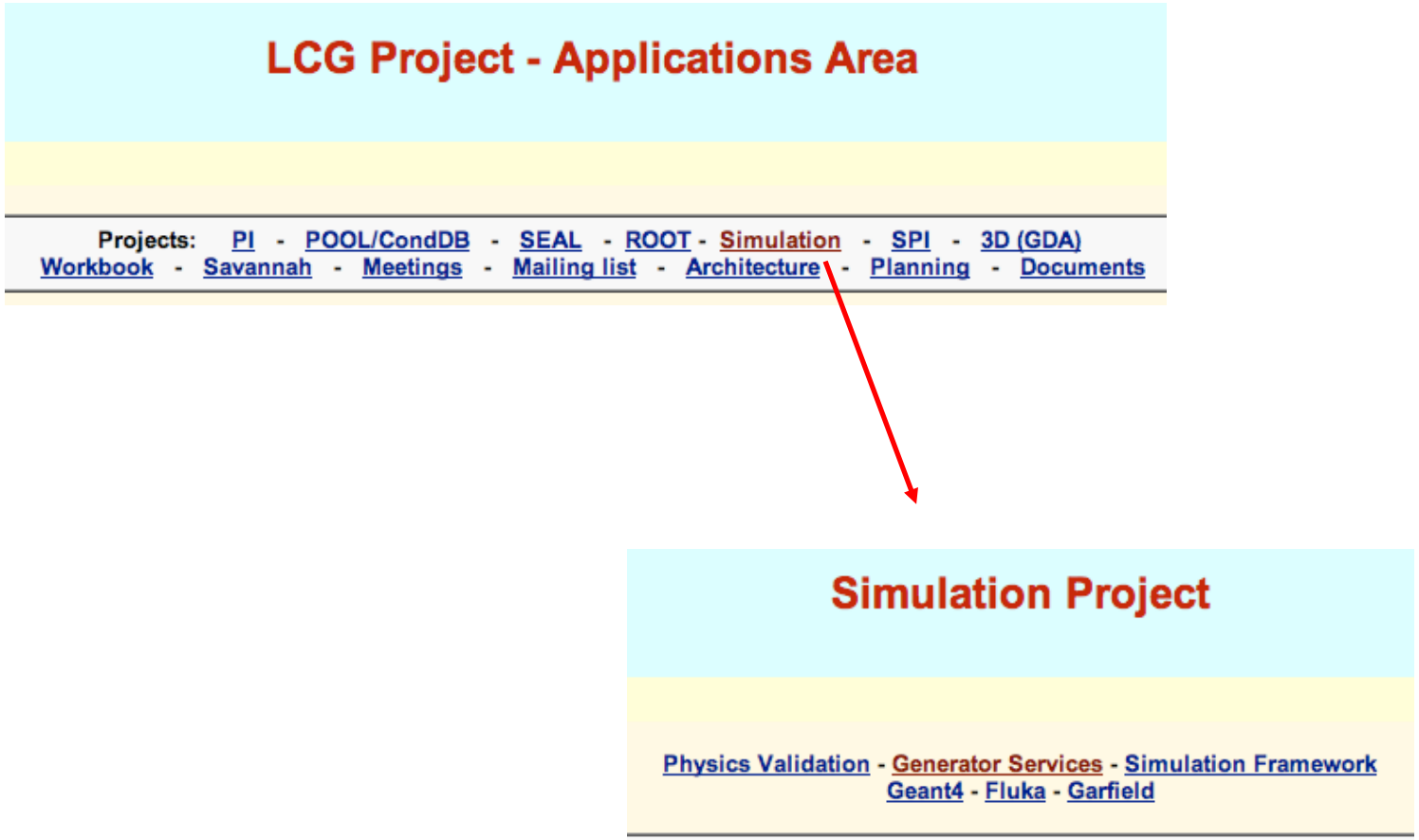
Outline

- Overview of LCG Generator Services project
- GENSER structure
- Generators validation
- Conclusion

LCG project


Project Structure
Boards
CRRB
OB
MB
GDB
CB
Committees
LHCC
SC2
Architects Forum
Project Planning
Documents
Dissemination
Related Projects
LCG Bulletin
Press & Media
Jobs
The Large Hadron Collider (LHC), currently being built at CERN near Geneva, is the largest scientific instrument on the planet. When it begins operations in 2007, it will produce roughly 15 Petabytes (15 million Gigabytes) of data annually, which thousands of scientists around the world will access and analyse.
The mission of the LHC Computing Project (LCG) is to build and maintain a data storage and analysis infrastructure for the entire high energy physics community that will use the LHC.
Project Overview
LCG
Worldwide LHC Computing Grid
Distributed Production Environment for Physics data Processing
Activities
Distributed Analysis (ARDA)
Grid Deployment
Security
Service Challenges
Physics Application Software
LCG Optical Private Network
Technical Design Report (TDR)
Status of WLCG (presentation at IEEE NSS Conference 06)
LCG Users
New Users
User Registration
Registered Users
User Support
Experiments Integration Support
LCG Sites
Getting Started
Software Releases
Site Guides and FAQ
Site Security
LCG Operations
Monitoring
Core Infrastructure Center
Security Incidents
LCG Bulletin
iSGTW

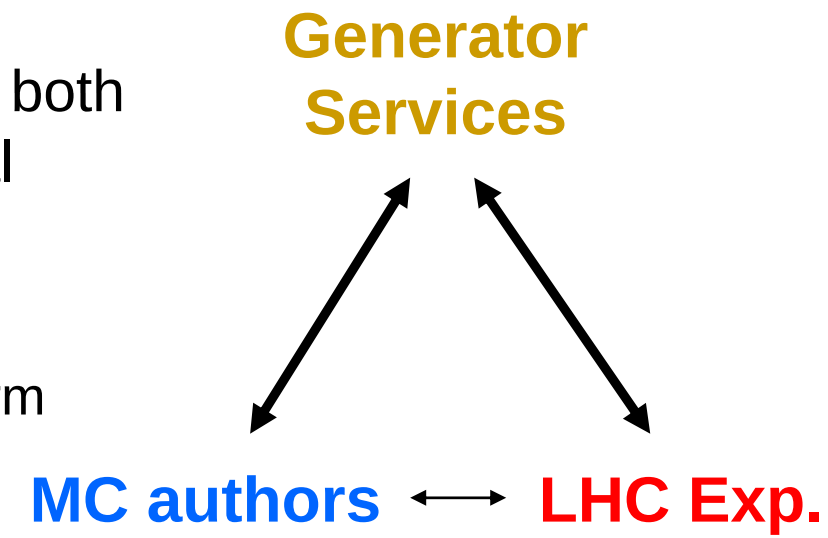
LCG Application Area Simulation Project



LCG Generator Services

http://lcgapp.cern.ch/project/simu/generator/

- mandate of the project:
  - "...to prepare validated LCG compliant (generators) code for both the theoretical and experimental communities at the LHC..."
- to avoid duplication of work
  - to build libraries for required platform
- to share experience between experiments
- to use common generators (tunings?)
- to offload authors from the 'basic support' duties



Project work packages

- generator libraries repository [GENSER]
- testing and validation of generators [VALIDATION]
- first level support [SUPPORT]
- event record and particle properties [HEPMC]
  - maintained by Lynn Garren (FERMILAB)
- event database [MCDB]

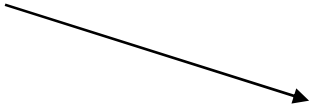
GENSER

- centralized installation of all the MC generators used by LHC experiments on all the LCG supported platforms
- common structure for all the generators
- ready to use libraries
- tarfiles with binaries
- tarfiles with sources

Repository structure (1/3)

```
/afs/cern.ch/sw/lcg/external/MCGenerators
                                        /pythia6
                                        /pythia8
                                        /herwig
                                        /herwig++
                                        /jimmy
                                        .....
                                        /distribution/..
```

LCG tar files with sources and binaries



- **For each generator:**

```
pythia6/411
       /412
       ....
```

Repository structure (2/3)

- **For each version:**

```
411/share
   /slc3_ia32_gcc323
   /slc4_amd64_gcc34
   ...
```

- **For each platform:**

```
slc4_amd64_gcc34/compile.log
                /config.mk
                /include/
                /lib/libpythia6.so
                /lib/archive/libpythia6.a
```

Repository structure (3/3)

- tarfiles:

```
/afs/cern.ch/sw/lcg/external/MCGenerators/distribution/

                                        pythia6-411-src.tgz
                                        pythia6-411-slc3_ia32_gcc323.tgz
                                        pythia6-411-slc4_amd64_gcc34.tgz
                                        pythia6-411-slc4_ia32_gcc34.tgz
```

Using GENSER

- to use libraries from AFS
  - obvious
- to use binary tarfiles
  - obvious
- to use source tarfiles
  - tar zxvf pythia6-413-src.tgz
  - cd pythia6/413
  - ./configure --help
  - ./configure --your-options
  - make
  - libraries go to pythia6/413/lib/

Available generators (1/2)

Overview of available MC event generators

deprecated	supported	not validated yet

alpgen	2.1.3	2.1.2	2.1.1									
cascade	1.2.10											
charybdis	1.003											
evtgenlhc	8.15.1	8.15	8.14									
herwig	6.510	6.510.2	6.510.3									
herwigpp	2.0.3	2.0.2	2.0.1									
hijing	1.383bs.2											
hydjet	1.2	1.1										
isajet	7.75	7.75.2	7.69	7.69.2								
jimmy	4.31	4.31.2	4.31.3	4.2								
lhapdf	5.3.0	5.2.3										
phojet	1.10	1.10.2										
photos	215	215.2	215.3									
pyquen	1.2	1.1										
pythia6	412	412.2	411	411.2	411.3	410	410.2	409	409.2	326	326.2	227	227.2
pythia8	095	090	080	070	060								
sherpa	1.0.10	1.0.9	1.0.8									
stagen	1.11											
tauola	27.121	27.121.2	27.121.3									
thepeg	1.0.1											
toprex	4.23											

Available generators (2/2)

- over 20 different generators available
  - FORTRAN and the new C++ generators
- new versions installed with minimal delay
  - for ex. six versions of Pythia6 already installed
- binaries provided for 3 platforms (Linux)
  - request to install some generators on Windows
- new generators added on experiments' request

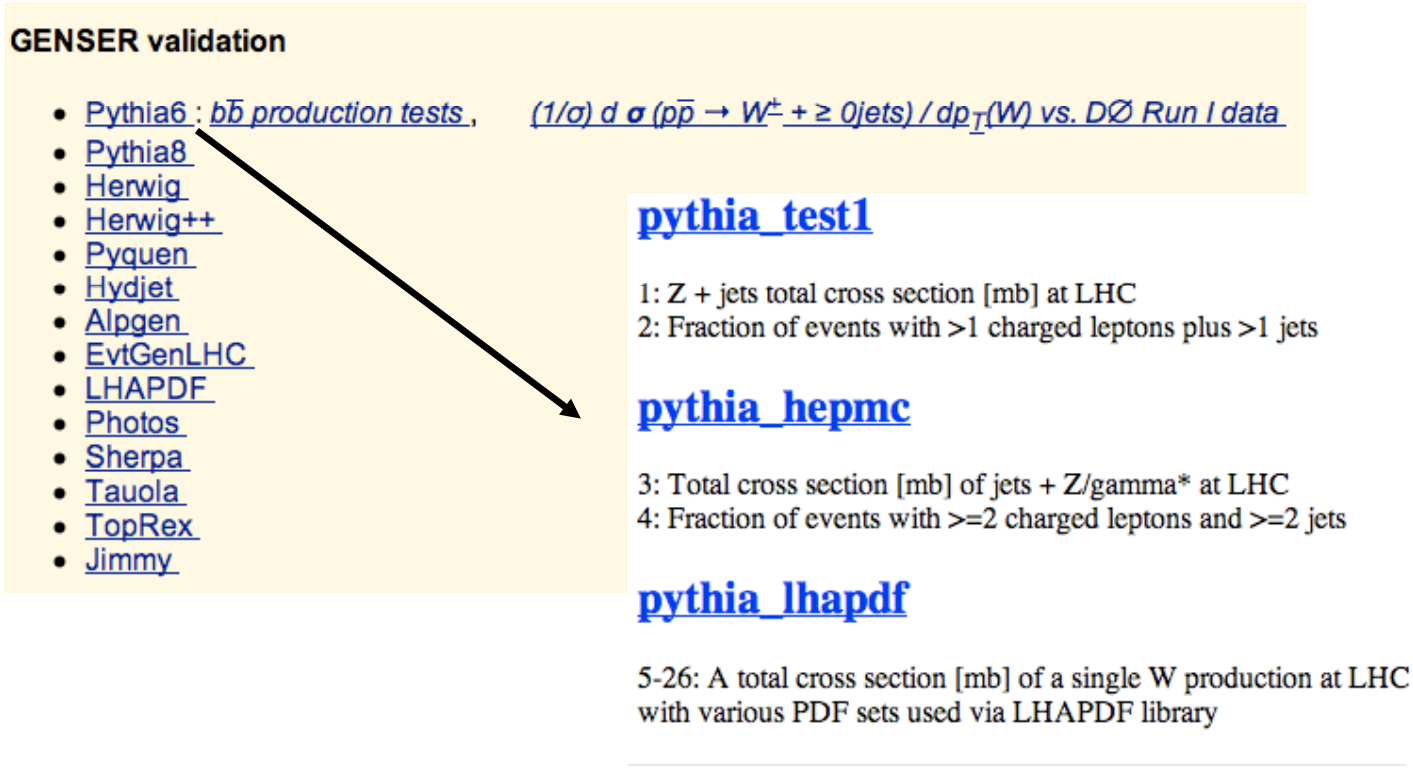
Testing and validation

- experiments used to independently test and validate each new version of the generator
  - clear duplication of work
- GENSER testing and validation
  - testing of generators on different platforms
  - comparing different (new) versions of each generator
  - physics validation (comparing to data)

GENSER testing

- simple tests
  - 'single number' output (charged multiplicity, etc)
- histogramming tests
  - distribution output (pT, etc)
  - needs to be linked with ROOT
- physics validation
  - implemented first analysis using Rivet
  - Rivet will be used for all future validation

GENSER simple tests

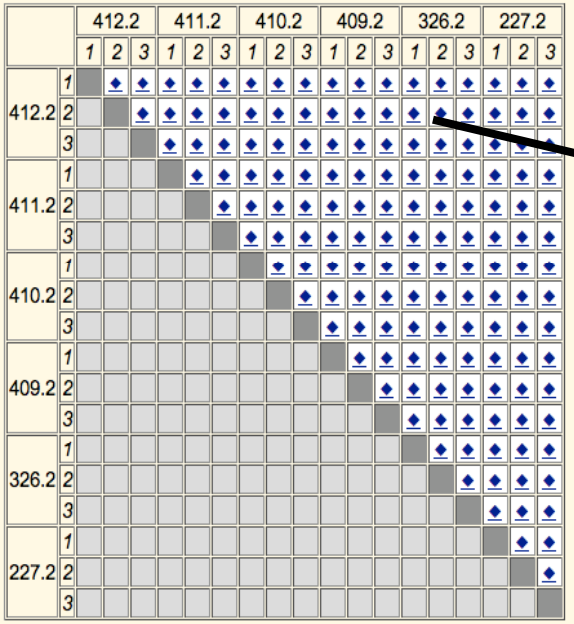


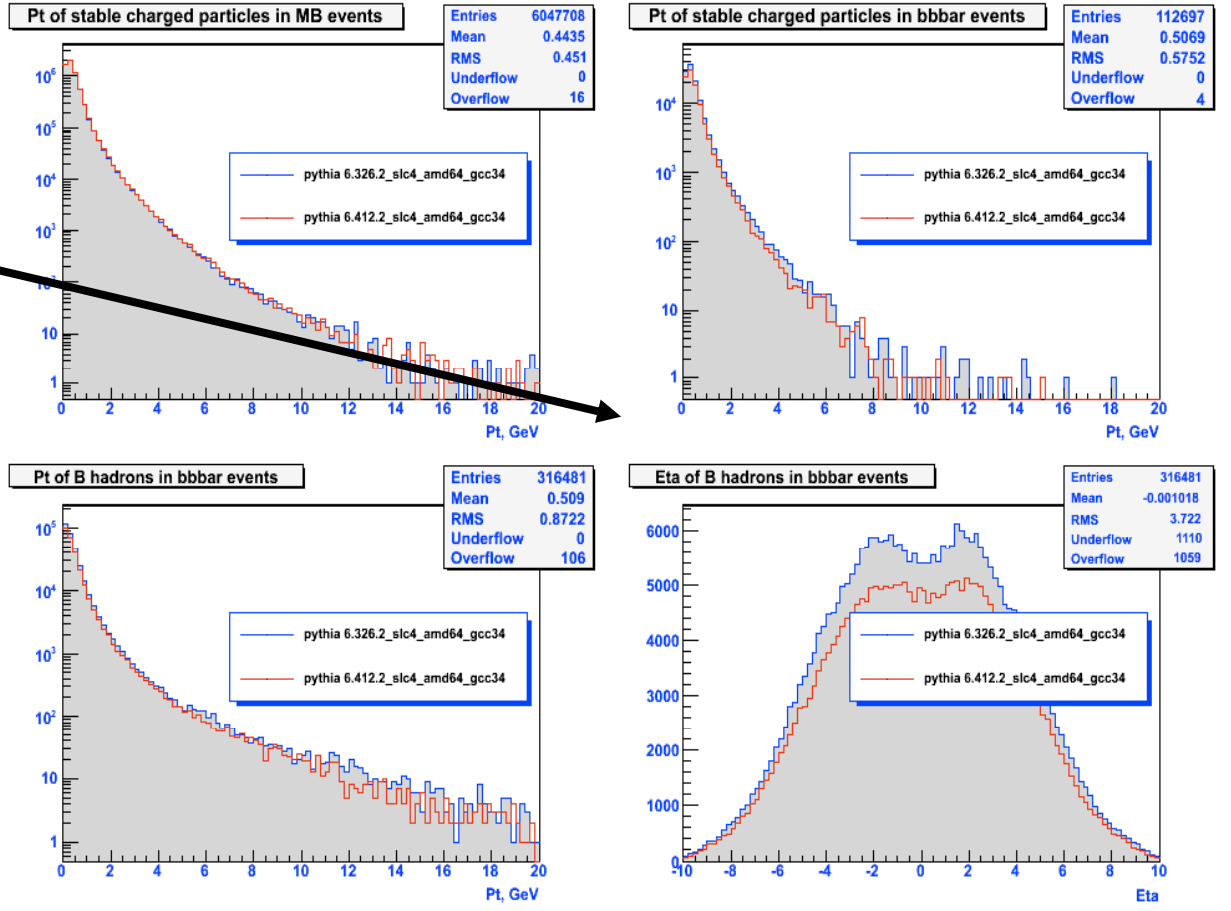
- at least one simple test per generator
- automatic checking between different versions of generators and platforms

GENSER distribution tests

Pythia6, $b\bar{b}$ production

1. slc3_ia32_gcc323
2. slc4_amd64_gcc34
3. slc4_ia32_gcc34





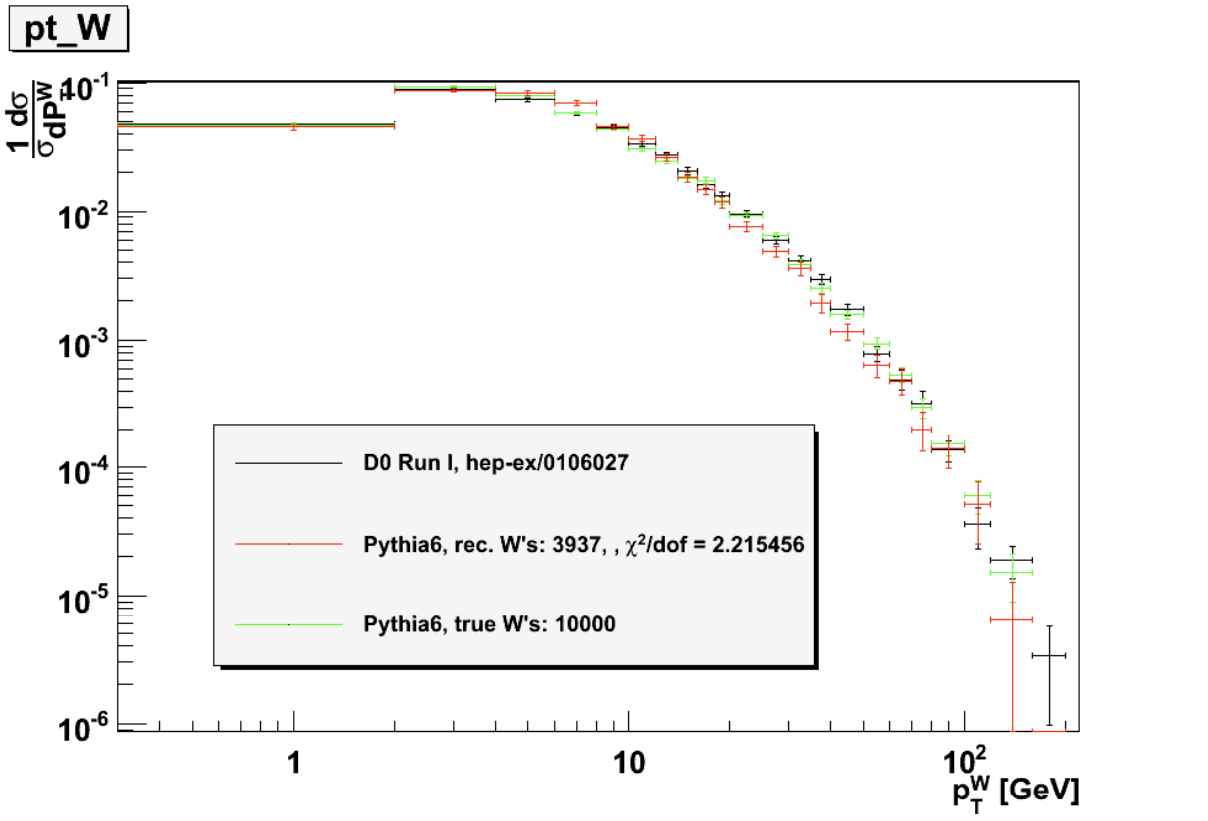
GENSER validation

MC vs. DØ Run I, $(1/\sigma)\, d\,\sigma\,(p\bar{p} \rightarrow W^{\pm} + \geq 0 jets) / dp_T(W)$

A comparison of $p_T(W^{\pm})$ distributions obtained with Pythia 6.411 vs. DØ Run I data:
Abazov, V.M., et al. [DØ Collaboration], hep-ex/0106027

A skeleton of comparison procedure is as follows:

1. The main routine linked to **Rivet** library instantiates HepMC::GenEvent. The analysis procedure utilizes selects events according to the criteria from the DØ
   - an isolated (as defined in the publication, see
   - missing p_T > 25 GeV.
2. The main routine launches an event cycle with each
3. Each event is passed to UserAnalysis::analyze() me W candidate to be put to a ROOT histogram.
4. Upon completion of the event cycle the routine calls
5. The histogram is read from the ROOT file and upon experimental data points from the publication. χ^2/d.c



Conclusions

- GENSER provides repository for MC generators for LHC experiments
- GENSER helps experiments to resolve particular issues related to generators
  - aim to participate more in tuning of generators (forum for a discussion between experiments)
- GENSER wants to fully rely on (to use) RIVET for the validation of the generators