Future
Status
History

Lund University
Department of Theoretical Physics
Torbjörn Sjöstrand

Whither PYTHIA?

Lund University

12 February 2004
Collider Physics Workshop
KITP Santa Barbara
Q: Why rewrite?
A: Need to cleanup!

Q: Why C++?
A: Only game in town!

My original ideas:

- Simple and robust structure
- Keep current physics unchanged
- Throw out-of-date alternatives
- Many minor improvements

Q: Origin
Q: Why rewrite?
A: Need to clean up!
Q: Why C++?
Many physics models:

- LUND CAMP PLURALISTIC
- HERWIG CAMP MONOLITHIC
- JETSET, PYTHIA
- LEPTO, LUCIFER
- FRIOTIO, LUCIAC
- ARIADNE, LDCM

Many programs:

- JETSET, PYTHIA
- RAPGAP, CASCADE
- LEPTO, LUCIFER
- FRIOTIO, LUCIAC
- ARIADNE, LDCM
- HERWIG, HIJING, ...
Evolution

Leif's idea:

before you worry about physics,
create a generic platform for event generation
"a language within the language"

The Rest: physics

basic structure must be physics-neutral

ThePEG: administration

HERWIG++ accepted and joined

PYTHIA7

ThePEG: administration

PyTHIA7
What is ThePEG?
What is in the Rest?

- Processes: QCD
- Showers: Initial- and final-state (old PYTHIA) (TS)
- Multiple interactions: none
- Beam remnants: Ariadne (LL)
- Fragmentation: Simple string (Marc Bertini)
- Decays: Most implemented (LL)
- Beam remnants: Ariadne (LL)
- Multiple interactions: none
- PDF: GRV 94 series (LL)
- Processes: QCD 2, e⁺e⁻ → qg (LL)

Mainly simple pieces done => almost all the hard work remains

Not useful for physics studies

- Need continuity => as above
- Not convenient to use postdocs
- The physics hurdle is as steep as the C++ learning curve
- Conversion effort: Everything takes longer and costs more

No work for graduate students

- Need continuity => as above
- Not convenient to use postdocs
- The physics hurdle is as steep as the C++ learning curve
- Conversion effort: Everything takes longer and costs more

Not useful for physics studies
Afreshstart

Lower priority than teaching, administration, master's and graduate students, answering PYTHIA questions, maintaining Fortran code...

Leif's interests are ThePEG, Ariadne, LDC,...

bad finances in Lund

...Leif's interests are ThePEG, Ariadne, LDC,...

maintaining Fortran code,...

and graduate students, answering PYTHIA questions,

Lower priority than teaching, administration, masters

Solution?: take a sabbatical and work "full-time"!

Objectives:

( Software development for experiments)


concentrate on physics, not administration

pure standard C++, no fancy programming tricks

independent of ThePEG (or anything else), but

written to be modular, i.e. easy to interface

interface to ThePEG later written by Leif(?)

A Fresh Start
New structure

Event record, parameters, data, PDF's, utilities

Particle decays

String fragmentation

Multiple interactions and beam remnants

Initial- and final-state showers

(Remaining resonance decays)

Les Houches Accord (style) Interface

internal

ThePEG

user

Hadron-level generator

borderline

Parton-level generators
Technical Notes (1)

Remaining resonance decays (Z, W, H, SUSY, ...):

- Internal or SUSY less Houches Accord decay tables

String fragmentation:

- Implement baseline model, minor physics improvements
- can use existing matrix-element matching code
- Implement the \(T^d\)-ordered algorithms
- Introduce L-CKKW-style mixing

Multiple interactions and beam remnants:

- Based on new scheme under development
- Ordered algorithms

Initial- and final-state showers:

- Primitive angular correlations
- Implement the \(T^d\)-ordered algorithms

- Ordered algorithms
- Internal or SUSY less Houches Accord decay tables

- High-mass strings

- Junction topologies

- Low-mass strings
Particle decays: updated decay tables (Bose-Einstein; overlap with fragmentation)

Event record, parameters, data, PDF's, utilities:
- PYTHIA-style event record with LHA colour tags
- LHAPDF parton densities
- SUSY evolution (use SLHA!) (WW/ZZ)
- colour reconnection
- SUSY evolution (use SLHA!)
- simple event analysis (for debugging)
- integrated manual/parameters/data in XML
- SUSY evolution (use SLHA!)
- Event record, parameters, data, PDF's, utilities:
- Bose-Einstein; overlaps with fragmentation
- update decay tables

Outside scope:
- many out-of-use options
- independent fragmentation
- old event annihilation machinery (QCD ME's)
Summary

- Complexity of problem underestimated (C++ & physics)
- "Slavelabour" not successful strategy (for me)
- Complexity of problem underestimated (C++ & physics)

Tentative schedule:

0 = Fall 2004
1 = Fall 2005
2 = Fall 2006
3 = Fall 2007

Year time hadron-level
begin new assault LHA-style input
incomplete draft

1 = Fall 2005
2 = Fall 2006
3 = Fall 2007
stable, debugged a few processes complete, buggy (?)

- don't forget Murphy's law

Summary

...
ThePEG defines a set of abstract handler classes for hard partonic sub-processes, parton densities, QCD cascades, hadronization, ...

Handlers
The class provides access to a lot of information, but it only has a pointer to a ParticleData, a LorentzTransform and a pointer to another object carrying the rest of the information (colour, spin etc.).

Some of this information can be user-defined by creating classes inheriting from e.g. the EventBase class. This information can then be accessed through dynamic casting. The EventBase class provides access to a lot of information. But it only has a pointer to a ParticleData, a LorentzTransform and a pointer to another object carrying the rest of the information (colour, spin etc.).
Running ThePEG

The end-user will use a setup program to be able to pick objects corresponding to different physics models to build up an EventGenerator, which then can be run interactively or off-line, or as a special slave program responding to different physics models. The end-user will use a setup program to be able to pick objects corresponding to different physics models to build up an EventGenerator, which then can be run interactively or off-line, or as a special slave program responding to different physics models. Currently there is only a rudimentary command-line interpreter. A flashy Graphical User Interface should be built on top of this repository. The repository is the central part of the setup phase. It handles a structured list of all available objects and allows the user to manipulate them.

The repository is the central part of the setup phase. It handles a structured list of all available objects and allows the user to manipulate them. Currently there is only a rudimentary command-line interpreter. A flashy Graphical User Interface should be built on top of this repository. The repository is the central part of the setup phase. It handles a structured list of all available objects and allows the user to manipulate them.

In the end of the run you will get a number of files with statistics and messages. And a LaTeX-file with references suitable for inclusion in an appendix of a paper.