



LUND UNIVERSITY



LCG Generator Services Meeting
CERN
28 July 2005

PYTHIA 8 — The First Release

Torbjörn Sjöstrand

CERN/PH and

Department of Theoretical Physics, Lund University

PYTHIA 8.040 released on 20 July 2005

Available on Pythia webpage

<http://www.thep.lu.se/~torbjorn/Pythia.html>

clicking on the “Future” link in the index

What is in it?

On To C++

Currently HERWIG and PYTHIA are successfully being used,
also in new LHC environments, using C++ wrappers

A1: Need to clean up!

Q: Why rewrite?

A2: Fortran 77 is limiting **Fortran 90**

A3: Young experimentalists will expect C++

PYTHIA7 project \implies **ThePEG**

Toolkit for High Energy Physics Event Generation

(L. Lönnblad; S. Gieseke, A. Ribon, P. Richardson)

HERWIG++: complete reimplementaion

(B.R. Webber; S. Gieseke, A. Ribon, P. Richardson, M. Seymour, P. Stephens, 3 new)

ARIADNE/LDC: to do ISR/FSR showers, multiple interactions

(L. Lönnblad; N. Lavesson)

SHERPA: in C++ from start, partly wrappers to PYTHIA Fortran

(F. Krauss; T. Gleisberg, S. Hoeche, A. Schaelicke, S. Schumann, J. Winter)

PYTHIA8: A fresh start

Problem: PYTHIA7 stalled, no other manpower

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

(⇒ baseline model, S. Mrenna & P. Skands join later ?)

Tentative schedule:

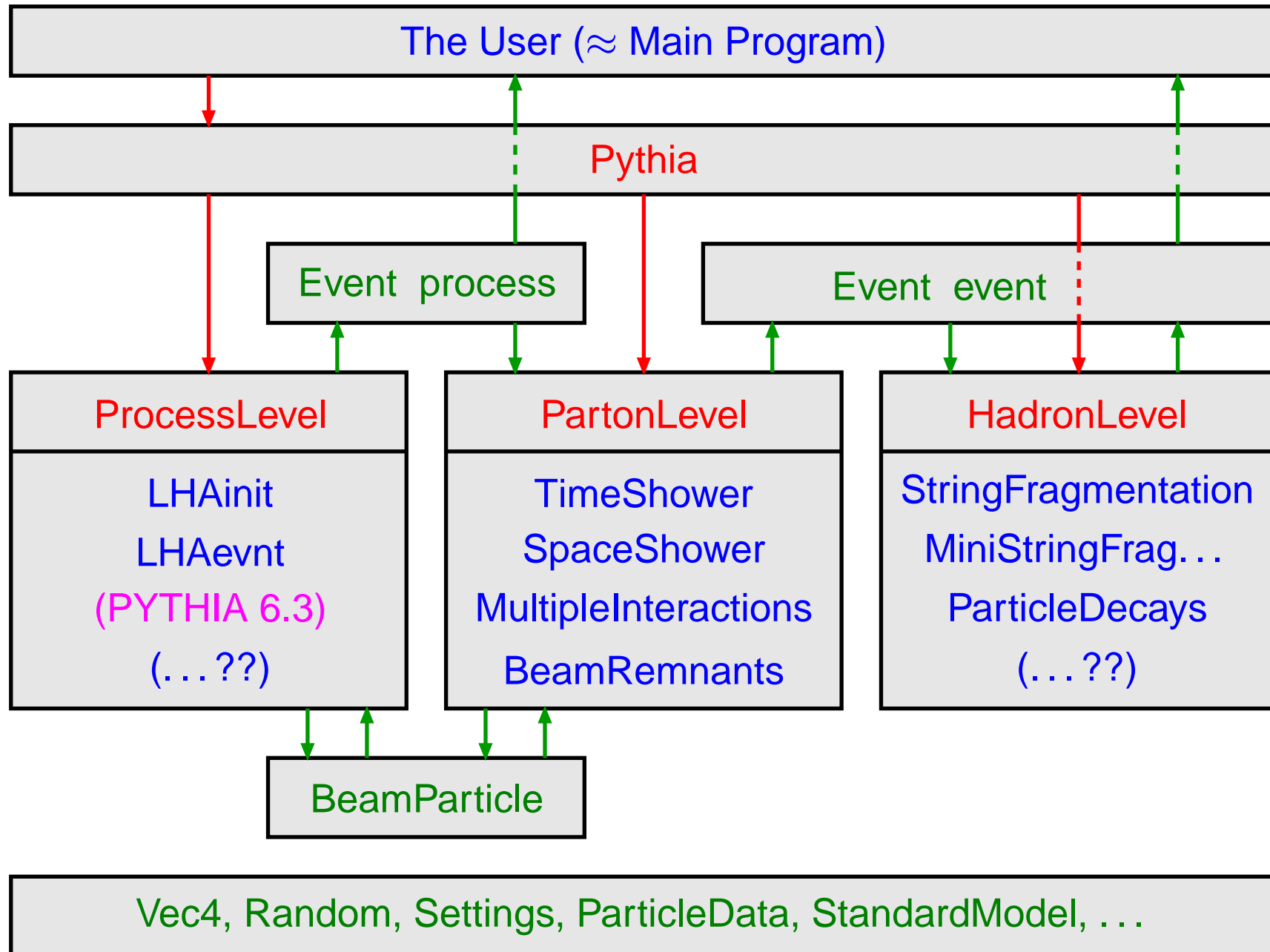
time	date	processes	final states
0 =	1 Sept. 2004	—	—
1 =	1 Sept. 2005	LHA-style input	incomplete draft
2 =	1 Sept. 2006	a few processes	complete, buggy(?)
3 =	1 Sept. 2007	more processes	stable, debugged

... but don't forget Murphy's law

Objectives:

- clean up, keep the most recent models
- Les Houches Accord style input central
- independent of ThePEG (or anything else), but
- interface to ThePEG later written by L. Lönnblad (?)

Current PYTHIA8 structure



Current PYTHIA8 status

Existing classes			Missing classes/topics
Process	LHAinit	**	Cross section administration Phase space selection Process matrix elements
Level	LHAevnt	**	
	(PYTHIA 6.3)	* * *	
Parton	TimeShower	**	Parton density libraries Resonance decays ThePEG input (?)
Level	SpaceShower	**	
	MultipleInteractions	**	
	BeamRemnants	**	
Hadron	StringFragmentation	**	MI/ISR/FSR interleaving colour flow models ME/PS matching
Level	MiniStringFrag. ...	**	
	ParticleDecays	**	
—	Event	**	Popcorn baryons updated decay tables Bose-Einstein
	BeamParticle	**	
	Vec4, Random	* * *	event analysis routines ... and much, much more
	Settings	**	
	ParticleData	**	

Distribution

Contents of Pythia 8.040 distribution:

no	Description	size
1	Introduction (.pdf)	20 pp
24	Header files (.h)	3,850 lines
22	Code files (.cc)	14,750 lines
1	PYTHIA 6.3 file (.f)	71,500 lines
25	Documentation files (.man)	4,700 lines
5	Sample main programs (.cc)	870 lines
3	Input to above	1,380 lines
1	Makefile	150 lines
<hr/>		
1	pythia8040.tar.gz (all)	1 MB

To get going: download \Rightarrow gunzip \Rightarrow tar xvf
 \Rightarrow make \Rightarrow run test programs(s)

Self-contained, but hooks to external programs for

- hard processes, Les Houches Accord style
- parton distribution functions
- decays (of some particles, e.g. τ , B^0 , B^+)
- random number generators (shared with other programs)

Event generation structure

1) Initialization step

- select process(es) to study
- modify physics parameters
- set kinematics constraints
- modify generator settings
- initialize generator
- book histograms

2) Generation loop

- generate one event at a time
- analyze it (or store for later)
- add results to histograms
- print a few events

3) Finishing step

- print deduced cross-sections
- print/save histograms etc.

```
#include "Pythia.h"
using namespace Pythia8;
Pythia pythia;
pythia.readString("command");
pythia.readFile("command.file");
pythia.init(idBeamA,idBeamB,eCM);
```

```
pythia.next();
pythia.process.list();
pythia.event.list();
int id = pythia.event[i].id();
```

```
pythia.statistics();
pythia.settings.listChanged();
```

Example of a main program

```
// Test program main06: study pTZ spectrum at the Tevatron.
#include "Pythia.h"
using namespace Pythia8;
int main() {
    // Generator. Process selection. Tevatron initialization. Histogram.
    Pythia pythia;
    pythia.readString("Pythia6:mse1 = 11");
    pythia.readString("Pythia6:ckin(1) = 80.");
    pythia.readString("PartonLevel:MI = off");
    pythia.readString("Beams:primordialKTwidth = 2.");
    pythia.init( 2212, -2212, 1960.);
    Hist pTZ("dN/dpTZ",100,0.,100.);
    // Begin event loop. Generate event. Skip if error. List first few.
    for (int iEvent = 0; iEvent < 10000; ++iEvent) {
        if (!pythia.next()) continue;
        if (iEvent < 2) pythia.event.list();
        // Loop over particles in event. Find last Z0 copy. Fill its pT.
        int iZ = 0;
        for (int i = 0; i < pythia.event.size(); ++i)
            if (pythia.event[i].id() == 23) iZ = i;
        pTZ.fill( pythia.event[iZ].pT() );
    }
    // End of event loop. Statistics. Histogram. Done.
    pythia.statistics();
    cout << pTZ;
    return 0;
}
```


Outlook

- **C++ PYTHIA 8 is coming along** ●

- ★ Roughly according to three-year plan so far! ★
- ★ ~ 1 sub-subversion per working week (backup) ★
- ★ Slowdown during autumn, then pick up again early 2006 ★
- ★ Release latest sub-subversion every 2–3 months ★
- ★ First useful — but incomplete — version in a year's time (?) ★
- ★ First production-quality release, 8.100, early 2007 (??)
- ★ Debugged and tuned by LHC startup (??) ★
- ★ Overtaking Fortran version usage by 2008–2009 (???) ★

- **Early feedback is most welcome** ●

- ★ Now is the time for any major course changes ★
- ★ In a year's time the structure will be frozen (?) ★