



CMS CPT week
CERN
12 May 2006

LUND UNIVERSITY

Status and Plans for PYTHIA 8

Torbjörn Sjöstrand

CERN/PH and

Department of Theoretical Physics, Lund University

PYTHIA 6 status

PYTHIA 6 still being actively developed and maintained:

- multiple interactions and underlying event, with
- transverse-momentum-ordered showers
- SUSY interfaces (SLHA) and simulation
- regular bug fixes and minor improvements
- plan to move to CEDAR HepForge (code management, bugtracking)

Currently PYTHIA 6.402:

- 73,600 lines of code (including comments/blanks)
- 580 page PYTHIA 6.4 Physics and Manual
T. Sjöstrand, S. Mrenna and P. Skands,
JHEP05 (2006) 026 [hep-ph/0603175]
- available on <http://www.thep.lu.se/~torbjorn/Pythia.html>
- together with sample main programs, old code, etc.

... but

- only add, never subtract
- ⇒ has become bloated and unmanageable
- is in Fortran 77, so not understood by young people

PYTHIA 8: A fresh start

Problem: PYTHIA 7 stalled, no other manpower

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

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

Tentative schedule:

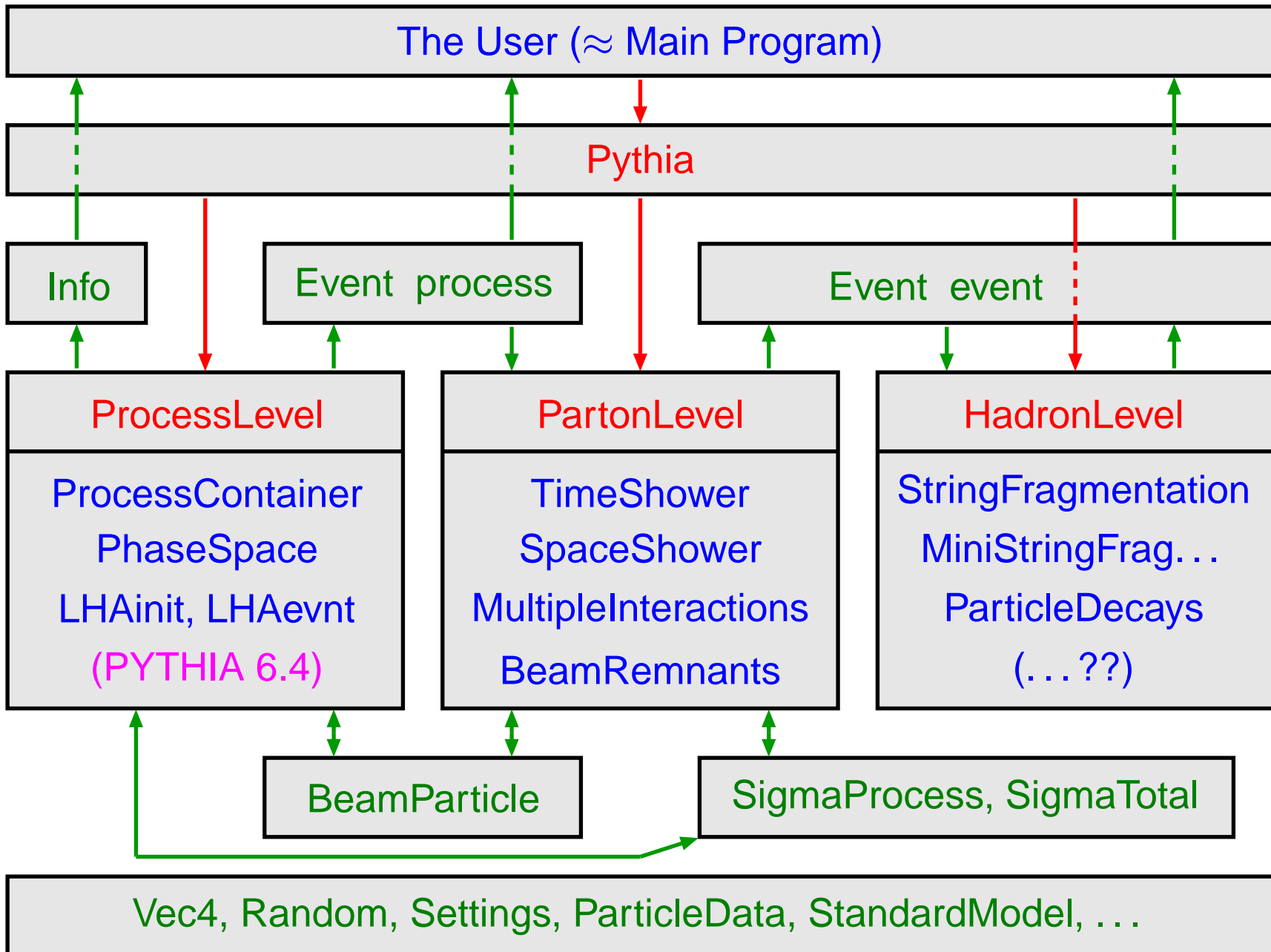
time	date	processes	final states
0 =	1 Sept. 2004	—	—
1 =	1 Sept. 2005	LHA-style input	incomplete draft
2 =	1 Sept. 2006	a few processes	complete, buggy(?)
3 =	1 Sept. 2007	more processes	stable, debugged

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

Objectives:

- clean up, keep the most recent models
- core program completely standalone, but
- Les Houches Accord style input central
 - interfaces to other libraries foreseen

Current PYTHIA 8 structure



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("filename");
pythia.init(idBeamA,idBeamB,eCM);
```

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

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

Settings and Particle Data

Can read in settings and particle data changes by

- `pythia.readString("command")`
- `pythia.readFile("filename")` with one `command` per line in file

Settings come in three kinds

- **Flags**: on/off switches, `bool`
- **Modes**: enumerated options, `int`
- **Parameters**: continuum of values, `double`

and `command` is of form `location:name = value`, e.g.

`PartonLevel:ISR = off` no initial-state radiation

`InFlux:nQuark = 4` do not consider b pdf's

`TimeShower:pTmin = 1.0` cut off final-state radiation at 1 GeV

To access **particle data**, instead `command` should be of form

`id:property = value` or `id:channel:property = value`, e.g.

`3122:mayDecay = no` do not allow Λ^0 to decay (no = off = false = 0)

`215:3:products = 211 111 111` to let $a_2^+ \rightarrow \pi^+ \pi^0 \pi^0$

Hard-process generation

Currently limited selection implemented internally,
but can use Fortran PYTHIA 6 library transparently via LHA interface.

Provide settings as before, with `command` of form
`Pythia6:variable = value`, e.g. `Pythia6:mset = 6`.
`variable` can be anything recognized by PYGIVE,
but only ones relevant for hard process are actually used.

Can also use Les Houches Accord for any other process,
input via runtime Fortran interface or via files.

Overview processes that can be generated:

Process	internal	PYTHIA 6	Les Houches
minbias/elastic/diffractive	+	-	-
QCD jets (incl. $c\bar{c}$, $b\bar{b}$)	+	+	+
prompt photons	+	+	+
other PYTHIA 6 processes	-	+	+
anything else (within limits)	-	-	+

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:mset = 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 the first year! ★
- ★ On hold during autumn 2005, now picking up speed again ★
- ★ ~ 1 sub-subversion per working week, currently 8.051 ★
- ★ Put recent versions on web at irregular intervals ★
- ★ First (almost) production-quality release, 8.100, early 2007 (?) ★
- ★ Debugged and tuned by LHC startup (??) ★
- ★ Overtaking Fortran version usage by 2009 (???) ★

- **Feedback is most welcome** ●

- ★ Basic structure is settling down ... ★
- ★ ... but still lot of room for changes ★
- ★ e.g. interfacing to other programs ★

- **MC4LHC workshop July 17-26** ●

- ★ will contain tutorials, if interest (sign up!) ★

Trying It Out

- Download `pythia8051.tgz` from <http://www.thep.lu.se/~torbjorn/Pythia.html>, link “Future”
- Unzip and expand with `tar xvzf pythia8051.tgz`
- Move to the thus created `pythia8051` directory
- Follow the `README` instructions (edit links to PYTHIA 6, HepMC)
- `make` will compile in ~ 4 minutes (half for PYTHIA 6)
- The `examples` subdirectory contains 14 sample main programs
(`make mainNN` and then `mainNN.exe > outfile`)
- The `pythia8051.pdf` file contains an introduction to the program
- Open `doc/Welcome.html` in a web browser for the full manual
(in the future: GUI with xml + Javascript ?)