MC-TESTER:

a tool for comparisons of HEP Monte Carlo Generators

Piotr Golonka

(FNPT UMM Krakow)

Tomasz Pierzchala (Silesia University, Katowice)

Zbigniew Was (INP Krakow)

ECFA-DESY Linear Collider Workshop

15-18 November, 2002

- MC-TESTER

- MC-TESTER
  - ⋆ purpose
  - ⋆ history
  - ⋆ incarnation

- MC-TESTER
  - ⋆ purpose
  - ⋆ history
  - ⋆ incarnation
- Example of analysis: 6-fermion final states

- MC-TESTER
  - ⋆ purpose
  - ⋆ history
  - ⋆ incarnation
- Example of analysis: 6-fermion final states
- HEPEvent library

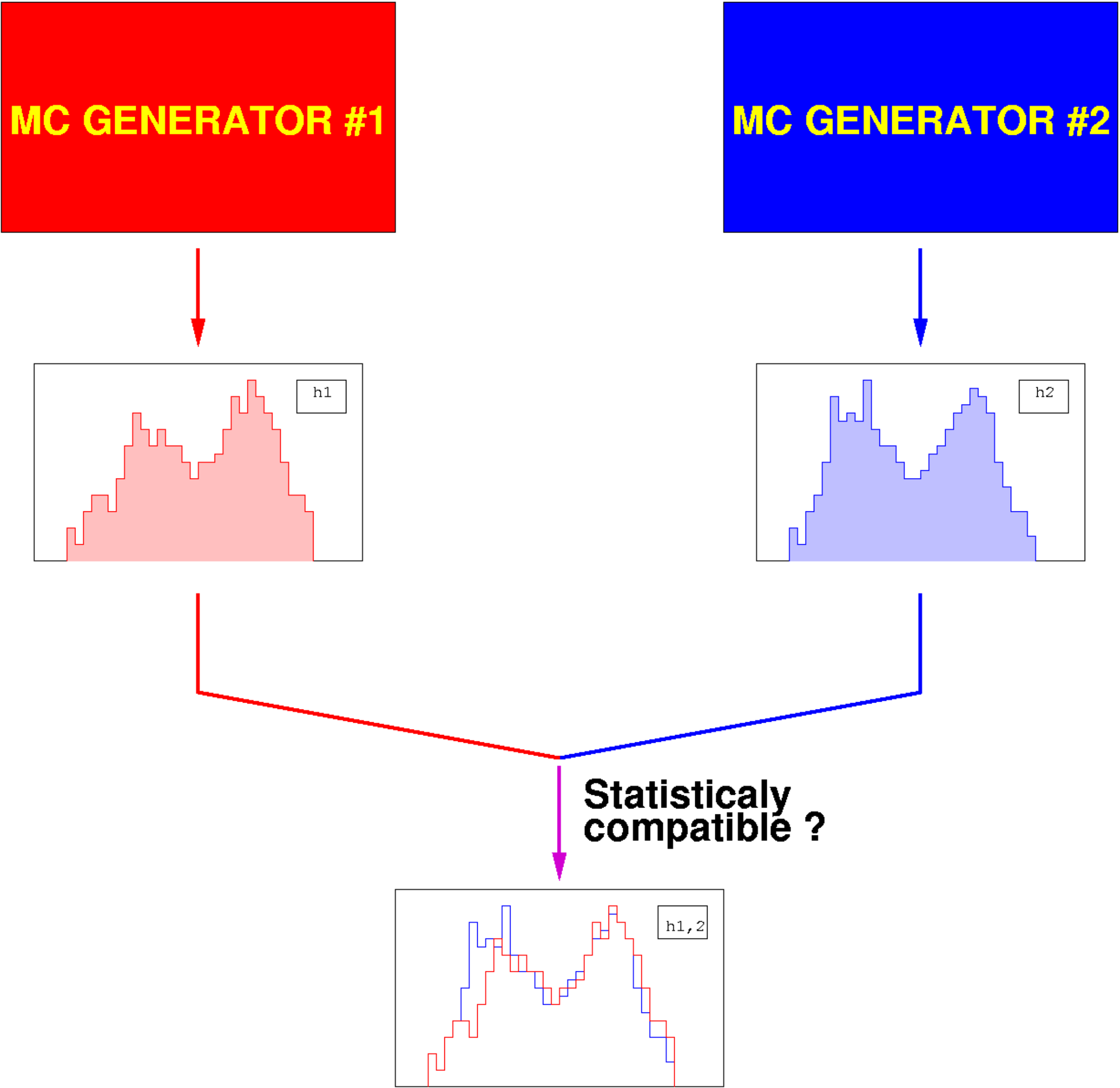
- MC-TESTER
  - ⋆ purpose
  - ⋆ history
  - ⋆ incarnation
- Example of analysis: 6-fermion final states
- HEPEvent library
- Example of analysis: τ lepton decay MC generators comparison.

- MC-TESTER
  - ⋆ purpose
  - ⋆ history
  - ⋆ incarnation
- Example of analysis: 6-fermion final states
- HEPEvent library
- Example of analysis: τ lepton decay MC generators comparison.
- Future improvements

Excuses ...

.. for computer-scientist view of Physics...

MC-TESTER: why?



MC-TESTER: history

MC-TESTER: history

- *TAUOLA*: τ lepton decay Monte-Carlo library by Z. Was:

MC-TESTER: history

- *TAUOLA*: τ lepton decay Monte-Carlo library by Z. Was:
  - ⋆ testing phase: checking for inconsistencies in invarant mass distribution

MC-TESTER: history

- *TAUOLA*: τ lepton decay Monte-Carlo library by Z. Was:
  - ⋆ testing phase: checking for inconsistencies in invarant mass distribution
  - ⋆ “visual”:

MC-TESTER: history

- *TAUOLA*: τ lepton decay Monte-Carlo library by Z. Was:
  - ⋆ testing phase: checking for inconsistencies in invarant mass distribution
  - ⋆ “visual”:

    inconsistencies in shape of histogrammes,

MC-TESTER: history

- *TAUOLA*: τ lepton decay Monte-Carlo library by Z. Was:
  - ⋆ testing phase: checking for inconsistencies in invarant mass distribution
  - ⋆ “visual”:

    inconsistencies in shape of histogrammes,
    difference between shape of their division and flat distribution

MC-TESTER: history

- *TAUOLA*: τ lepton decay Monte-Carlo library by Z. Was:
  - ⋆ testing phase: checking for inconsistencies in invarant mass distribution
  - ⋆ “visual”:

    inconsistencies in shape of histogrammes,
    difference between shape of their division and flat distribution
  - ⋆ performed for all possible mass invariants in every decay channel

MC-TESTER: history

- *TAUOLA*: τ lepton decay Monte-Carlo library by Z. Was:
  - ⋆ testing phase: checking for inconsistencies in invarant mass distribution
  - ⋆ “visual”:

    inconsistencies in shape of histogrammes,
    difference between shape of their division and flat distribution
  - ⋆ performed for all possible mass invariants in every decay channel
  - ⋆ hundreds of histogrammes to be compared!

MC-TESTER incarnation

- authors: ***P. Golonka, T. Pierzchala, Z. Was***
- development: November 2000 - July 2002
- version 1.0 released: September 2002
- documentation: hep-ph\0210252:
  “MC-TESTER: a universal tool for comparisons of Monte Carlo predictions for particle decays in high energy physics”

MC-TESTER features:

MC-TESTER features:

- works on decay trees produced by two (or more) HEP MC event generators

MC-TESTER features:

- works on decay trees produced by two (or more) HEP MC event generators
- uses all supported event records (see further slides)

MC-TESTER features:

- works on decay trees produced by two (or more) HEP MC event generators
- uses all supported event records (see further slides)
- identifies decay channels of given particle (τ, H, *etc)*

MC-TESTER features:

- works on decay trees produced by two (or more) HEP MC event generators
- uses all supported event records (see further slides)
- identifies decay channels of given particle (τ, H, *etc)*
- creates/analyzes histogrammes of all invariant masses for every decay channels

MC-TESTER features:

- works on decay trees produced by two (or more) HEP MC event generators
- uses all supported event records (see further slides)
- identifies decay channels of given particle (τ, H, *etc)*
- creates/analyzes histogrammes of all invariant masses for every decay channels
- produces plots of histogrammes

MC-TESTER features:

- works on decay trees produced by two (or more) HEP MC event generators
- uses all supported event records (see further slides)
- identifies decay channels of given particle (τ, *H*, *etc*)
- creates/analyzes histogrammes of all invariant masses for every decay channels
- produces plots of histogrammes
- determines statistical compatibility for every two histogrammes: calculates “Shape Difference Parameter” (SDP)

MC-TESTER features:

- works on decay trees produced by two (or more) HEP MC event generators
- uses all supported event records (see further slides)
- identifies decay channels of given particle (τ, H, *etc)*
- creates/analyzes histogrammes of all invariant masses for every decay channels
- produces plots of histogrammes
- determines statistical compatibility for every two histogrammes: calculates “Shape Difference Parameter” (SDP)
- 3 pre-defined algorithms for SDP; user-provided as well

MC-TESTER -aided analysis:

MC-TESTER -aided analysis:

- For two (or more) MC generators (simulation systems), calls to MC-TESTER needs to be inserted to code, libraries linked

MC-TESTER -aided analysis:

- For two (or more) MC generators (simulation systems), calls to MC-TESTER needs to be inserted to code, libraries linked
- Each simulation needs to be run, producing decay trees.

MC-TESTER -aided analysis:

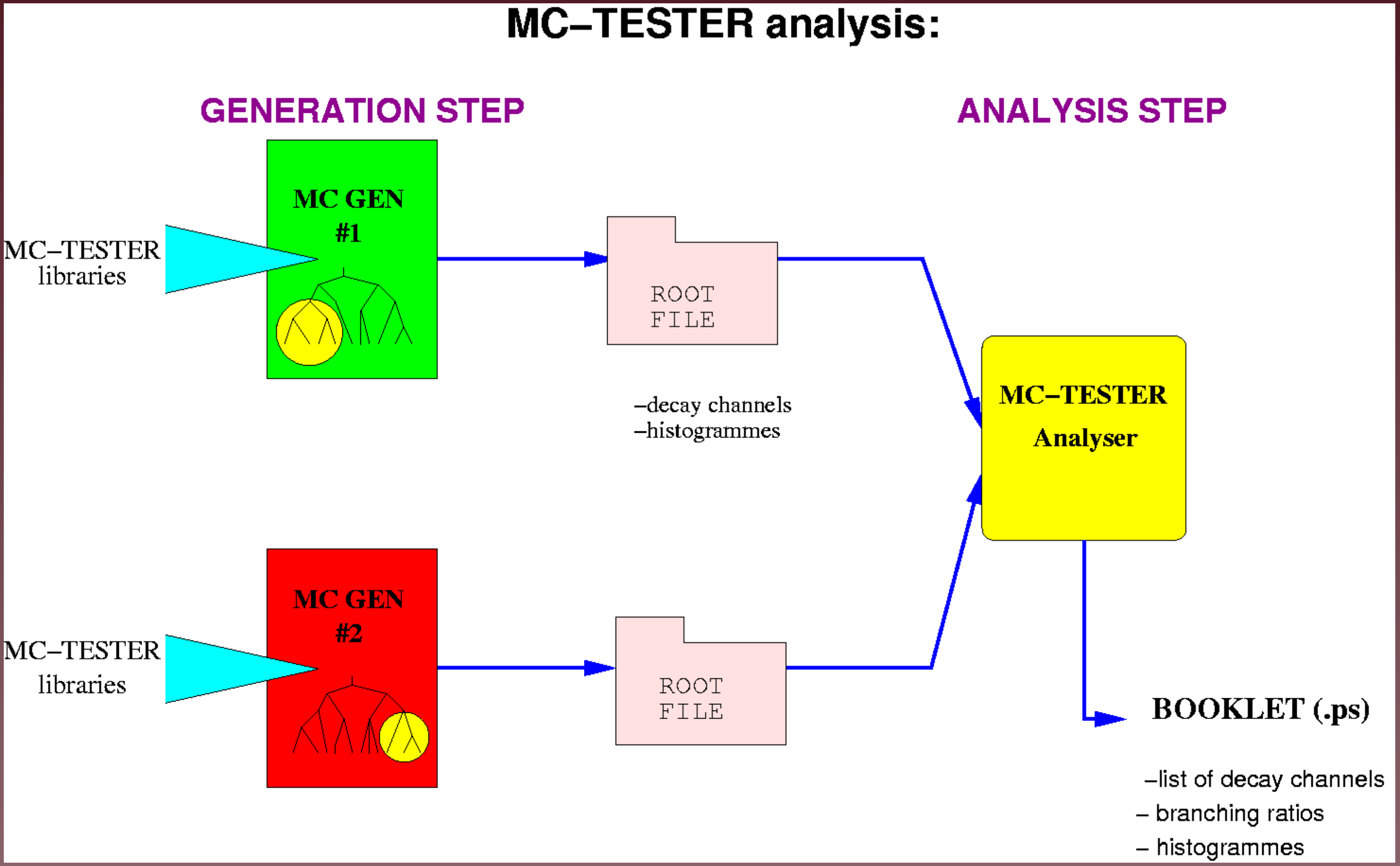
- For two (or more) MC generators (simulation systems), calls to MC-TESTER needs to be inserted to code, libraries linked
- Each simulation needs to be run, producing decay trees.
- Each decay tree is analysed by MC-TESTER routine, information/histogrammes of invariant masses stored in files.

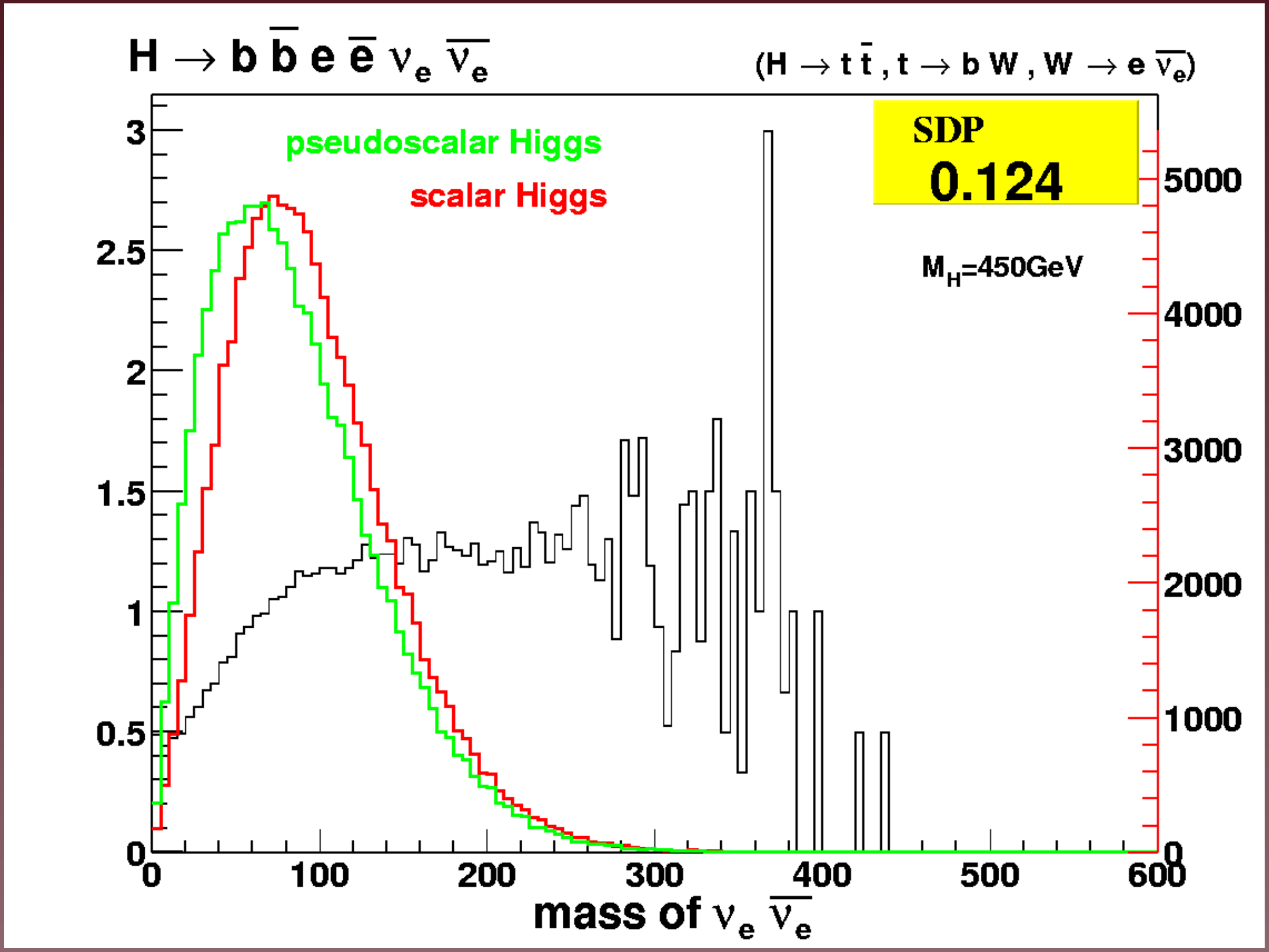
MC-TESTER -aided analysis:

- For two (or more) MC generators (simulation systems), calls to MC-TESTER needs to be inserted to code, libraries linked
- Each simulation needs to be run, producing decay trees.
- Each decay tree is analysed by MC-TESTER routine, information/histogrammes of invariant masses stored in files.
- MC-TESTER analysis step: data files from two production runs are analysed: histogrammes for all matching decay channels analysed, plotted, SDP calculated

MC-TESTER -aided analysis:

- For two (or more) MC generators (simulation systems), calls to MC-TESTER needs to be inserted to code, libraries linked
- Each simulation needs to be run, producing decay trees.
- Each decay tree is analysed by MC-TESTER routine, information/histogrammes of invariant masses stored in files.
- MC-TESTER analysis step: data files from two production runs are analysed: histogrammes for all matching decay channels analysed, plotted, SDP calculated
- booklet with the results of analysis is produced.


MC–TESTER analysis:
GENERATION STEP
ANALYSIS STEP
MC–TESTER libraries
MC GEN #1
ROOT FILE
–decay channels
–histogrammes
MC–TESTER libraries
MC GEN #2
ROOT FILE
MC–TESTER Analyser
BOOKLET (.ps)
–list of decay channels
– branching ratios
– histogrammes


$H \rightarrow b \bar{b} e \bar{e} \nu_e \bar{\nu}_e$
$(H \rightarrow t \bar{t}, t \rightarrow b W, W \rightarrow e \bar{\nu}_e)$
pseudoscalar Higgs
scalar Higgs
SDP
0.124
M_H=450GeV
0
0.5
1
1.5
2
2.5
3
0
1000
2000
3000
4000
5000
0
100
200
300
400
500
600
mass of $\nu_e \bar{\nu}_e$

MC-TESTER requirements:

MC-TESTER requirements:

- ROOT library

MC-TESTER requirements:

- ROOT library
- LaTeX

MC-TESTER requirements:

- ROOT library
- LaTeX
- Linux system with g77 and g++ compilers.(RedHat 6.1, 7.2, 8.0 tested)

MC-TESTER requirements:

- ROOT library
- LaTeX
- Linux system with g77 and g++ compilers.(RedHat 6.1, 7.2, 8.0 tested)
- A few tens of MB of disk space.

MC-TESTER requirements:

- ROOT library
- LaTeX
- Linux system with g77 and g++ compilers.(RedHat 6.1, 7.2, 8.0 tested)
- A few tens of MB of disk space.
- Event records supported by HEPEvent library (see next slide).

MC-TESTER requirements:

- ROOT library
- LaTeX
- Linux system with g77 and g++ compilers.(RedHat 6.1, 7.2, 8.0 tested)
- A few tens of MB of disk space.
- Event records supported by HEPEvent library (see next slide).

MC-TESTER requirements:

- ROOT library
- LaTeX
- Linux system with g77 and g++ compilers.(RedHat 6.1, 7.2, 8.0 tested)
- A few tens of MB of disk space.
- Event records supported by HEPEvent library (see next slide).

WARNING: decay channels with large multiplicities (>7)produces thousands of histogrammes!

MC-TESTER requirements:

- ROOT library
- LaTeX
- Linux system with g77 and g++ compilers.(RedHat 6.1, 7.2, 8.0 tested)
- A few tens of MB of disk space.
- Event records supported by HEPEvent library (see next slide).

WARNING: decay channels with large multiplicities (>7)produces thousands of histogrammes! Analysis may take long time, intermediate data files and booklet may be large. There may exist problems with memory allocation.

HEPEvent library.

HEPEvent library.

- Unified, C++ access to various HEP MC Event Record in F77

HEPEvent library.

- Unified, C++ access to various HEP MC Event Record in F77
- First implementation: “Photos+”

HEPEvent library.

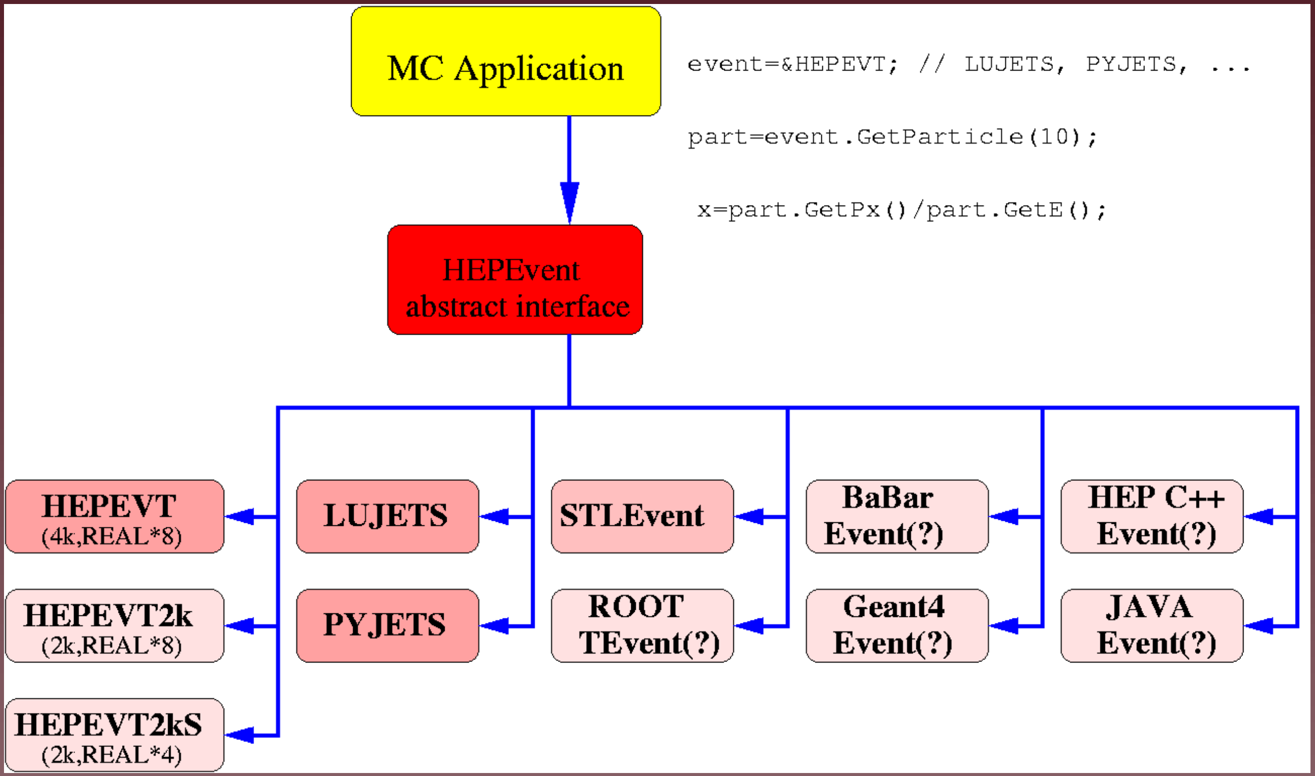
- Unified, C++ access to various HEP MC Event Record in F77
- First implementation: “Photos+”
- Currently Supports:

HEPEvent library.

- Unified, C++ access to various HEP MC Event Record in F77
- First implementation: “Photos+”
- Currently Supports:
  - ⋆ HEPEVT
  - ⋆ LUJETS (i.e. Pythia 5)
  - ⋆ PYJETS (i.e. Pythia 6)
  - ⋆ Experimental C++ event structures
- In future:

HEPEvent library.

- Unified, C++ access to various HEP MC Event Record in F77
- First implementation: “Photos+”
- Currently Supports:
  - ⋆ HEPEVT
  - ⋆ LUJETS (i.e. Pythia 5)
  - ⋆ PYJETS (i.e. Pythia 6)
  - ⋆ Experimental C++ event structures
- In future:
  - ⋆ C++, Java(?) event records.
  - ⋆ simplified access to various “flavours” of F77 event records (precision, number of entries)



MC Application

HEPEvent
abstract interface

HEPEVT
(4k,REAL*8)

HEPEVT2k
(2k,REAL*8)

HEPEVT2kS
(2k,REAL*4)

LUJETS

PYJETS

STLEvent

ROOT
TEvent(?)

BaBar
Event(?)

Geant4
Event(?)

HEP C++
Event(?)

JAVA
Event(?)

```
event=&HEPEVT; // LUJETS, PYJETS, ...

part=event.GetParticle(10);

 x=part.GetPx()/part.GetE();
```

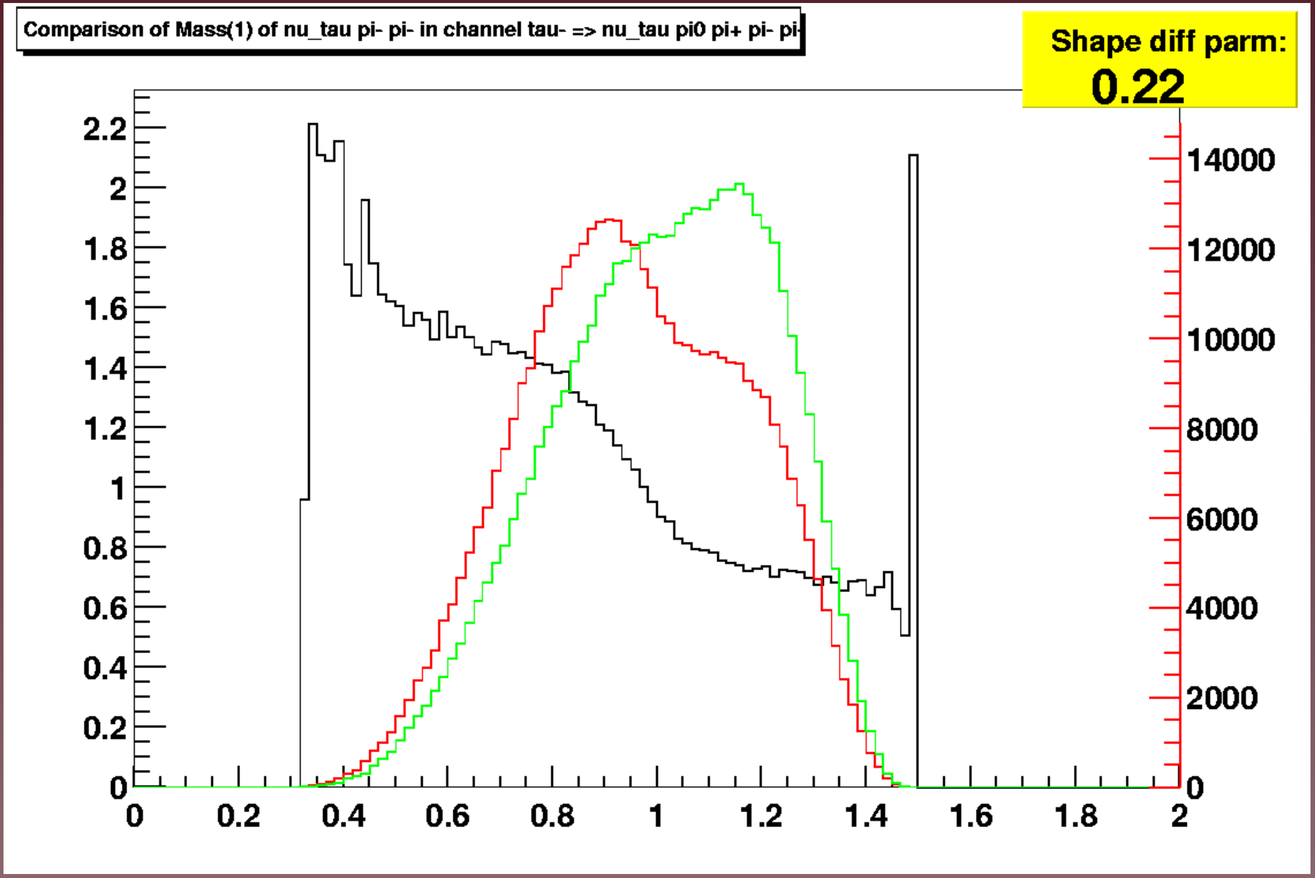
Example:

τ lepton decay generators comparison:

- two versions of TAUOLA library with different initializations:
  - ⋆ CLEO
  - ⋆ ALEPH
- PYTHIA

Task: identify and compare the decay channels of the τ lepton produced by these generators.

Ex: single plot



Accesibility:

- The program and the documentation are available from the webpage:
  `http://cern.ch/Piotr.Golonka/MC/MC-TESTER`

Future extensions:

- Support for other event record standards

Future extensions:

- Support for other event record standards
  C++

Future extensions:

- Support for other event record standards
  C++
  JAVA

Future extensions:

- Support for other event record standards
  C++
  JAVA

- Support for other parameters (e.g accolinearity)

Future extensions:

- Support for other event record standards
  C++
  JAVA

- Support for other parameters (e.g accolinearity)
- Limit on depth of analysed decay tree

Future extensions:

- Support for other event record standards
  C++
  JAVA

- Support for other parameters (e.g accolinearity)
- Limit on depth of analysed decay tree
- Switch off analysis of decays of certain particles (e.g. π_0)

Conclusions and outlook

Conclusions and outlook

- MC-TESTER may be a useful tool for analysis performed for linear colliders studies

Conclusions and outlook

- MC-TESTER may be a useful tool for analysis performed for linear colliders studies
- It easily integrates with already existing Monte Carlo environments

Conclusions and outlook

- MC-TESTER may be a useful tool for analysis performed for linear colliders studies
- It easily integrates with already existing Monte Carlo environments
- Support for all major event record standards is provided

Conclusions and outlook

- MC-TESTER may be a useful tool for analysis performed for linear colliders studies
- It easily integrates with already existing Monte Carlo environments
- Support for all major event record standards is provided
- It may help to find observables for physics studies

Acknowledgments

- **Malgorzata Worek**, for providing the Higgs decay library
- **Swagato Bernjee, Stanislaw Jadach and Wieslaw Placzek** for valuable remarks ans suggestions.