HERWIG++ progress report

Stefan Gieseke

University of Cambridge
Cavendish Laboratory

for the Herwig++ team:

SG, Philip Stephens, Bryan Webber *(Cambridge)*
Alberto Ribon ($\rightarrow$ CERN), Mike Seymour *(Manchester)*

- Introduction, why C++.
- Parton Shower
  - new variables
  - new multiscale shower
- Decays, hard interactions
- Hadronization
- Future

MC Event Generators

Hard Matrix Elements

Parton Shower

Hadronization

$\longrightarrow$ Frank's fantastic figure

Why ++?

- C++ is standard in the Unix/Linux world
- Experiments use it
- Maintenance
- Encapsulation of code
- Easy to implement new physics

Interplay with Pythia7

won't re-invent the wheel

share administrative overhead, common to event generators

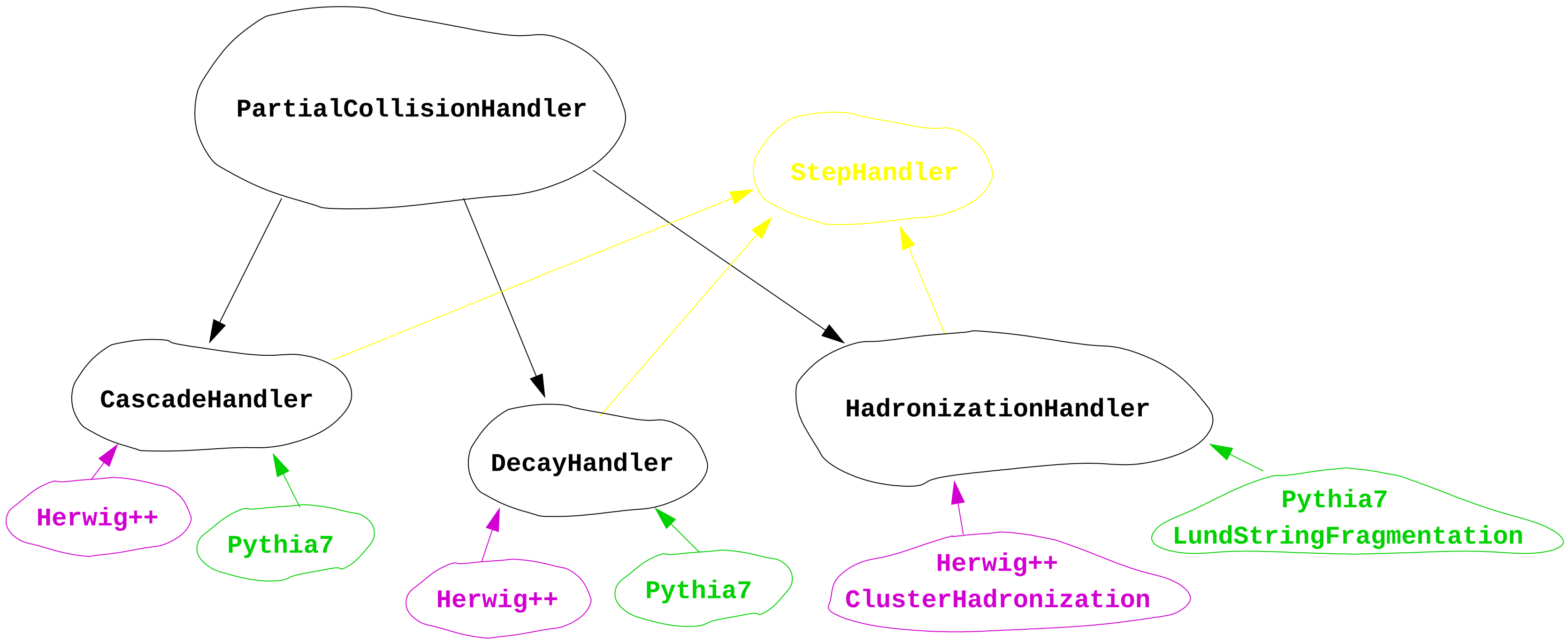
independent *physics* implementation

large but very flexible implementation taking full advantage of the object oriented paradigms

Common basis for Pythia7/Herwig++:

- lack of independence
- miss the possiblity to test codes against each other
- physics, however, is still independent
- beneficial for the user to have the same framework (even a graphical interface?)

PartialCollisionHandlers



New evolution variables

Kinematics to allow better treatment of heavy particles, avoiding

- overlapping regions in phase space, in particular for soft emissions
- dead cones

We choose $\tilde{q}^2$ as new evolution variable,

$$\tilde{q}^2 = \frac{\boldsymbol{p}^2}{z^2(1-z)^2} + \frac{m^2}{z^2}$$

with the agument of running α_S chosen according to

$$\alpha_S(z^2(1-z)^2\tilde{q}^2)$$

angular ordering

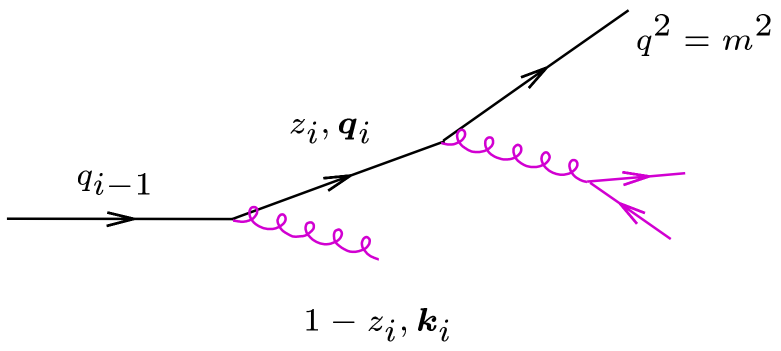
$$\tilde{q}_{i+1} < z_i\tilde{q}_i \qquad \tilde{k}_{i+1} < (1-z_i)\tilde{q}_i$$

Technically: *reinterpretation* of known evolution variables, i.e. the branching probability for $a \to bc$ still is

$$dP(a \to bc) = \frac{d\tilde{q}^2}{\tilde{q}^2}\frac{C_i\alpha_S}{2\pi}P_{bc}(z)\,dz$$

$\longrightarrow$ Sudakov's etc. technically remain the same!

Kinematics



Sudakov basis p, n with $p^2 = m^2$, $n^2 = 0$,

$$q_i = \alpha_i p + \beta_i n + \boldsymbol{q}_i$$

Longitudinal splitting

$$\alpha_i = z_i \alpha_{i-1}$$

transverse momenta reconstructed from $\boldsymbol{p}_i$,

$$\boldsymbol{q}_i = \boldsymbol{p}_i + z_i \boldsymbol{q}_{i-1} \qquad \boldsymbol{k}_i = -\boldsymbol{p}_i + (1 - z_i)\boldsymbol{q}_{i-1}$$

Azimuthal angle φ still chosen randomly or as a result of *azimuthal spin correlations* (planned) β_i from virtuality q_i

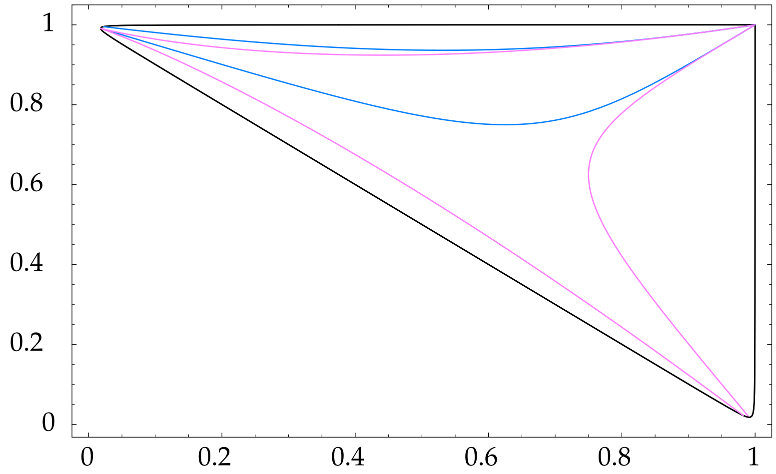
$$\beta_i = \frac{\boldsymbol{q}_i^2 + q_i^2 - \alpha_i^2 m^2}{2p \cdot n \alpha_i}$$

with recursive reconstruction of virtualities from

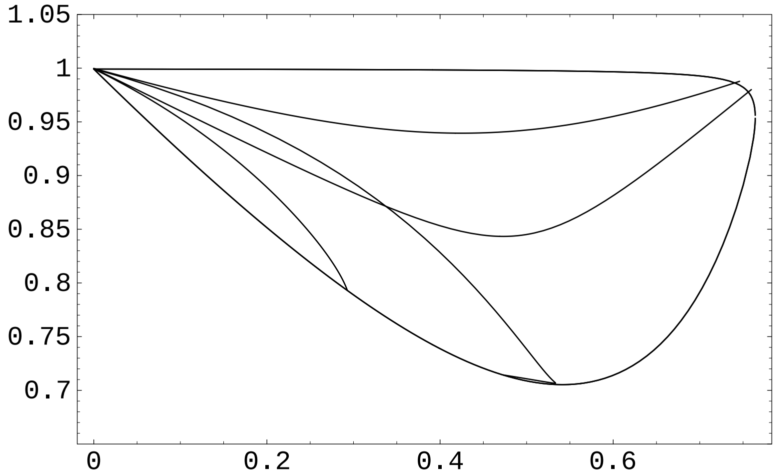
$$q_{i-1}^2 = \frac{q_i^2}{z_i} + \frac{k_i^2}{1 - z_i} + \frac{\boldsymbol{p}_i^2}{z_i(1 - z_i)}$$

Phase space coverage

$e^+e^- \to q\bar{q}g$

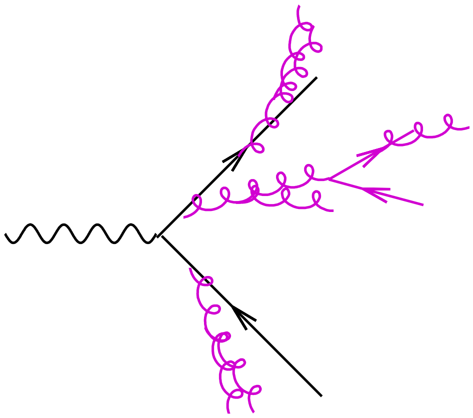


$t \to Wbg$



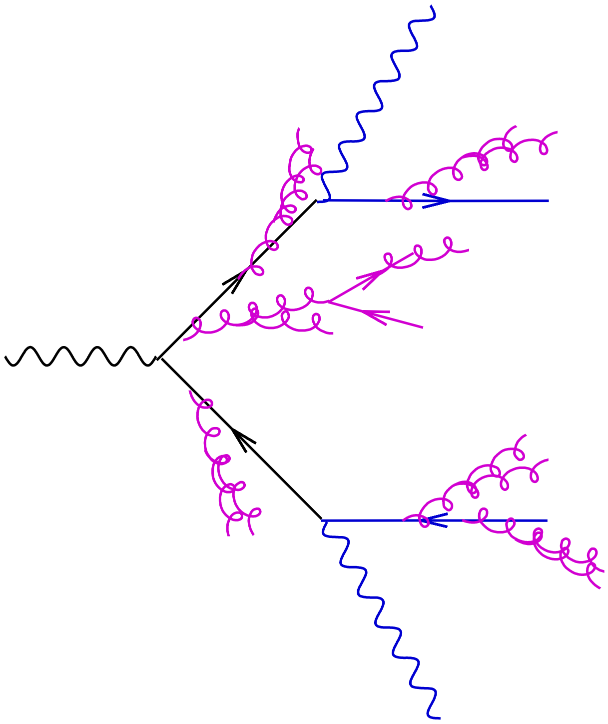
- smooth coverage of the soft region is achieved by choosing appropriate limiting values for the parton shower
- soft ME correction can be applied without double counting (N.B. tricky for radiation from b in t-decay)

Old Shower 2



1. hard process $e^+e^- \to t\bar{t}$
2. radiation from t, $\bar{t}$ ($\hat{s} \to \mu_0$)

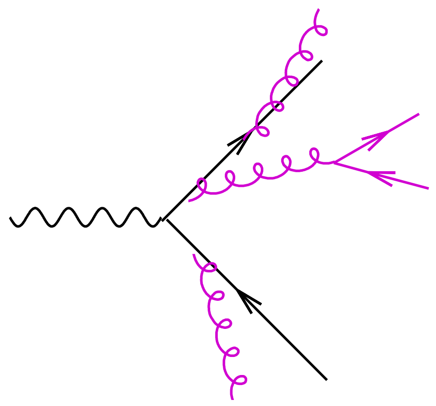
Old Shower 4



1. hard process $e^+e^- \to t\bar{t}$
2. radiation from t, $\bar{t}$ $(\hat{s} \to \mu_0)$
3. decay $t \to Wb$, $\bar{t} \to W\bar{b}$,
4. radiation from b, $\bar{b}$ $(m_t \to \mu_0)$

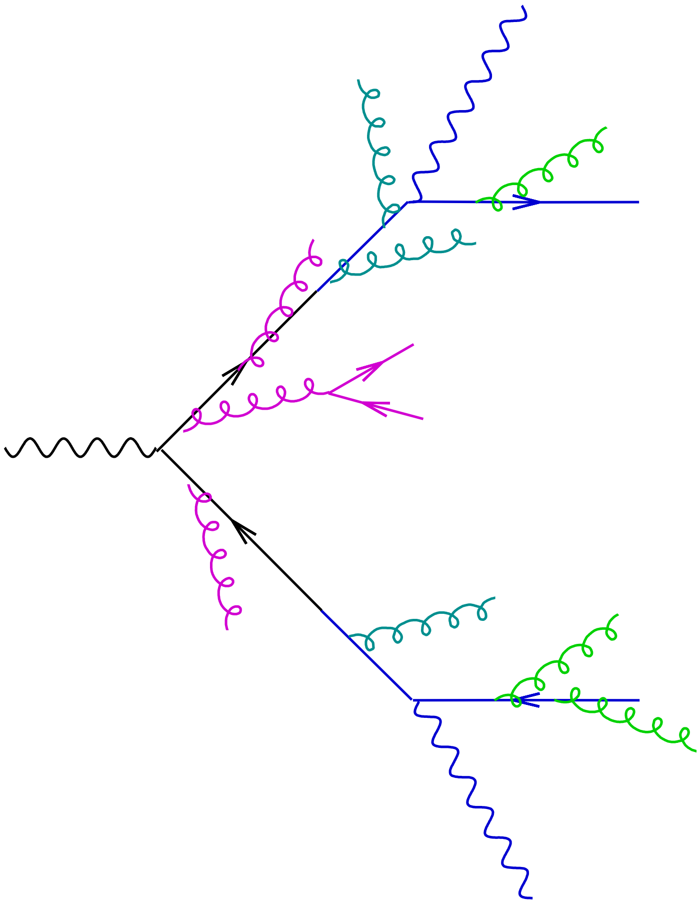
Radiation down to μ_0, ignoring the finite width Γ_t.

Multiscale Shower 2



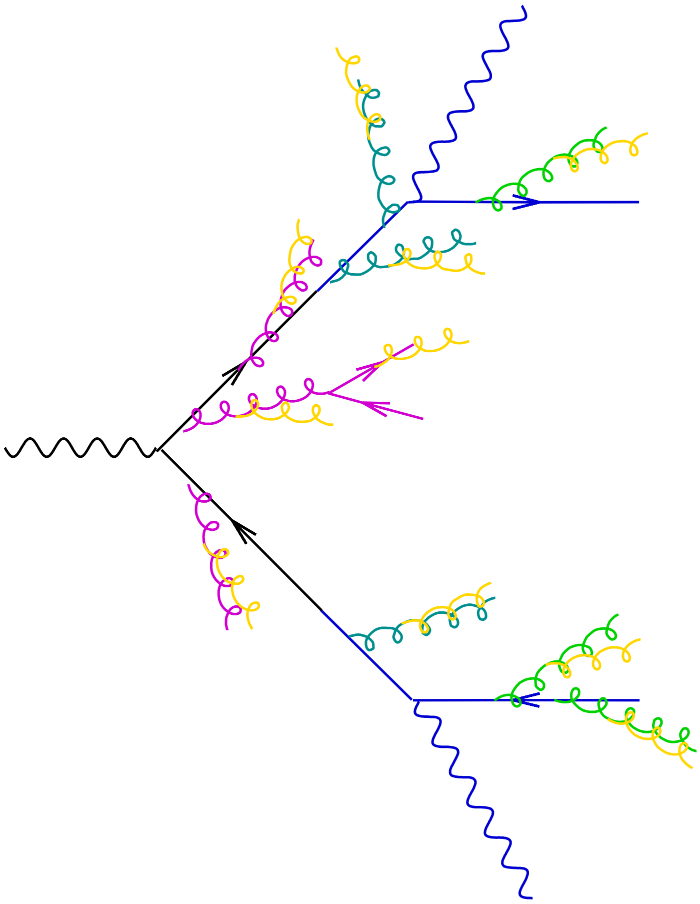
1. hard process $e^+e^- \to t\bar{t}$
2. `showerNormally()` ($\hat{s} \to \Gamma_t$) from t, $\bar{t}$

Multiscale Shower 5



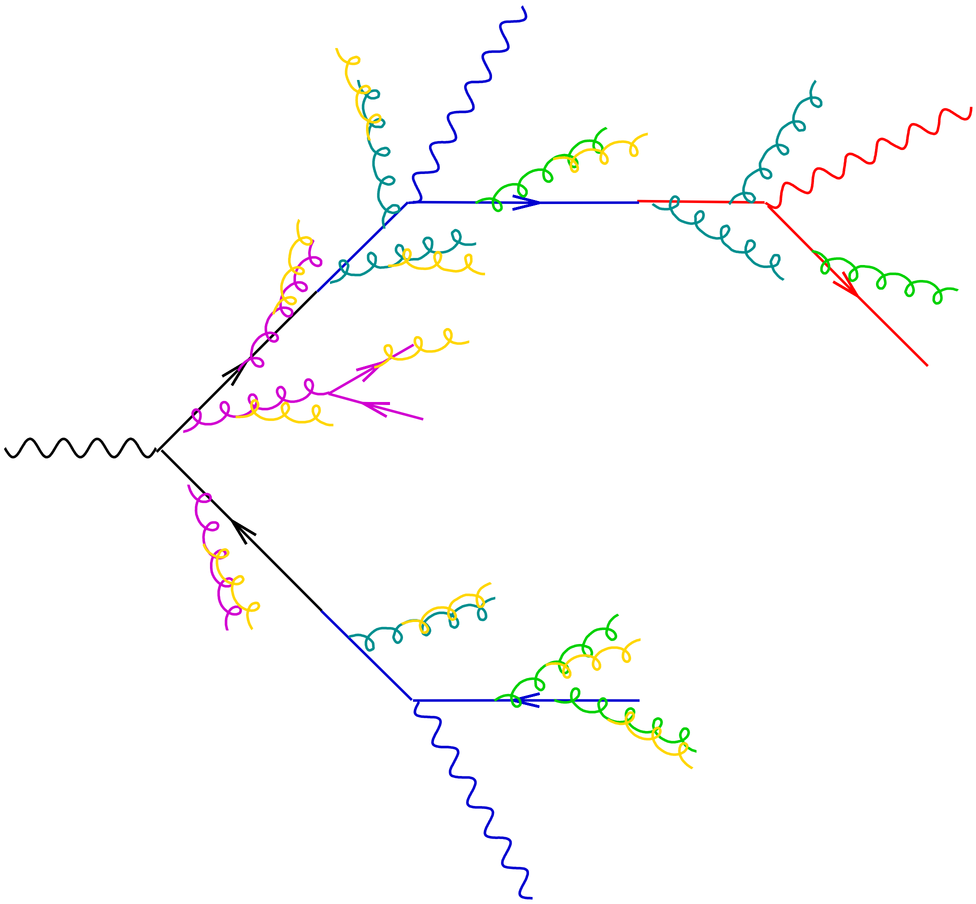
1. hard process $e^+e^- \to t\bar{t}$
2. `showerNormally()` ($\hat{s} \to \Gamma_t$) from t, $\bar{t}$
3. decay $t \to Wb$, $\bar{t} \to W\bar{b}$,
4. `showerDecay()` ($m_t \to \Gamma_t$) from t, $\bar{t}$ (DIS-like)
5. `showerDecay()` ($m_t \to \Gamma_t$) from b, $\bar{b}$

Multiscale Shower 6



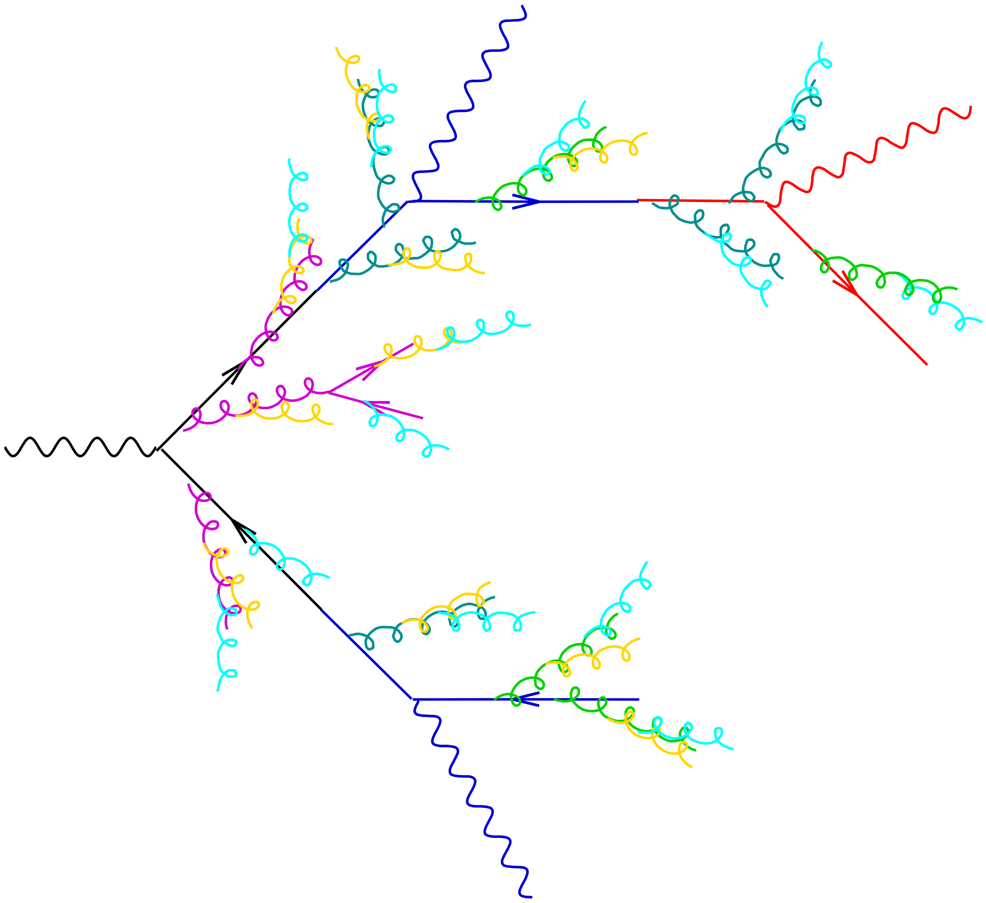
1. hard process $e^+e^- \to t\bar{t}$
2. `showerNormally()` $(\hat{s} \to \Gamma_t)$ from t, $\bar{t}$
3. decay $t \to Wb$, $\bar{t} \to W\bar{b}$,
4. `showerDecay()` $(m_t \to \Gamma_t)$ from t, $\bar{t}$ (DIS-like)
5. `showerDecay()` $(m_t \to \Gamma_t)$ from b, $\bar{b}$
6. `showerGlobally()` $(\Gamma_t \to \Gamma_b)$ everywhere

Multiscale Shower 8



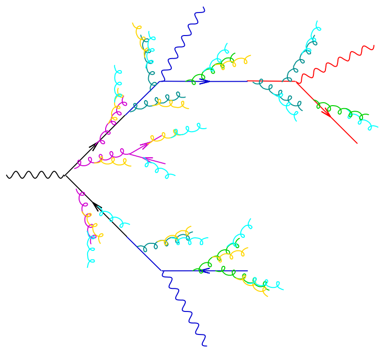
1. hard process $e^+e^- \to t\bar{t}$
2. `showerNormally()` ($\hat{s} \to \Gamma_t$) from t, $\bar{t}$
3. decay $t \to Wb$, $\bar{t} \to W\bar{b}$,
4. `showerDecay()` ($m_t \to \Gamma_t$) from t, $\bar{t}$ (DIS-like)
5. `showerDecay()` ($m_t \to \Gamma_t$) from b, $\bar{b}$
6. `showerGlobally()` ($\Gamma_t \to \Gamma_b$) everywhere
7. suppose b is some unstable particle, $b \to \text{whatever};$ $\mu_0 \to \Gamma_b$
8. `showerDecay()` ($m_b \to \Gamma_b$) from b, $\bar{b}$ (DIS-like) and `showerDecay()` ($m_b \to \Gamma_b$) from b', $\bar{b}'$

Multiscale Shower 9



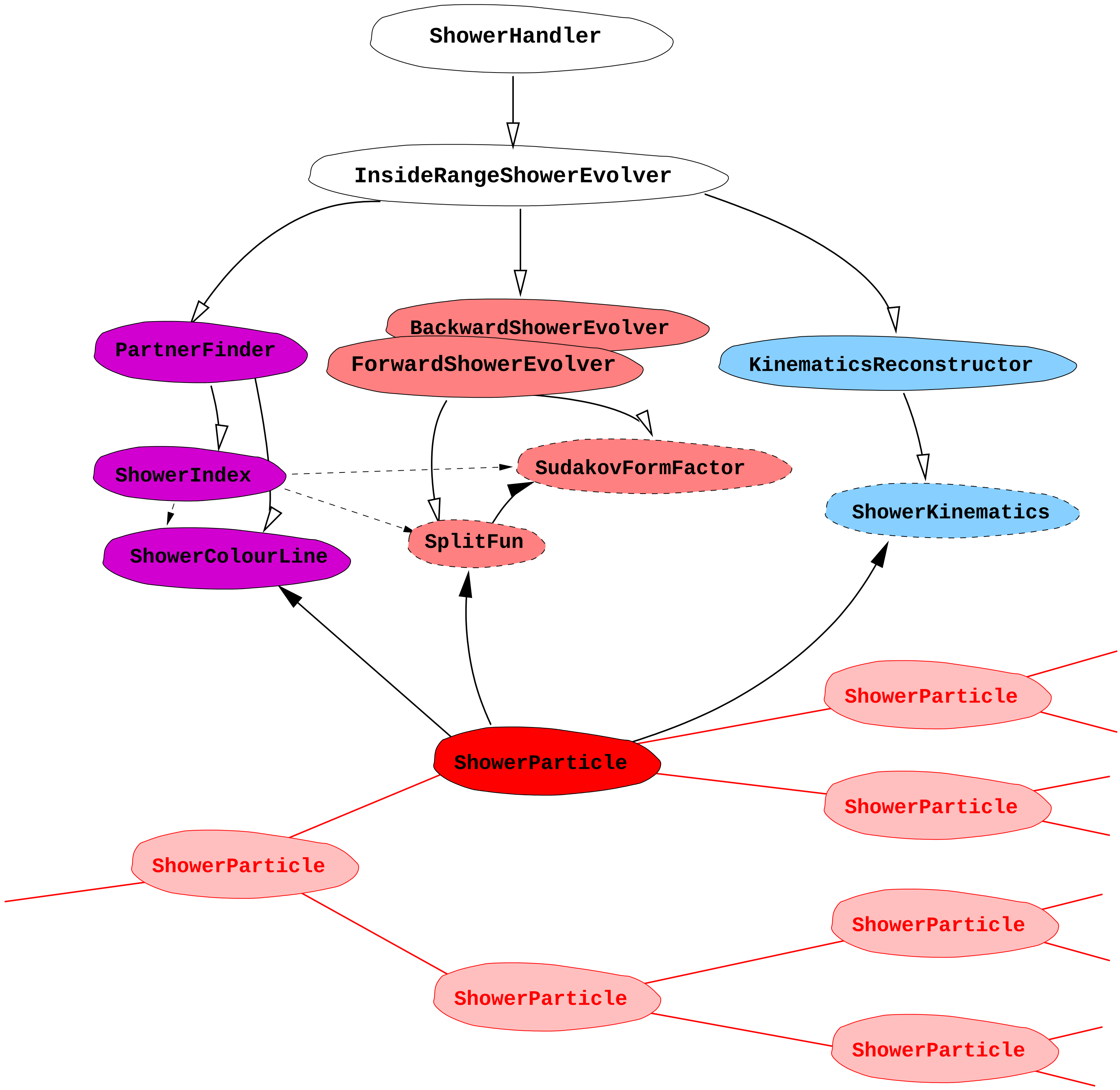
1. hard process $e^+e^- \to t\bar{t}$
2. `showerNormally()` $(\hat{s} \to \Gamma_t)$ from t, $\bar{t}$
3. decay $t \to Wb$, $\bar{t} \to W\bar{b}$,
4. `showerDecay()` $(m_t \to \Gamma_t)$ from t, $\bar{t}$ (DIS-like)
5. `showerDecay()` $(m_t \to \Gamma_t)$ from b, $\bar{b}$
6. `showerGlobally()` $(\Gamma_t \to \Gamma_b)$ everywhere
7. suppose b is some unstable particle, $b \to$ whatever; $\mu_0 \to \Gamma_b$
8. `showerDecay()` $(m_b \to \Gamma_b)$ from b, $\bar{b}$ (DIS-like) and `showerDecay()` $(m_b \to \Gamma_b)$ from b', $\bar{b}'$
9. `showerGlobally()` $(\Gamma_b \to \mu_0)$ everywhere

Multiscale Shower

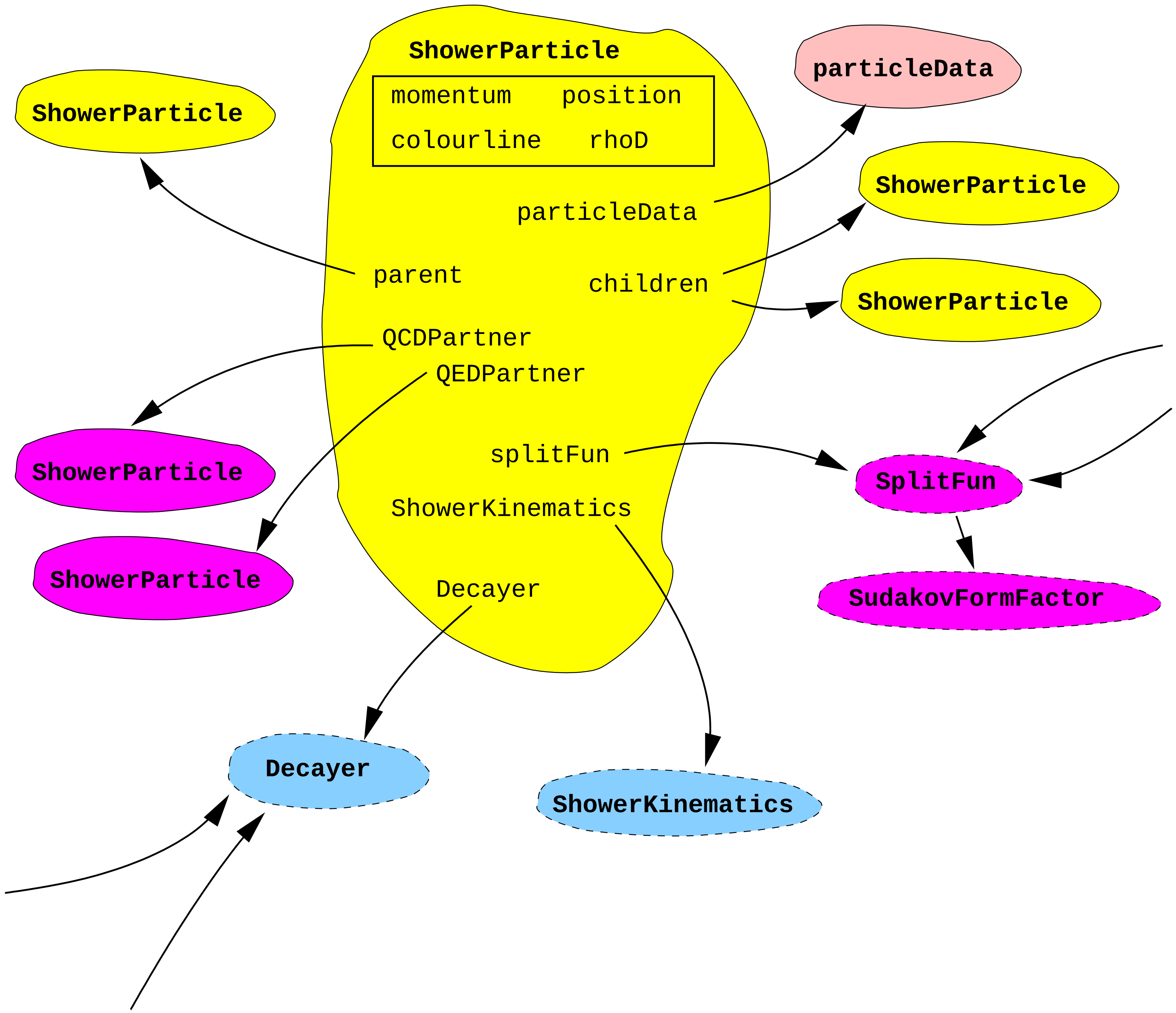


- Classes for multiscale shower are implemented and algorithms are designed appropriately.
- 'concrete' code yet incomplete.
- `FORTRAN HERWIG` should be reproduced with appropriate switches (`multiScaleShower=OFF`) and (`decayBeforeShower=ON`) for SUSY particles.
- SUSY particles might shower as well.
- photon radiation already implemented
- EW interactions foreseen, can be included easily

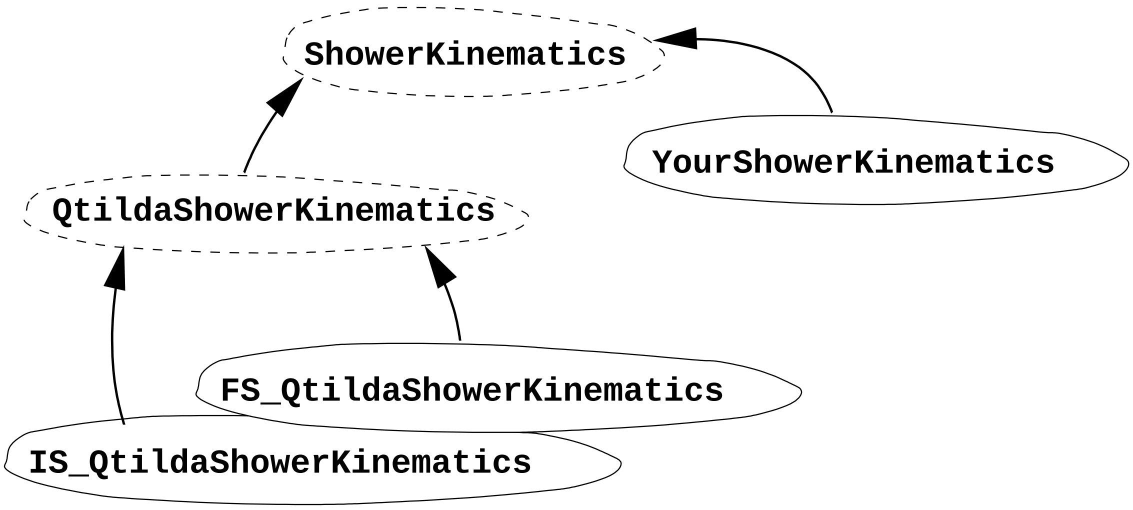
Shower Classes



ShowerParticle



ShowerKinematics

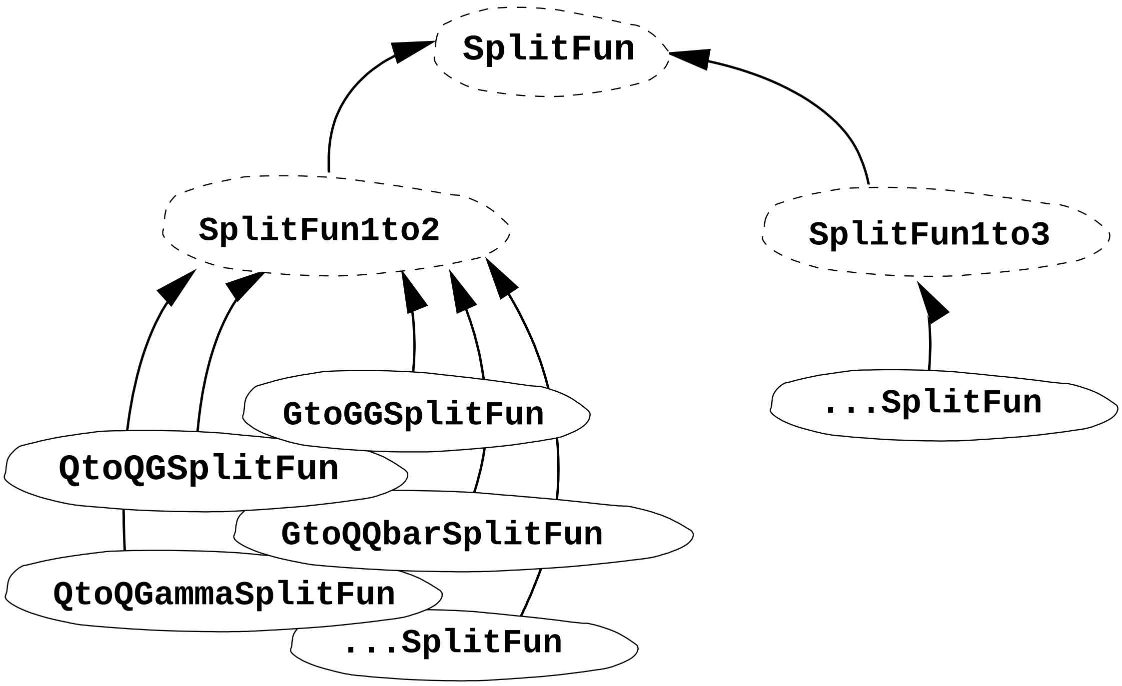


- Every splitted particle has its own `ShowerKinematics`
- methods include e.g.
  - `updateParent()`
  - `updateChildren()`
  - `updateLast()`

  to reconstruct the kinematics of the respective relatives of a particle during various stages of the evolution.
- highly abstract $\longrightarrow$ allows for a simple extension to include different sets of evolution variables.

$$\tilde{q}, \mathrm{angle}, \mathrm{virtuality}, \ldots$$

SplitFun

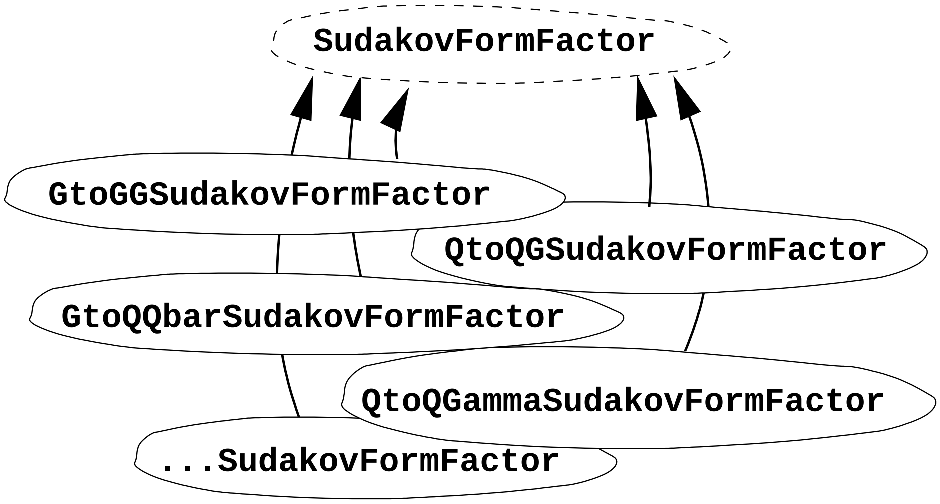


- code written abstract with $1 \to n$ splittings in mind.
- only the base `SplitFun1to2()` used so far. Implemented are the standard splittings with

$$\texttt{showerIndex()} \in \{\mathrm{QCD}, \mathrm{QED}\}.$$

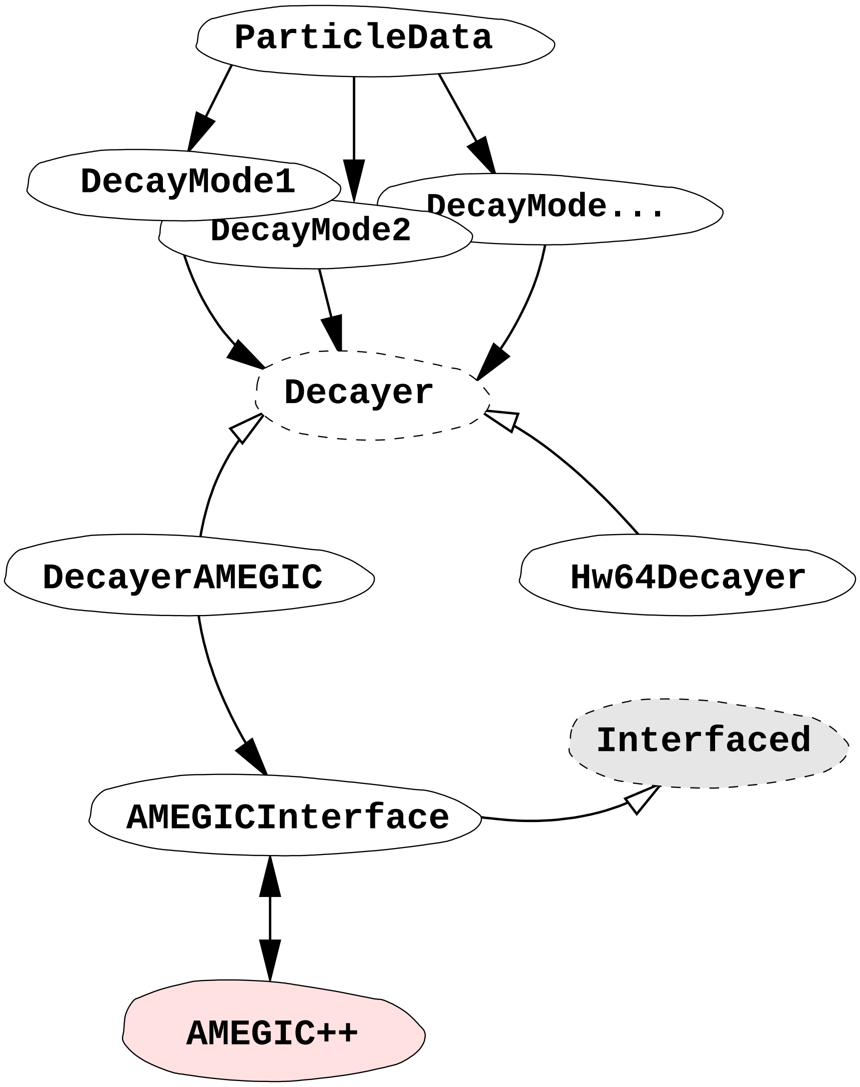
  I.e. $q \to qg$, $g \to gg$, $g \to q\bar{q}$ and $q \to q\gamma$.
- q might be heavy using the *new* $\tilde{q}$-evolution.
- new interactions easy to include.
- getting a new particle (e.g. a $\tilde{g}$) to shower is only a small administrative task ('switch on') once the appropriate `SplitFun()` is set up.

SudakovFormFactor



- Every `SplitFun` has a pointer to its appropriate `SudakovFormFactor`.
- For any type of interaction (`ShowerIndex`=QCD, QED, EW, . . .) there will be a separate pointer.
- Different interactions *compete* in the decision for a new branching: nearest to recent scale wins $\rightarrow$ flexibility.

Decays



- `FORTRAN HERWIG` is reproduced with `Hw64Decayer` using the same Matrix element codes as before (will be used for hadronic decays right now)
- `DecayerAMEGIC` gets final states for a decay mode directly from `AMEGIC++`
- works fine in principle, further tests required

Hard interactions

- some basic ME's included in Pythia7
- check them/provide our own extended set
- `AMEGIC++` may provide arbitrary ME's for multiparton final states
- ME corrections are foreseen in parton shower.
  Not yet implemented
- other authors can easily include their own matrix elements ($\rightarrow$ *safety* of OO code)

Hadronization

- Cluster Hadronization is designed, implemented and debugged
- works fine, however, further running with the full event generator might discover yet hidden bugs
- aside, also the Lund String model is implemented already in Pythia7 and will work together with `Herwig++`
- this requires that final state gluons are on-shell
  $\rightarrow$ foreseen in the design of the shower

Conclusions/Future

- Herwig++ is on its way!
  - new variables under control
  - very general shower design complete
  - implementation of shower, first `FORTRAN HERWIG`-like, ongoing and appears to work fine so far
  - multiscale shower will be developed continuously
- partonic decays from `AMEGIC++` work
- hadronic decays `FORTRAN HERWIG`-like work
- cluster hadronization complete

a very preliminary schedule is

- end of 2002: β-version for LEP1-events with `FORTRAN HERWIG` features.
- 2003: complete multiscale shower, first release?
- 2004: a version for LHC?