The RAVE / VERTIGO Vertex Reconstruction Toolkit

Winfried Mitaroff,
Fabian Moser and
Wolfgang Waltenberger

Institute of High Energy Physics
Austrian Academy of Sciences
Nikolsdorfer Gasse 18
A-1050 Vienna, Austria, Europe

A detector-independent toolkit (RAVE) for the reconstruction of interaction vertices is being created, along with a supplementary framework (VERTIGO) for testing, analyzing, and debugging. It aims at both finding (pattern recognition) and fitting (statistical estimation) the vertices. Main design goals are ease of use, high integrability into existing software, extensibility, and generality. The toolkit is coded in C++, but interfaces for other languages (e.g. Java, Python) are a desirable future goal.

Main goals

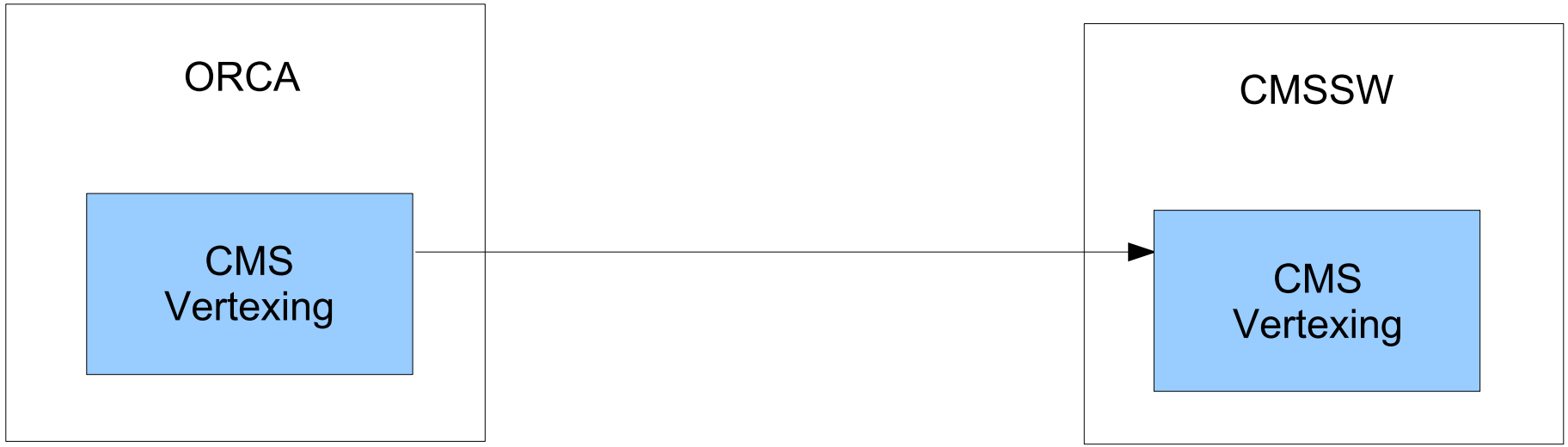
1. Creation of an extensible, detector-independent toolkit (RAVE) for vertex reconstruction:

- *RAVE = "Reconstruction in abstract vertexing environments"*;
- Starting point was the old CMS software (ORCA) which is currently being refactored and ported into the new CMS framework (CMSSW);
- The principal algorithmic asset are robustified estimators based on adaptive filters, thus downweighting the influence of outlying tracks;
- The toolkit is required to stay source-code compatible with the CMS software. It is intended to be first used with the ILC detector concepts;
- RAVE includes both vertex finding (the pattern recognition task of track bundling) and vertex fitting (estimation of vertex parameters).

2. Creation of a simple stand-alone framework (VERTIGO), enabling fast implementation, debugging, and analyzing of RAVE's algorithms:

- *VERTIGO = "Vertex reconstruction and interfaces to generic objets"*;
- Tools include Visialization, Histogramming, a Vertex Gun for generating test events, and a Data Harvester & Seeder for flexible I/O.
- VERTIGO is able also to emulate various detector setups.

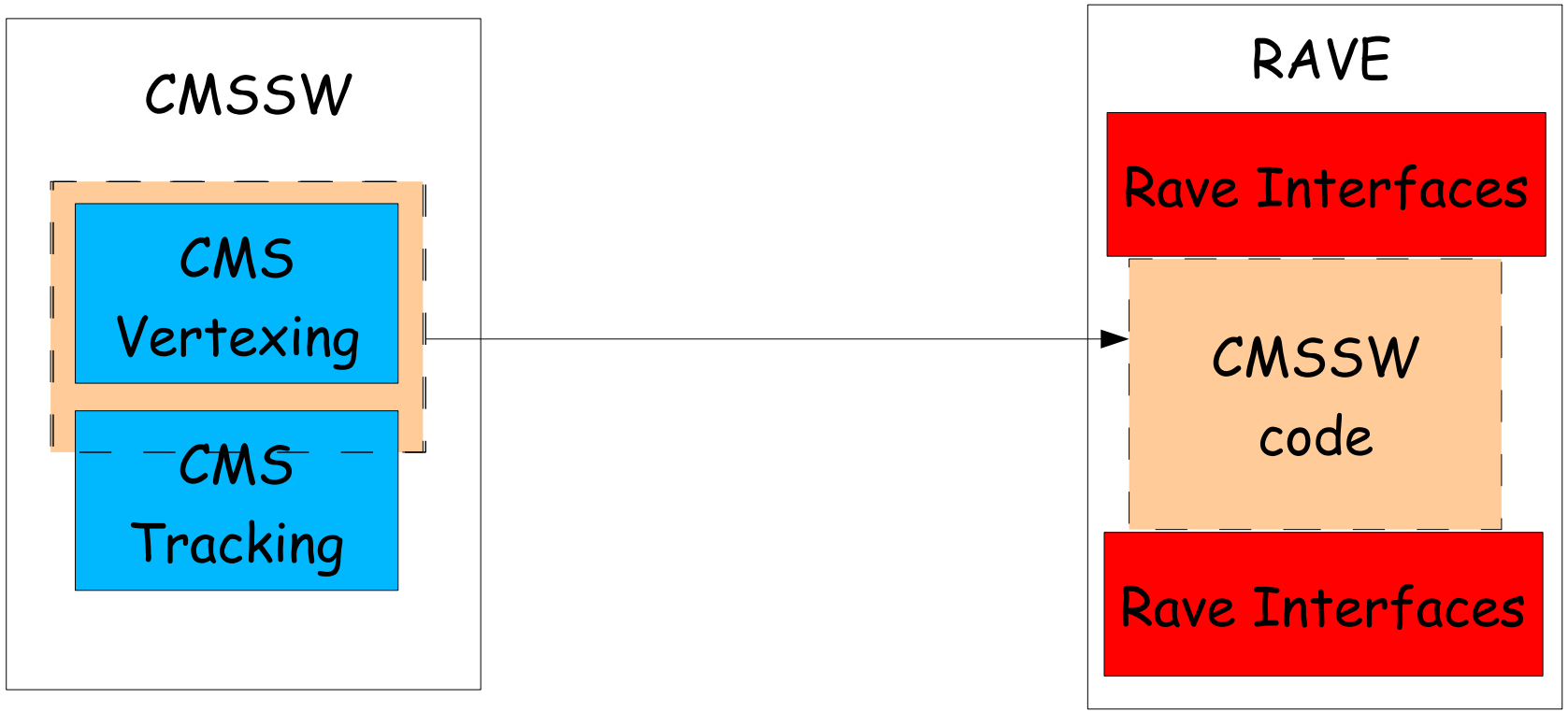
Step 1: porting from ORCA to CMSSW



Status:

- The central classes, Kalman filter, and adaptive filter have been ported;
- Sigificant tests in CMSSW have not been possible so far, because of unavailability of reconstructed tracks due to a ROOT persistency bug.

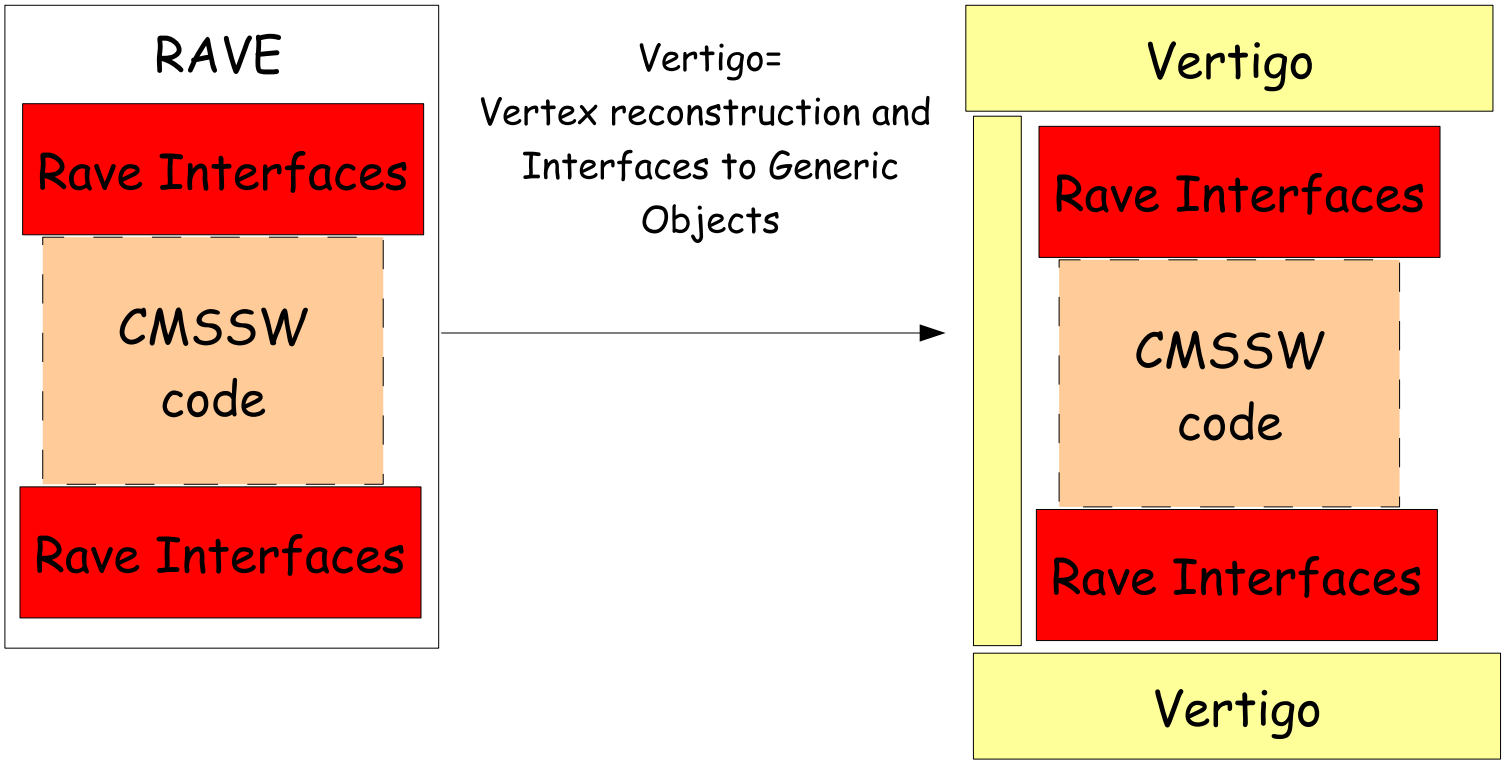
Step 2: encapsulation of CMS vertexing



Status:

- Finished.

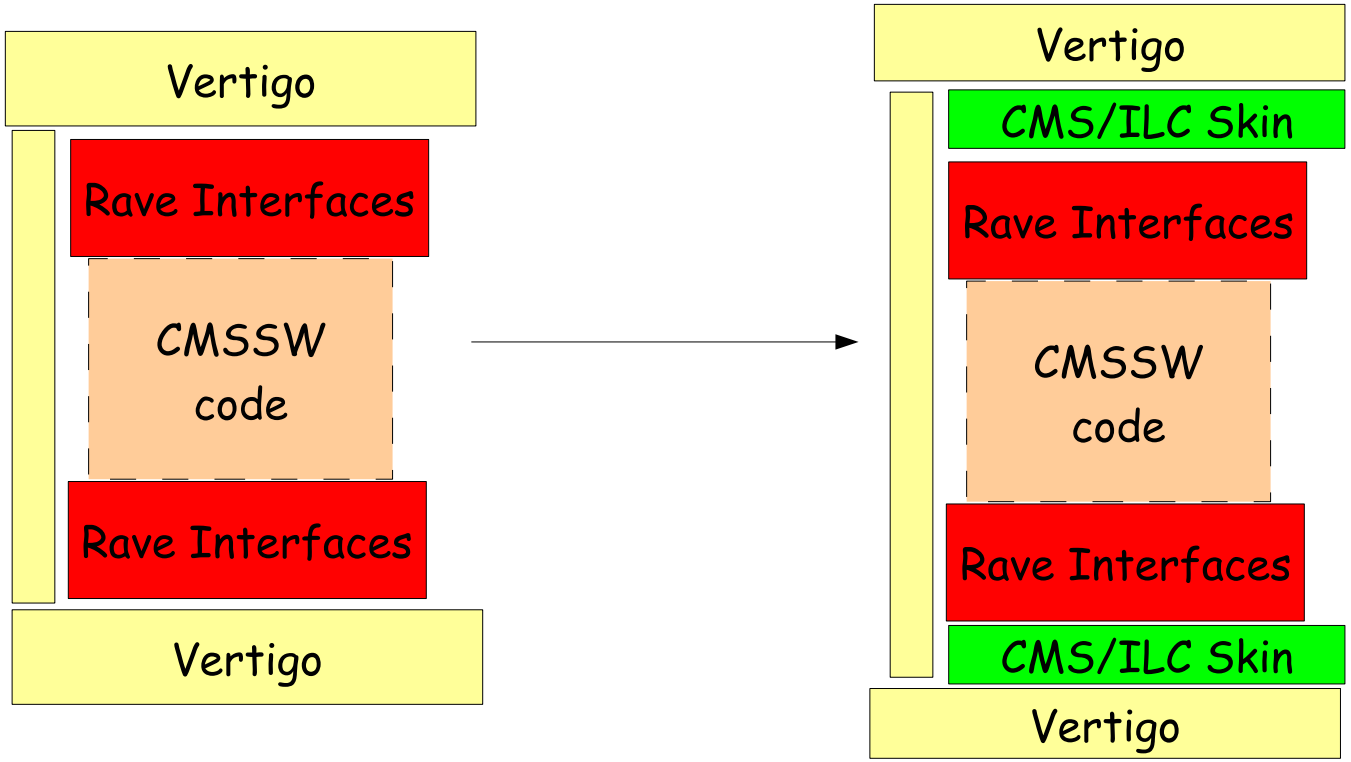
Step 3: creation of the stand-alone framework



Status:

- Finished. Tests which haven't been possible with CMSSW (see step 1) have been performed with VERTIGO !

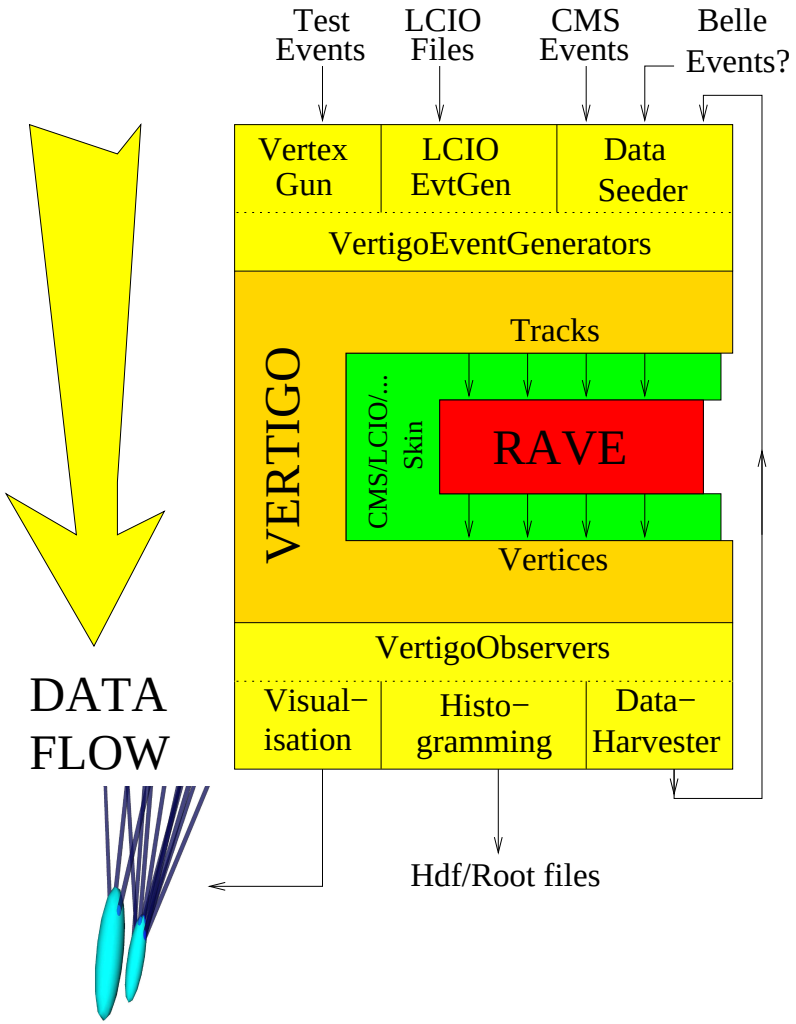
Step 4: VERTIGO emulating detector setups



Status:

- CMS skin available without material effects;
- A simple ILC detector skin may be available within a few weeks.

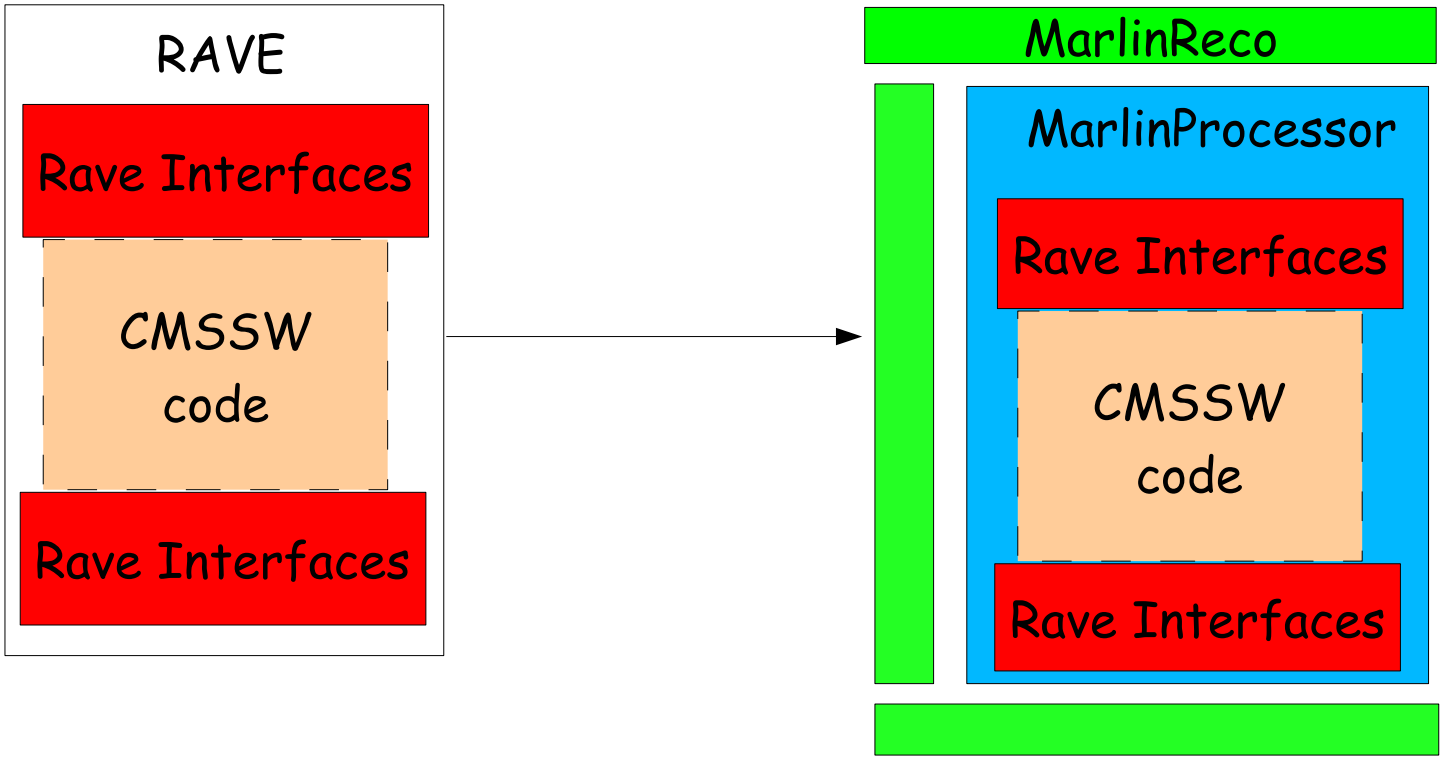
Event generation and analysis tools



Status:

- Comparable with vertexing in ORCA;
- Visualization, histogramming, vertex gun, etc.

Step 5: Embedding RAVE into ILC software



Status:

- At present aiming only at C++ software;
- Not yet started: waiting for MarlinReco to produce fully reconstructed tracks !

Plans for the future

Near future:

- Creation of a simple "VERTIGO skin" for ILC detectors (LDC, SiD) via an LCIO event generator;
- Creation of a simple "VERTIGO skin" for the BELLE detector (interface to "Panther tables" via the DataSeeder/Harvester);
- Embedding RAVE into the ILC "C++" reconstruction software (MarlinReco).

Mid-term future:

- Embedding RAVE into the ILC "Java" reconstrction software (org.lcsim) by means of a wrapper;
- Embedding RAVE into the new reconstruction software of the proposed Super-BELLE detector.

Maintenance & development:

- The HEPHY Institute of Vienna is committed to the maintenance, documentation and distribution of the RAVE and VERTIGO packages;
- The RAVE toolkit will permanently be augmented by new high-quality algorithms, as soon as available (e.g. Ben Jeffery's ZVTOP);
- Other HEP experiments using a tracking detector are welcome to contribute with a "VERTIGO skin" of their own, or alternatively to embed RAVE into their reconstruction software;
- Caveat: use of algorithms developed by CMS is subject to the "CERN public licence" (CPL).

Contributors

The Vienna team:

- Wolfgang Waltenberger: leading developer, core algorithms, CMS interface;
- Fabian Moser (undergraduate student): core algorithms, ILC interface;
- Winfried Mitaroff: ILC interface, integration with Java;
- Gerald Richter (graduate student): BELLE interface.

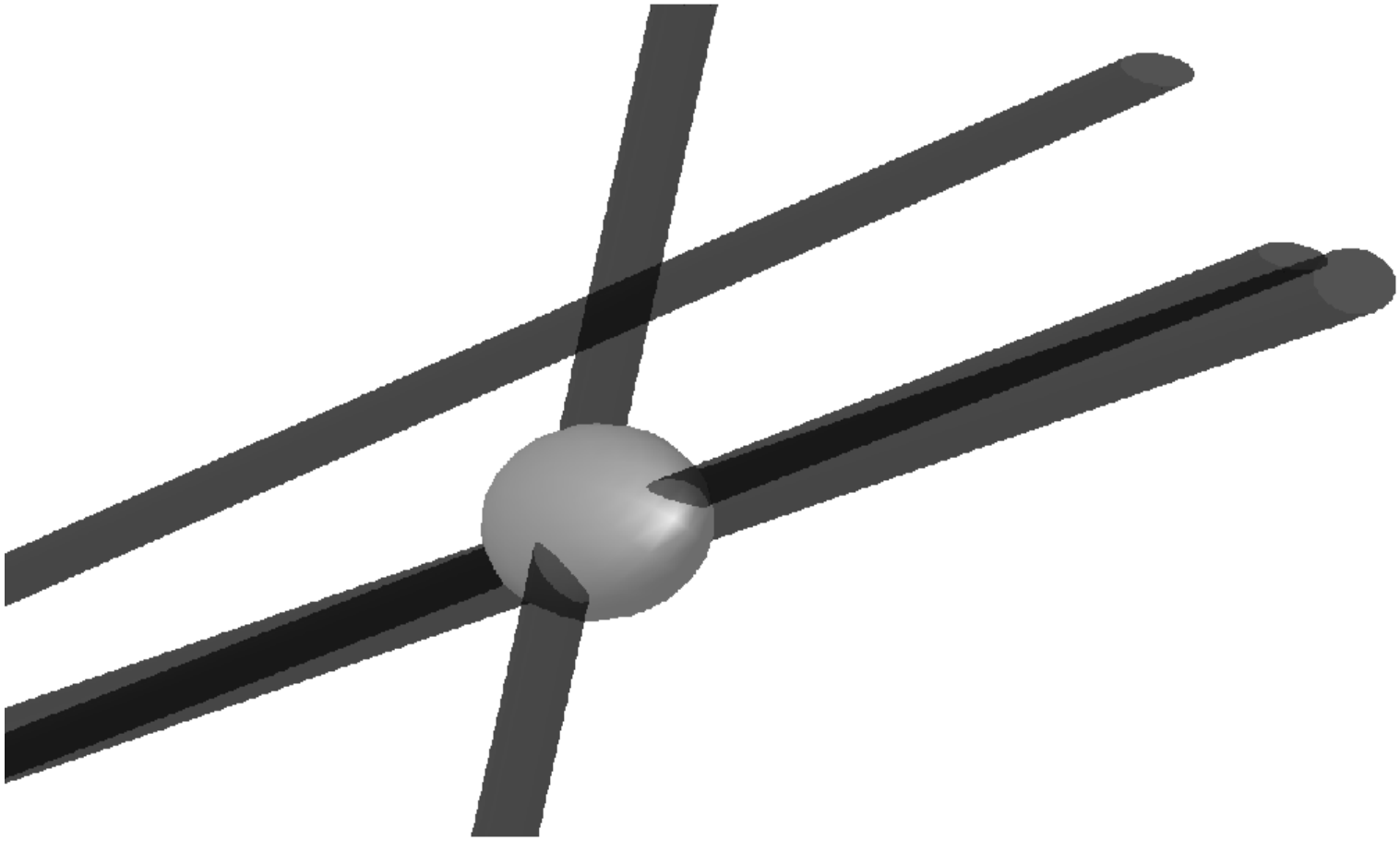
External contributors:

- The CMS vertexing community (Brussels, CERN, Lyon, Zurich, . . .);
- Hopefully in future: the ILC "C++" and "Java" software communities.

Spread the word:

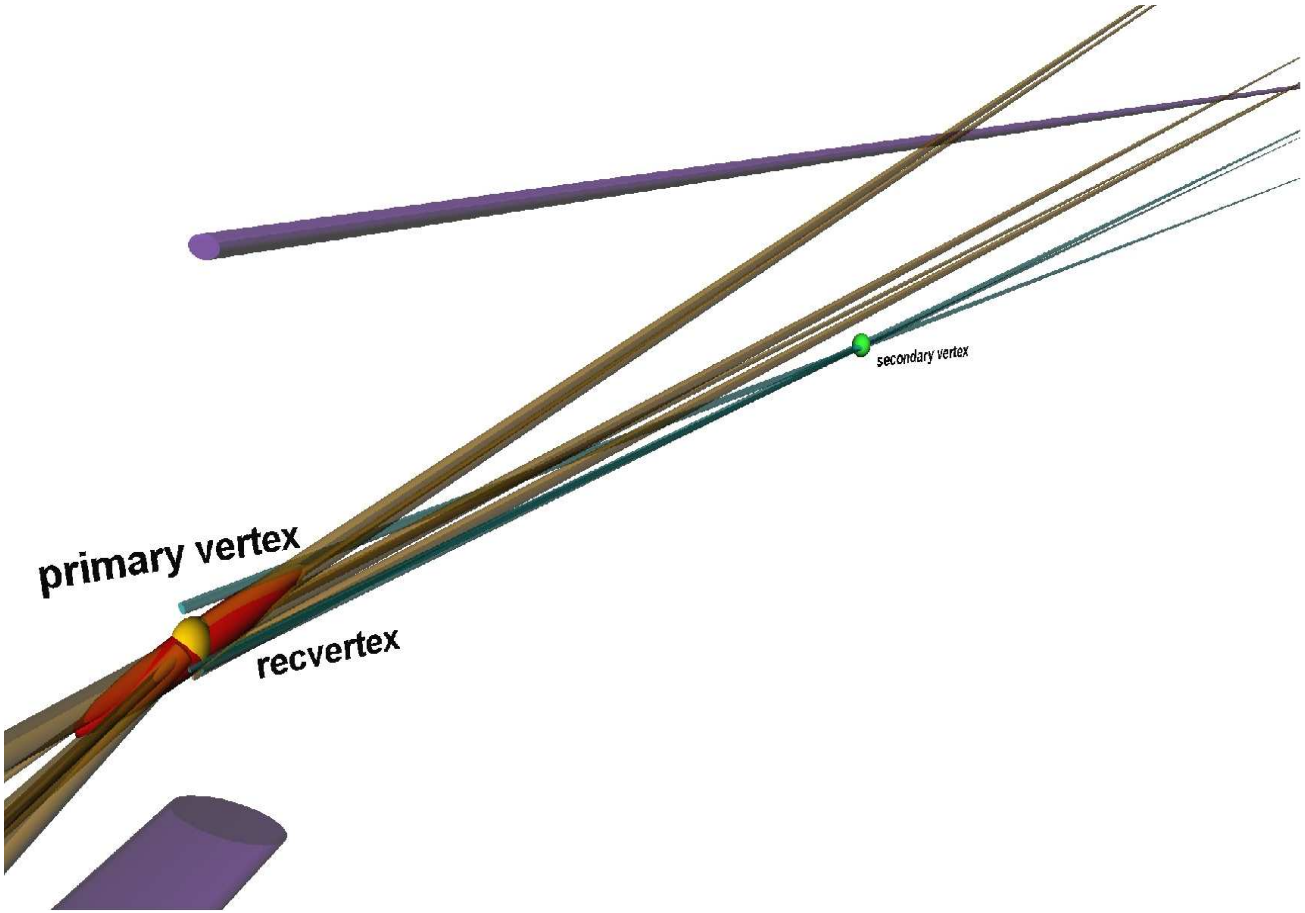
- Earlier presentations: LCWS 2004 (Paris), CHEP 2004 (Interlaken), . . . ;
- *Development was stalled in 2005 because of WW's 1 year civil service, and WM having been busy with organizing the ECFA WS 2005 in Vienna;*
- Recent presenation: ILC Software Meeting 2006 (Cambridge);
- Next presentations: ILC Software Meeting 2006 (DESY), ECFA WS 2006 (Valencia), IEEE Nuclear Science Symposium 2006 (San Diego).

Visualization example (1)



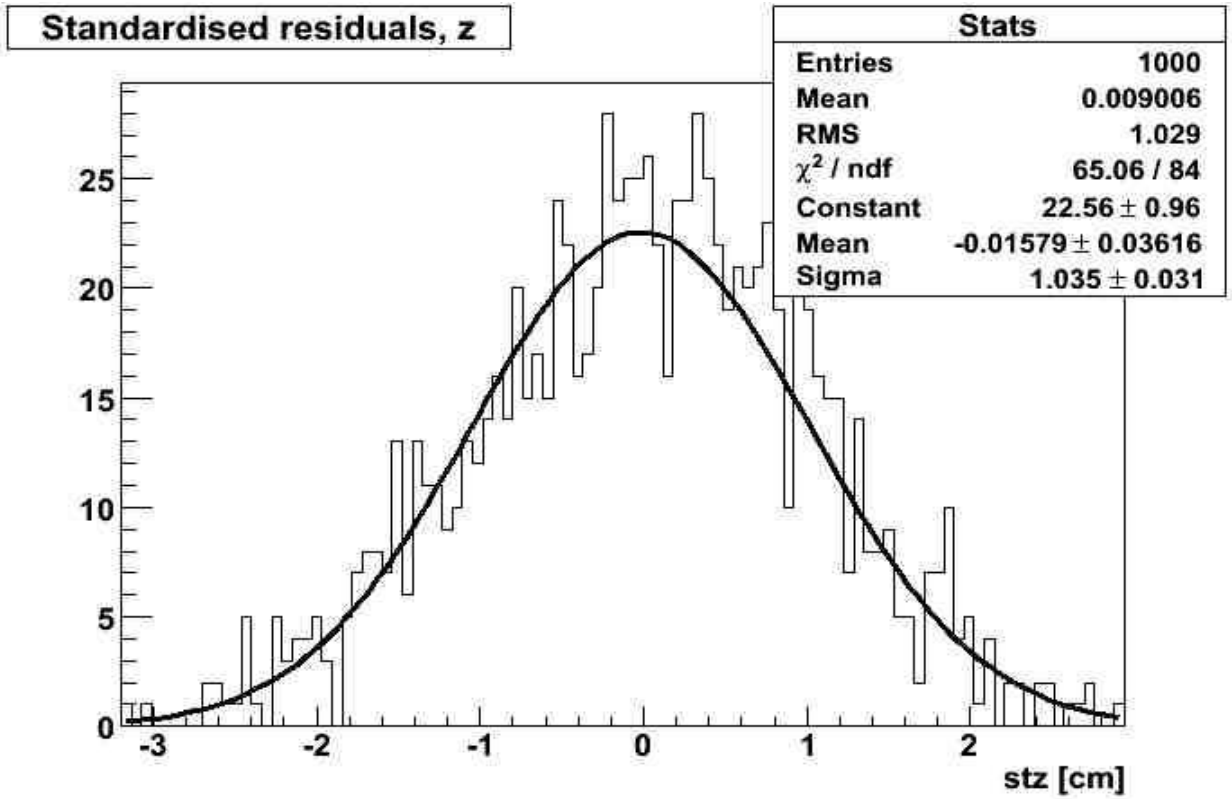
ORCA 8.2.0 $b\bar{b}$ event ($B_s \rightarrow J/\psi\Phi \rightarrow \mu^-\mu^+K^+K^-$), the B_s vertex fitted adaptively.

Visualization example (2)



ORCA 8.2.0 $c\bar{c}$ event, the primary vertex (red) fitted adaptively.

Plot from a RAVE release test



Normalized residuals (pulls) of adaptively fitted vertices; example event on the left.

The Vertexing Circle's credo

"The method of Least Squares is seen to be our best course when we have thrown overboard a certain portion of our data -- a sort of sacrifice which has often to be made by those who sail the stormy seas of Probability."

F. Y. Edgeworth, 1887

"Let us not throw away data all too hastily. Instead, let us weight and reweight the data, consider and reconsider alternative models. Only if we must, at the latest possible stage, shall we distinguish between 'in' and 'out', between signal and noise."

The CMS vertexing circle, 2004