A VERTEX RECONSTRUCTION TOOLKIT AND INTERFACE TO GENERIC OBJECTS (VERTIGO)

W. MITAROFF and W. WALTENBERGER
Institute of High Energy Physics, Austrian Academy of Sciences
Nikolsdorfer Gasse 18, A-1050 Vienna, Austria, EU

A proposal is made for the design and implementation of a detector-independent Vertex Reconstruction Toolkit (VERTIGO). It aims at re-using existing state-of-the-art algorithms for geometric vertex finding and fitting by both linear and robust estimation methods; kinematic constraints will also be included for the benefit of complex multi-vertex topologies. The design is based on modern object-oriented techniques. A core (RAVE) is surrounded by a shell of interfaces and a set of analysis & debugging tools. The implementation follows an open source approach and is easily adaptable to future standards.

1 Motivation and Goals

In the offline data reduction chain, the early stages – local pattern recognition and track fitting – are highly detector-dependent, whereas the next stage – vertex reconstruction (finding and fitting) is almost fully detector-independent. Vertex fitting with kinematic constraints may rather be subject to the requirements of a subsequent physics analysis.

Why looking for a toolkit? Geometric vertex finding and fitting must not compromise the high spatial resolution of modern vertex detectors. This goal can be achieved by new, sophisticated methods beyond the traditional least squares or Kalman filter estimators, using robust, non-linear, mostly adaptive algorithms. It is not desirable for each new detector to re-code vertex reconstruction from scratch – provided there exists an adequate, reliable and easy-to-use TOOLKIT.

As a good point to start from, we propose taking out vertex reconstruction from the CMS general reconstruction software ORCA, thus providing the basic stock for the core of such a toolkit. However, the core must be complemented by flexible interfaces and a modular set of analysis & debugging tools.

2 Design Concepts of VERTIGO

A draft version of the overall design – core, interfaces and optional packages – is shown in figure 1.

2.1 The RAVE core

The core, called RAVE ("Reconstruction Algorithms for Vertices"), is to become a collection of the best algorithms available for vertex reconstruction – finding, fitting and kinematics; starting with the packages developed by CMS [1], but open for entries by other parties. The code is to be based on C++ and HEP-wide (albeit not CERN-specific) OO standards.

Candidate core algorithms include packages of general tools (e.g. clustering), for vertex fitting (e.g. the deterministic annealing filter – DAF), and for vertex finding (e.g. the "apex point" method). At present, the list is dominated by algorithms implemented in the CMS offline reconstruction (ORCA), but non-CMS candidates exist (e.g. ZVTOP [2]). New entries of first-class algorithms are highly welcome.

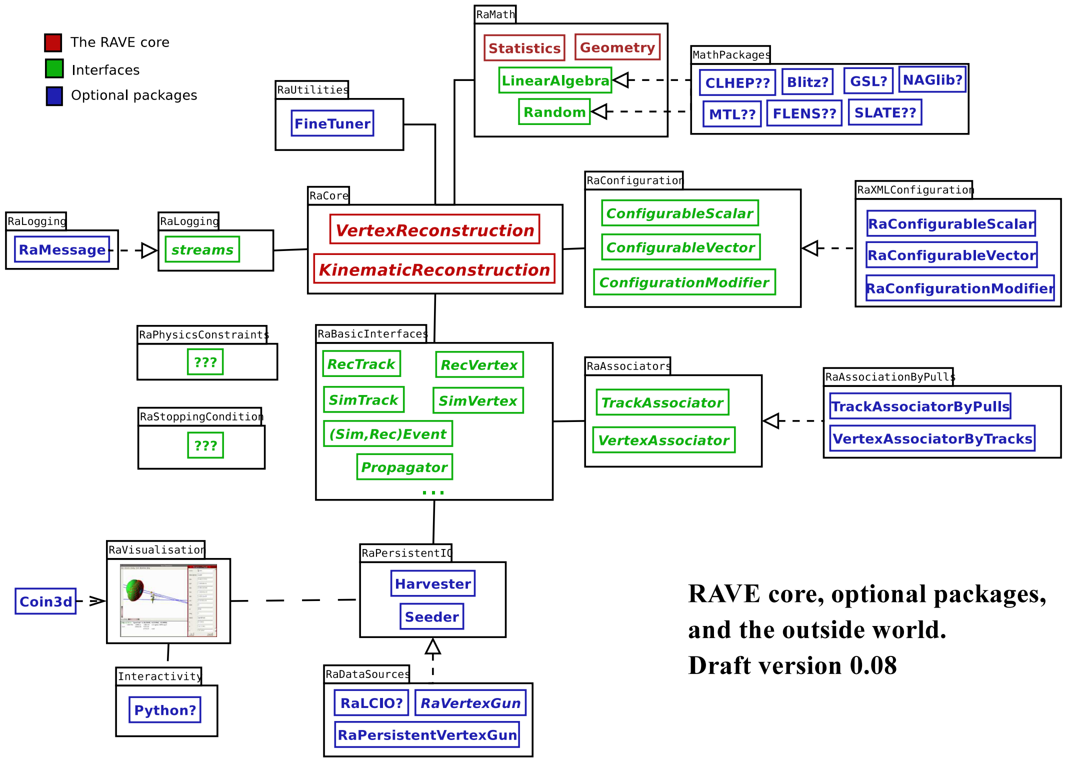

Figure 1: VERTIGO overall design (draft).

Documentation (based on Doxygen) of the algorithms, including information about their scope of application, will be provided. The proper choice of algorithms is also supported by the SKIN concept (see below).

2.2 Shell of interfaces

Access from/to the outside world will exclusively proceed via a "shell" of interfaces surrounding the core. These interfaces make use of adaptors in order to keep a high level of abstraction; good design will be the key of success.

2.3 Analysis & debugging tools

Analysis & debugging tools are optional packages, containing those parts of code which might be helpful without being strictly necessary. Prototypes of a few packages have already been written: the framework for a stand-alone realisation of VERTIGO, a persistency storage solution, data sources, and a visualisation tool; but much more work is still to be done. Extensive use of open standards will minimize the burden of development for this part of the toolkit.

The persistency storage solution was originally based on top of ROOT; it is currently being extended by more standard-compliant alternatives (AIDA and XML). Data sources include a "vertex gun", interfaces to LCIO, etc. All I/O is handled through a "data harvesting" concept (which may possibly be integrated as front/end in AIDA): object → STL map → ASCII/ROOT/AIDA file ("harvester") and vice versa ("seeder"). The STL mapping is heterogeneous: it handles int/double/string objects as multi-type.

Visualisation is deliberately kept simple for the sake of detector-independence. It follows the model-view-controller (MVC) paradigm and is based on COIN3D. Object data are accessed as multi-type STL maps: at present only indirectly from a file through the seeder; in future maybe also directly through the harvester and a TCP stream. Interactivity is at present limited to manipulators on graphic objects. The tool may later be augmented with full-scale interactivity, to be provided by PYTHON (or some other scripting language).

Proper choice of a math library package (including linear algebra) is crucial for the efficiency and reliability of the toolkit. CLHEP appears to be the only choice freely available today, but there are serious doubts about its reliability. NAGlib is a reliable alternative, but may be too expensive for users outside of campus licence agreements. Generic (template) libraries would be our preferred choice; candidates exist, e.g. MTL, GSL, FLENS, SLATE, BLITZ (all GPL-licenced), but none providing the full functionality required.

2.4 The SKIN concept

Different experiments will use different sets of the optional packages. A package may be part of and shipped with VERTIGO; or it may be maintained by the particular experiment, and VERTIGO provides only the appropriate interface.

An experiment-specific set of packages is called a SKIN. Examples are a stand-alone skin (called the "framework"), CMS skin, TESLA skin, LCD skin, etc. Pre-defined skins may easily be selected by the user. Maintenance and distribution of the toolkit will be supported by a CVS repository at HEPHY Vienna.

3 Conclusions and Outlook

This is (according to our knowledge) among the first large-scale attempts of refining a substantial part of reconstruction software into a detector-independent toolkit. Interests in using the toolkit, once it will be released, have been expressed by CMS, ATLAS, LHCb, BELLE and Linear Collider (TESLA, LCD) collaborators, with more to follow. Close collaboration among the contributing laboratories is welcome and will be essential for success.

References

1. R. Frühwirth et al.: New vertex reconstruction algorithms for CMS, *Proc. CHEP 2003, La Jolla (Cal, USA)*, SLAC-R-636 / eConf C0303241.
2. D.J. Jackson: *Nucl. Instrum. Methods* A**388** (1997) 247.