Dmitry Onoprienko

Kansas State University

for

SiD Tracking Group

Tracking Software Status & Plans

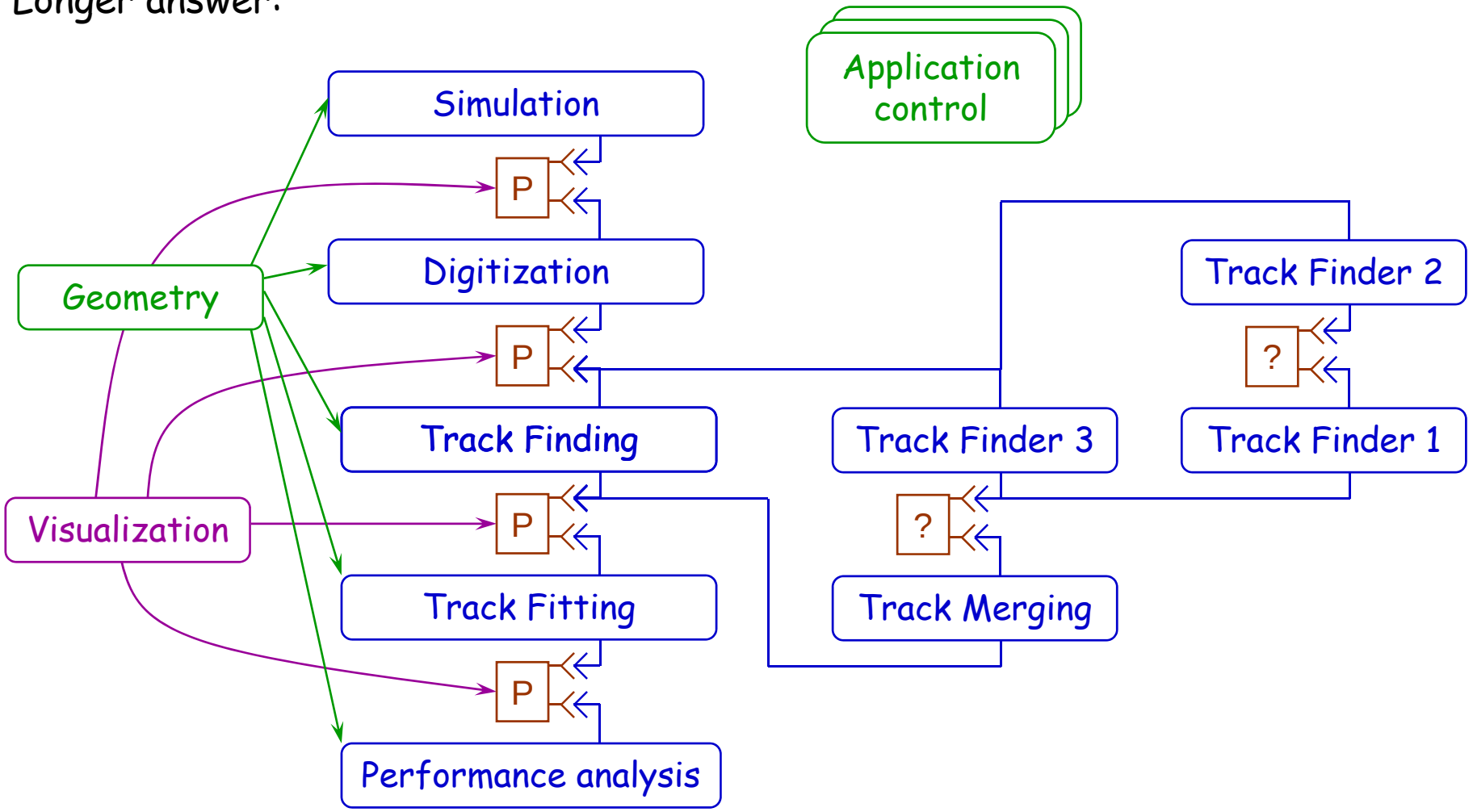
SiD Workshop

Fermilab, April 2007

Tracking software – what is needed ?

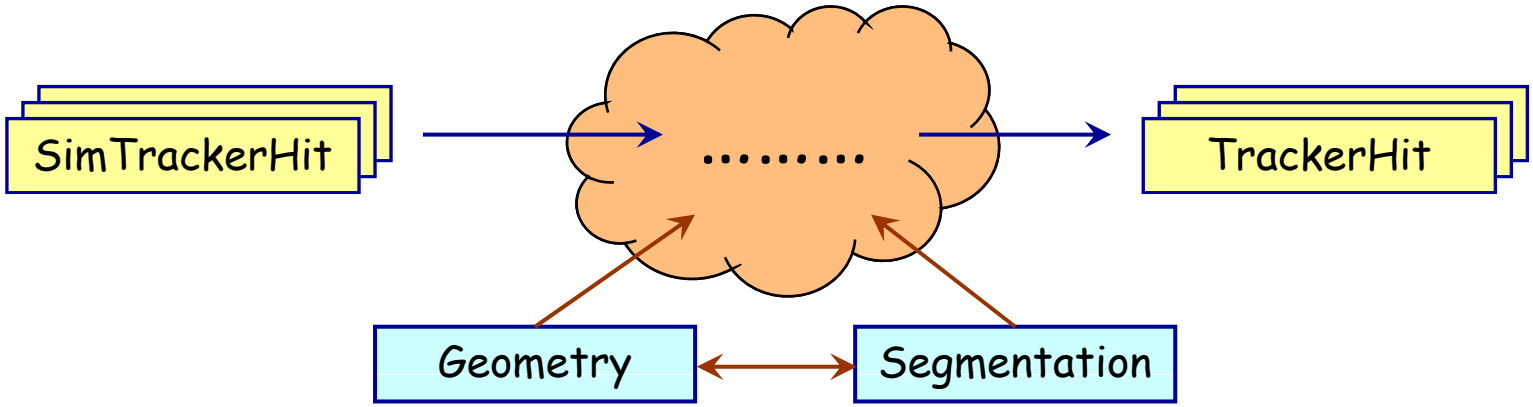
Short answer: algorithms and infrastructure to optimize the detector and characterize its performance.

Longer answer:



Digitization

- Simulation of charge deposition in sensors.
- Simulation of tracker segmentation and readout electronics.



Available algorithms :

- **CCD Pixels** (Nick Sinev)

  Package: **org.lcsim.mc.CCDSim**

  Fully functional.

- **Silicon strips** (Tim Nelson)

  Package: **org.lcsim.contrib.SiStripSim**

  Work in progress.

Reconstruction - Track Finders

- **Vertex Seeded Tracking** (Fred Blanc, Steve Wagner)
  - Package: **org.lcsim.contrib.SODTracker**
  - Find track seed in vertex detector and pick up hits in outer tracker
  - Uses MC truth for vertex seeds and runs on SimTrackerHits at the moment
- **Conformal Mapping** (Norman Graf)
  - Packages: **org.lcsim.recon.tracking.trf*****
  - Conformal mapping of circular trajectory to a straight line
  - Uses alternative tracking infrastructure
- **Calorimeter Seeded Tracking** (Dima Onoprienko, Eckhard von Toerne)
  - Package: **org.lcsim.recon.cat**
  - Find MIP stubs in the calorimeter and extrapolate them into tracker, picking up hits
  - Can be used to reconstruct long-lived particles (K_S^0, Λ, etc.)
- **Standalone Outer Tracking** (Tim Nelson, Bruce Schumm)
  - Package: **org.lcsim.contrib.tracking**
  - Find track candidates using outer barrel tracker
- **Standalone Outer Tracking** (Rich Partridge) [Under development]
  - Package: **org.lcsim.contrib.seedtracker**
  - Flexible selection of seed layers and track selection criteria.

SeedTracker

Rich Partridge

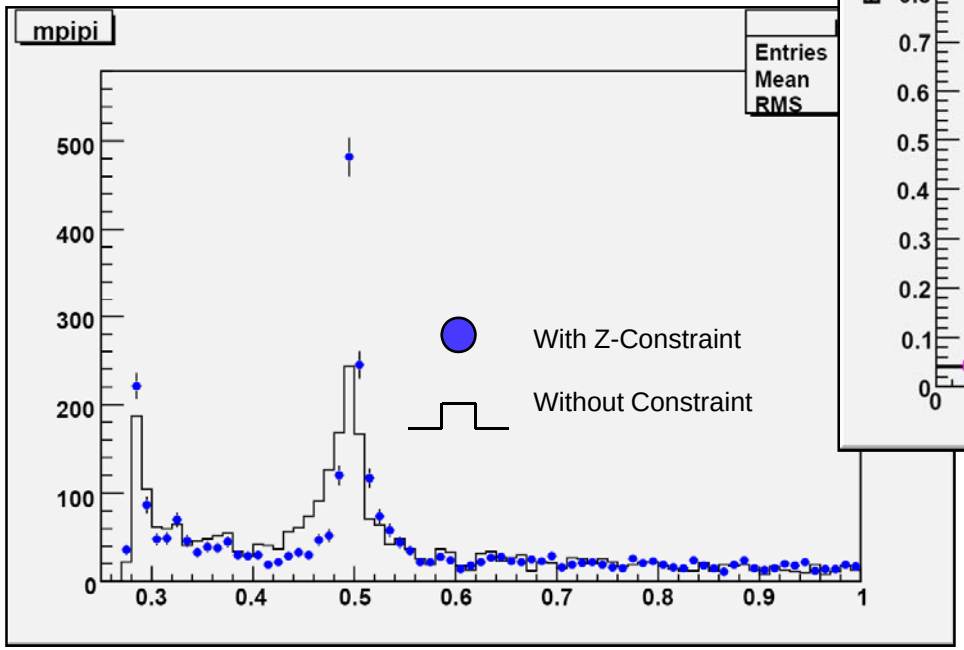
- Primary goal is to provide track seeds from the vertex detector for use by outer tracker algorithms
- Provides brute-force search of all three hit combinations in seed layers consistent with track selection criteria
- To reduce combinatoric-driven fakes, requires additional hits in confirmation layers
- Algorithm is driven by a set of "strategies"
  - Each strategy defines specific layers to use for seeds, confirmation hits, etc.
  - Strategy also specifies track selection criteria (cuts on P_T, DCA, χ^2, number of hits, etc.)
  - Multiple strategies allows track finding in barrel, disk, or combinations of barrel and disk layers
- Code currently works on single tracks, but needs to be tuned up for more complex events
  - Goal is to have these wrinkles worked out during the next few weeks.

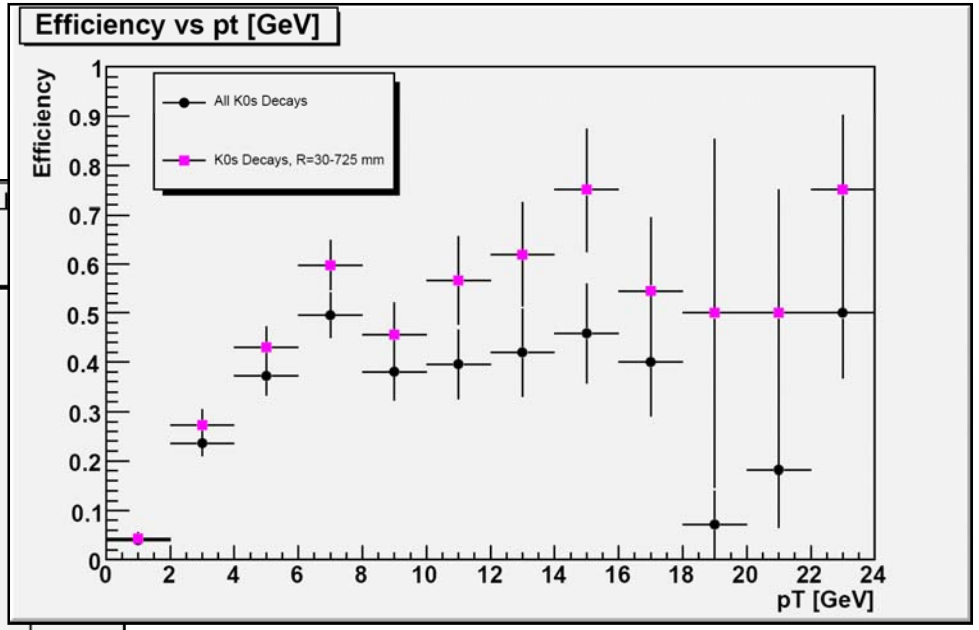
Calorimeter Assisted Tracking

Eckhard von Toerne
Dima Onoprienko

- Moved to production area: **org.lcsim.recon.cat**
  expect many changes in the next few weeks; fully functional version will always be available
- Updated the package to make use of new framework features
- Improved efficiency for K_S^0 reconstruction in hadronic events:

Invariant Mass M(ππ) in t-tbar events @ 500 GeV





Near-term plans:

Performance characterization & tune-up

Comparison of different tracker geometry and segmentation options.

Switching to new geometry and infrastructure classes as they become available.

Reconstruction - Fitters

Available algorithms :

- **Weight matrix** (Nick Sinev)
  - Package: **org.lcsim.contrib.NickSinev.tracking.wmfitter**
  - Complete, functional
- **Kalman filter** (Fred Blanc, Steve Wagner)
  - Packages: **org.lcsim.contrib.KFFitter**
  - Work in progress
- **Kalman filter** (Norman Graf)
  - Package: **org.lcsim.recon.tracking.trffit**
  - Designed to work with TRF toolkit
- **… and a few packages have their own private fitters.**

KFFitter

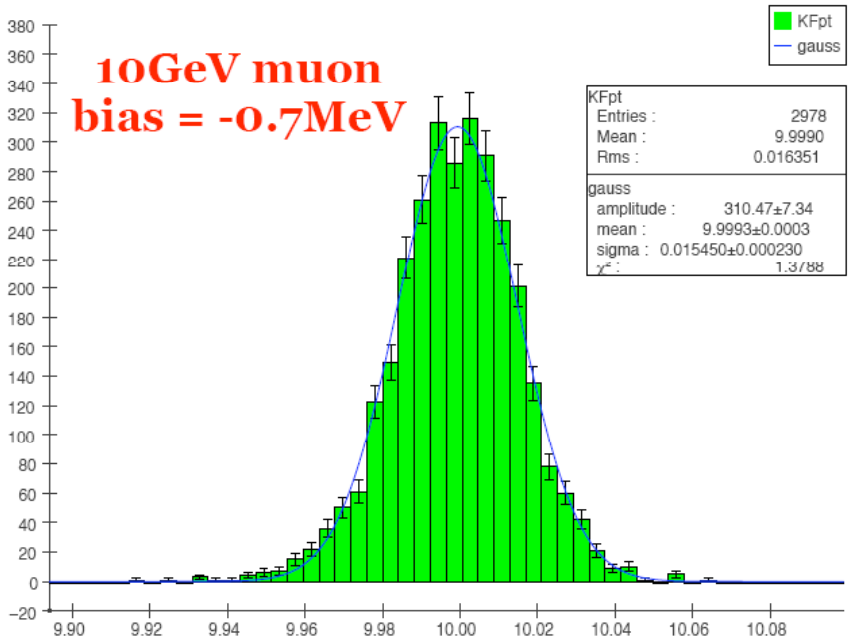
Steve Wagner, Fred Blanc

KFFitter loosely inspired by BABAR Kalman fitter
[D.Brown CHEP'97 http://www.ifh.de/CHEP97/abstract/a341.htm]

Tested (with SODTracker) on single muons
http://ilcagenda.linearcollider.org/conferenceDisplay.py?confId=1438

Current limitations :

- Works in Barrel only
- Runs on SimTrackerHits (+ smearing)
- Hard-coded geometry (cylinders)



Plans:

- extend to endcaps
- use geometry package
- run on TrackerHits
- improve smoothing and error treatment algorithms
- use real pattern recognition instead of a cheater in forming VTX seeds (SODTracker)
- combine KFFitter and SODTracker to use Kalman filter in selecting hits

Analysis – Monte Carlo Tools & Cheaters

Lots of private tools and cheaters – not all maintained – this is an area where some clean-up and consolidation could be useful.

org.lcsim.contrib.NickSinev.ztracking

org.lcsim.contrib.onoprien.mcTrackFinder

org.lcsim.recon.cheater

org.lcsim.recon.tracking.cheat

org.lcsim.recon.ztracking.cheater

……

Many high-level interfaces for hits and reconstructed objects do not provide easy access to MC truth – the information needs to be extracted following a chain of references to their precursors.

We need to maintain a set of standard tools for doing this.

Analysis – Benchmarking & Efficiency Tests

Goals :
- Compare algorithms
- Optimize detector design

Eventually, we want to benchmark on physics. However, having a standardized set of tools for simple performance tests would be highly useful at this point.

- Efficiency, purity, angular/Pt dependence, etc.
- Single particles, hadronic events, tracks in the jet core, etc.
- Effect on PFA.

A number of packages are used by different groups :

Santa Cruz: see Chris Meyer's talk at http://scipp.ucsc.edu/~schumm/nlc/sim/chris_meyer_0307.ppt for the kind of studies they are going. The resolution numbers in the talk were obtained with crude SODTracker fitter, now switching to KFFitter.

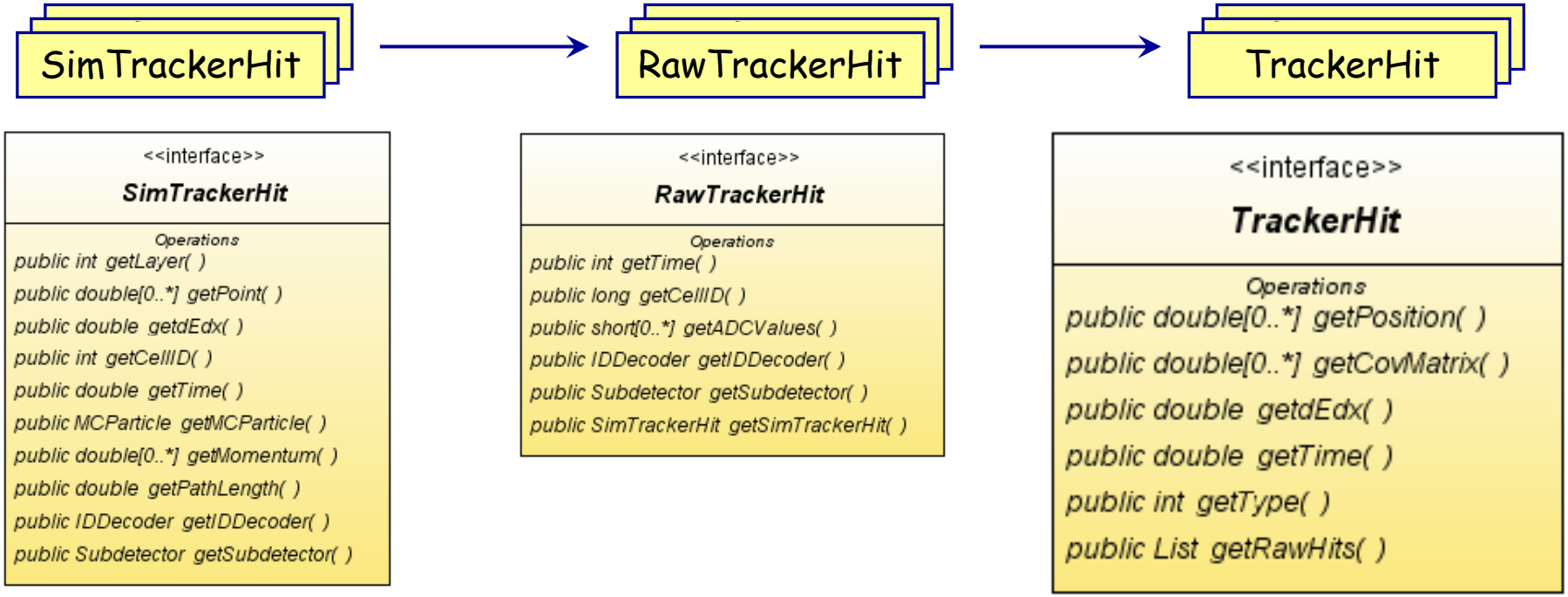
Oregon: see Nick Sinev's weight matrix fitter resolution studies: http://www-sid.slac.stanford.edu/vertexing/resolution/sidmay06.htm

Kansas State: org.lcsim.contrib.onoprien.tester package – algorithm independent track finder performance testing suite and K_S^0 reconstruction tests.

Others …

Not always easy to compare results.

Tracking infrastructure – currently used hit objects

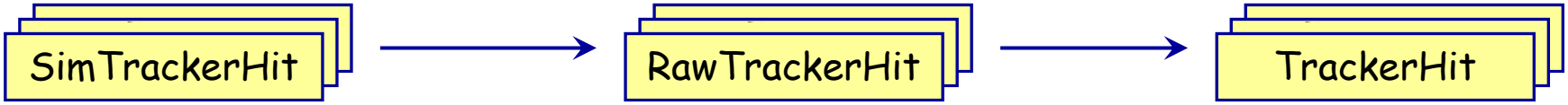


TrackerHit issues :

- No clear way to access MC truth
- No access to geometry
- No way to describe a non-point-like object like a silicon strip.
- Unsafe and ambiguous return types.

To work around these issues, various packages introduced their own custom extensions, resulting in incompatibilities between different tools.

Tracking infrastructure – currently used hit objects



Other missing ideas :

Clusters of hit strips/pixels.

Given a cluster or hit, how to navigate to associated geometry information ? Given a geometry object, how to find associated hits and clusters ?

Existing position info is in global coordinates but the native information of strips/pixels is in local coordinates.

Concept of "Hit on a Track":

- Some corrections can only be made after a cluster is associated with a track.
- How to handle cluster splitting ?

Classes that organize hit objects for fast access by pattern recognition and fitting code.

Clear definition of the point after which processing of real data and processing of MC events proceed along the same path.

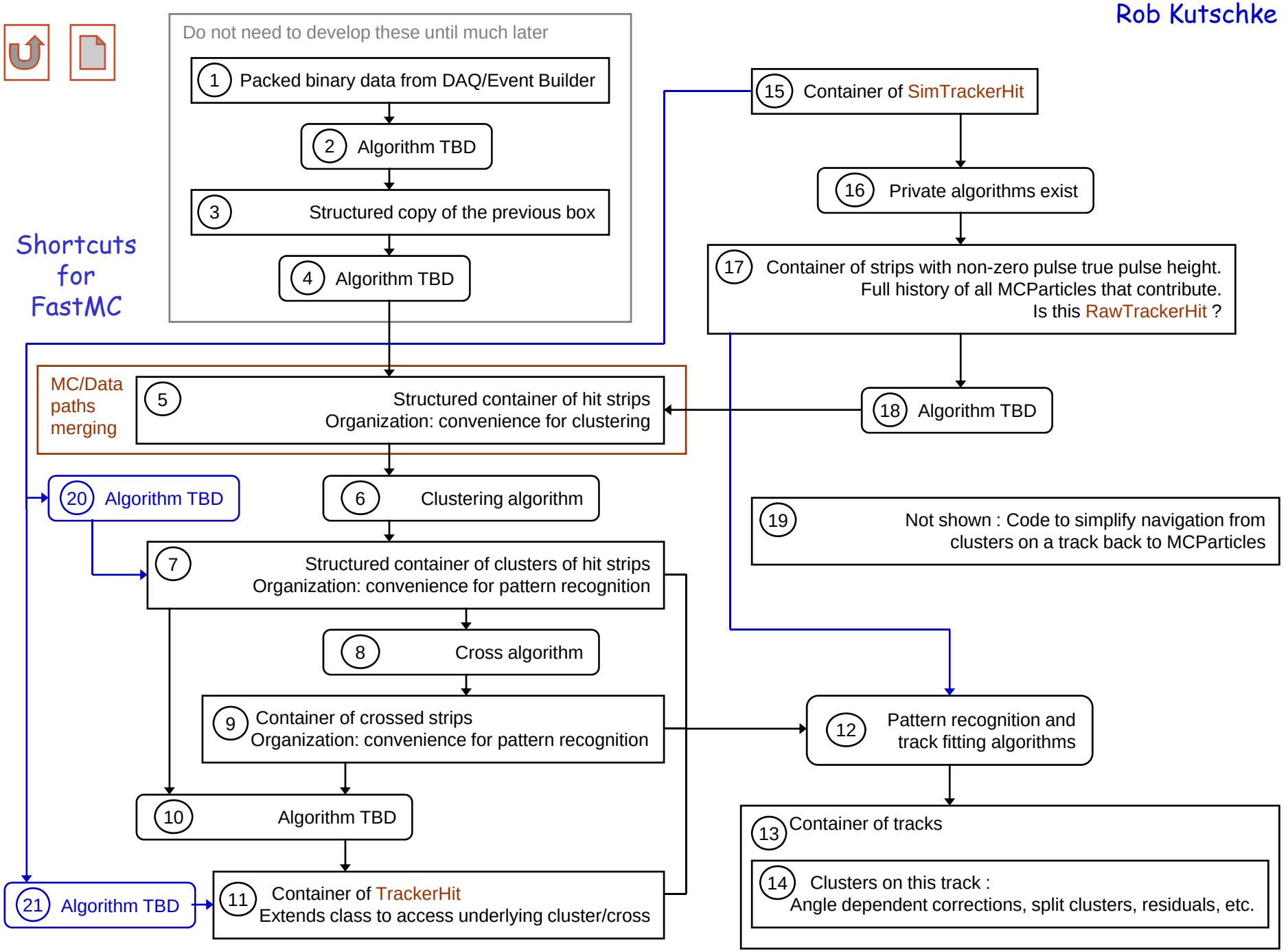
- We want to use the same framework for MC, test beam data and SiD data.

Defined mechanism for overlapping events, or adding noise to the event.

Access to MC truth is sometimes incomplete and awkward to use.

Fermilab software group sketched a proposal :

Rob Kutschke



Fermilab proposal was discussed at Sim/Reco meeting on March 27.

More details and discussion:
http://ilcagenda.linearcollider.org/conferenceDisplay.py?confId=1465

The discussion is ongoing – feedback is very welcome.

Fermilab group (Rob Kutschke, Caroline Milstene, Hans Wenzel) is available to work on this.

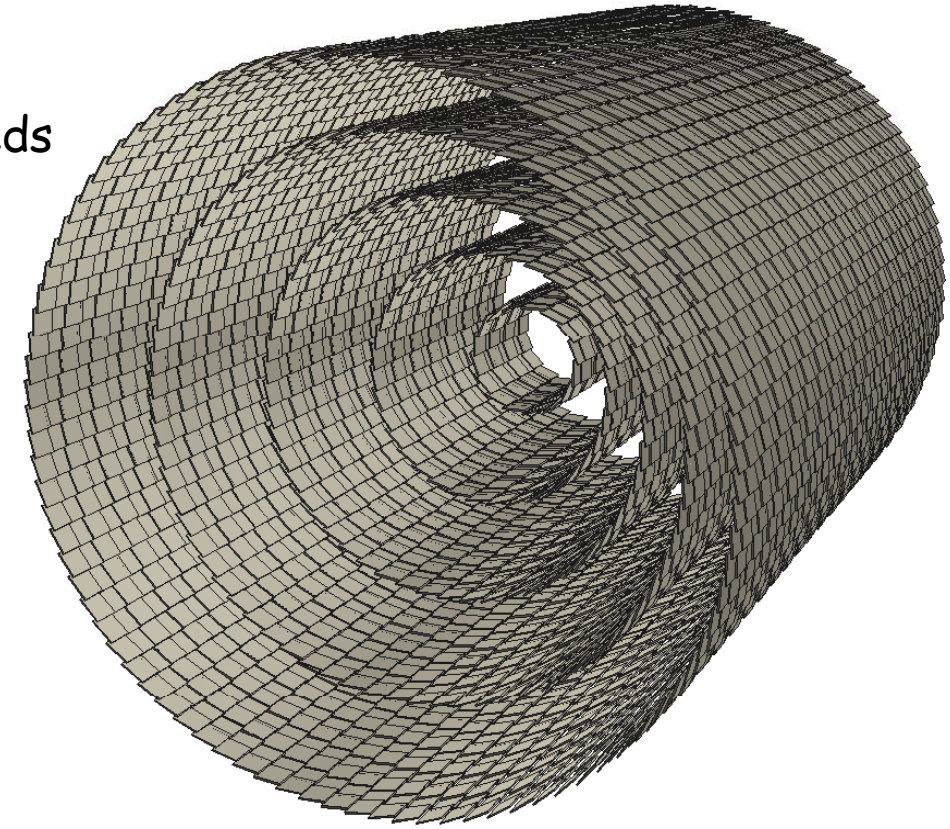
Get in touch with them if you would like to be in the loop…

New Geometry System

Jeremy McCormick
Tim Nelson

More realistic tracker simulation is needed to understand tracking performance as a function of sensor design and module layout

- overlapping planar silicon
- realistic detector segmentation



To be useful, this complex geometry needs to be represented in reconstruction.

Current system :

Flat list of subdetectors, no hierarchy

No geometrical objects smaller than subdetectors; nothing to attach parameters, conditions, and data to.

No infrastructure for conversions between local and global coordinates, finding detector components given geometrical position, and so on.

As a result, algorithm authors code their own – mutually incompatible –solutions.

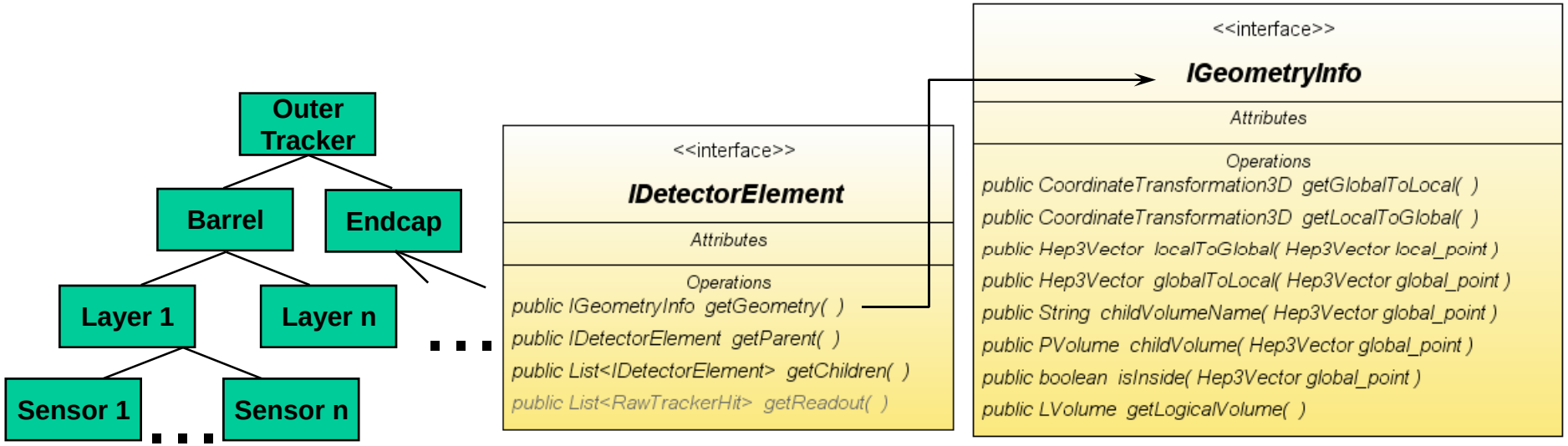
New Geometry System

Jeremy McCormick
Tim Nelson

Geometry hierarchy – tree of logical volumes.

Logical volume = solid + material + positioned subcomponents (other logical volumes)
Provides navigators, coordinate conversions, etc.

Functional hierarchy – tree of DetectorElement (DE) objects



Ties into geometrical hierarchy for access to info and utilities, but the structure can be different:

- skip layers of geometrical hierarchy
- create DE that do not correspond to logical volumes but contain other DEs that are logically connected.

New Geometry - Status & Migration Strategy

Jeremy McCormick
Tim Nelson

Compact geometry description remains the same

Current Detector and all existing Subdetector classes will inherit from DetectorElement

- Existing functions remain; new capabilities come for free
- Can use existing IDDecoder infrastructure for identification of hits with their DetectorElements
- Most, if not all, existing Subdetectors are now converted including MultiLayerTracker, DiskTracker and SiTrackerBarrel.

Interfacing to SiStripSim digitization package (in progress):

- SiStripSim will implement IDetectorElement, and produce RawTrackerHits from simulated hits.

Plan to interface to Norman's TRF toolkit.

Improved system of hit and track classes is needed to take advantage of this geometry – need to coordinate with Fermilab group.

Snapshot – Who's Doing What

(apologies to those whose work in not included)

Brown / SLAC
(Rich Partridge)
Ongoing:
• Tracking infrastructure
• seedtracker track finder

Oregon
(Nick Sinev)
Ongoing:
• Tracker resolution studies
• Pixel digitization

Santa Cruz
(Bruce Schumm, Chris Meyer, Tyler Rice, Lori Stevens)
Ongoing:
• Tracking performance studies with SODTracker and AxialBarrelTracker
• Non-prompt track reconstruction
Plans:
• Include other track finders/fitters into the study

Colorado
(Steve Wagner, Fred Blanc)
Ongoing:
• Kalman filter fitter package
• SODTracker track finder

SLAC
(Jeremy McCormick, Tim Nelson)
Ongoing:
• Silicon strip digitization
• New geometry system

SLAC
(Norman Graf)
Ongoing:
• TRF toolkit
• Tools & Infrastructure

Fermilab Software Group
(Rob Kutschke, Caroline Milstene, Hans Wenzel)
Ongoing:
• Tracking infrastructure improvements
Plans:
• Forward tracking
• Vertexing involved in b-jec/c-jet separation

Kansas State
(Dima Onoprienko, Eckhard von Toerne. Tim Bolton, others plan to get actively involved)
Ongoing:
• Calorimeter assisted tracking
• Tracking infrastructure improvements
Plans:
• Non-prompt track reconstruction and tracker design optimization studies

Get in touch with Marcel or Rich if you would like to get involved !

Summary

Significant progress in the last few months.

Tracking infrastructure issues that have been a bottleneck for quite a while are being attacked.

Much more capable geometry system is being developed.

Numerous algorithms improvements.

Tracking software tools for detector optimization are still far from perfect, but are perfectly capable of addressing at least some of the design questions – we need more people to use those tools.

We're in the middle of an active development period on many fronts – stay in touch with developers and other users. Forum is a good place to ask questions:

users : http://forum.linearcollider.org/index.php?t=threadt&frm_id=3&rid=0

developers: http://forum.linearcollider.org/index.php?t=threadt&frm_id=41&rid=3