Track Trigger Software Overview

Track Trigger Integration WG

22.10.2012

Nicola Pozzobon

Università degli Studi di Padova
INFN – Sezione di Padova

Outline

- Why simulation tools
- Status of available simulation tools (some of them)
- Pending issues with simulation tools

Check also the TT Simulation tutorial available from the 24th July 2012 Tracker DPG TT WG

What are sim tools meant for

- Evaluation of a tracker itself
- Evaluation of a tracker *within* CMS
- Evaluation of track trigger algorithms
- Comparison of full simulation with predictions from other sources
- Evaluation of data-flow constraints
- Feedback to/from HW design

What are the available tools?

- tkLayout standalone tool for basic performance estimate with parametric models
- Verilog simulations of track trigger architecture
- CMS FullSim with custom tracker geometry for everything else

Why a G4 simulation in CMSSW?

Despite being slow and non-optimal when rough and fast answers are needed

- It can satisfy all described needs
- It can validate results from independent modeling and evaluation
- It allows the simulation and evaluation of nasty effects (secondaries, loopers, etc …)
- It allows the combination of tracker *and* other subsystems

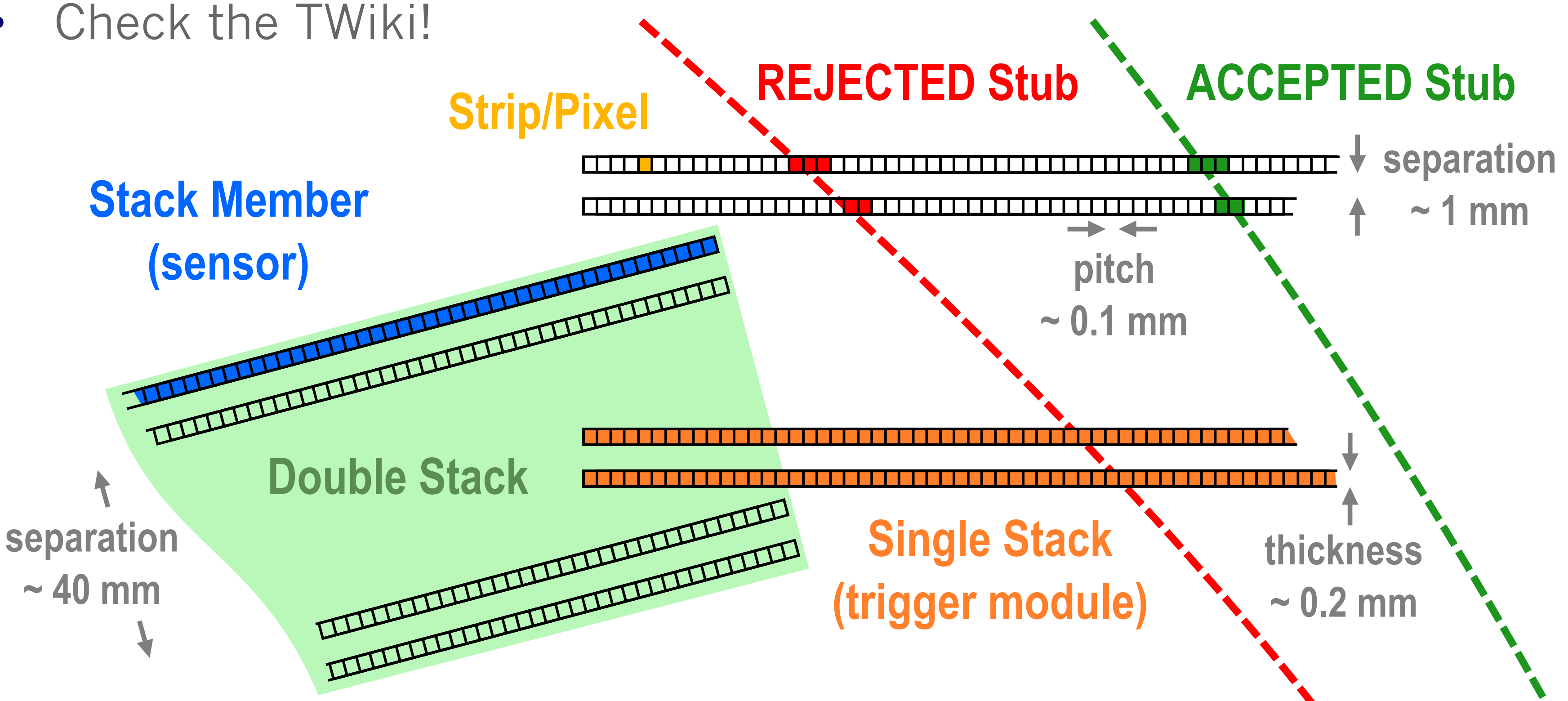
CMSSW Track Trigger: where?

https://twiki.cern.ch/twiki/bin/viewauth/CMS/SLHCTrackerTriggerSWTools

- SLHCUpgradeSimulations/Geometry
- SLHCUpgradeSimulations/Utilities
- SLHCUpgradeSimulations/L1TrackTrigger
- SimDataFormats/SLHC

Avoid misuse of vocabulary!

- Both sensors in a Stack belong to the same Layer
- Stubs in consecutive Layers can be paired into Tracklets
- One cannot name a Stub or a Cluster or any other L1 TT object without specifying **HOW** it is built
- Check the TWiki!



CMSSW Track Trigger modules

SLHCUpgradeSimulations/Geometry:

- Geometry files and basic plugins for topology handling

SLHCUpgradeSimulations/Utilities

- Tools to call paired sensors “stacked modules”

SLHCUpgradeSimulations/L1TrackTrigger

- Algorithms to build Clusters, Stubs, Tracklets, Tracks

SimDataFormats/SLHC

- Data formats describing Clusters, Stubs, Tracklets, Tracks → this is where the “vocabulary” comes from!

Other (TT-modified) modules

The general idea is not to mess up the existing modules:

- Minimize the work, use what is already available
- Do not affect what is currently used in CMSSW for validation, data-taking, data analysis, etc ...

DataFormats/SiPixelDetId

- Re-assignment of empty bits in PXB and PXF DetId

Geometry/TrackerNumberingBuilder

- Must follow new features of DataFormats/SiPixelDetId

Geometry/TrackerCommonData

- New positioning algorithms to use tkLayout XML

Tracker geometry

- It includes Phase 1 Pixels as in CMSSW_4_2_8_SLHCtk3
- TT was born around barrel-like pixelated layers of trigger modules complementing a standard strip tracker
- The LongBarrel concept layout came then together with a hierarchic L1 track finding idea being developed at FNAL
- Frozen: from 2_2_6 to 4_2_8_SLHCTk3
- This was a limit for TT

Geometry-induced limitations

- SLHCUpgradeSimulations/Utilities could only call pairs of PXB sensors “stacks” → FIXED
- SLHCUpgradeSimulations/L1TrackTrigger could only handle signals from PXB-PXB “stacks” → FIXED
- Ambiguous sorting of parent/child structures, with non predictable indexing of modules → FIXED
- Difficult to address inner/outer sensor → FIXED (tricky)
- Geometry XML files could not be easily configurable
- There was a request for Strip modules, mixed modules, Endcap modules
- TT software could not fit those requests

A new geometry is difficult to get

- TT needs two different containers to describe the tracker: TrackerGeometry and StackedTrackerGeometry
- StackedTrackerGeometry contains objects which are safely used and *only pointed* by TrackerGeometry
- Difficult part: build TrackerGeometry from XML
- tkLayout was designed to export the geometry into files that cope with CMSSW requirements
- XML format, 7 files for definition of volumes, module positioning, material assignment and sensor topology

tkLayout and CMS FullSim

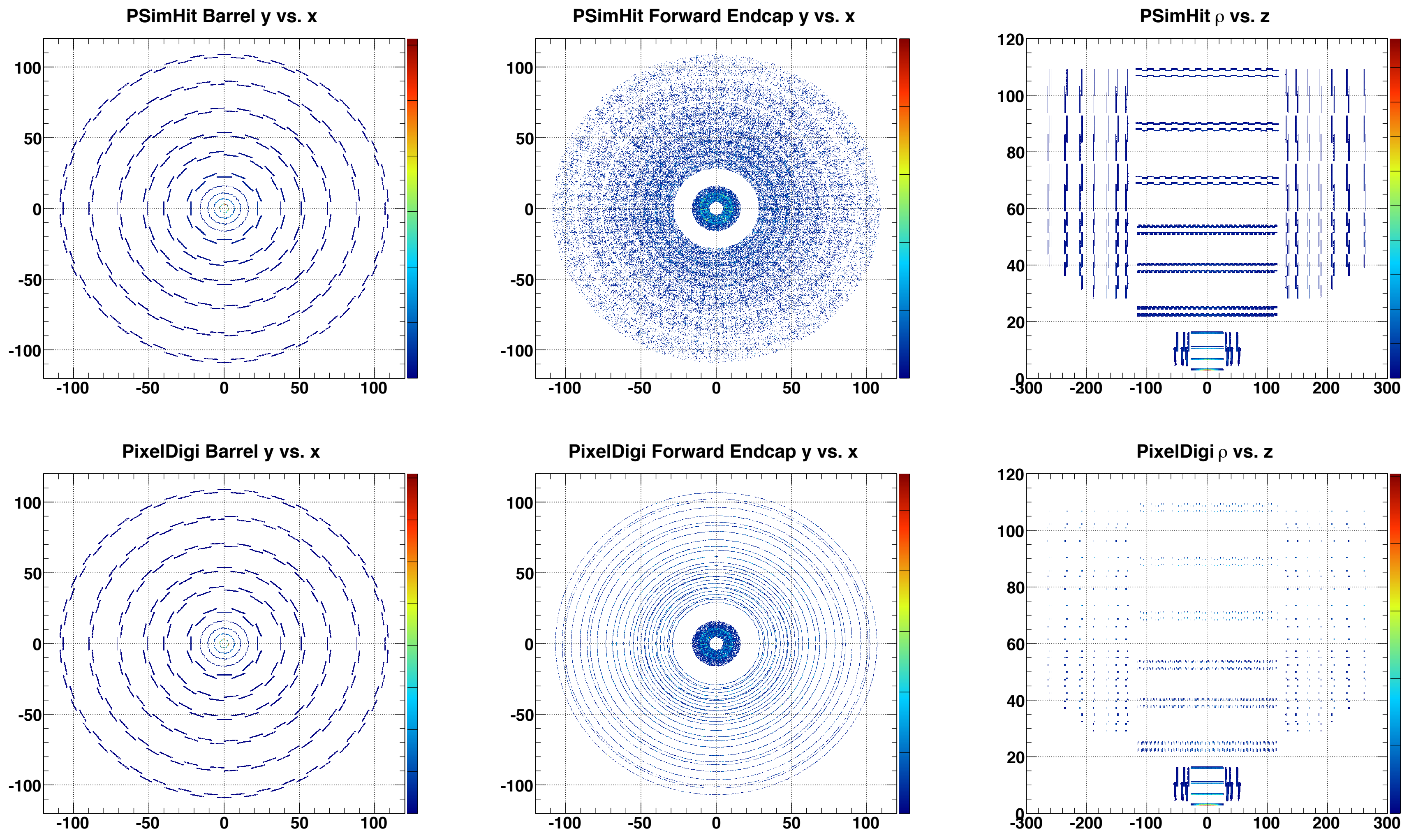
- tkLayout tool XML exportation is based on templates
- Not touched since long, long time ago … stuck to old constraints and templates
- Enormous manual editing of output XML files was needed
- There was a general requirement of flexible TT tools and of testing any layout, so debugging started
- Also the management supported the fact that new geometry files should be provided by tkLayout
- Generic geometry → flexible TT tools

tkLayout debugging

- Volume boundaries
- Module boundaries
- Sensor spacing
- Module and sensor orientation
- Module 2D and 3D overlap
- Keywords for DetId assignment
- Custom version of few CMSSW classes out of TT specific modules
- Particularly difficult for Endcaps

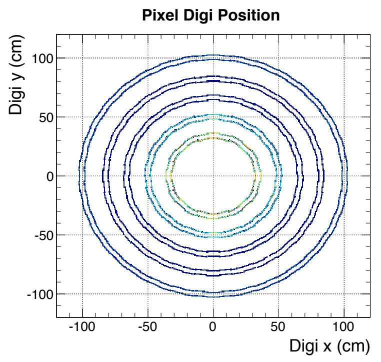
Current status of tkLayout/CMSSW

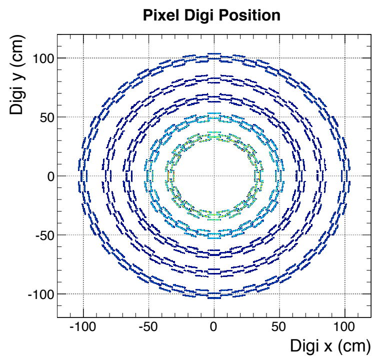
- (Almost) completed, few more bugs to be fixed

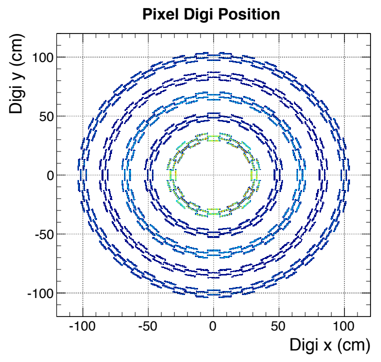


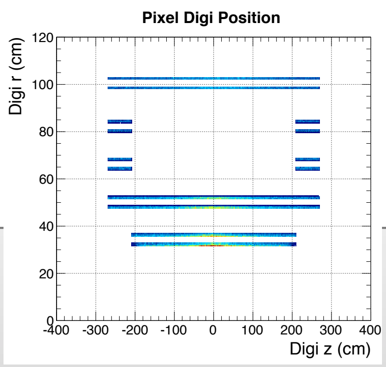
A new LongBarrel from tkLayout

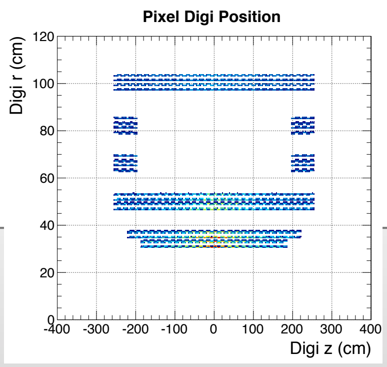
- All the "FIXED" in the limitations list were born in trying to make a new tkLayout-based LongBarrel work, test version released in 4_2_8_SLHCtk3 and available from CVS
- Basic guideline: keep changes to CMSSW modules an classes to the least necessary amount

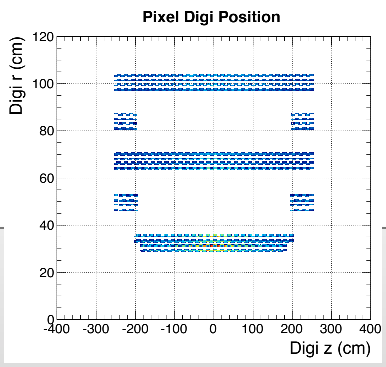












LongBarrel from tkLayout features

- Old LongBarrel: 0.098 mm x 1 mm pixels, 1 mm sensor separation, 0.100 mm thick sensors, main stack layers at 32-36 cm, 48-52 cm and 98.5-102.5 cm
- **New LongBarrel: 0.094 mm x 1.4 mm pixels, 1.2 mm sensor separation, 0.100 mm thick sensors, main stack layers at 32-36 cm, 48-52 cm and 98.4-102.4 cm (tolerance: few tens of mm)**
- We have also **LongBarrelSwapped**, with second main double stack at 64.2-68.2 cm

Geometry

SLHCUpgradeSimulations/Geometry/data/LongBarrel/

- 6 XML files needed to describe the geometry (7, if different pixfwd.xml is needed)
- the PixelSkimmedGeometry.txt file containing a table of DetId, Rows, Columns, ROCs

SLHCUpgradeSimulations/Geometry/test/

- dumpGeom_cfg.py, which is used to create the Fireworks geometry file
- writeFile_*_cfg.py, which writes the PixelSkimmedGeometry.txt file

Geometry

SLHCUpgradeSimulations/Geometry/python/

- Digi_*_cff.py, needed by the Digitizer
- LongBarrelSwapped_cmsSimIdealGeometryXML_cff.py LongBarrelSwapped_cmsSimIdealGeometryXML_cfi.py, to load the Geometry, including all subsystems

SLHCUpgradeSimulations/L1TrackTrigger/test/

- PrintStackInfo_*_cfg.py to print a summary of the tracker part made of trigger modules

Data formats and Producers

Class name: **L1Tk**Stub *(Example with Stubs)*

Class implementation:

- SimDataFormats/SLHC/interface/L1TkStub.h
- SimDataFormats/SLHC/src/L1TkStub.cc
- SimDataFormats/SLHC/src/classes.h and classes_def.xml

NOTE: *latest CVS tag is common with L1DTTrigger and L1CaloTrigger data formats, in order to allow using L1TrackTrigger at the same time*

Data formats and Producers

The Builder is an is an EDProducer which defines an input/output scheme and the interface to data formats, regardless of the particular algorithm

- SLHCUpgradeSimulations/L1TrackTrigger/interface/L1TkStubBuilder.h

The Builder loads a specific algorithm which is also an EDProducer, based on a specific “reference” class

- SLHCUpgradeSimulations/L1TrackTrigger/interface/ and src/HitMatchingAlgorithm.h and HitMatchingAlgorithmRecord.h,
- SLHCUpgradeSimulations/L1TrackTrigger/src/HitMatchingAlgorithm.cc and ES_HitMatchingAlgorithm.cc

Data formats and Producers

Some Algorithms are already available, but custom ones can be used, provided they respect all the constraints

- SLHCUpgradeSimulations/L1TrackingTrigger/interface/ HitMatchingAlgorithm_globalgeometry.h and many others

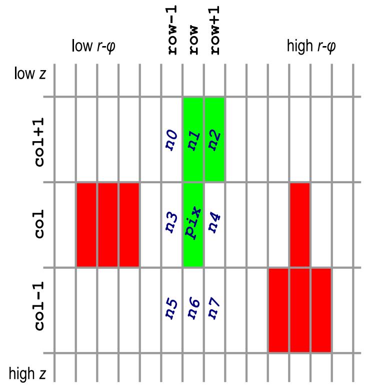
Specific parameters are contained in dedicated cfi files and set via edm::InputTag and ESPrefer

- SLHCUpgradeSimulations/L1TrackingTrigger/python/ Stub_cfi.py
- SLHCUpgradeSimulations/L1TrackingTrigger/python/ HitMatchingAlgorithmRegister_cfi.py

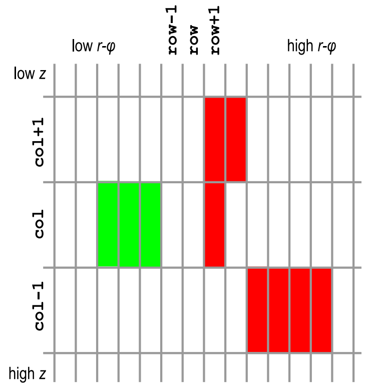
If you want your algorithms to be included in the official CMSSW packages, please contact me

DN-2012-003

- Algorithms are described in detail therein
- Reference Cluster algorithm: “2d”
  - Clusters within sensor, across ROCs
  - Max size is 2x2
  - *Duplicates and fakes still to be understood: major performance problem by now*



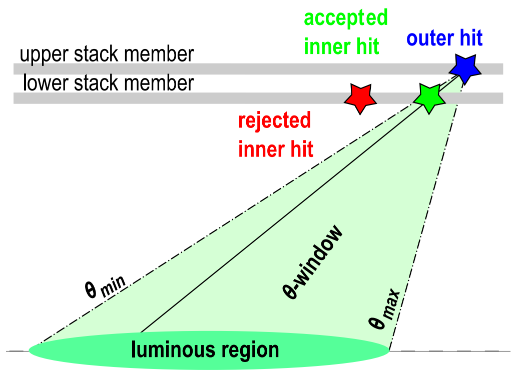
2d

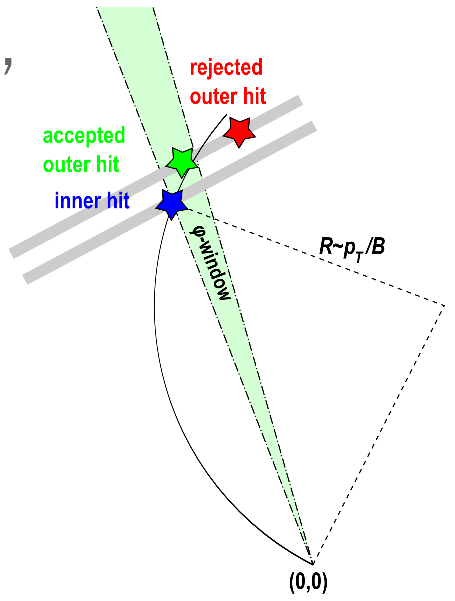


broadside

DN-2012-003

- Reference Stub algorithm: “pixelray”
  - Stubs within stacked module
  - Variation to “globalgeometry” one *(pictures)*, using only global coordinates and trigonometry
  - $\Delta\varphi = \Delta R \times cB \times 0.5E{-}9/p_T$ [cm, s, GeV/c, T]
  - I am working also on another algorithm, making use of local coordinates
  - News ASAP





DN-2012-003

- Only one Tracklet algorithm: “globalgeometry”
  - Tracklets: within double stack
  - same approach as Clusters → Stubs, only global coordinates, Stubs are matched to each other instead of Clusters
  - $p_T = cB \times 0.5E-9 \times ((R^2+r^2-2Rr \times \cos(\Delta\varphi)))^{1/2} / \sin(\Delta\varphi)$ [cm, s, GeV/c, T]
  - $p_Z = p_T \times (Z - z) / \Delta R \rightarrow$ hence η
  - $z_{VTX} = Z - R \times (Z - z) / \Delta R \rightarrow$ LINEAR APPROXIMATION OF TRAJECTORY!

Tracking at L1

- One ideal approach in CVS code
- One recent tracking algorithm by Anders and Emmanuele, available as a standalone tool running on ASCII files with Stub information from FullSim
- Currently being interfaced to L1TrackTrigger module and data formats to be embedded in the FullSim workflow

Main pending problems

- Duplicate and fake Clusters
  - 3 pixel width was more likely from old simulations
- Duplicate and fake Stubs
  - With pixelated sensors, forward “broadside” clusters
- Stub rates with realistic MB
  - Old LongBarrel had very simplified services
- Threshold tuning
- Simplified local coordinate-based algorithm (Nicola P.)
- Complete embedding of Cornell L1 Track finding in CMSSW (Emmanuele S.)
- Have a single release for Phase 1 (Pixel + HCAL), where to plug Phase 2 tracker (Eric B.)
- Have a single tag for all L1*Trigger data formats
  - L1Tk, L1DT, L1Calo ok! *L1CSC missing…*

Concluding remarks

- Stay tuned: HN and TWiki
  - First validation plots of "local" stub algorithm to be released soon
  - These weeks, many updates are being committed to CVS, so there might be a bit of consecutive alerts sent to HN
- Keep track of what is happening in TT WG (Tk DPG)