Status of Full Simulator for JLC Detectors Based on Geant4

2002/7/12 ACFA LC Workshop
K. Hoshina
representing ACFA-Sim group

Purpose of Full Simulator

1) To check and finalize the present detector design, taking into account requirements from
   - Physics
   - Beam-related background
   - etc.
2) To tune quick simulator for physics study

Why are we developing a new simulator?

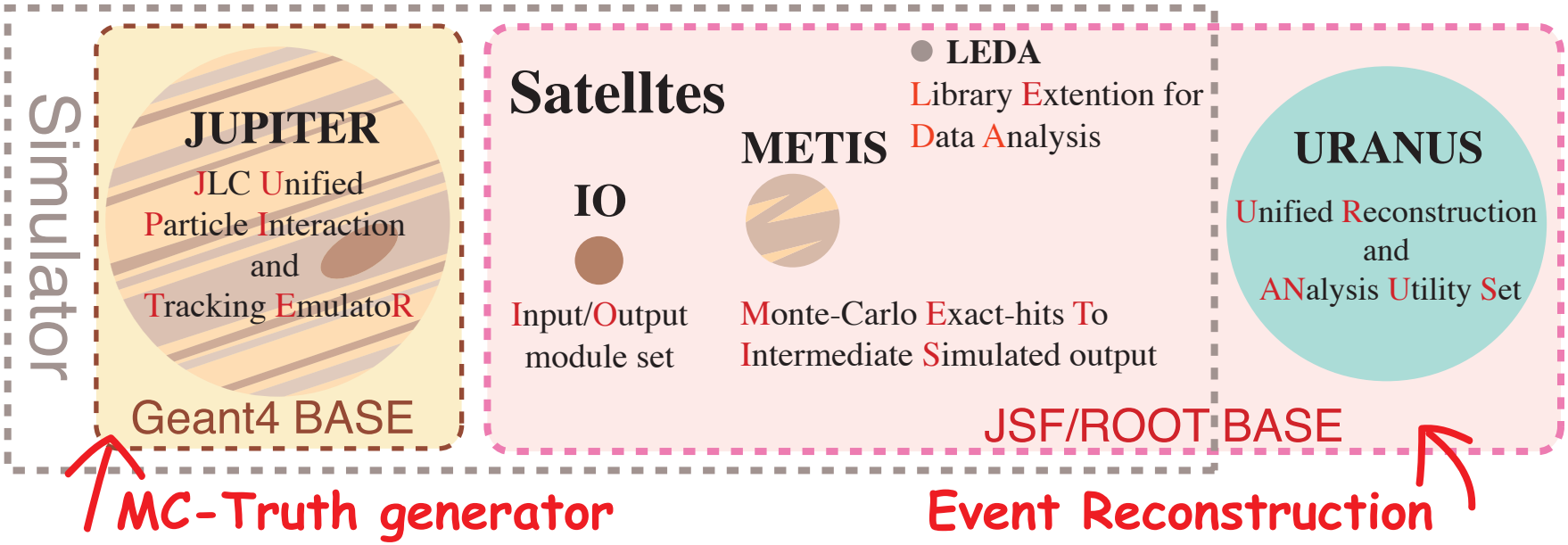
We have a full simulator named JIM/JLCSIM, which is based on a FORTRAN program, Geant3.

However, to keep up with large scale data analysis and simulation at JLC, it is desireble to use

- Object Oriented technology
- Framework which can deal with Terabytes raw data per event

→ Develop all programs with C++ and ROOT based analysis-packages !!

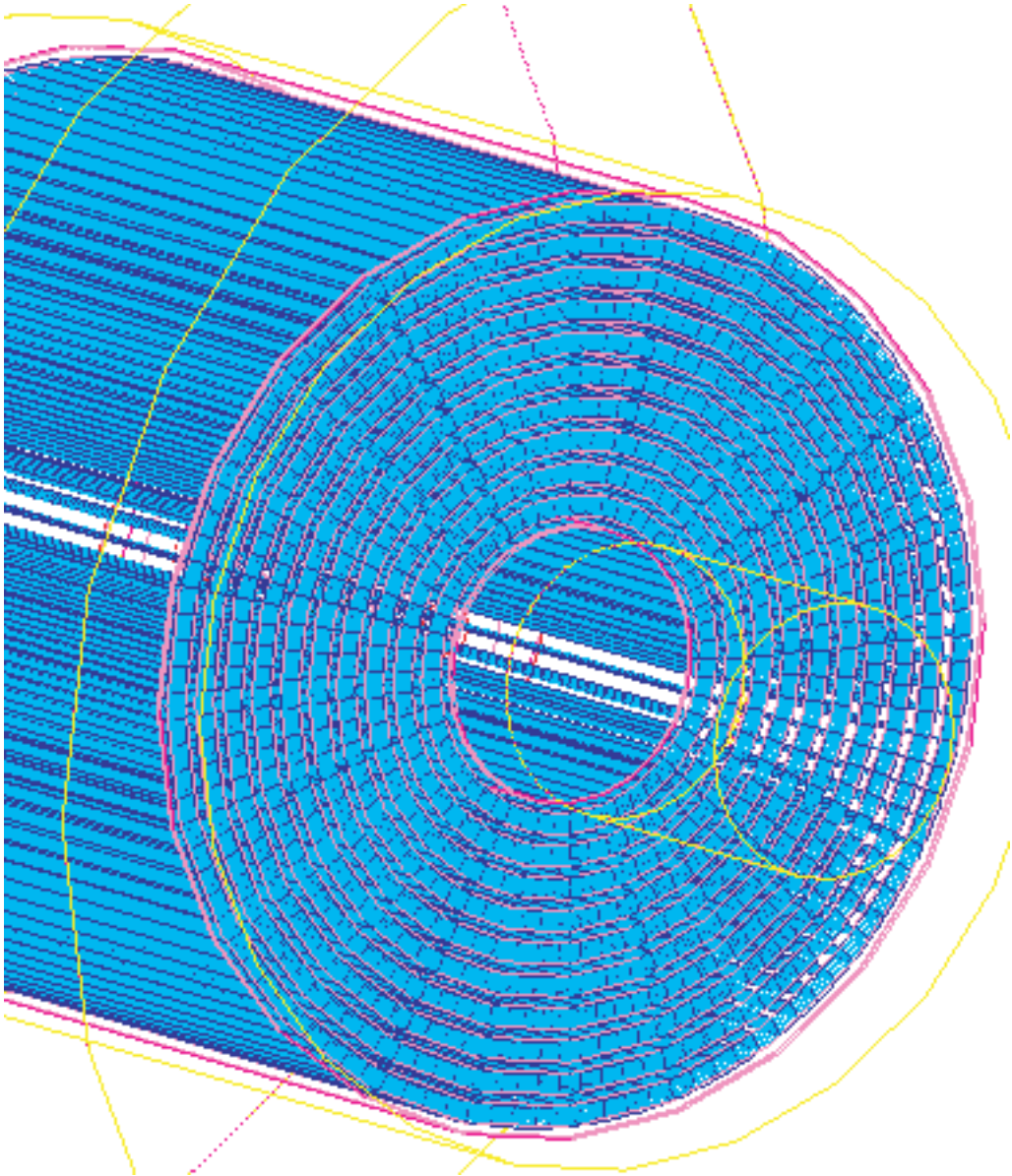
We divided the new simulator into following parts...



Requirements

1) Modularized components
   - To enable simultaneous development by sub-groups
   - To facilitate easy modifications of detector configurations
   - To help implement detailed structure with small effort
2) Framework
   - To unify class design in order to minimize user-written source code
   - To unify class interfaces

JUPITER =
JLC Unified Particle Interaction and Tracking EmulatoR



Role
Generates Monte-Carlo Truth

Features
Easy update
Easy Install/Uninstall
Powerful Base Classes (Framework)

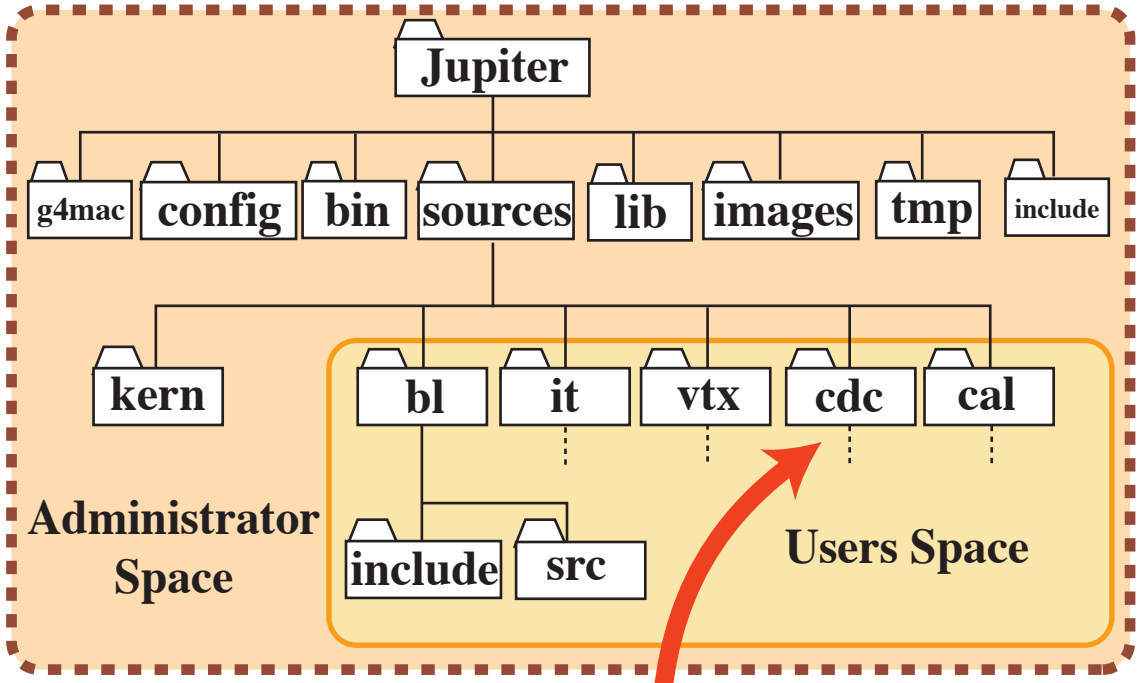
Present status
CAL, CDC, Intermediate Tracker, VTX, Interaction Region, Beam Delivery Region

Future project
Interface to XML or CAD

Each sub-group is assigned its own directory...

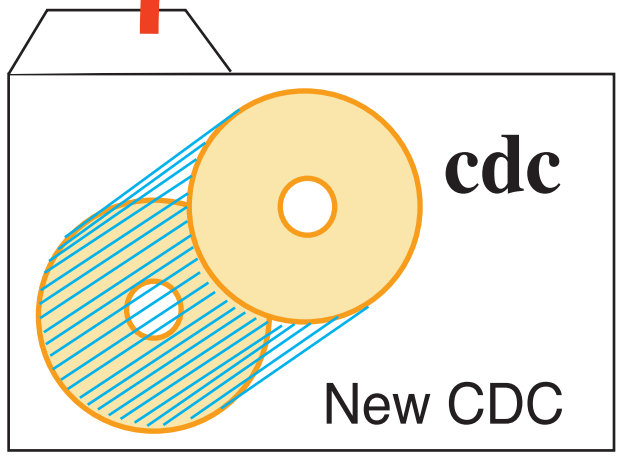
Development concerning the sub-detector component is confined to that directory

Installation of a new detector or updating of a existing detector can be done by just putting or replacing the corresponding directory without any influence on other sub-groups



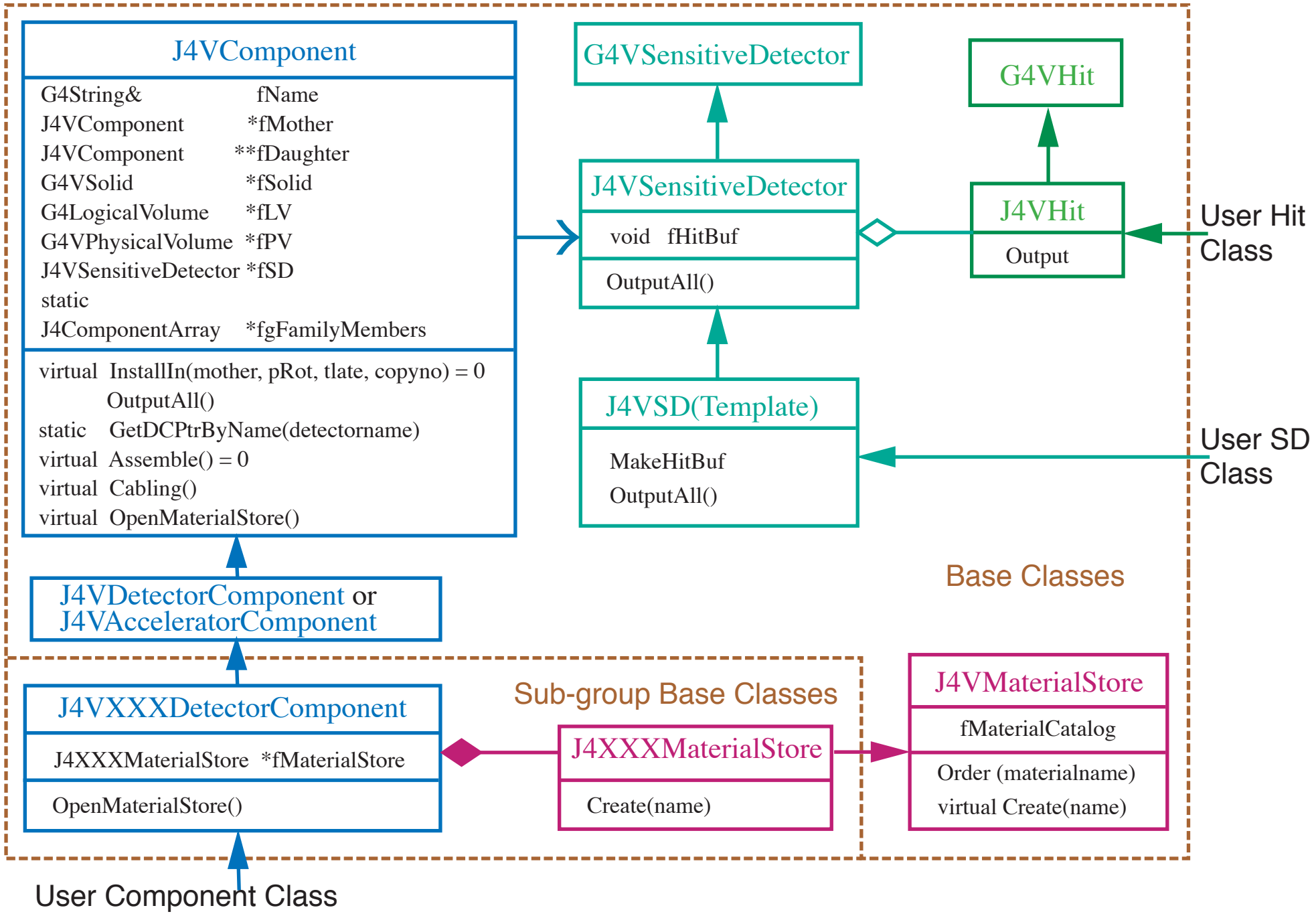
Easy Update!

Replace your dirctory, then update will finish immediately !



JUPITER : Base classes

ACFA-Sim



J4VDetectorComponent
(abstract class)

Geometry
Material
Position*

Name
Assemble()
InstallIn()

PtrArray to Daughter
DetectorComponent

Ptr to Mother
DetctorComponent

Derived from

J4VCDCDetectorComponent
(abstract class)

Geometry
Material
Position*

InstallIn()
Switch On or Off

members , functions and
Material Store
for sub-group

Derived from

J4CDCDriftRegion
Geometry
Material
Position* = null

InstallIn(J4CDCCell* cell);

J4CDCCell : public
J4VCDCDetectorComponent

J4CDCDriftRegion
Geometry
Material
Position*
= positionptr

Position

Geometry
Material
Position* = null

InstallIn(J4CDCLayer* layer)

J4CDCLayer : public
J4VCDCDetectorComponent

J4CDCCell

Example

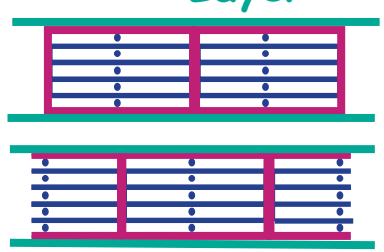
Drift Region

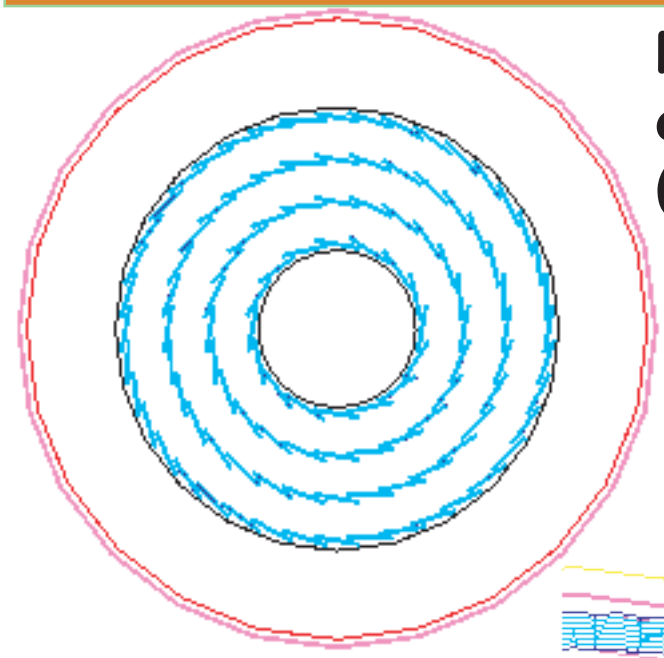
InstallIn(....)

Drift Cell

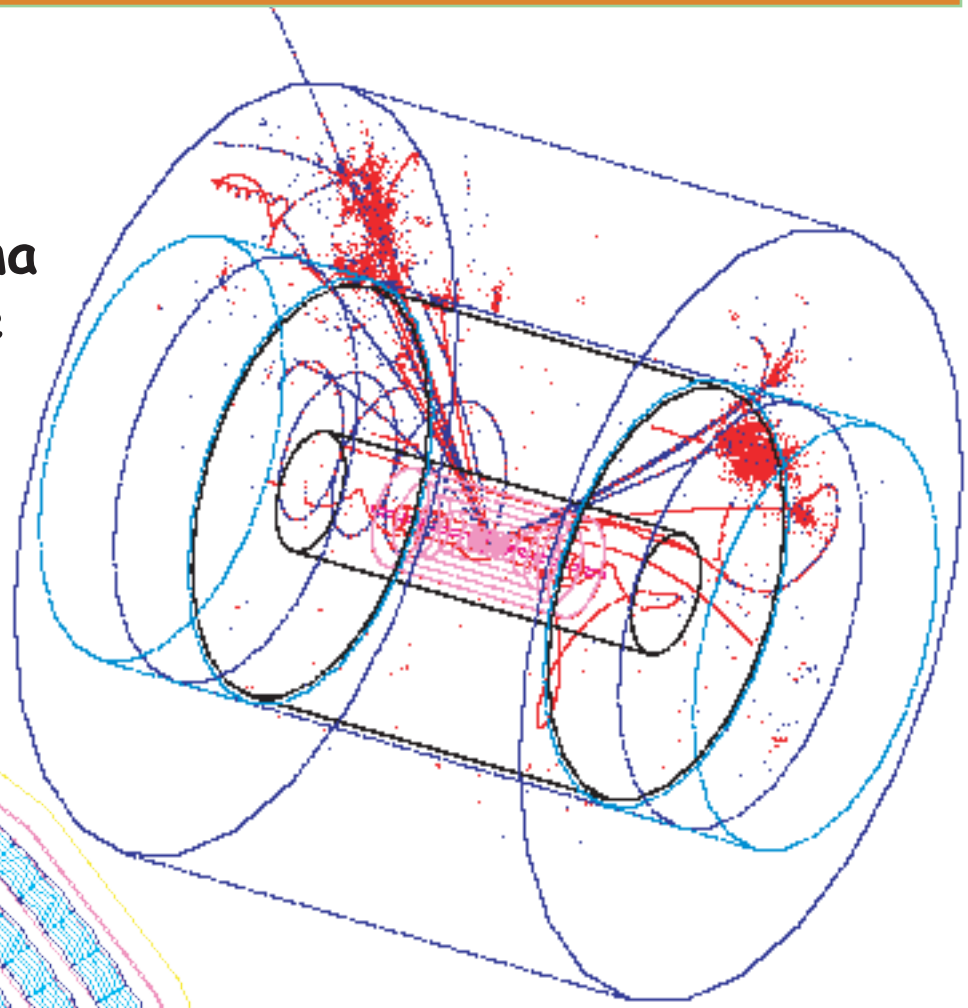
InstallIn(....)

Layer

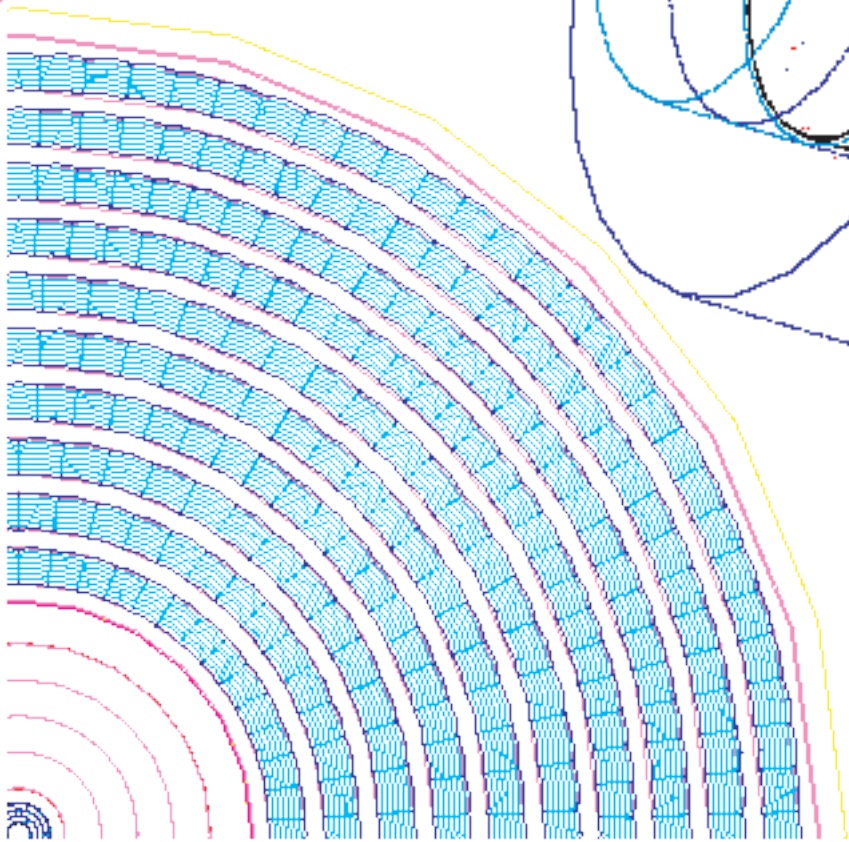




R-phi section
of VTX
(installed by
Aso-lab@Toyama
National College
of Maritime
Technology)



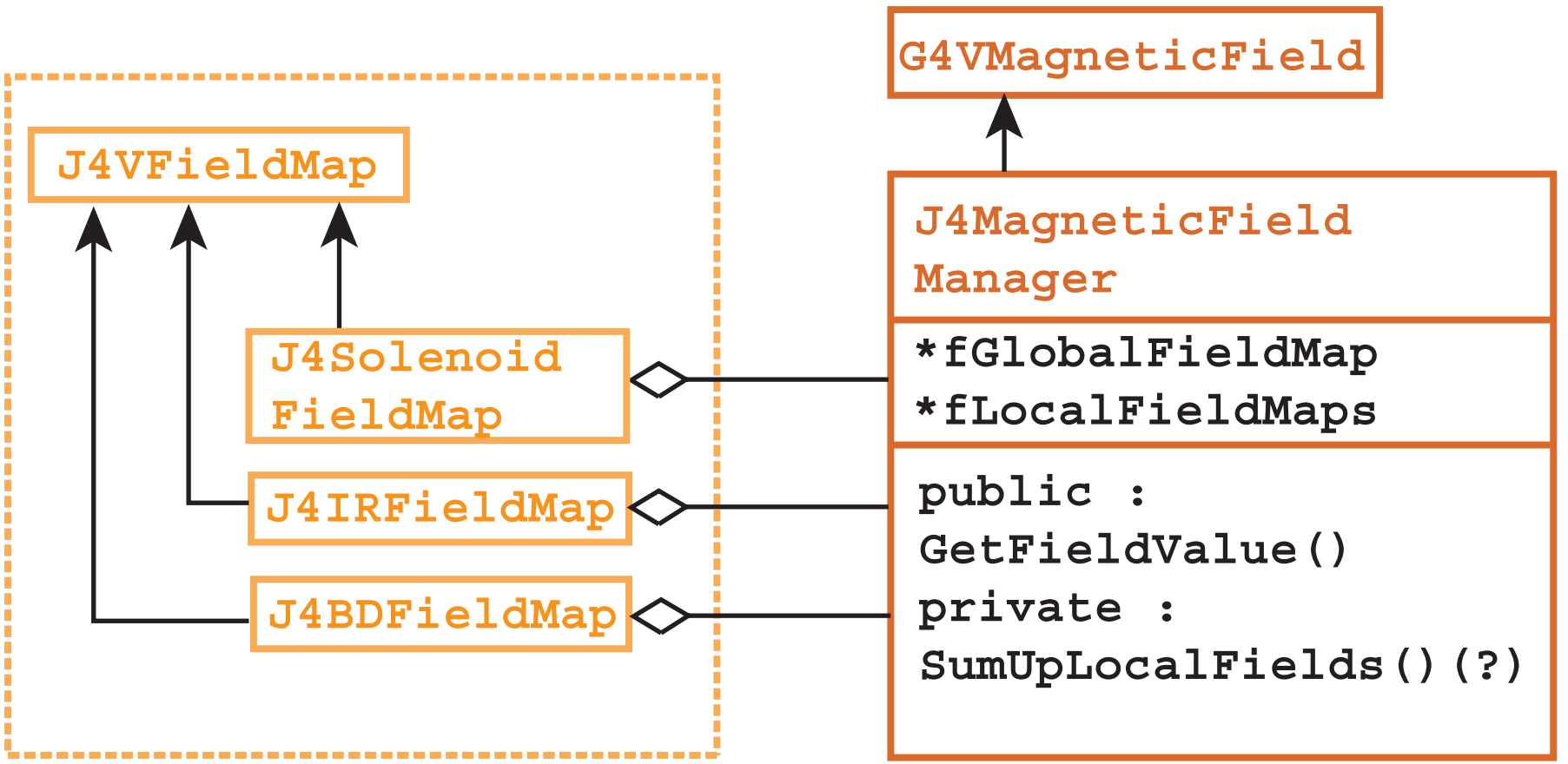
CDC:
Layer 10
Cell 36~108
□ □ /Layer
Wire 5/Layer
□ □ r=15um



Event display of
e+e- -> Z^0H

$\sqrt{s}$ = 350GeV

For beam-beam background study, JUPITER is equiped with Accelerator and Magnetic components.



Components with B-field attribute have field map in local coodinate system (local field map)

Sum up local field maps and prepare a global Field map at the beginning of program execution

Satellites are event-reconstruction simulation modules...

Input/Output
module set

IO - Input / Output module set

Role

Converts JUPITER's output to ROOT format

Feature

Gives flexible interface to JUPITER output format

Future project

Support I/O of interim results from METIS and URANUS

Monte-Carlo Exact-hits
To
Intermediate Simulated
output

METIS - Monte-Carlo Exact-hit To Intermediate Simulated output

Role

Provides module set for simulating event reconstruction

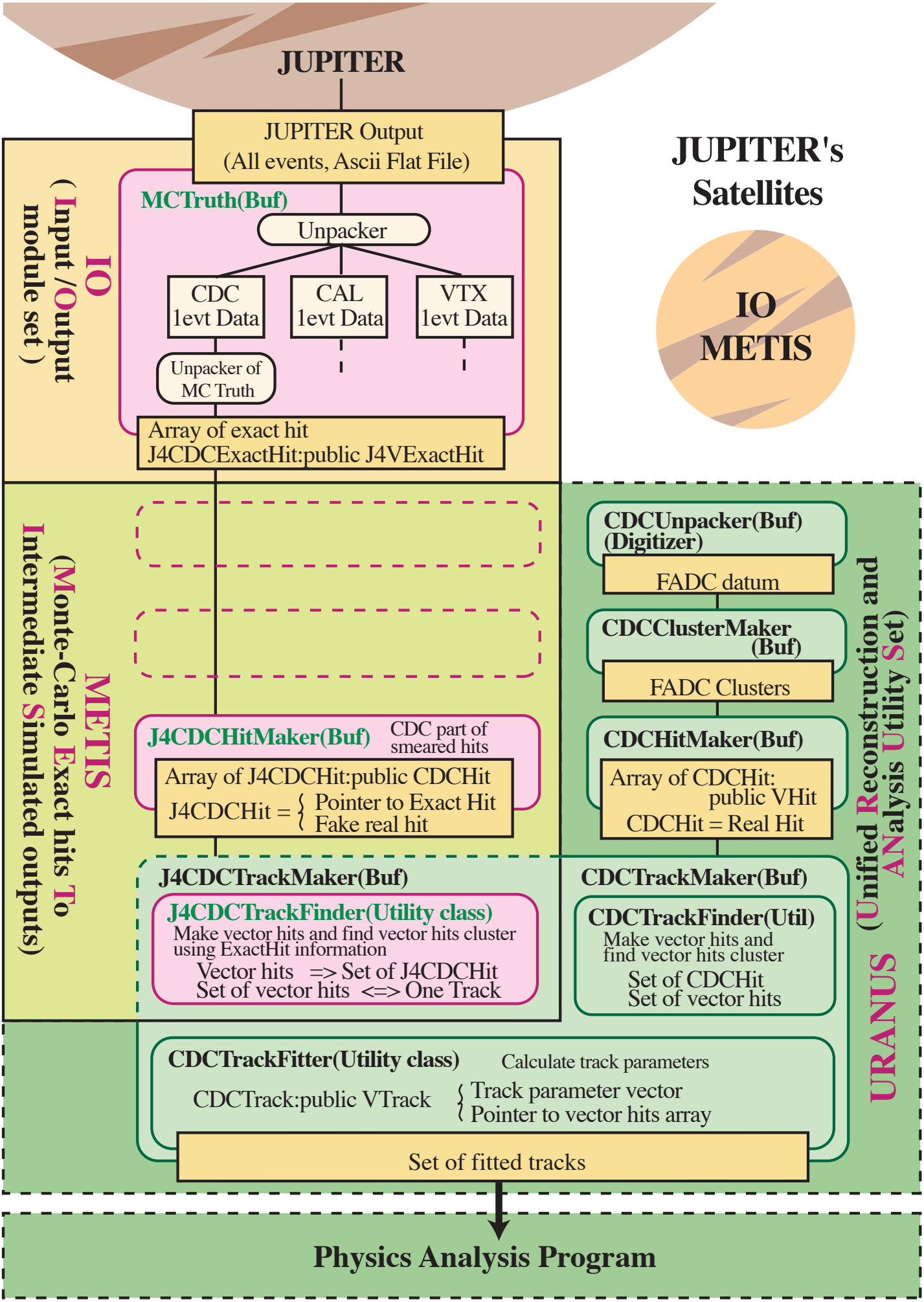
Features

Each module inherits from the classes of URANUS (HitMaker, TrackFinder, etc.)

Easy switching of simulation levels

Future project

GUI interface for simulation level switching


JUPITER
JUPITER Output
(All events, Ascii Flat File)
JUPITER's
Satellites
IO
METIS
IO
(Input /Output
module set)
MCTruth(Buf)
Unpacker
CDC
1evt Data
CAL
1evt Data
VTX
1evt Data
Unpacker of
MC Truth
Array of exact hit
J4CDCExactHit:public J4VExactHit
METIS
(Monte-Carlo Exact hits To
Intermediate Simulated outputs)
CDCUnpacker(Buf)
(Digitizer)
FADC datum
CDCClusterMaker
(Buf)
FADC Clusters
J4CDCHitMaker(Buf)
CDC part of
smeared hits
Array of J4CDCHit:public CDCHit
J4CDCHit = Pointer to Exact Hit
Fake real hit
CDCHitMaker(Buf)
Array of CDCHit:
public VHit
CDCHit = Real Hit
URANUS
(Unified Reconstruction and
ANalysis Utility Set)
J4CDCTrackMaker(Buf)
J4CDCTrackFinder(Utility class)
Make vector hits and find vector hits cluster
using ExactHit information
Vector hits => Set of J4CDCHit
Set of vector hits <=> One Track
CDCTrackMaker(Buf)
CDCTrackFinder(Util)
Make vector hits and
find vector hits cluster
Set of CDCHit
Set of vector hits
CDCTrackFitter(Utility class)
Calculate track parameters
CDCTrack:public VTrack
Track parameter vector
Pointer to vector hits array
Set of fitted tracks
Physics Analysis Program

2-Track Separation

ACFA-Sim

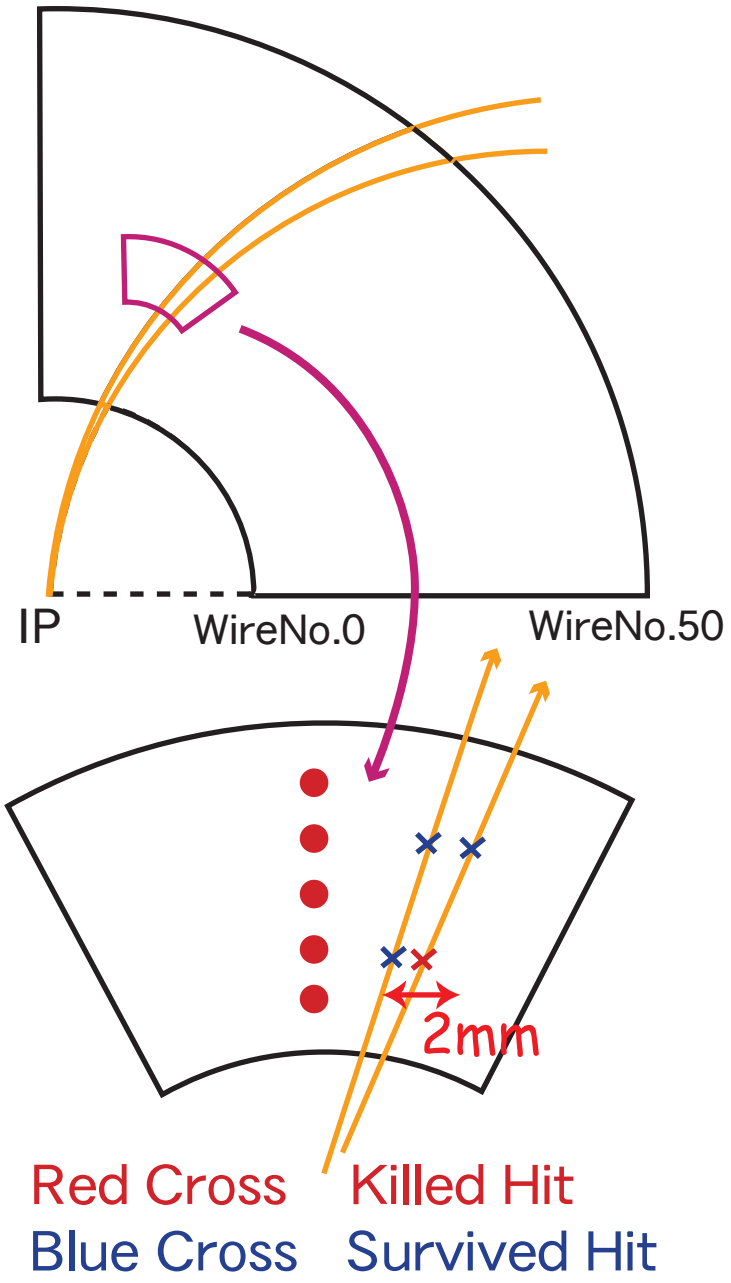
Using JUPITER and Satellites,
let's study momentum resolution at
B=3T when 2-Track Separation = 2mm

As the most pessimistic case:

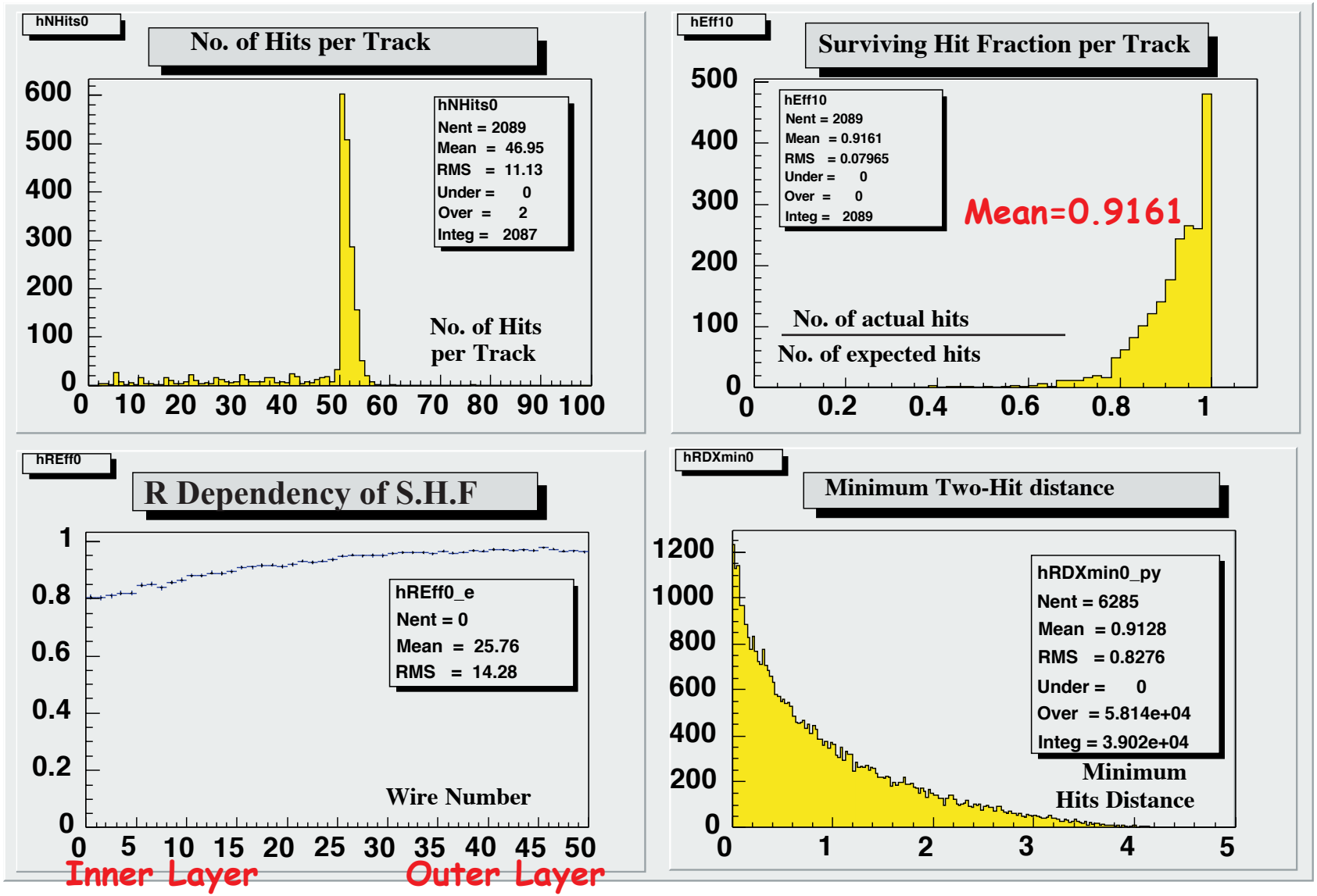
1) Delete any hits that comes after a previous hit when it is within 2mm from the first one
2) Fit surviving hits and calculate momentum

Check Points

1) 2-hit Separation dependence of momentum resolution
2) Effect of IP constraint on momentum resolution



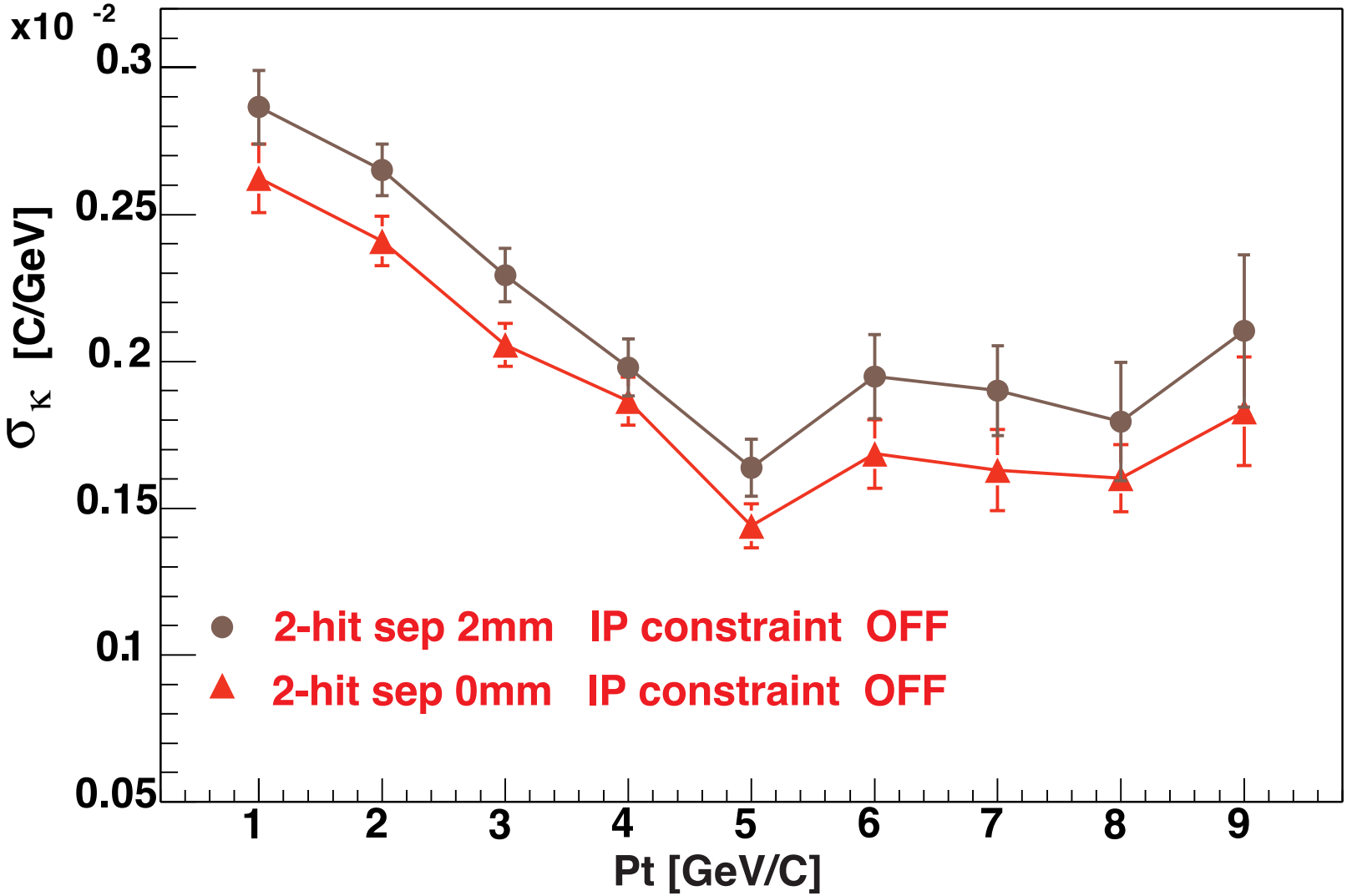
Surviving Hit Fraction (ZH @350GeV, pt > 1GeV) ACFA-Sim



More than 90% of hits will survive, however, we will lose more hits in inner layers thereby losing lever arm

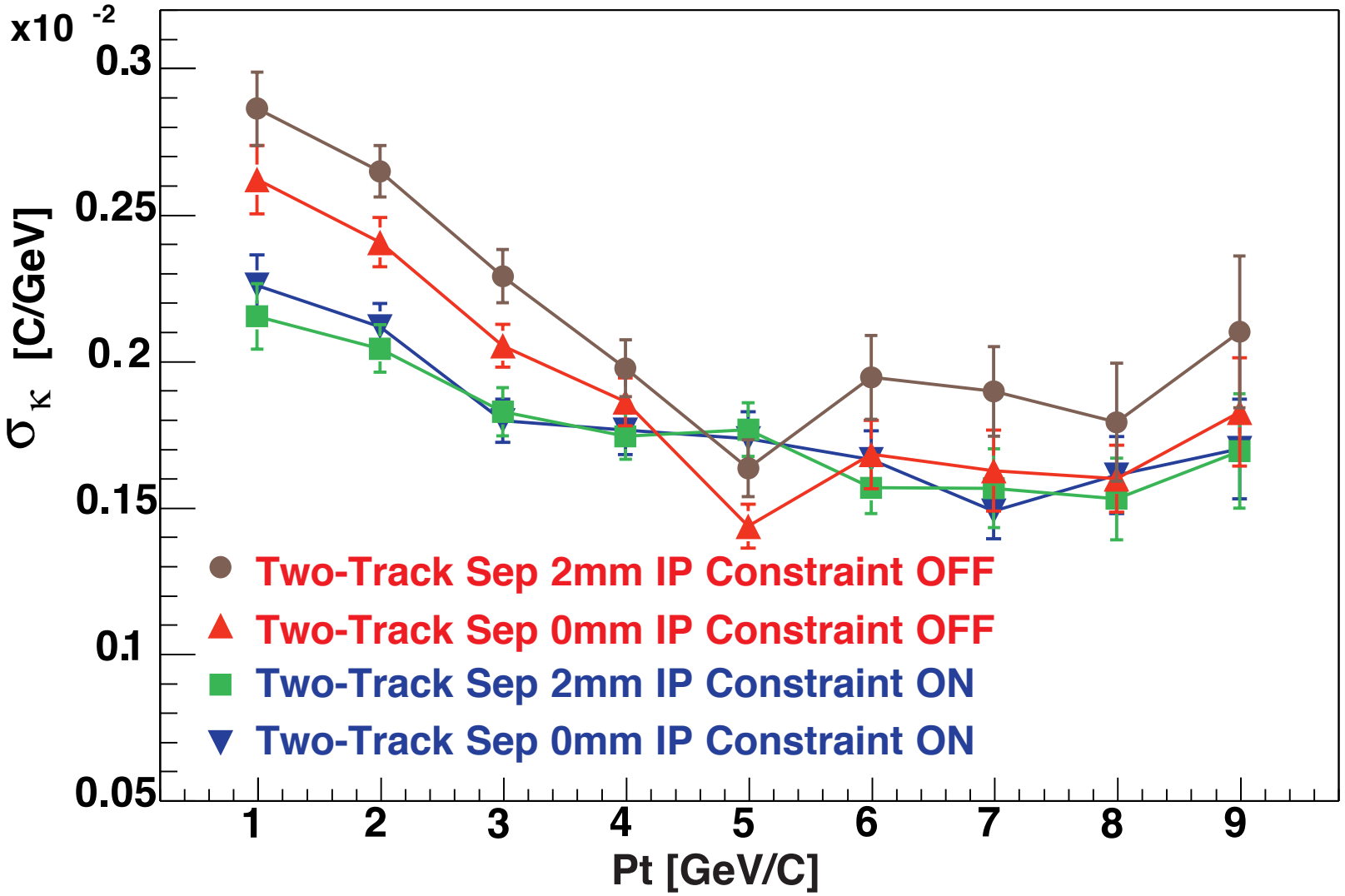
compare the following two cases:

- 2-hit separation = 2mm
- 2-hit separation = 0mm(no saturation effect)



Try to recover momentum resolution by adding IP constraint

- IP constraint ≅ VTX+CDC combined



1) A basic framework for detector simulation and analysis (JUPITER, its Satellites, and URANUS) based on GEANT4 and ROOT/JSF has been developed and is now being used to develop a full JLC detector simulator.

2) Using this simulator, we checked surviving hit fraction for the CDC. Under the condition of 2-hit separation of 2mm, we lose 10% of hit points on average and 20% at the innermost layer, implying loss of lever arm length and consequently deterioration of momentum resolution. However, the momentum resolution can be recovered by applying the IP constraint, which has, empirically, the same effect as including information from inner trackers.

3) Now JUPITER has precise structures of VTX and the CDC built in, however, some detector components are still missing.
Recently, beamline implementation has also begun to simulate beam-related background.

Come and Join Us !

Special thanks to ASO-san, ACFA-Sim members and workshop organizers