RecPack

a general reconstruction toolkit

CHEP06
Mumbai, 15/02/06

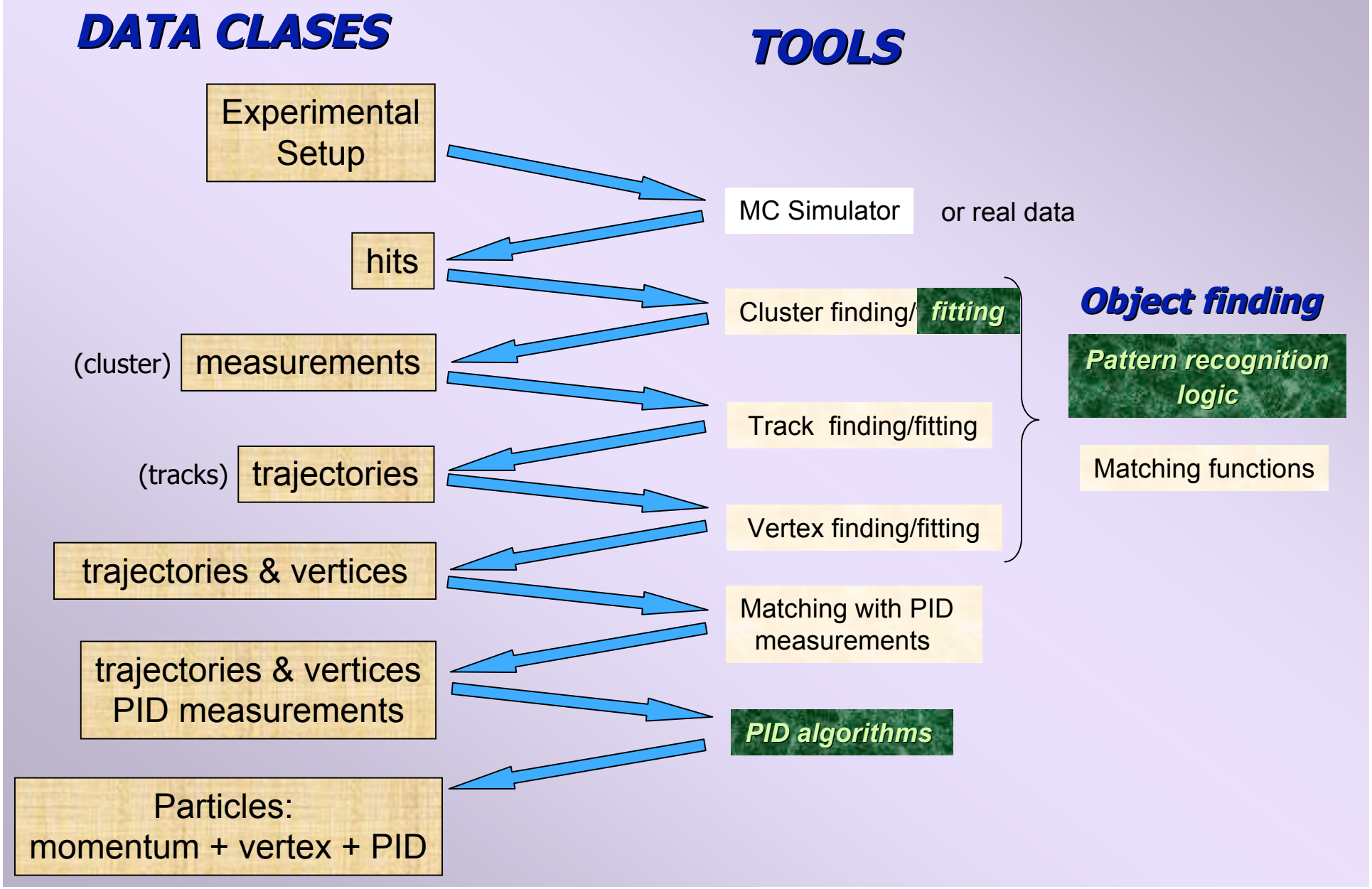
Anselmo Cervera Villanueva
Université de Genève

with:
Juan José Gómez Cadenas
José Ángel Hernando Morata

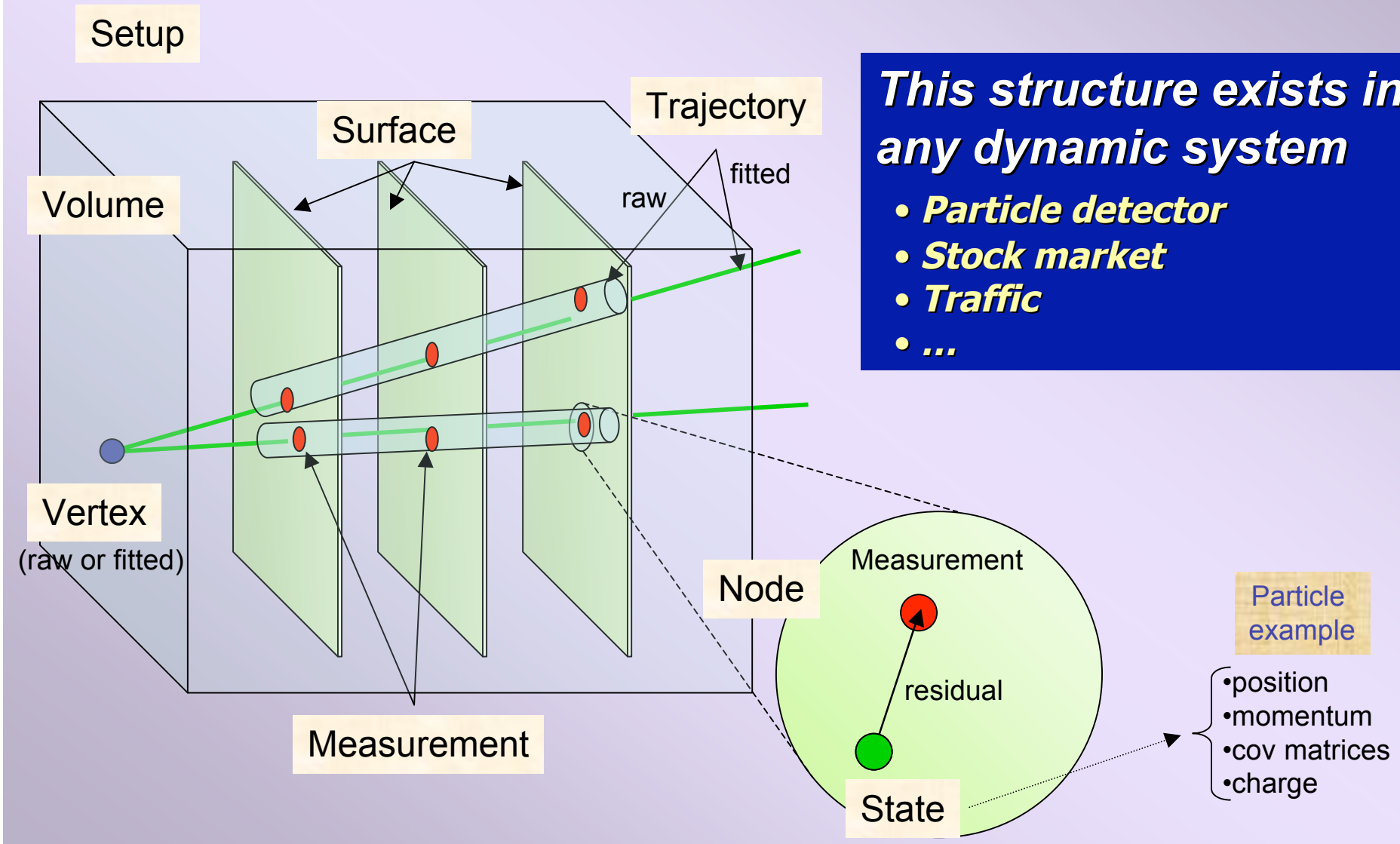
Why General Reconstruction Packages ?

- In general, every HEP experiment implements its own reconstruction software
- This was the case in the past for:
  - Simulation software
  - Analysis tools
  - Data bases
  - …
- But we have learned that this is not optimal
- Instead code reuse and general packages are preferred solutions

Reconstruction in HEP



RecPack data classes



This structure exists in any dynamic system

- ***Particle detector***
- ***Stock market***
- ***Traffic***
- ***...***

Measurement and State

class ***HyperVector :***

vector() *HepVector*

covariance() *HepMatrix*

CLHEP

Only external dependency

class ***Measurement :***

type()

HyperVector()

surface() *optional*

volume()

class ***State :***

type()

HyperVector()

type(key)

HyperVector(key)

class ***Node :***

status()

measurement()

state()

residual()

quality(key)

surface()

class ***Trajectory :***

status()

nodes()

quality(key)

end_vertex()

ini_vertex()

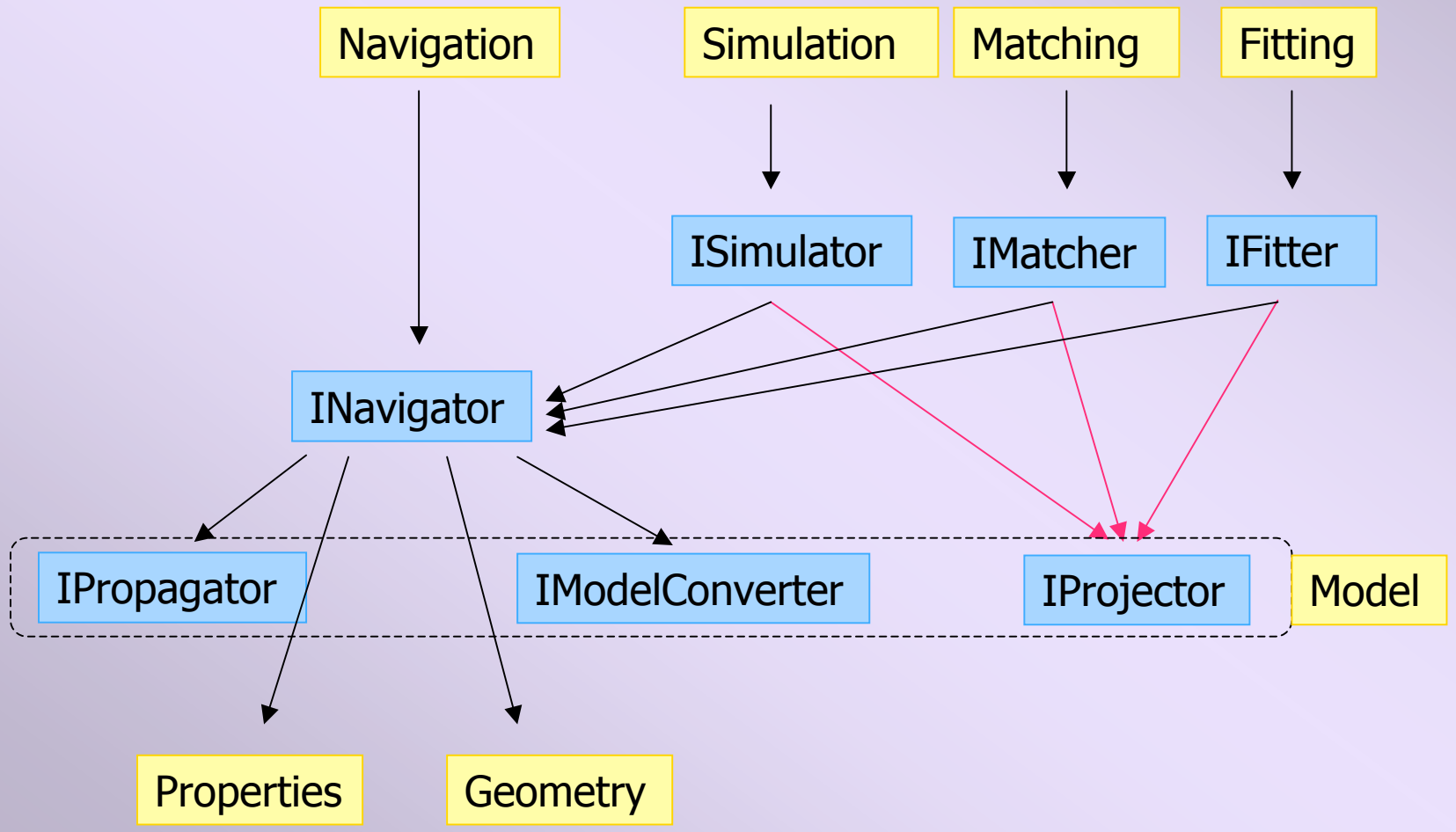
RecPack services

RecPack MANAGER

service name	Description
Model	mathematical equations for: propagation, projection, intersection with surfaces, random noise (ms), model corrections (eloss), model conversions, etc
Fitting	track fitting vertex fitting *cluster fitting*
Geometry	purely geometrical volumes and surfaces
Navigation	propagate to any surface or length taking into account the volume hierarchy
Property	Volume/surface properties (needed for navigation)
Matching	matching functions between trajectories, measurements, states, vertices etc. *pattern recognition methods*
Simulation	Simulate trajectories and measurements for debugging purposes

RecPack tools

- Services contain and manage the tools
- Tools operate over the data classes
- They are pure interfaces ⇒ extendible



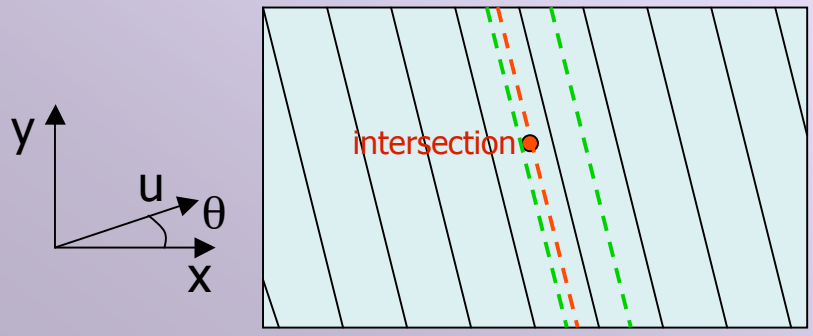
BUROCRACY
MATHEMATICS
SETUP

Propagator and Projector

- The propagator:

	Sub-tools
Computes distance to surface	*SurfaceIntersector*
Propagates state vector to that distance	*ModelEquation*
Applies model corrections (energy loss, etc)	*ModelCorrection*
Propagates the covariance matrix	*Propagator*
Adds random noise (multiple scattering, eloss fluctuations, etc)	*Noiser*

- The projector:
  - Converts the propagated state into a virtual measurement



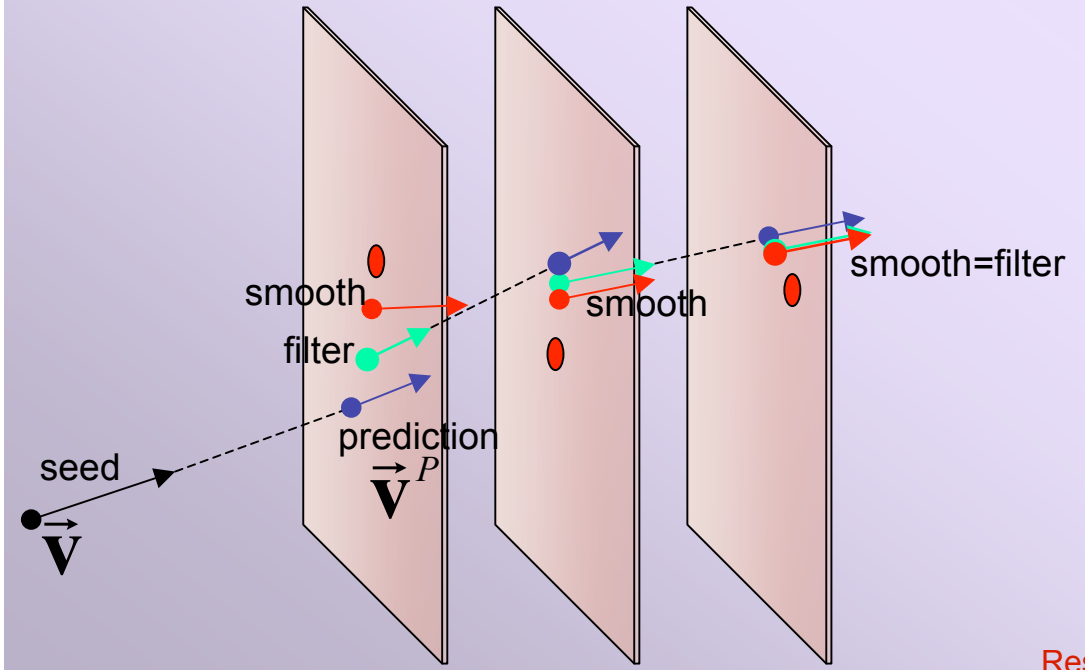
Measurement (virtual) $(u) =$ *Projector* $(\cos\theta \quad \sin\theta \quad 0 \quad 0 \quad 0)$ *State Vector* $\begin{pmatrix} x \\ y \\ x' \\ y' \\ \frac{q}{p} \end{pmatrix}$

Fitting: Kalman Filter

- Used for track and vertex fitting by most of HEP experiments
- Easy to include random noise processes (*multiple scatt.*) and systematic effects (*eloss*)
- It is a local and incremental fit

→
- simultaneous pattern recognition and fitting
- detection of outliers

Any other fitting algorithm can be plugged in



IPropagator

$$\vec{v}^{\,P} = f(\vec{v}) \xrightarrow[linear]{} \vec{v}^{\,P} = \mathbf{F}\vec{v}$$

Transport matrix

$$C^{P} = FCF^{T} + \mathbf{Q}$$

Noise matrix (ms)

IProjector

$$\vec{w} = h(\vec{v}) \xrightarrow[linear]{} \vec{w} = \mathbf{H}\vec{v}$$

Projection matrix

$$\vec{r} = \vec{w} - \vec{m}$$

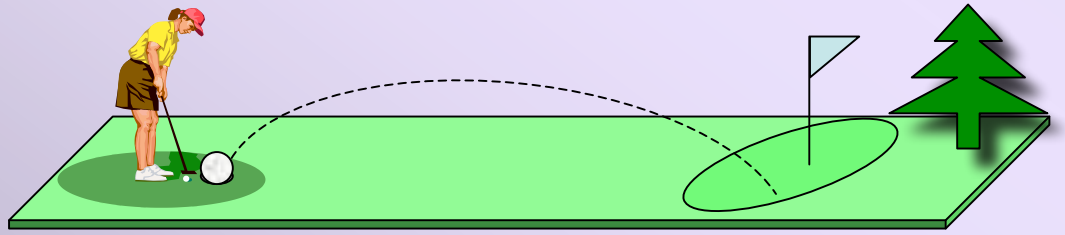
Residual = prediction-measurement

Model tools

Models	Noise estimators	Model corrections	surface intersectors	projectors
straight line in any dimension	multiple scattering	energy loss	plane	2D
helix in variable B field	*energy loss*		cylinder	3D
			sphere	rϕ

All these pieces live in independent containers & *implement the corresponding interface*

→ ***Extensions are straight forward***



parabola	wind fluctuations	wind	earth surface	green	2D

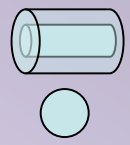
Geometry (I)

- Reconstruction processes can use a simplified geometry
- because measurement errors hide geometrical details:
  - screws, …
- The important objects are:
  - Measurement surfaces
  - Simplified volumes delimiting regions of different propagation properties:
    - *Radiation length, interaction length, density*
    - *Magnetic or electric field*
- This allows fast reconstruction using analytical extrapolations whenever is possible (homogeneous properties) and dynamic stepping otherwise

Geometry (II)

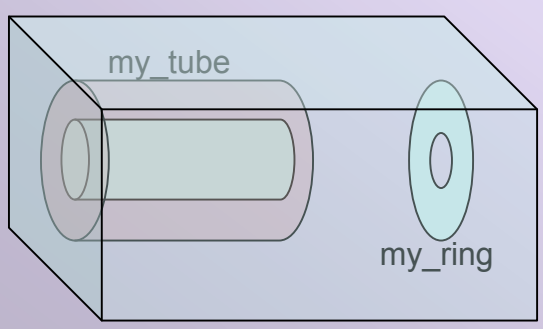
add_volume(“parent”, “my_box”, “box3D”, pos, axes, size)

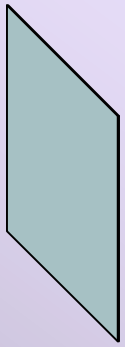
- tube
- sphere
- ….

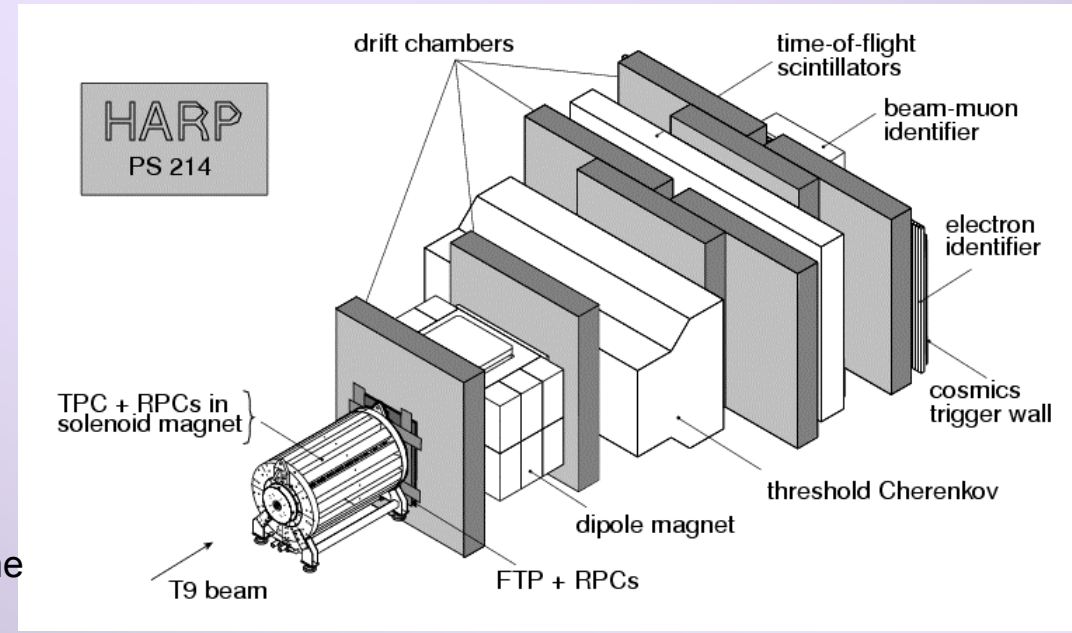


add_surface(“parent”, “my_plane”, “rectangle”, pos, axes, size)

- ring
- ring_sector
- cylinder_sector
- …

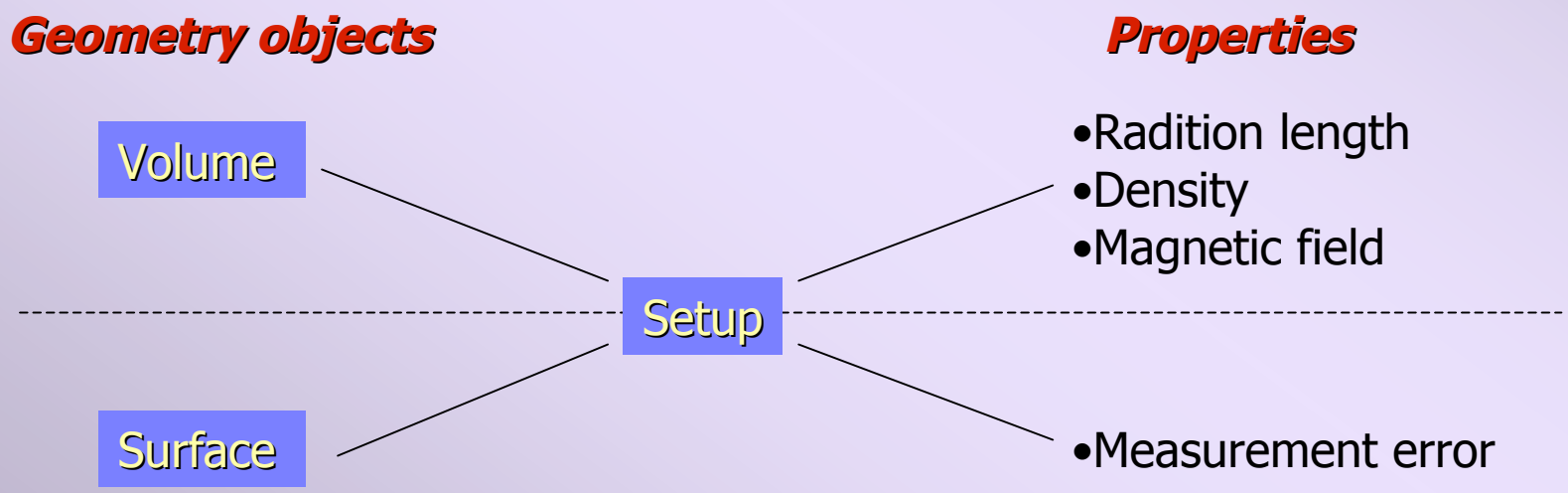






Properties

- Volumes and surfaces are purely geometrical:
  - Position, direction and size
  - Information about the volume hierarchy
- Properties are associated indirectly via the Setup class



Navigation

- A ***Navigator*** propagates states in a Setup via steps

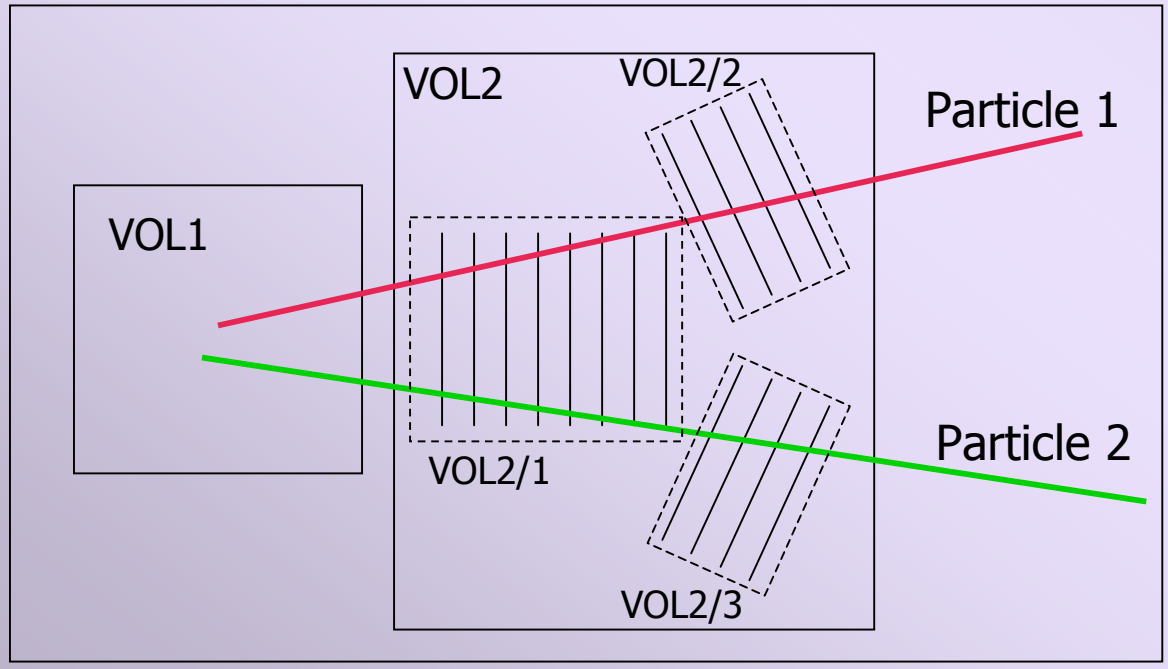
  navigation().propagate(state, length)

  navigation().propagate(state, surface)

- Before and after each step a list of ***Inspectors*** is called:
  - Change volume properties
  - Model conversion
  - Sum up intermediate path lengths
  - Set length of the next step
  - Counters
  - …
- Inspectors can be associated to any volume or surface
- User can:
  - Add its own inspector
  - Define analytic intersection for a given model and a surface type
    - Most relevant are provided by default
  - Establish a sequence of volumes and surfaces to intersect to speed up the propagation → *Navigation logic*

Navigation Logic

- In the most general case all possible surfaces seen by the current volume are intersected. Then the closest one is selected
- This is not efficient when the number of surfaces is large
- Instead navigation logics and trees



- VOL1
- VOL2
  - If !(VOL2/1) stop
  - If VOL2/1
    - S1-S8
    - If VOL2/2
      - S1-S4
    - If VOL2/3
      - S1-S4
    - !(VOL2/1 || VOL2/1)
      - stop

Clients and versions

RecPack-2

- ***HARP*** (CERN), where ***RecPack*** was born
- ***MICE*** (RAL)
- ***MuScatt*** (FNAL)
- ***MIPP*** (FNAL), recently requested

RecPack-1

- ***SciBar*** detector, which is part of ***K2K*** (Japan)
- Inspiration for tracking model in ***LHCb*** (CERN)
- Design of future neutrino experiments:
  - ***SuperNOvA*** (US)

RecPack0

- ***T2K*** (Japan)

All experiments will update to RecPack0 in few months

Future extensions for HEP

- Extension of vertex concept → *singularity*
  - Vertex, kink, decay
- And trajectory → *dynamic system*
  - Collection of trajectories connected through singularities
- Common pattern recognition logics:
  - *3D tracking:* TPC, …
  - *2D tracking:* planar, cylindrical, spherical, …
  - *Clustering:* 2D uncorrelated, 2D correlated, 3D, …
- Common PID algorithms
- GUI: (HEP and non-HEP)
  - Detector design and geometry maker
  - Event display
  - Visual debugger: interactive reconstruction (probably via Python)

Conclusions

- ***RecPack*** is a c++ reconstruction toolkit
- It provides the common tools of any reconstruction program ⟶ avoids reinventing the wheel !!!
- Its modular structure allows extensions in any direction

data types ⟶ volumes, surfaces, measurements, …
tools ⟶ models, navigators, simulators, fitters, …

- It is setup independent
- It is being successfully used by several HEP experiments
- Interested people (users or developers) may contact:

Anselmo.Cervera@cern.ch
Juan.Jose.Gomez.Cadenas@cern.ch
Jose.Angel.Hernando@cern.ch

New collaborators:
- ***Malcolm Ellis***
- ***Federico Sánchez***
- ***Javier Muñoz***

Coming soon:
- ***Detailed documentation***
- ***WWW page***