ReconstructedParticle Photons

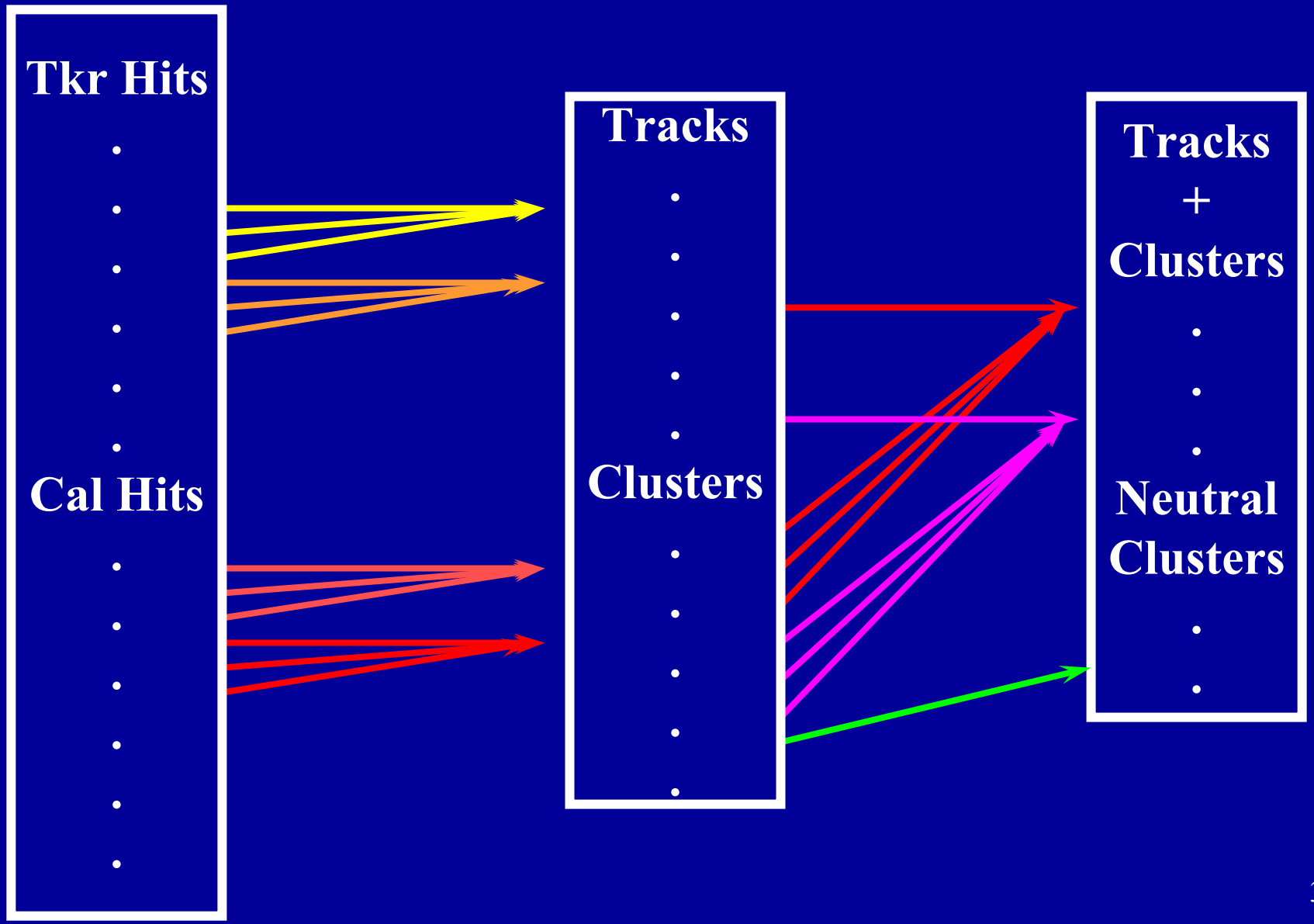
Norman Graf

March 15, 2004

Problem Statement

- In order to implement the energy flow algorithm as a part of the event reconstruction, we need a framework within which we can communicate between packages.
- Need common definitions (interfaces) for constituents of final reconstructed particles.

Reconstruction Flow



Design Specifications

- All reconstructed particles, simple and composite, are of the same base type.
- Kinematics and identity of a ReconstructedParticle should be independent.
- Identity of a ReconstructedParticle given by data member, not by the concrete class type.
- The identity of a ReconstructedParticle may be undefined.
- When defined it should be easy to change
  - after application of alternative ID algorithm.

ReconstructedParticle I

- A class which encapsulates the behavior of an object which can be used for physics analysis.
  - mirrors MCParticle
- Kinematics determined by track momentum or calorimeter cluster energy at time of creation.
- ID determined later by particle ID algorithms, e.g. track dE/dx, cluster shape, or combination of detector element variables.
  - could entertain multiple hypotheses.

ReconstructedParticle II

- Can also be created from combinations of other ReconstructedParticles.
- e.g. Photon can be single EM cluster without associated track, or combination of e^+ and e^-, each composed of an EM cluster and a matching track.
- Resonances, when identifiable.
- Jets are also ReconstructedParticles.

ParticleType

- Encapsulates information about known types of particles, e.g.
  - name
  - mass
  - charge
  - …
- Not limited to physics particles, could also simply consider "EFlow" particles, e.g. "Neutral EM", "Neutral Hadron", "Charged Hadron", etc.

ParticleId

- Combines a ParticleType and the probability for the id given by a ParticleTypeIdentifier.
- Also contains a GUID to allow the identification to be reviewed at a later time.
- ReconstructedParticle should contain all the information needed by a ParticleTypeIdentifier to return a ParticleId.

ReconstructedParticle attributes

- Collection of calorimeter Cells and/or Clusters
- Collection of Tracks
- Collection of ParticleIds (sorted by probability)
- Mass
- Charge
- Kinematics
- Collection of ReconstructedParticles of which this is composed
- ReconstructedParticle of which this is a constituent
  - (Flag to indicate whether this is a final state)

ParticleTypeIdentifier

- An interface used to provide particle type identification for a ReconstructedParticle.
- Any class implementing ParticleTypeIdentifier is required to provide a constructor taking a single String as argument.
- This provides a mechanism to recreate the identification at some future time using class reflection.
- ParticleId identify(ReconstructedParticle part);

Prototype Reconstruction

```
public ReconstructedParticleJob(double radius, double
    seedEmin, double clusEmin, String hmxName, double
    clusEmin, double chisqmin, double trackdistmin)
    {
        // Smear Tracker hits with resolution
        add(new SmearDriver());
        // Find tracks
        add(new TrackReco());
        // build up the eflow event
        // sets up and populates the CalorimeterHitMap
        add(new EflowEventBuilder());
```

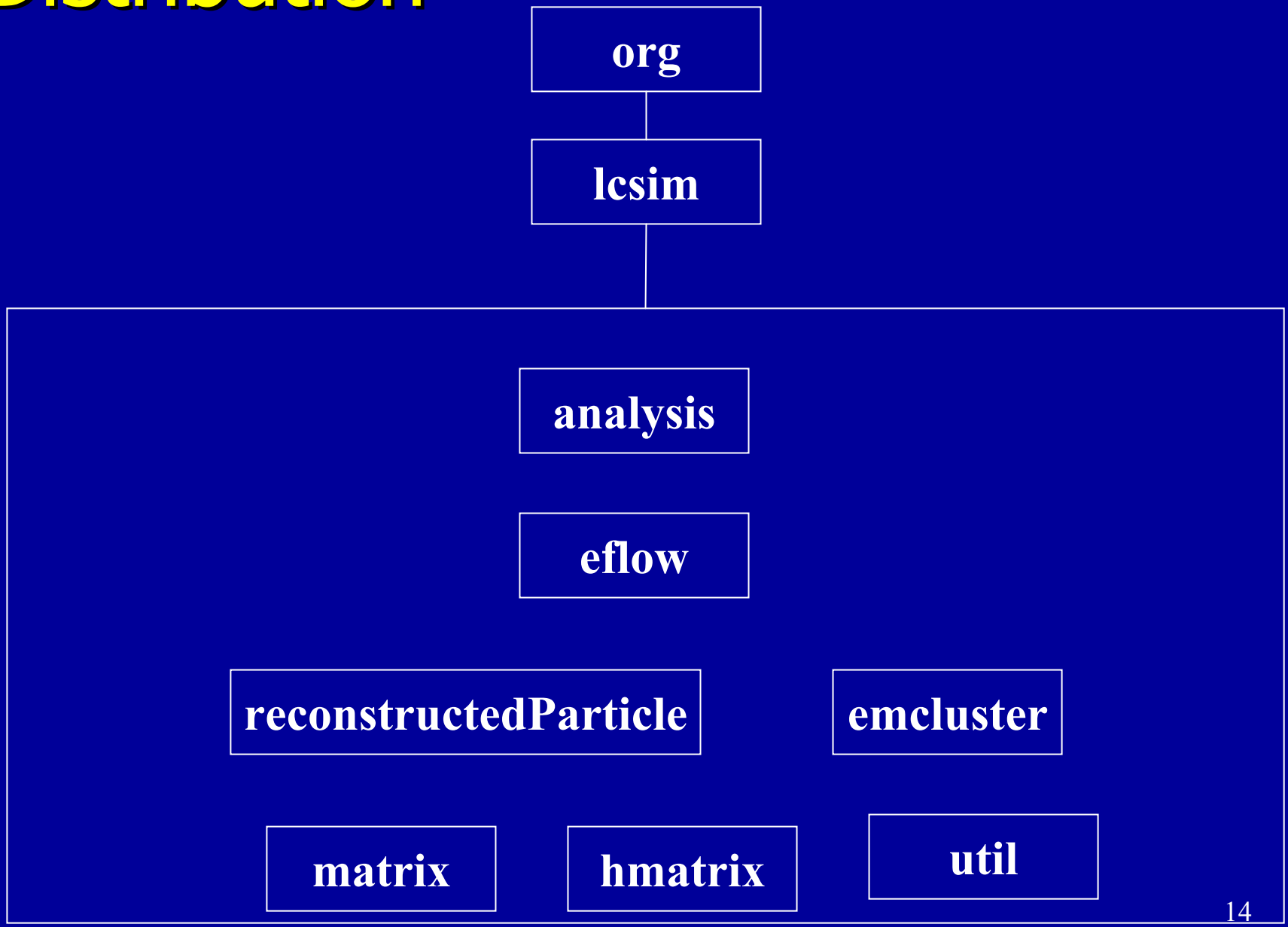
Prototype Reconstruction

```
// Find EM clusters using a simple cone algorithm
add(new EMConeClusterBuilder(radius, seedEmin, clusEmin));
// Construct and identify the ReconstructedParticles
// Photons, electrons, pi0
add(new EMParticleFinder(hmxName,clusEmin,chisqmin,trackdistmin));
// Analysis!
add(new ReconstructedParticleAnalyzer());
}
```

Reference Implementation

- I have provided a reference implementation for this proposal.
- As this is a draft proposal, and also to avoid any conflicts, I have placed all the code in packages, org.lcsim…
- Full source code, documentation, a build.xml file for ant, and a jar file containing the compiled classes can be found at:
- http://www-sldnt.slac.stanford.edu/nld/reconstruction/EnergyFlow/orglcsim_full.zip

Distribution



Documentation

- Javadoc-generated API is available at:
- http://www-sldnt.slac.stanford.edu/nld/reconstruction/EnergyFlow/docs/

Usage

- For now, simply copy the lcsim-1.0.jar file into your JAS extensions directory. This will then automatically be placed in your CLASSPATH when developing within JAS.
- Can also simply put jar file into your CLASSPATH
- Can also point your CLASSPATH to head of source distribution, orglcsim/src

Example Analyses

- Examples for analyzing EM clusters and reconstructed photons. Both have main method to allow line-command analysis.
  - Example code has hard-coded filenames!
  - Example code uses JAS2 style histograms!
  - Will fix this!
- EMClusterJob will process and analyze EMClusters.
- ReconstructedPhotonJob will process and analyze the full chain.

Summary

- ReconstructedParticle design proposed.
- Separate kinematics and identity.
- Clean interface at this level allows much closer collaboration and easier extension.
- Will continue to work on implementation and documentation.
- Welcome feedback and participation in design.