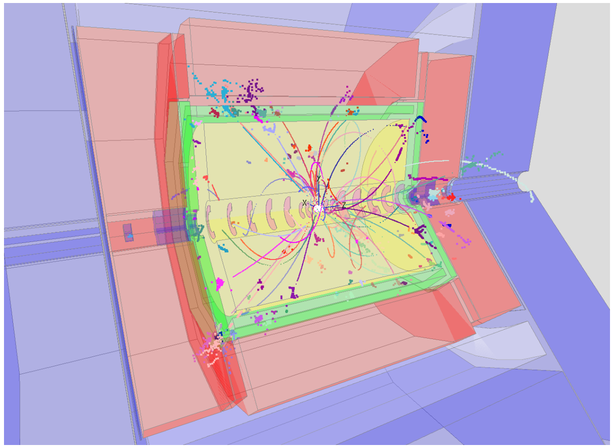
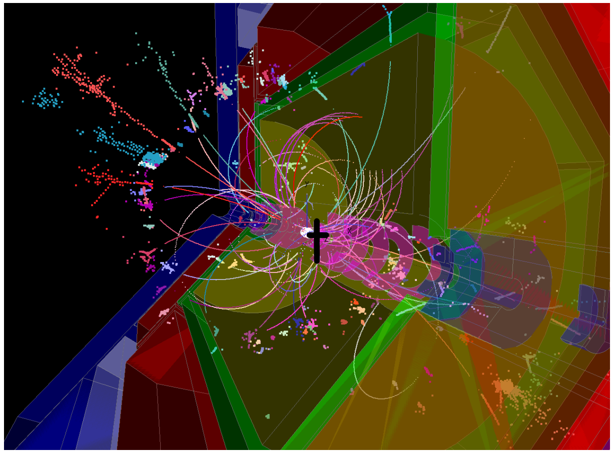
Plans of the Software group towards the DBD



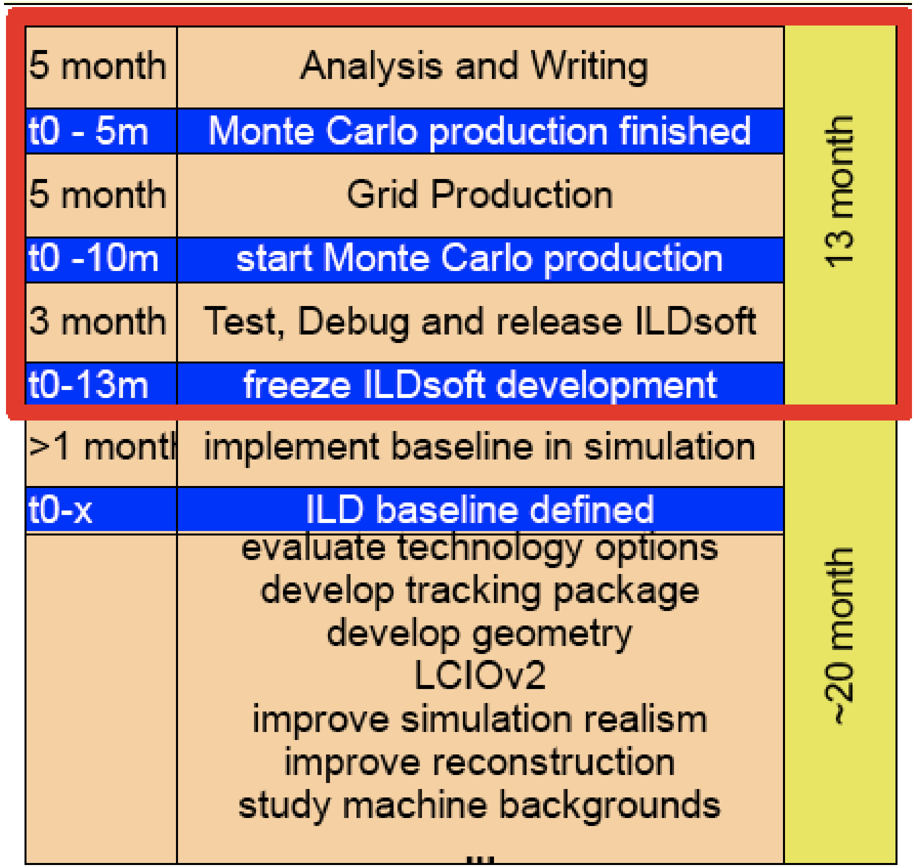


Frank Gaede, DESY
ILD Meeting 2011
Orsay, May 23-25, 2011

Outline

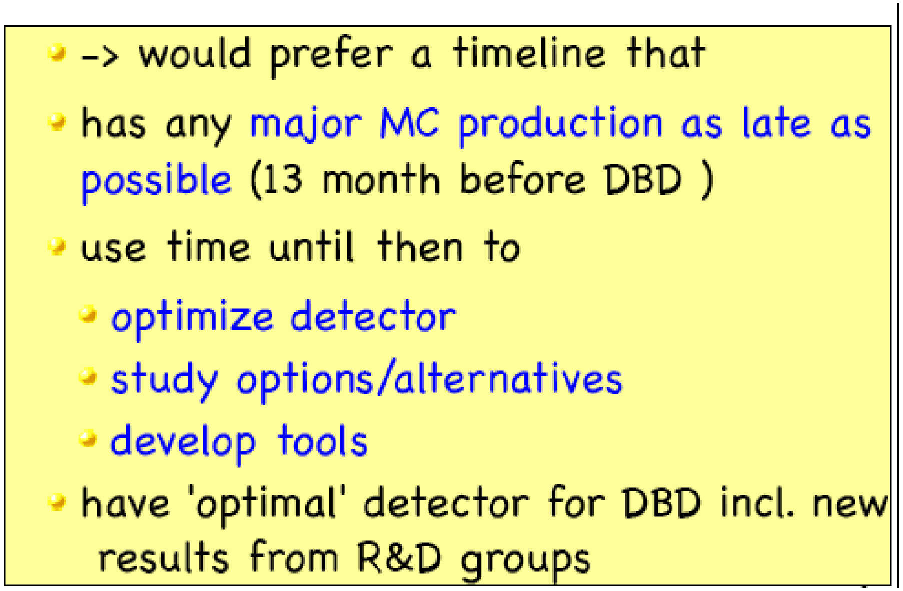
- Introduction – Timeline
- Core Software
    - LCIO v2
    - Grid production system
- Tracking
    - TPC
    - Si-Tracking
- DBD Monte Carlo production

ILD software timeline



5 month	Analysis and Writing	13 month
t0 - 5m	Monte Carlo production finished	
5 month	Grid Production	
t0 -10m	start Monte Carlo production	
3 month	Test, Debug and release ILDsoft	
t0-13m	freeze ILDsoft development	
>1 montl	implement baseline in simulation	~20 month
t0-x	ILD baseline defined	
	evaluate technology options develop tracking package develop geometry LCIOv2 improve simulation realism improve reconstruction study machine backgrounds ...	

agreed timeline for ILD software:



- -> would prefer a timeline that
- has any major MC production as late as possible (13 month before DBD)
- use time until then to
  - optimize detector
  - study options/alternatives
  - develop tools
- have 'optimal' detector for DBD incl. new results from R&D groups

- **need to define simulation models very soon - ideally now**
- need time to integrate, test and debug the code
- develop reconstruction software (tracking, PFA, flavor tag) before end of year

Towards LCIOv2

- goal is to improve LCIO while still being backward compatible
- planned/requested features:
- **direct access to events** **-> Done**
- **partial reading of events** -> postponed
- **splitting of events over files** -> postponed
- **storing of (arbitrary) user classes** -> currently not planned
- **simplify using LCIO with ROOT** **-> Done**
  - (ROOT macros, TTreeViewer, I/O (?) ,...)
- **improving the event data model** **-> Under Way (this talk)**
  - (1d,2d hits, tracks/trajectories)

- Had LCIO developers meeting in Eugene after ALCPG 2011
  - made significant progress in defining the EDM improvements
  - **implementation already started - hope to release this summer**

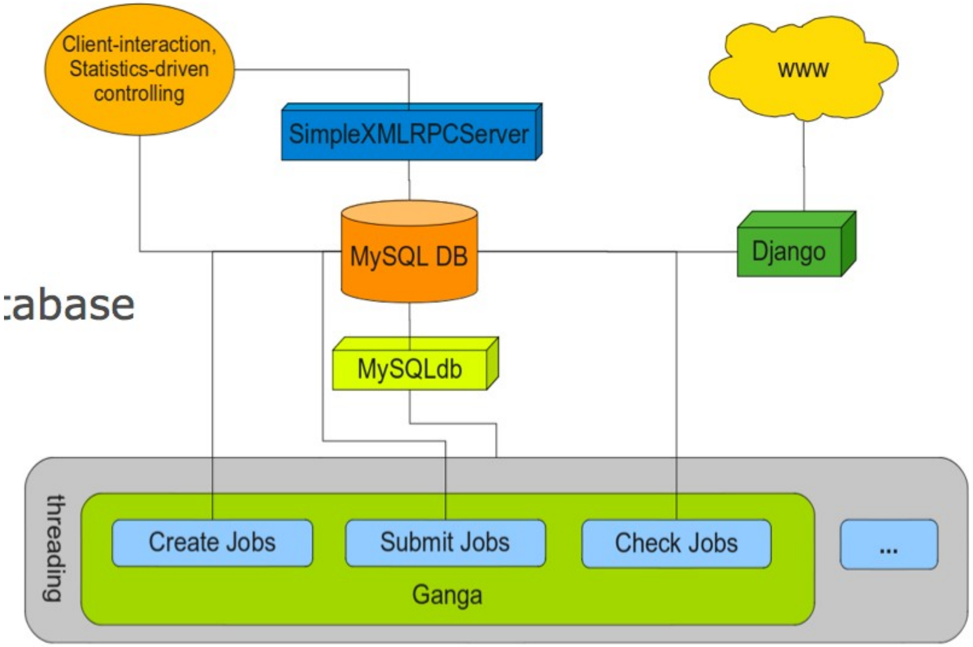
LCIO v2 - EDM improvements

- MCParticle
  - float[3] getSpin()
  - int[2] getColorFlow()
  - SimProcessName (in collection)
- (Sim)TrackerHit
  - getCellID0(), getCellID1()
  - getLayerNumber()
- SimCalorimeterHit
  - float[3] getStepPosition(int i)
  - e.g. for SDHcal digitization !
- Cluster
  - float getEnergyError()
  - float getTime() - new request

- Track
  - store multiple TrackStates
  - AtIP, AtfFirstHit, AtLasthit, AtCalorimeter, AtVertex, Other
  - TS* getClosestTrackState(x,y,z)
  - TS* getTrackState(int location)
- agreed to introduce six new TrackerHit classes
  - PlanarDisk1D, Planar1D, Cylindrical1D
  - PlanarDisk2D, Planar2D, Cylindrical2D
  - have u, du, pos1, pos2 (strip begin end) for 1D
  - have u, v + cov(u,v) + cylinder/plane parameters for 2D
  - implement 3D TrackerHit interface

Grid production system

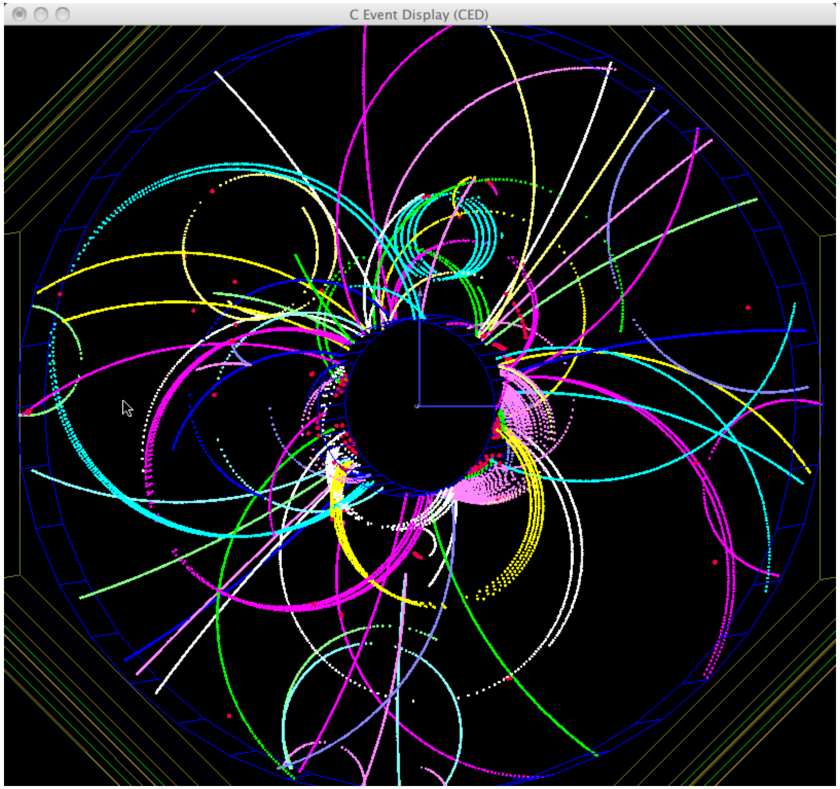
- have developed a Grid production system at DESY (J.Engels)
- tested for some smaller scale MC production
- 'ready' to use for DBD production
- need to setup DB and web-interface for data catalogue



- core software group (DESY) happy to lead DBD production
- any help is of course very welcome
- need to allocate Grid resources
- -> need to define samples soon for estimation of resources

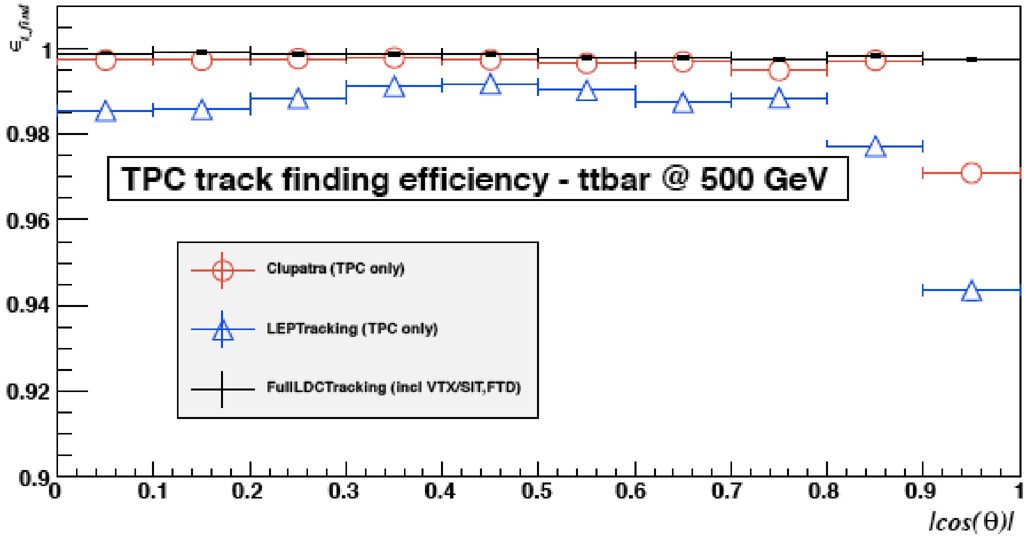
- **offer from CLIC group to contribute to production with DIRAC system - of course very welcome**
  - need to see how best to share the tasks

Clupatra: topological TPC pat-rec



- use **NN-clustering in full TPC**
  - merge hits that have dist<3cm
- in merged clusters (duplicate pad rows) cluster in **pad row ranges** (15 rows) - **outside inwards** to find clean track stubs
- ***extend clean stubs with Kalman fitter***
  - pick up matching hits fwd & bwd if delta(chi2) < 35.
  - update track state
- ***force leftover clusters into one, two or three tracks*** (depending on pad row multiplicity)
- ***merge curler segments:***
  - delta(R), delta(xc,yc) and delta(tanL) < 10%

- track finding efficiency better than previous algorithm (based on LEP tracking code)
- NB: no fully reconstructed tracks yet -> might loose a bit due to quality cuts
- next steps:
  - fully reconstruct tracks
  - merge with Si-Tracks (hits)



Tracking - next steps

- started to develop IMarlinTrack:
  - interface between LCIO and Track fitters (Kalman filters):
    - Delphi, KalTest and possibly others in the future
  - allows track fitting and development of pat rec code
  - access to several track states (atIP, atFirst/LastHit,....)

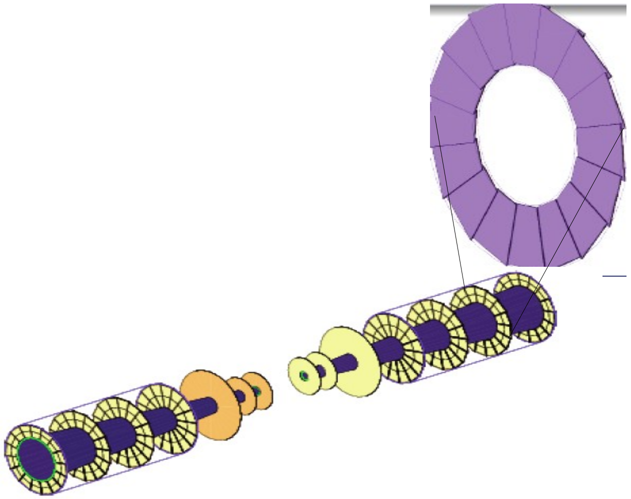
- TPC tracking
  - add propagation through inner material
  - track collections with quality cuts
  - study effects of background
    - pair-bg and gamma-gamma

- include Si-Trackers (VXD, SIT, ETD, SET, FTD)
  - pick up hits for TPC tracks (central)
  - need pattern recognition for FTD (Vienna)
  - develop standalone Si-Tracking (if time allows)

Si-Tracking - issues

- recently new Mokka simulation drivers added

  for SIT, ETD, SET, FTD
  - increased level of detail/realism
  - single wavers, support, gaps, ...



- need digitization code for this
  - realistic strip hits (1D) !?
  - has to come from R&D groups
- navigation in Track Fit:
  - KalTest currently has no 'bounded planes' (wafers)
  - need ordering of layers -> difficult to deal with overlaps (VXD)
  - -> will address these together with KalTest developers
  - track extrapolation to rotated planes (FTD) based on simple Newtonian method
  - would prefer to have design with wafers at fixed z (staggered)
    - under investigation by R&D experts
- need to be practical in order to stay on DBD timeline...

ILD simulation models for DBD

- need to define simulation model(s) for the DBD
  - for central Monte Carlo production of benchmark and other physics channels incl. background
  - can't afford **massive** production with many models
  - if we would like to have MC production for more than one model, we need to cut down on
    - # channels, lumi, background,...
  - validating the complete sw chain for a mass production for one models is a considerable effort
  - man power is smaller than for LOI
  - **cannot be done without support from R&D groups**

- requirements for subdetector software in simulation model:
  - improved realism wrt. LOI
    - gaps, cables, services and imperfections
  - tested and debugged Mokka driver exists
    - that writes proper GEAR parameters
  - tested and debugged digitizer code exists
  - tested and debugged reconstruction code exists, that
    - has demonstrated the physics performance that is needed
    - validated up by tbeam
  - is approved by corresponding R&D group
  - maintained and supported by authors !